

EFEKTY UCZENIA SIĘ I TREŚCI PROGRAMOWE ZAJĘĆ

Kierunek: **Informatyka kwantowa**
Poziom studiów: **studia pierwszego stopnia**

Nazwa zajęć: **Zarządzanie produktem**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie pojęcie „produktu” i „zarządzania produktem”, charakter pracy product managera.

w zakresie umiejętności:

1. ma opanowane podstawowe techniki i narzędzia product managera

Treści programowe dla zajęć:

Produkt, zarządzanie produktem – wprowadzenie, definicje i historia, wspaniałe produkty.

Cykl życia produktu.

Product Manager a Product Owner: produkt w świecie metodologii zwinnych (Scrum, skalowany Scrum np. SAFE).

Praca z zespołem, role w zespole we współczesnej firmie informatycznej, tzw. “empowered product teams”.

Business Cases oraz elementy marketingu.

Użytkownicy produktu: segmentacja i persony.

Wymagania: pochodzenia, analiza, zarządzanie.

Mapy drogowe (roadmaps).

Prototypowanie: cele, rodzaje prototypów.

Dostępność (accessibility).

Analiza zachowań użytkowników w skali.

Zarządzanie produktami różnego typu (backend, frontend, produkty nieinformatyczne).

Narzędzia i systemy Product Managera.

Etyka w świecie produktu.

Rekrutacja w świecie produktu.

Nazwa zajęć: **Elementy algebry i teorii liczb**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna podstawowe definicje struktur algebraicznych i ich typowe przykłady.

2. zna pojęcie grupy i potrafi wymienić podstawowe przykłady.

3. zna pojęcie pierścienia i ciała.

4. zna rozszerzony algorytm Euklidesa, chińskie twierdzenie o resztach.

5. zna podstawowe algorytmy wielomianowe.

6. zna podstawowe algorytmy teoriogrupowe.

w zakresie umiejętności:

1. potrafi posługiwać się podstawowymi strukturami algebraicznymi i wykonywać na nich proste rachunki.

2. potrafi wykonywać obliczenia na grupach permutacji.

3. potrafi wykonywać obliczenia na ciałach skończonych oraz algebrach Boole'a.

4. potrafi obliczyć za pomocą pakietu numerycznego bazę Groebnera i rozwiązać za jej pomocą układ równań algebraicznych.

Treści programowe dla zajęć:

Podstawowe struktury algebraiczne, język algebry, motywujące przykłady.

Podstawowe pojęcia z teorii grup wraz z przykładami z permutacji. Omówienie maszyny szyfrującej Enigma jako przykład wykorzystania grup permutacji.

Definicje i przykłady pierścieni oraz ciał. Pierścień liczb całkowitych, macierzy oraz ich zastosowania.

Ciało liczb wymiernych, rzeczywistych i zespolonych oraz ciała skończone. Podstawowe wykorzystanie tych struktur.

Ciała skończone, algebry Boole'a. Podstawowe rachunki i zastosowania w informatyce.

Algorytmy całkowitoliczbowe i ich zastosowania w informatyce. Sito Eratostenesa, testy pierwszości, algorytm Euklidesa, chińskie twierdzenie o resztach.

Algorytmy wielomianowe. Elementy rachunków symbolicznych.

Wstęp do rachunku symbolicznego: bazy Groebnera i ich zastosowania.

Nazwa zajęć: **Elementy logiki i teorii mnogości**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna spójniki logiczne, pojęcie formuły KRZ, tautologii KRZ, formuły niespełnialnej, zna podstawowe tautologie KRZ.
2. zna pojęcie logicznego wynikania, schematu wnioskowania, wnioskowania dedukcyjnego.
3. zna koniunkcyjną i alternatywną postać normalną formuł KRZ.
4. zna pojęcie i zastosowanie systemu sekwentowego.
5. zna regułę rezolucji zdaniowej i jej zastosowania.
6. zna podstawowe pojęcia rachunku predykatów pierwszego rzędu.
7. rozumie pojęcie uzgadniania wyrażeń prostych.
8. zna i rozumie regułę rezolucji liniowej i jej zastosowania.
9. zna podstawowe pojęcia teorii mnogości.
10. zna pojęcie iloczynu kartezjańskiego, relacji, zna rodzaje relacji, relacje równoważnościowe, relacje porządkujące.
11. zna podstawowe pojęcia związane z modelami teorii formalnych.

w zakresie umiejętności:

1. potrafi zbudować schemat logiczny zdania, potrafi sprawdzać tautologie metodą tablicy i metodą nie wprost.
2. potrafi sprawdzać niezawodność schematów wnioskowania, potrafi sprowadzać formuły do KPN i APN oraz sprawdzać ich tautologiczność i spełnialność.
3. potrafi posługiwać się systemem sekwentowym jako systemem automatycznego dowodzenia.
4. stosuje regułę rezolucji do sprawdzania niespełnialności zbioru klauzul, przeprowadza dowody rezolucyjne.
5. potrafi budować kwantyfikatorski schemat zdania, potrafi zapisywać definicje i twierdzenia w symbolice logicznej.
6. potrafi wyprowadzać prawa rachunku predykatów metodą przekształceń formuł.
7. potrafi sprawdzić unifikowalność wyrażeń, znajduje unifikator.
8. potrafi zapisywać bazy wiedzy za pomocą klauzul Horna i rozumie zasady wnioskowania metodą rezolucji.
9. potrafi pisać procedury w Prologu, stawia poprawnie cele w Prologu.
10. potrafi wyprowadzać prawa działań na zbiorach za pomocą aksjomatu ekstensjonalności i ilustrować je na diagramach Venna.
11. potrafi wyprowadzać prawa iloczynu kartezjańskiego, potrafi przedstawiać relacje na wykresie, potrafi wyprowadzać prawa algebry relacji.
12. potrafi sprawdzać, czy relacja jest relacją równoważności, potrafi wyznaczać klasy abstrakcji relacji równoważności.
13. potrafi sprawdzać, czy relacja jest relacją porządkującą i liniowo porządkującą, potrafi wyznaczać element największy i najmniejszy, elementy maksymalne i minimalne, kres górny i dolny.

w zakresie kompetencji społecznych:

1. jest gotów/gotowa do zrozumienia znaczenia logiki w informatyce i kształtowaniu życia społecznego.
2. jest gotów/gotowa do dalszego samodzielnego kształcenia w zakresie logiki i teorii mnogości.

Treści programowe dla zajęć:

Spójniki logiczne, schemat logiczny zdania, wartościowania, tautologie rachunku zdań, równoważność logiczna.

Formuły spełnialne, wynikanie logiczne, logiczna reguła wnioskowania, podstawianie w rachunku zdań, koniunkcyjna i alternatywna postać normalna.

Systemy aksjomatyczne rachunku zdań, poszukiwanie dowodu w systemie sekwentowym.

Metoda rezolucji.

Kwantyfikatory, formuły rachunku predykatów, interpretacje, prawa, logiczne reguły wnioskowania.

Wyprowadzanie praw rachunku predykatów, rachunek predykatów z równością.

Uzgadnianie.

Klauzule Horna, procedury i programy Horna, zapytania.

Rezolucja liniowa.

Prolog: składnia, programy, zastosowania. Struktury danych, rekurencja i sterowanie wnioskowaniem w Prologu.

Podstawowe pojęcia teorii mnogości, własności inkluzji, zbiory skończone, zbiór wyznaczony przez warunek, aksjomat ekstensjonalności, działania na zbiorach.

Para uporządkowana, iloczyn kartezjański, relacje, algebra relacji.

Relacje równoważności, klasy abstrakcji, zasada abstrakcji.

Relacje porządkujące i liniowo porządkujące, zbiory uporządkowane, element największy i najmniejszy, maksymalny i minimalny, kres górny i dolny, porządek produktowy i leksykograficzny.
Modele teorii formalnych.

Nazwa zajęć: **Optymalizacja**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i potrafi posługiwać się terminologią związaną z problemami optymalizacyjnymi
2. zna i rozumie klasyczne problemy optymalizacji dyskretnej.
3. zna i rozumie problemy optymalizacji liniowej.

w zakresie umiejętności:

1. potrafi przeanalizować, dobrać odpowiednią metodę oraz rozwiązać klasyczne zadania optymalizacji dyskretnej..
2. potrafi formułować oraz rozwiązywać różne problemy optymalizacji liniowej.

Treści programowe dla zajęć:

Programowanie liniowe - metoda simplex (oraz informacyjnie metody: graficzna, punktu wewnętrznego oraz epsolidalna).

Programowanie całkowitoliczbowe - metoda relaksacji programu liniowego.

Klasyczne problemy optymalizacji na grafach.

Przepływy w sieciach.

Heurystyczne algorytmy optymalizacji dyskretnej (metody wspinania, symulowanego wyżarzania, listy tabu oraz gentyczna)

Nazwa zajęć: **Język angielski B21**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. tworzyć ustne wypowiedzi na przygotowane tematy, prezentować i argumentować swoje stanowisko oraz innych osób na tematy związane ze swoim otoczeniem jak i na tematy ogólno-akademickie.
2. czytać ze zrozumieniem teksty w języku angielskim charakterze ogólnym jak i akademickim, związane z kierunkiem studiów, oraz analizować ich treść i wybierać niezbędne informacje.
3. zrozumieć oryginalny materiał audio lub wideo na większość tematów dotyczących życia codziennego, kulturalnego i społecznego, na poziomie ogólnym jak i wychwycić niezbędne szczegóły.
4. przygotować i wygłosić prezentację na wybrany temat.
5. opracować teksty oraz wypowiedzi dotyczące życia społecznego, uniwersyteckiego i zawodowego.
6. redagować wybrane teksty w stylu formalnym.
7. uzupełniać i doskonalić nabytą wiedzę i umiejętności.

Treści programowe dla zajęć:

Przegląd i utrwalenie umiejętności w zakresie posługiwania się formami i funkcjami czasów gramatycznych odpowiednich dla poziomu B2.

Inne struktury gramatyczne potrzebne do wyrażania różnorodnych treści i opinii: okresy warunkowe typ 1,2,3 oraz mieszane; struktury gramatyczne 'wish,'get used to/used to, past modals, formy bezokolicznikowe i imiesłowowe.

Słownictwo dotyczące problematyki współczesnego świata w zakresie następujących tematów: ekstremalne sytuacje, refleksja na temat planów życiowych, terapeutyczna funkcja muzyki, higiena snu, komunikacja niewerbalna oraz wybrane słownictwo akademickie i specjalistyczne związane z kierunkiem studiów.

Strategie efektywnego czytania w celu zrozumienia ogólnego sensu wypowiedzi w tekstach popularno-naukowych oraz specjalistycznych; domyślanie się znaczenia nieznanymi słów w zakresie bloków tematycznych określonych w treści 3.

Strategie efektywnego słuchania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanymi słów w zakresie bloków tematycznych określonych w treści 3.

Udzielanie odpowiedzi, udział w dyskusji oraz wyrażanie różnorodnych funkcji językowych w zakresie tematyki określonej w treści 3.

Redagowanie wybranych typów tekstów formalnych.

Nazwa zajęć: **Kryptografia z elementami algebry**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Zna strukturę algebraiczną związaną z systemem kryptograficznym.
2. Zna współczesną terminologię kryptologiczną.

w zakresie umiejętności:

1. Potrafi zaimplementować działania na elementach struktur algebraicznych.
2. Wykorzystuje twierdzenia matematyczne w analizie systemów kryptograficznych.
3. Analizuje bezpieczeństwo algorytmów kryptologicznych. Potrafi wskazać wady i zalety danego rozwiązania kryptologicznego.
4. Potrafi obliczyć złożoność obliczeniową algorytmów wykorzystywanych do konstrukcji systemów kryptologicznych.

Treści programowe dla zajęć:

Działania wewnętrzne w zbiorze i ich własności. Zasadnicze pojęcia teorii grup – podgrupy, warstwy. Operacje elementarne na bitach. Notacja wielkie O. Algorytmy czasu wielomianowego oraz wykładniczego ze względu na liczbę bitów danych.

Podstawowe protokoły kryptograficzne. Bezpieczeństwo doskonałe. Bezpieczeństwo obliczeniowe. Kandydaci na funkcje jednokierunkowe. Problem logarytmu dyskretnego w grupie. Problem i-tego bitu logarytmu dyskretnego.

Rząd elementu w grupie. Grupy cykliczne. Homomorfizm grup. Jądro oraz obraz homomorfizmu. Grupa ilorazowa i twierdzenie o izomorfizmie.

Algorytmy szyfrowania z kluczem publicznym. Algorytm ElGamala i RSA. Obliczanie logarytmu dyskretnego metodą wielkich i małych kroków. Równoważność między problemem faktoryzacji a łamaniem RSA.

Użycie klucza publicznego na przykładzie GPG. Generowanie, przedłużanie i wygaszanie kluczy. Szyfrowanie, podpisywanie, weryfikacja plików i maili. Budowanie łańcucha zaufania – podpisywanie kluczy.

Ciała skończone i ich własności. Reszty i niereszty kwadratowe w grupach skończonych. Obliczanie pierwiastków w ciałach skończonych.

Krzywe eliptyczne nad ciałem skończonym. Generowanie krzywej eliptycznej. Kodowanie wiadomości na krzywej. Algorytm obliczanie n-tej wielokrotności punktu krzywej eliptycznej.

Algorytm ElGamala na krzywej eliptycznej. Problem logarytmu dyskretnego na krzywej eliptycznej. Protokoły uzgadniania kluczy. Protokół Diffiego-Hellmana na krzywej eliptycznej. Wykorzystanie ECDH na przykładzie openssl.

Symetryczny protokół szyfrowania. Standard szyfrowania danych DES i jego modyfikacje. Tryby pracy szyfrów blokowych.

Pierścienie. Jedności pierścieni. Obraz i jądro homomorfizmu pierścieni. Pierścienie wielomianów. Algorytm Euklidesa dla wielomianów. Ideały i pierścienie ilorazowe.

Zaawansowany standard szyfrowania danych AES. Transformacje SubByte i MixColumn w AES Użycie AES do szyfrowania plików i dysków na przykładzie cryptsetup.

Funkcje hashujące. Atak urodzinowy. Gdzie występują, czym się charakteryzują, użycie ich we własnym kodzie.

Uwierzytelnienie. Poufność, wiarygodność i zaufanie: w przeglądarce internetowej. SSL/TLS na przykładzie openssl w budowaniu VPN, na przykładzie openssl.

Nazwa zajęć: **Chmura obliczeniowa**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. wie na czym polega wirtualizacja zasobów.
2. zna definicję przetwarzania chmurowego.
3. zna modele przetwarzania danych: infrastruktura jako usługa (IaaS), platforma jako usługa (PaaS), oprogramowanie jako usługa (SaaS).
4. zna modele wdrożeń: chmura publiczna, chmura prywatna, chmura hybrydowa.
5. zna normy dotyczące przetwarzania danych w chmurze.
6. rozumie na czym polega bezpieczeństwo chmury (PL/UE)

w zakresie umiejętności:

1. potrafi porównać modele przetwarzania danych.
2. potrafi sprawdzić zgodność z normami.

Treści programowe dla zajęć:

Wirtualizacja zasobów

Definicja przetwarzania chmurowego

Modele przetwarzania danych:

- Infrastruktura jako usługa (IaaS)
- Platforma jako usługa (PaaS)
- Oprogramowanie jako usługa (SaaS)

Porównanie modeli

Modele wdrożeń – chmura publiczna/prywatna/hybrydowa

Bezpieczeństwo chmury (PL/UE)

Normy i zgodność z normami (w praktyce)

Nazwa zajęć: **Architektura komputerów**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. wymienia i opisuje podstawowe elementy oraz mechanizmy działania systemu komputerowego.
2. porównuje najważniejsze typy architektury pamięci, listy rozkazów oraz komunikacji.
3. tłumaczy matematyczne podstawy kodowania liczb, logiki matematycznej oraz jej sprzętowej realizacji.
4. wyjaśnia zasady pisania, asemblowania i uruchamiania programów w języku assembler dla wybranych platform.

w zakresie umiejętności:

1. rozróżnia wartość liczbową zapisaną w różnych systemach kodowania.
2. stosuje poznaną wiedzę do przeanalizowania oraz napisania programu w assemblerze i uruchomienia go na wybranych platformach.
3. rozróżnia typ architektury zbioru rozkazów na podstawie trybów adresowania dostępnych w assemblerze dla danej platformy.

Treści programowe dla zajęć:

Architektura systemu komputerowego (podstawowe komponenty, magistrala systemowa, sygnały sterujące).

Architektura procesora (podstawowe bloki i rejestry, model programowy procesora).

Organizacja pamięci i tryby adresowania.

Cykl rozkazowy procesora, podprogramy i przerwania.

Układy wejścia-wyjścia.

Operacje arytmetyczno logiczne, dostęp do bitów.

Kodowanie liczb, matematyczne podstawy układów cyfrowych.

Architektura mikrokontrolera 8051 oraz platformy Raspberry Pi.

Assembler mikrokontrolera 8051 oraz platformy Raspberry Pi, implementacja pętli i rozgałęzień w kodzie assemblerowym.

Nazwa zajęć: **Komputery kwantowe - algorytmy i zastosowania**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna i rozumie algorytmy i narzędzia wysokopoziomowe pozwalające na zastosowanie komputera kwantowego do modelowania i rozwiązywania problemów optymalizacyjnych, problemów wyszukiwania i problemów SAT.
2. zna i rozumie narzędzia wysokopoziomowe pozwalające na zastosowanie komputera kwantowego do wyznaczania: stanów podstawowych i wzbudzonych w problemach struktury elektronowej i wibracyjnej, wyznaczania momentu dipolowego układu, rozwiązywania modelu Isinga i modelu Fermiego-Hubbarda na sieciach.
3. zna i rozumie narzędzia wysokopoziomowe pozwalające na zastosowanie komputera kwantowego do modelowania i rozwiązywania problemów rynków finansowych takich jak optymalizacja portfela i pozyskiwanie prawdziwych lub generowanych maszynowo danych do eksperymentów finansowych.
4. zna i rozumie narzędzia pozwalające na zastosowanie komputera kwantowego do uczenia maszynowego z użyciem jąder kwantowych i kwantowych sieci neuronowych w zastosowaniach do regresji i klasyfikacji.

w zakresie umiejętności:

1. potrafi zastosować komputer kwantowy lub symulator do modelowania i rozwiązywania problemów optymalizacyjnych oraz automatycznej konwersji między różnymi reprezentacjami problemu (np. Quantum Approximate Optimization Algorithm QAOA, Grover Adaptive Search).
2. potrafi zastosować komputer kwantowy lub symulator do wyznaczania: stanów podstawowych i wzbudzonych w problemach struktury elektronowej i wibracyjnej, pomiaru moment dipolowego układu, rozwiązywania modelu Isinga i modelu Fermiego-Hubbarda na sieciach.

3. potrafi zastosować komputer kwantowy lub symulator do modelowania i rozwiązywania problemów rynków finansowych takich jak optymalizacja portfela i pozyskiwanie prawdziwych lub generowanych maszynowo danych do eksperymentów finansowych.

4. potrafi zastosować komputer kwantowy lub symulator do uczenia maszynowego z użyciem jąderek kwantowych i kwantowych sieci neuronowych w zastosowaniach do regresji i klasyfikacji.

Treści programowe dla zajęć:

Modelowanie i praktyczne aspekty implementacyjne wybranych kwantowych protokołów i algorytmów na komputerze kwantowym i symulatorze kwantowym w środowisku Qiskit.

Druga kwantyzacja i odwzorowania między reprezentacjami problemów rozwiązywanych przez komputery kwantowe, związek między kwantową optymalizacją a wyszukiwaniem.

Modelowanie i rozwiązywanie problemów optymalizacyjnych i problemów SAT z użyciem praktycznych algorytmów kwantowych na komputerze i symulatorze kwantowym. Środowisko Qiskit Optimization.

Wykorzystanie komputera kwantowego i symulatora w uczeniu maszynowym w modelach uczenia z jądrem kwantowym lub kwantowymi sieciami neuronowymi. Środowisko Qiskit Machine Learning.

Wykorzystanie komputera kwantowego i symulatora do wyznaczania stanów podstawowych i wzbudzonych w problemach struktury elektronowej i wibracyjnej, wyznaczanie momentu dipolowego układu, rozwiązywanie modelu Isinga i modelu Fermiego-Hubbarda na sieciach. Środowisko Qiskit Nature.

Komputer kwantowy jako narzędzie do modelowania i rozwiązywania problemów rynków finansowych takich jak optymalizacja portfela i pozyskiwanie prawdziwych lub generowanych maszynowo danych do eksperymentów finansowych. Środowisko Qiskit Finance.

Nazwa zajęć: Wstęp do informatyki

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Wie, czym się zajmuje i jaką odpowiedzialność ponosi informatyk oraz zna podstawowe pojęcia z zakresu informatyki jako nauki.

2. Zna i rozumie pojęcie algebry Boole'a i jego znaczenie w informatyce.

3. Zna sposoby reprezentacji liczb stało- i zmiennoprzecinkowych w pamięci komputera (znak-moduł, uzupełnieniowa do dwóch, standard IEEE 754) oraz pojęcia z nimi związane (cyfry poprawne i znaczące, pojęcia nadmiaru i niedomiaru, epsilon maszynowy). Wie, jak przebiegają działania na liczbach zmiennopozycyjnych i rozumie działanie algorytmu sumacyjnego Kahana.

4. Zna i rozumie podstawowe pojęcia z teorii informacji (ilość informacji, entropia informacyjna źródła, redundancja, kod zwarty, algorytm Huffmana).

5. Zna podstawowe techniki przetwarzania niskopoziomowego, pojęcie bramki logicznej, sumatora, maszyny Turinga. Zna ogólne zasady pracy procesora.

6. Zna podstawowe elementy i zastosowanie XML, DTD, XSD, XSL (XSLT i XPath). Zna podstawy HTML i CSS. Zna formaty JSON, YAML i CSV.

7. Zna zasady formatowania tekstu przy użyciu LaTeXa.

8. Zna podstawowe funkcjonalności arkuszy kalkulacyjnych (różne rodzaje adresowania, funkcje, formuły, tabele przestawne, Solver). Zna zasady operowania ramkami danych w języku Python.

9. Zna i rozumie podstawowe zagadnienia związane z sieciami komputerowymi (topologie sieci, routing, sieć Internet, protokoły sieciowe, DNS).

10. Zna wybrane nowoczesne usługi Internetu (rozwiązania chmurowe, Google – Drive, Serach, Colaboratory, Wolfram Alpha).

11. Zna podstawowe problemy i pojęcia z dziedzin Big Data, uczenia maszynowego i uczenia głębokiego.

12. Zna podstawowe pojęcia z zakresu bezpieczeństwa danych i kryptografii (dane wrażliwe, szyfrowanie, deszyfrowanie, kryptografia symetryczna i asymetryczna, ogólne informacje o algorytmie RSA).

w zakresie umiejętności:

1. Potrafi dokonywać konwersji liczb między różnymi systemami pozycyjnymi.

2. Potrafi wygenerować kod zwarty za pomocą algorytmu Huffmana.

3. Potrafi zapisać schematy funkcji logicznych przy użyciu bramek logicznych. Potrafi konstruować proste układy logiczne.

4. Potrafi reprezentować dane w strukturalizowany sposób za pomocą formatu XML. Potrafi utworzyć schemat XML przy użyciu DTD i XSD. Potrafi przy pomocy XSLT formatować dokumenty XML. Potrafi za pomocą XPath lokalizować elementy w dokumentach XML. Umie stworzyć strony internetowe przy pomocy HTML i CSS.

5. Potrafi tworzyć tekst matematyczno-informatyczny, rysunki, prezentacje multimedialne w LaTeXu.

6. Potrafi za pomocą arkusza kalkulacyjnego rozwiązywać proste zadania, tworzyć wykresy, reprezentować dane. Potrafi wykorzystać dodatek Solver do rozwiązywania zagadnień programowania liniowego. Potrafi wykorzystywać tabele przestawne do prezentowania i filtrowania danych. Potrafi filtrować i eksplorować ramki danych i tablice Numpy w języku Python.

7. Potrafi opisać algorytm komunikacji w protokołach sieciowych. Potrafi zabezpieczyć hasłem stronę przez protokół HTTP.

8. Potrafi wykorzystać wybrane narzędzia internetowe (Google – Drive, Serach, Colaboratory, Wolfram Alpha).

9. Potrafi szyfrować i deszyfrować wiadomości przy użyciu oprogramowania GPG.

w zakresie kompetencji społecznych:

1. Ma świadomość społecznej odpowiedzialności pracy informatyka.

2. Zna i stosuje ogólne zasady etyki w pracy informatyka.

Treści programowe dla zajęć:

Przedmiot informatyki i jej historia. Kierunki współczesnej informatyki.

Algebry Boole'a i systemy pozycyjne.

Architektura i działanie komputera (Maszyna Turinga, bramki logiczne, procesor).

Reprezentacja liczb w pamięci komputera (znak-moduł, uzupełnieniowa do dwóch, stała- i zmiennoprzecinkowa, standard IEEE 754, pojęcia nadmiaru i niedomiaru, epsilon maszynowy), działania na liczbach zmiennopozycyjnych, algorytm Kahana. Reprezentacja danych graficznych, model RGB, kod ASCII.

Teoria informacji (ilość informacji, entropia informacyjna źródła, redundancja, kod zwarty, algorytm Huffmana).

Systemy przechowywania i prezentacji danych. Języki XML, XSLT, XPath, XML Schema. Formaty JSON, YAML, CSV. HTML i CSS.

Edycja i prezentacja wiedzy z nauk ścisłych za pomocą LaTeX'a i Beamera. Jupyter notebook.

Arkusze kalkulacyjne i ich zastosowania (różne rodzaje adresowania, funkcje, formuły, wykresy, tabele przestawne, programowanie liniowe z zastosowaniem dodatku Solver). Ramki danych (DataFrames).

Sieci komputerowe (topologie sieci, routing, sieć Internet, protokoły sieciowe, DNS).

Nowoczesne usługi Internetu i rozwiązania chmurowe (Google – Drive, Serach, Colaboratory; Wolfram Alpha).

Podstawowe wiadomości dotyczące Big Data, uczenia maszynowego i uczenia głębokiego (rodzaje baz danych, DBMS, uczenie nadzorowane i nienadzorowane, sieci neuronowe).

Bezpieczeństwo danych i kryptografia (podstawowe pojęcia, kryptografia symetryczna i asymetryczna, ogólne informacje o RSA, generowanie kluczy za pomocą GPG).

Etyka pracy informatyka.

Nazwa zajęć: Systemy operacyjne

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Posiada podstawową wiedzę na temat idei oraz algorytmów wykorzystywanych w systemach operacyjnych w przeszłości i obecnie.

2. Zna podstawowe fakty dotyczące budowy wybranych przedstawicieli systemów operacyjnych z rodziny Unix oraz Windows.

w zakresie umiejętności:

1. Umie posługiwać się podstawowymi poleceniami systemów operacyjnych z rodziny Unix oraz Windows, służącymi do wykonywania operacji na plikach, procesach/wątkach i urządzeniach wejścia/wyjścia.

2. Potrafi przeczytać ze zrozumieniem/napisać skrypt w języku powłoki BASH, wykorzystujący podstawowe konstrukcje sterujące tej powłoki dostępne w systemach z rodziny Unix.

3. Potrafi przeczytać ze zrozumieniem/napisać program w języku C zawierający wywołania funkcji systemowych Unixa dotyczących procesów, plików i sygnałów.

4. Umie wykonać podstawowe czynności związane z administrowaniem systemem operacyjnym z rodziny Unix i Windows.

5. Wykorzystuje podstawowe fakty oraz algorytmy związane ze współbieżnym wykonywaniem procesów/wątków.

6. Potrafi zastosować w praktyce podstawowe algorytmy przydziału procesów do procesorów.

w zakresie kompetencji społecznych:

1. Jest gotów/gotowa odnaleźć odpowiednie informacje w dokumentacji technicznej lub zasobach internetowych dla wybranego systemu operacyjnego.

2. Ma świadomość znaczenia i roli systemów operacyjnych w informatyce, rozumie potrzebę dalszego kształcenia w tym zakresie.

Treści programowe dla zajęć:

Wprowadzenie do tematyki SO, Podstawowe definicje i pojęcia: funkcje systemu operacyjnego; unix z perspektywy użytkownika: pojęcia i polecenia dotyczące systemu plików (plik, katalog, montowanie, prawa do plików).

UNIX z perspektywy użytkownika c.d.: pojęcia dotyczące procesów (proces, polecenie ps, sygnały, deskryptory, stdin/out, procesy macierzysty/potomny, pierwszo/drugo-planowy, potoki, łącza) powłoka bash - krótki przegląd.

Instalacja i konfiguracja systemu LINUX, rozruch systemu, podstawowe usługi systemowe i sieciowe.

UNIX z perspektywy programisty: krótki przegląd funkcji systemowych; UNIX z perspektywy administratora: pliki z katalogu /etc, poziomy działania, konfiguracja podstawowych usług.

Historia rozwoju systemów operacyjnych (motywacja idei wieloprogramowości i podziału czasu); klasyfikacja architektur systemów operacyjnych (monolityczne, warstwowe, z mikrojądrem itp); założenia sprzętowe (pojęcia: magistrala, kontroler, port, przerwanie, DMA, ...), architektura x86.

Zarządzanie procesami oraz wątki (diagram stanów procesu, kolejki BKP, planiści, SJF, RR), zagadnienia systemów wieloprocesorowych.

Systemy plików (atrybuty pliku, katalogi, dowiązania twarde i symboliczne + ich implementacje) implementacja systemu plików, przydział listowy, FAT, indeksowy; struktury danych kernela związane z plikami (w tym omówienie idei VFS); omówienie i porównanie konkretnych systemów plików Linuxa i Windows: ext2/3/4, xfs, ntfs, sieciowe systemy plików: CIFS/SMB, SSHFS, NFS ; woluminy linuxowe (lvm2) i windowsowe; dyski RAID.

Obsługa urządzeń wejścia i wyjścia, pliki specjalne blokowe i znakowe, pojęcie sterownika sprzętowego i programowego, zasada działania sys. we/wy; struktura warstwowa sterowników (sieci i sys. plików), struktury danych dotyczące we/wy w linuxie (tablice deskryptorów, tablice plików, tablice i-węzłów) nietypowe operacje we/wy (asynchroniczne, nieblokujące, funkcje select(), itp.) obsługa sieci (interfejsy sieciowe); obsługa urządzeń USB.

Współbieżność, synchronizacja procesów: semafony, sem. binarne, monitory, problemy współbieżności (sekcja krytyczna, producent/konsument, czytelnicy i pisarze, n-filozofów itp.).

Rodzaje pamięci, hierarchia pamięci, cache; ochrona sprzętowa; przegląd składników sys. op.

Zarządzanie pamięcią operacyjną; w tym także: tworzenie programów, bibl. dynamiczne, przydzielanie pamięci procesom i jakie to rodzi problemy; sposoby zarządzania pamięcią: rej. przesunięcia, stronicowanie, segmentacja, rozwiązania stosowane w procesorach x86 i późniejszych, pamięć wirtualna, algorytmy wyszukiwania "ramki ofiary" (FIFO, LRU).

Wirtualizacja: pojęcie i typy wirtualizacji, pojęcie hiperwizora, przegląd oprogramowania służącego do wirtualizacji. Instalacja i konfiguracja maszyny wirtualnej, instalacja systemu LINUX.

Bezpieczeństwo systemu komputerowego - standardy, uprawnienia, zabezpieczanie logowania i kont użytkowników, zabezpieczenia kryptograficzne, bezpieczeństwo sieciowe, moduły autoryzacyjne (np. PAM). Konfiguracja podstawowych usług sieciowych i systemowych.

Omówienie architektury LINUX i Android. Instalacja modułów, instalacja oprogramowania ze źródeł, gcc, make, biblioteki linkowane statycznie i dynamicznie. LINUX - wprowadzenie do funkcji systemowych - zarządzanie procesami, zarządzanie systemem plików. LINUX - moduły jądra, implementacja prostego sterownika.

Omówienie architektury Windows. Windows - zarządzanie systemem, PowerShell/TWAPI. Windows - współbieżność, mutexy, zarządzanie procesami, POSIX.

Obsługa systemu LINUX - Powłoka BASH, obsługa systemu plików, zmienne środowiskowe, stdin/stdout, obsługa procesów, sygnały, łącza, potoki, filtry, skrypty.

Współbieżność, problem sekcji krytycznej, semafony (BACI). Klasyczne problemy współbieżności (BACI). Współbieżność, monitory (BACI). LINUX - współbieżność – semafony. LINUX - współbieżność - IPC, pamięć dzielona, łącza nazwane i nienazwane.

Nazwa zajęć: Inżynieria oprogramowania

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Zna problemy pojawiająca się podczas tworzenia oprogramowania.
2. Zna różne podejścia do tworzenia oprogramowania i związane z nimi środowiska projektowo-programistyczne.

3. Zna metody i narzędzia pozwalające na zbieranie wymagań oraz projektowanie oprogramowania zgodnie z metodyką obiektową.
4. Zna metody i narzędzia wykorzystywane w procesie budowy oprogramowania.
5. Zna metody, techniki i narzędzia wykorzystywane w procesie projektowania systemów informatycznych.
6. Zna techniki prowadzące do otrzymania oprogramowania wysokiej jakości.
7. Zna wybrane wzorce projektowe.

w zakresie umiejętności:

1. Potrafi stworzyć projekt prostego systemu informatycznego w wybranym narzędziu (np. w języku UML).
2. Potrafi oceniać przydatność różnych podejść do rozwoju oprogramowania i związanych z nimi środowisk programistycznych do rozwiązywania problemów pojawiających na różnych etapach budowy oprogramowania.
3. Potrafi projektować oprogramowanie zgodnie z metodyką obiektową.
4. Potrafi oceniać przydatność metod i narzędzi wykorzystywanych w procesie budowy oprogramowania oraz wybrać i zastosować właściwą metodę i narzędzia do rozwiązania typowych zadań związanych z rozwojem oprogramowania.
5. Potrafi zgodnie z zadaną specyfikacją zaprojektować oraz zbudować prosty system informatyczny, używając właściwych metod, technik i narzędzi.
6. Potrafi stosować techniki prowadzące do otrzymania oprogramowania wysokiej jakości.
7. Potrafi posługiwać się wzorcami projektowymi w procesie budowy oprogramowania.

w zakresie kompetencji społecznych:

1. Jest świadomy/a roli inżynierii oprogramowania w procesie budowy oprogramowania wysokiej jakości.
2. Rozumie proces ewolucji narzędzi i metod inżynierii oprogramowania.

Treści programowe dla zajęć:

Czym jest oprogramowanie, problemy pojawiające się przy rozwoju oprogramowania.

Cykl życia oprogramowania z uwzględnieniem różnych modeli. Wyróżnienie składowych procesu rozwoju oprogramowania. Analiza modeli cyklu życia oprogramowania na przykładzie metodyki hybrydowej (Rational Unified Process) oraz zwinnej (SCRUM).

Analiza wymagań: wymagania funkcjonalne i niefunkcjonalne, specyfikacja wymagań, proces pozyskiwania wymagań (scenariusze, przypadki użycia, user stories), zarządzanie wymaganiami. Zbieranie wymagań przy pomocy przypadków użycia oraz user stories

Projektowanie systemu informatycznego. UML w projektowaniu. Projektowanie interakcji (diagramy interakcji), modelowanie struktury systemu (model logiczny, diagram klas, diagram wdrożenia), modelowania zachowania (diagram stanów, diagram aktywności).

Architektura systemu informatycznego, wzorce architektoniczne (klient-serwer, aplikacje wielowarstwowe), wzorce projektowe (MVC, ORM). Wpływ architektury systemu na tworzone oprogramowanie (przykłady).

Analiza i projektowanie obiektowe: model wiedzy dziedzinowej, identyfikacja klas, przypisywanie obiektom odpowiedzialności, wielokrotne użycie.

Testowanie oprogramowania. Rodzaje testów. Testy prowadzone przez programistę (testy jednostkowe, testy modułowe, testy integracyjne). Test-driven development. Ciągła integracja. Scenariusze testowe i testy akceptacyjne. Testy użyteczności.

Utrzymanie i pielęgnacja oprogramowania.

Zarządzanie projektem informatycznym: zarządzanie zespołem, planowanie prac, zarządzanie jakością, zarządzanie wersjami.

Wykorzystanie narzędzi wspierających rozwój oprogramowania (repozytorium kodu, issue tracker, bug tracker).

Realizacja grupowego mini-projektu w wybranej metodyce. Realizacja w dużym zespole z podziałem na role.

Nazwa zajęć: **Algorytmy i struktury danych**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna i rozumie podstawowe konstrukcje algorytmiczne
2. zna i rozumie podstawowe proste i złożone struktury danych, w tym struktury dynamiczne
3. zna i rozumie podstawowe techniki projektowania algorytmów

4. zna i rozumie pojęcia poprawności oraz złożoności czasowej i pamięciowej algorytmu
5. zna i rozumie podstawowe algorytmy rozwiązujące klasyczne problemy informatyczne
6. zna i rozumie pojęcie problemu NP-zupełnego i NP-trudnego

w zakresie umiejętności:

1. potrafi zapisać podstawowe konstrukcje algorytmiczne w pseudokodzie i wybranym języku programowania
2. potrafi opisać podstawowe proste i złożone struktury danych, w tym struktury dynamiczne, w pseudokodzie i wybranym języku programowania
3. potrafi zastosować podstawowe techniki projektowania algorytmów do rozwiązywania wybranych problemów informatycznych
4. potrafi ocenić poprawność oraz złożoność czasową i pamięciową poznanych algorytmów
5. potrafi napisać w wybranym języku programowania implementacje poznanych algorytmów

w zakresie kompetencji społecznych:

1. jest gotów/gotowa do zrozumienia znaczenia algorytmiki w informatyce i kształtowaniu życia społecznego
2. jest gotów/gotowa do dalszego samodzielnego kształcenia algorytmicznego

Treści programowe dla zajęć:

Język algorytmiczny: pojęcie zmiennej, instrukcja przypisania, instrukcje warunkowe, iteracje, operatory.

Pojęcie struktury tablicowej: przykłady i implementacje prostych algorytmów dla problemów algorytmicznych na tablicach 1 i 2-wymiarowych, wyszukiwanie liniowe i binarne.

Pojęcie procedury: deklaracja, parametry formalne, wywołanie, przykłady prostych procedur i funkcji.

Rekurencja i programowanie dynamiczne: pojęcie rekurencji, przykłady procedur rekurencyjnych, własność optymalnej podstruktury, programowanie dynamiczne, przykłady algorytmów wykorzystujących programowanie dynamiczne.

Analiza algorytmów: notacja asymptotyczna, złożoność czasowa i pamięciowa algorytmu, złożoność optymistyczna, pesymistyczna i średnia, twierdzenie o rekurencji uniwersalnej, klasy złożoności, dowodzenie poprawności konstrukcji wybranych algorytmów.

Algorytmy sortowania: sortowanie przez wstawianie, bąbelkowe, przez scalanie, szybkie, przez zliczanie.

Stosy, kolejki, listy: pojęcie zbioru dynamicznego, tablicowa implementacja stosu i operacje na stosie, tablicowa implementacja kolejki i operacje kolejkowe, listy z dowiązaniem, operacje na listach, listy z wartownikiem.

Pojęcia teorii grafów: graf prosty, drzewa i ich podstawowe własności, drzewa ukorzenione, ważone grafy skierowane, reprezentacja grafów w komputerze.

Kopce: podstawowe operacje na kopcach binarnych, sortowanie przez kopcowanie, implementacja kolejki priorytetowej.

Drzewa wyszukiwań binarnych: podstawowe własności, operacje słownikowe, metody przechodzenia drzewa BST.

Algorytmy grafowe: przeszukiwanie grafu wszerz i w głąb, znajdowanie najkrótszych ścieżek.

Metoda z nawrotami i metoda zachłanna: problem n hetmanów, problem plecakowy, problem wyboru zajęć.

Problemy trudne obliczeniowo: klasy P i NP, problemy NP-zupełne i NP-trudne.

Nazwa zajęć: Modelowanie procesorów kwantowych

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna matematyczny opis kwantowego oscylatora harmonicznego.
2. zna budowę i matematyczny opis transmonu.
3. zna źródła dekoherencji w układach nadprzewodzących.
4. zna zjawiska fizyczne pozwalające traktować najniższe stany transmonu jako kubit albo jako kutrit.
5. zna fizyczne mechanizmy pozwalające na kontrolowanie stanu nadprzewodzącego kubitu i kutritu.
6. zna sposoby sprzęgania transmonów.
7. zna podstawowe sposoby realizacji dwukubitowych bramek kwantowych na transmonach.

w zakresie umiejętności:

1. potrafi zmierzyć parametry fizyczne transmonu takie jak: częstość, stała tłumienia, przesunięcie dyspersyjne, przesunięcie Starka i czas koherencji.
2. potrafi zmierzyć stałą tłumienia nadprzewodzącego rezonatora oraz stałą sprzężenia rezonatora z nadprzewodzącym kubitami.

3. potrafi zaprojektować układy nadprzewodzące i obliczyć szacunkowe wartości kluczowych parametrów tych układów, takich jak częstości kubitów, stałe tłumienia, czy stałe sprzężenia.
4. potrafi wyznaczyć hamiltonian zaprojektowanego układu nadprzewodzącego.

Treści programowe dla zajęć:

Kwantowy rezonator LC i złącze Josephsona.

Transmon jako nadprzewodzący kubit oraz różne jego warianty.

Kontrolowanie stanu transmonu za pomocą wzbudzenia.

Transmon jako kutrit; pomiar oraz kontrolowanie stanu kutritu.

Metody pomiaru parametrów transmonu takich jak częstość, stała tłumienia, przesunięcie dyspersyjne, przesunięcie Starka oraz czas koherencji.

Pojemnościowe i indukcyjne sprzęganie transmonów.

Klasa QComponent (standardowe komponenty układów nadprzewodzących) oraz klasa QDesign (układy jednopłaszczyznowe i wielopłaszczyznowe).

Realizacja w układzie nadprzewodzącym jednokubitowych i dwukubitowych bramek kwantowych.

Nazwa zajęć: Analiza matematyczna z zastosowaniami 1

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. rozumie pojęcie sumy szeregu i potrafi badać zbieżność szeregów liczbowych i potęgowych. Zna podstawowe kryteria zbieżności i ich ograniczenia w obliczeniach numerycznych.
2. rozumie potrzebę badań ciągłości funkcji i granic funkcji w punkcie. Zna i rozumie zastosowania w praktyce informatycznej własności funkcji ciągłych (np. własność Darboux).
3. zna metody rozwijania funkcji w szeregi potęgowe i rozumie ich zastosowania.
4. zna i rozumie zastosowania pojęcia pochodnej, oblicza pochodne i zna podstawowe zastosowania w informatyce, np. w analizie błędów.
5. zna wzór Taylora i rozumie jego zastosowania do lokalnego przybliżania funkcji wielomianami.
6. zna pojęcie całki i całki Riemanna i rozumie jak je użyć w odpowiednich sytuacjach. Zna podstawowe zastosowania takich całek w informatyce.
7. zna i rozumie stosowanie podstawowych metod całkowania i wie jak je wykorzystać do obliczeń prostych całek.
8. zna zastosowania całek niewłaściwych i splotu w informatyce.

w zakresie umiejętności:

1. potrafi prowadzić obliczenia analityczne i obliczenia numeryczne całek.
2. potrafi zastosować całki niewłaściwe w pewnych zastosowaniach informatycznych i ma umiejętność dostrzegania ich w takim kontekście.
3. potrafi przeanalizować dostępne materiały w celu dostrzeżenia matematycznych podstaw informatyki.
4. ma umiejętność wykorzystania własności funkcji ciągłych.
5. ma umiejętność wykorzystania pochodnej do badania monotoniczności funkcji oraz obliczania granic.
6. potrafi zastosować wiedzę matematyczną do analizy problemów informatycznych.

w zakresie kompetencji społecznych:

1. jest gotów/gotowa do wskazywania matematycznych podstaw algorytmów informatycznych i opracowywania takich materiałów.

Treści programowe dla zajęć:

Powtórzenie informacji o ciągach i funkcjach elementarnych. Ciągi i funkcje w informatyce. Elementy analizy błędów.

Szeregi liczbowe. Suma szeregu. Zbieżność i bezwzględna zbieżność szeregu. Kryteria zbieżności. Wprowadzenie do obliczeń sum szeregów, problem przybliżonego obliczania sumy szeregu. Obliczanie sum i badanie ich zbieżności, kryterium Leibnitz'a i reszty szeregu. Oszacowania wartości za pomocą szeregów, ograniczenia badań zbieżności na komputerze.

Granica i ciągłość funkcji jednej zmiennej rzeczywistej. Punkt skupienia zbioru. Granica funkcji w punkcie. Ciągłość funkcji (np. spline) i ciągłość jednostajna funkcji. Własność Darboux. Twierdzenie Weierstrassa o kresach. Ciągłość funkcji w przestrzeniach metrycznych – podstawy.

Rachunek różniczkowy funkcji jednej zmiennej rzeczywistej. Pochodna i jej sens geometryczny. Interpretacja geometryczna pochodnej. Liniowe przybliżanie funkcji (lokalne). Rola wzoru Taylora w szacowaniu błędów. Wzory Taylora dla wybranych funkcji. Przybliżanie lokalne funkcji wielomianami.

Rozwiązywanie równań nieliniowych metodami iteracyjnymi (metody bisekcji, stycznych, siecznych i inne – podstawy matematyczne). Szybkość zbieżności, obliczenia dla pierwiastków zespolonych.

Rachunek całkowy funkcji jednej zmiennej. Funkcja pierwotna i całka nieoznaczona. Podstawowe metody całkowania. Całka Riemanna (konstrukcje Darboux i Riemanna).

Całkowanie numeryczne. Metody prostokątów, trapezów. Podstawowe informacje o aproksymacji wielomianowej. Metoda Simpsona, kwadratury, rząd metody, oszacowanie błędów. Inne metody całkowania.

Całka niewłaściwa i jej zastosowania w informatyce. Splot i jego wersja dyskretna oraz przykłady zastosowań (np. splot sygnałów czy w grafice komputerowej).

Nazwa zajęć: Pracownia programowania

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Zna i rozumie architekturę nowoczesnych aplikacji internetowych.
2. Zna i rozumie protokoły i formaty wymiany danych w sieci Internet.
3. Zna i rozumie sposoby mapowania i przechowywania danych w bazach danych.
4. Zna i rozumie rolę i zadania aplikacji backendowej.
5. Zna i rozumie rolę i zadania aplikacji frontendowej.

w zakresie umiejętności:

1. Potrafi zaprojektować i zaimplementować serwis internetowy oparty na niezależnej aplikacji backendowej, frontendowej i bazie danych.
2. Potrafi implementować przepływ danych w aplikacji.
3. Potrafi implementować programy w oparciu o nowoczesne narzędzia i IDE.

w zakresie kompetencji społecznych:

1. Rozumie ewolucję projektowania aplikacji i zmiany technologiczne w procesie ich projektowania w ostatnich latach.

Treści programowe dla zajęć:

Przygotowanie środowiska programistycznego, wybór edytora, repozytorium kodu, sposobu budowania aplikacji oraz logowania komunikatów.

Debugowanie i testowanie kodu w wybranym środowisku programistycznym.

Przetwarzanie strumieniowe, paradygmat programowania funkcyjnego, kolekcje.

Serializacja i deserializacja danych w postaci JSON i XML.

Mapowanie relacyjno - obiektowe (ORM), praca na modelu obiektywnym, zapytania SQL w modelu ORM.

Tworzenie i praca z REST API. Postman jako narzędzie komunikacji z restowym API.

Platforma programistyczna do tworzenia serwisu opartego na REST API z przykładową implementacją. Serwery aplikacji, osadzanie projektu na serwerze aplikacji, konfiguracja serwera, monitorowanie i debugowanie osadzonej aplikacji.

Tworzenie serwisu WWW dla serwera korzystającego z REST API, przykładowa implementacja.

Integracja wielu serwisów – autoryzacja za pomocą OAuth i zewnętrznego API.

Integracja modułów projektu.

Konsultacje projektowe.

Nazwa zajęć: Platforma chmurowa Amazon Web Services

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. rozumie koncepcję chmury obliczeniowej i jej znaczenie w informatyce.
2. rozumie podstawowe aspekty bezpieczeństwa i zgodności platformy AWS oraz model współdzielonego bezpieczeństwa.
3. rozumie podstawowe aspekty wdrażania i działania usług w chmurze AWS.
4. rozumie jak przenieść zasoby z infrastruktury lokalnej do infrastruktury chmurowej.
5. rozumie czym jest chmura AWS i jaka jest jej podstawowa globalna infrastruktura.

w zakresie umiejętności:

1. potrafi zdefiniować modele rozliczeń, zarządzania kontem i naliczania kosztów chmury obliczeniowej.
2. potrafi zidentyfikować źródła dokumentacji lub pomocy technicznej.
3. potrafi rozróżnić infrastrukturę lokalną i chmurową.
4. potrafi zastosować kluczowe usługi na platformie AWS w typowych obszarach zastosowania.
5. potrafi uruchomić oraz skonfigurować podstawowe usługi chmury AWS.

Treści programowe dla zajęć:

Koncepcja i struktura chmury obliczeniowej

Podstawowe usługi w chmurze obliczeniowej AWS

Zabezpieczanie i monitorowanie w chmurze

Zarządzanie danymi

Zarządzanie i optymalizacja chmury obliczeniowej

Nazwa zajęć: Algebra liniowa z zastosowaniami 2. Podstawy obliczeń kwantowych
Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka
w zakresie wiedzy:

1. zna pojęcia przestrzeni i podprzestrzeni wektorowej, podprzestrzeni liniowej rozpiętej przez układ wektorów (powłoki liniowej układu wektorów), zna pojęcie bazy przestrzeni liniowej oraz macierzy przejścia z bazy do bazy, zna zastosowania tych pojęć do opisu stanu układów fizycznych.
2. zna pojęcie iloczynu tensorowego przestrzeni liniowych, iloczynu Kroneckera operatorów liniowych i notację Diraca, zna przykłady zastosowań tych pojęć do opisu stanu złożonych układów fizycznych.
3. zna pojęcia związane z zagadnieniem własnym i własności macierzy podobnych, zna własności spektralne wybranych klas macierzy, zna przykłady zastosowań tych pojęć w opisie stanu i ewolucji układów fizycznych.
4. zna metody pozwalające na przybliżone rozwiązywanie wybranych zagadnień matematycznych i rozumie ograniczenia wynikające z ich stosowania i rozumie znaczenie fizyki układu realizującego daną metodę.
5. zna i rozumie twierdzenia algebry liniowej (m.in. Cayleya-Hamiltona, Jordana) oraz przykłady zastosowania w fizyce.
6. rozumie potrzebę wyznaczania rozkładów macierzy (w tym rozkładów trójkątnych i SVD - według wartości szczególnych) i rozumie powiązania z opisem układów fizycznych.
7. posiada wiedzę dotyczącą podstaw przestrzeni euklidesowych.
8. zna definicję i własności norm wektorowych i macierzowych i potrafi wskazać ich zastosowanie w fizyce.
9. zna podstawy teorii form kwadratowych oraz przykłady wielkości fizycznych opisywanych jako normy kwadratowe.

w zakresie umiejętności:

1. stosuje poznane na zajęciach metody numeryczne do rozwiązywania zagadnień sformułowanych w języku algebry liniowej.
2. potrafi formułować i rozwiązywać zagadnienie własne, zna własności spektralne wybranych klas macierzy. Potrafi podać przykłady zagadnienia własnego w fizyce.
3. potrafi sprawdzić, czy wektor należy do danej powłoki liniowej i zweryfikować, czy dane powłoki liniowe są równe; umie obliczyć jądro i obraz przekształcenia liniowego oraz wyznaczyć ich bazę i wymiar. Potrafi podać przykłady przekształcenia liniowego w fizyce.
4. potrafi zweryfikować określoność form kwadratowych i wyznaczyć ich postaci kanoniczne. Potrafi podać przykłady form kwadratowych w fizyce.
5. potrafi sprawdzić, czy przekształcenie liniowe jest ortogonalne, czy macierz jest ortogonalna lub unitarna, podać własności macierzy ortogonalnej lub unitarnej. Potrafi zweryfikować czy przekształcenie wektora stanu fizycznego nie łamie prawa ruchu.
6. potrafi sprawdzić, czy odwzorowanie jest normą wektorową; wylicza ogólnie znane normy wektorowe i macierzowe (dla małych wymiarów); stosuje normy wektorowe lub macierzowe do rozwiązywania wybranych zagadnień matematycznych i fizycznych.
7. rozwiązuje układy liniowych równań algebraicznych opisane macierzą wysokiego stopnia.
8. potrafi odczytać informację zawartą w rozkładzie SVD (według wartości szczególnych) i zrekonstruować macierz wyjściową z użyciem minimalnej wymaganej liczby elementów macierzy rozkładu, podać zastosowania rozkładu SVD w fizyce, matematyce i informatyce

Treści programowe dla zajęć:

Powłoka liniowa układu wektorów; twierdzenia związane z powłoką liniową układu wektorów; przykłady powłok liniowych. Macierz przekształcenia liniowego w bazach. Współrzędne obrazu wektora poprzez przekształcenie liniowe a macierz przekształcenia liniowego. Związek macierzy przekształcenia liniowego z macierzą przejścia z bazy do bazy. Definicja jądra i obrazu przekształcenia liniowego, wyznaczanie bazy jądra i obrazu przekształcenia liniowego.

Wartości i wektory własne. Wielomian i równanie charakterystyczne. Krotność algebraiczna i geometryczna wartości własnej. Przestrzeń własna. Promień spektralny. Zależności pomiędzy wyznacznikiem i śladem macierzy, a wartościami własnymi. Własności macierzy podobnych w kontekście zagadnienia własnego. Twierdzenie Cayleya-Hamiltona. Diagonalizacja macierzy. Twierdzenie Jordana. Wielomian minimalny i dzielniki elementarne.

Metody iteracyjne obliczania wartości własnych.

Formy kwadratowe, badanie określoności. (+rozkład Cholesky'ego-Banachiewicza jako przykład poprawy algorytmu rozkładu macierzy na czynniki trójkątne w związku z jej dodatnią określonością.)

Przestrzenie liniowe z iloczynem skalarnym. Przekształcenia ortogonalne, unitarne. Normy wektorowe. Algorytm ortogonalizacji Grama-Schmidta. Normy macierzowe. Wskaźnik uwarunkowania zadania na przykładzie wskaźnika uwarunkowania macierzy.

Metody iteracyjne rozwiązywania układów liniowych równań algebraicznych.

Wartości szczególne macierzy, rozkład SVD (według wartości szczególnych) i jego zastosowania.

Iloczyn tensorowy przestrzeni liniowych, iloczyn Kroneckera transformacji liniowych, notacja Diraca dla iloczynu tensorowego.

Nazwa zajęć: **Elektryczność i magnetyzm**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. posiada wiedzę z zakresu elektryczności i magnetyzmu, w szczególności zna podstawowe pojęcia oraz prawa elektromagnetyzmu (Gausa, Ampere'a, Faradaya, równania Maxwella).

2. zna elektryczne i magnetyczne własności materii.

3. rozumie i wyjaśnia wybrane zjawiska z dziedziny elektryczności i magnetyzmu.

w zakresie umiejętności:

1. Potrafi samodzielnie rozwiązywać problemy/zadania z zakresu elektromagnetyzmu w oparciu o nabytą wiedzę z wykładu i opanowany aparat matematyczny (wynikający z programu studiów).

2. Rozumie i potrafi opisać wybrane zjawiska z zakresu elektromagnetyzmu.

Treści programowe dla zajęć:

ELEKTROSTATYKA: ładunek elektryczny, kwantyzacja ładunku, zasada zachowania ładunku elektrycznego; prawo Coulomba, obliczanie sił działających w układach ładunków punktowych; pojęcie pola elektrycznego (obliczanie natężenia pól elektrycznych dla układu ładunków punktowych), pole elektryczne dipolaenergia układu ładunków, ruch ładunku w polu elektrycznym, dipol w polu elektrycznym; Strumień pola wektorowego, rozkłady ciągłe ładunków – gęstość liniowa, powierzchniowa i objętościowa, natężenie pola dla różnych ciągłych rozkładów ładunku; prawo Gausa w postaci całkowitej i różniczkowej; Potencjał pola elektrycznego, powierzchnie ekwipotencjalne, natężenie pola a potencjał, napięcie, energia pola elektrycznego, praca i energia w polu elektrycznym; kondensatory i pojemność, dielektryki w polu elektrycznym

PRĄD ELEKTRYCZNY: natężenie i gęstość prądu elektrycznego, prądy stacjonarne i prawo zachowania ładunku; prawo Ohma, oporność właściwa materiałów i opór elektryczny, zależność temperaturowa dla oporu, praca i moc prądu elektrycznego, obwody elektryczne, źródła SEM, uogólnione prawo Ohma, I i II prawo Kirchhoffa, obwody RC; przewodnictwo elektryczne w metalach, elektrolitach i gazach, siła elektromotoryczna, obwody prądu stałego, prawa Kirchhoffa, rozpraszanie energii przy przepływie prądu

POLE MAGNETYCZNE: definicja pola magnetycznego, strumień pola magnetycznego, ruch ładunku elektrycznego w polu magnetycznym, siła Lorentza, wyznaczenie stosunku e/m, efekt Halla, cyklotrony i synchrotrony; siła działająca w polu magnetycznym na przewodnik z prądem, pole magnetyczne wytwarzane przez przewodnik z prądem, prawo Ampere'a, prawo Biot-Savarta, siła działająca między przewodnikami z prądem, ramka z prądem w polu magnetycznym – moment siły, dipolowy moment magnetyczny, pole magnetyczne solenoidu, pola ładunków w ruchu, pola wirowe, twierdzenie Stokes'a, potencjał wektorowy

INDUKCJA ELEKTROMAGNETYCZNA: prawo Faradaya, reguła Lenza, indukcyjność, samoindukcja, indukcja wzajemna; prądy wirowe, siła elektromotoryczna w pręcie poruszającym się w jednorodnym polu magnetycznym, generator prądu zmiennego, transformator, energia i gęstość energii pola magnetycznego

RÓWNANIA MAXWELLA: Prawo Gausa dla pola magnetycznego, prąd przesunięcia, równania Maxwella w postaci całkowitej i różniczkowej, fale elektromagnetyczne i transport energii, doświadczenie Hertza.

OBWODY PRĄDU ZMIENNEGO: drgania elektromagnetyczne w obwodzie LC; drgania obwodu RLC, tłumienie w obwodzie RLC, drgania wymuszone i rezonans w obwodach RLC; obwody prądu zmiennego, moc i energia w obwodach prądu zmiennego, transformatory

POLA ELEKTRYCZNE I MAGNETYCZNE W MATERII: polaryzacja dielektryka, podatność elektryczna, polaryzowalność atomów i molekuł, piezoelektryczność i elektrostrykcja, ferroelektryki; pole magnetyczne pętli z prądem, prądy elektryczne w atomie i orbitalny moment magnetyczny, spin elektronu i spinowy moment magnetyczny, doświadczenie Einsteina-de Hassa; magnetyzacja, podatność magnetyczna, paramagnetyzm, diamagnetyzm, ferromagnetyzm, histereza magnetyczna, magnetyzm ziemski

Nazwa zajęć: **Sztuczna inteligencja**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Rozumie pojęcie sztuczna inteligencja oraz potrafi wskazać jej działy.
2. Zna historię sztucznej inteligencji.
3. Zna założenia podejścia agentowego w definiowaniu problemów sztucznej inteligencji
4. Zna metody rozwiązywania problemów poprzez przeszukiwanie przestrzeni stanów.
5. Rozumie zasady prowadzenia wnioskowania w języku logiki.
6. Zna metody wnioskowania nieprecyzyjnego.
7. Zna podstawowe narzędzia i techniki reprezentacji wiedzy.
8. Zna metody uczenia maszynowego.
9. Zna sposoby wykorzystania sztucznych sieci neuronowych.
10. Zna obszary zastosowania sztucznej inteligencji.

w zakresie umiejętności:

1. Potrafi zastosować metody rozwiązywania problemów poprzez przeszukiwanie przestrzeni stanów.
2. Umie zastosować zasady prowadzenia wnioskowania w języku logiki.
3. Potrafi wykorzystać metody wnioskowania nieprecyzyjnego.
4. Umie reprezentować wiedzę z wykorzystaniem podstawowych narzędzia i techniki reprezentacji wiedzy.
5. Umie zastosować metody uczenia maszynowego.
6. Umie zastosować sztuczne sieci neuronowe.

w zakresie kompetencji społecznych:

1. Rozumie odpowiedzialność twórców systemów sztucznej inteligencji.
2. Rozumie problemy etyczne sztucznej inteligencji.
3. Rozumie proces ewolucji narzędzi i metod sztucznej inteligencji
4. Jest gotów do określania roli sztucznej inteligencji w procesach gospodarczych oraz życiu społecznym.

Treści programowe dla zajęć:

Określenie czym jest sztuczna inteligencja. Różne sposoby jej definiowania. Rys historyczny. Dziedziny sztucznej inteligencji.

Podejście agentowe w definiowaniu problemów sztucznej inteligencji. Implementacja środowiska agentowego.

Rozwiązywanie problemów poprzez przeszukiwanie przestrzeni stanów. Reprezentacja przestrzeni stanów. Niepoinformowane i poinformowane strategie przeszukiwania. Heurystyki. Implementacja przestrzeni stanów i strategii przeszukiwania przy rozwiązywaniu problemów.

Wnioskowanie w języku logiki. Wykorzystanie w sztucznej inteligencji klasycznego rachunku zdań, logiki pierwszego rzędu, logiki rozmytej. Wnioskowanie w systemach sztucznej inteligencji.

Wnioskowanie nieprecyzyjne (reguły rozmyte, przybliżone, Baysowskie).

Reprezentacja wiedzy. Wybrane formalizmy reprezentacji wiedzy i związane z nimi struktury danych.

Uczenie maszynowe nadzorowane i nienadzorowane. Klasyfikacja i predykcja. Miary w uczeniu maszynowym. Wybrane metody uczenia maszynowego: drzewa decyzyjne, regresja liniowa, regresja logistyczna.

Sztuczne sieci neuronowe. Głębokie sieci neuronowe. Wykorzystanie sieci neuronowych w systemach sztucznej inteligencji .

Wybrane obszary zastosowań sztucznej inteligencji.

Odpowiedzialności twórców systemów sztucznej inteligencji, problemy etyczne, przyszłość sztucznej inteligencji.

Nazwa zajęć: Seminarium dyplomowe - zdalne

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. Potrafi przedstawić ustnie kilkudziesięciominutową prezentację na zadany temat na odpowiednim poziomie merytorycznym
2. Potrafi zredagować szczegółowy konspekt prezentacji oraz przygotowywanej pracy inżynierskiej.
3. Potrafi wyszukiwać materiały w bazach danych i zasobach bibliotecznych niezbędne do przygotowania prezentacji oraz pracy inżynierskiej.
4. Potrafi dokonać właściwej i krytycznej oceny oraz selekcji materiału zebranego do prezentacji i pracy inżynierskiej

w zakresie kompetencji społecznych:

1. Rozumie konieczność systematycznej pracy, stałego uzupełniania i aktualizowania posiadanej wiedzy

2. Potrafi formułować i objaśniać najważniejsze pojęcia i twierdzenia z działów informatyki i fizyki bezpośrednio związanych z tematyką seminarium oraz pracą inżynierską
3. Potrafi sprawnie posługiwać się językiem technicznym

Treści programowe dla zajęć:

Kształtowanie umiejętności prezentowania informacji na temat różnych aspektów inżynierii oprogramowania

Raportowanie i prezentowanie wykonania prac związanych z tematyką pracy inżynierskiej oraz dostarczenie końcowej wersji samodzielnie przygotowanej pracy inżynierskiej o charakterze systematyzującym i projektowym lub aplikacyjnym.

Nazwa zajęć: **Języki formalne i złożoność obliczeniowa**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie podstawowe pojęcia teorii języków formalnych.
2. zna formalne modele obliczeń (automaty skończenie stanowe, automaty ze stosem, maszyny Turinga) i ich własności.
3. rozumie pojęcie gramatyki generatywnej oraz zna podstawowe rodzaje gramatyk generatywnych, ich własności i ich związki z automatami. Zna pojęcie wyrażenia regularnego, zna i rozumie jego związek z językami regularnymi.
4. zna i rozumie pojęcie problemu decyzyjnego oraz rozstrzygalnego problemu decyzyjnego. Zna podstawowe algorytmy rozstrzygania dla języków formalnych.
5. zna modele maszyn Turinga równoważne z modelem podstawowym i rozumie na czym polegają różnice między modelami. Zna i rozumie tezę Churcha.
6. zna miary złożoności obliczeniowej, zna podstawowe klasy złożoności obliczeniowej i rozumie relacje między nimi.
7. zna i rozumie pojęcie problemów trudnych i zupełnych dla danej klasy problemów. Rozumie konsekwencje rozstrzygnięcia hipotezy $P=NP$.
8. zna zalety i możliwości oraz ograniczenia w zastosowaniach automatów w pracy programisty.

w zakresie umiejętności:

1. umie wykonywać operacje na językach formalnych.
2. potrafi konstruować automaty i gramatyki oraz znajdować wyrażenia regularne dla zadanych języków.
3. umie wykonać determinizację automatu skończenie stanowego.
4. potrafi znajdować postaci normalne dla zadanych gramatyk i umie je wykorzystać.
5. umie zastosować lematy o pompowaniu dla języków regularnych i bezkontekstowych.
6. umie zastosować algorytm CYK dla rozstrzygnięcia problemu słowa dla języków bezkontekstowych.
7. umie pisać programy na maszynie Turinga akceptujące/rozstrzygające języki i liczące funkcje, oraz oszacować złożoność programów.
8. potrafi zanalizować redukcje problemów decyzyjnych.
9. umie stosować automaty w pracy programistycznej.

Treści programowe dla zajęć:

Pojęcia podstawowe: alfabet, słowo, język. Operacje na językach.

Automaty skończenie stanowe (wersja deterministyczna, niedeterministyczna i z pustym przejściem), język akceptowany przez automat skończenie stanowy.

Determinizacja automatu skończenie stanowego i jej koszt. Informacja o automacie minimalnym.

Lemat o pompowaniu dla języków regularnych i jego zastosowanie.

Wyrażenia regularne i języki oznaczane przez te wyrażenia. Twierdzenie Kleene'ego.

Gramatyki typu 3 (regularne) i ich równoważność z automatami skończenie stanowymi.

Gramatyki bezkontekstowe i automaty ze stosem. Postaci normalne dla gramatyk bezkontekstowych.

Lemat o pompowaniu dla języków bezkontekstowych. Języki poza klasą CF.

Pojęcie (rozstrzygalnego) problemu decyzyjnego. Problemy rozstrzygalne i nierozstrzygalne dla języków regularnych i bezkontekstowych. Algorytm CYK.

Maszyna Turinga - model podstawowy i modele równoważne. Akceptowanie i rozstrzyganie języków przez maszynę Turinga. Języki akceptowane i rozstrzygane przez maszynę Turinga. Problemy decyzyjne jako języki.

Maszyna Turinga jako model obliczeń numerycznych - informacja o tezie Churcha.

Złożoność czasowa i pamięciowa maszyny Turinga. Klasy czasowej i pamięciowej złożoności obliczeniowej.

Redukowalność, NP-trudność i NP-zupełność. Twierdzenie Cooka-Levina i jego znaczenie. Przykłady problemów NP-zupełnych.

Hipoteza P=NP i konsekwencje jej prawdziwości/fałszywości. Twierdzenie Savitcha.
Wybrane zagadnienia z programowania parserów
Automat skończenie stanowy w pracy programisty - struktura w kodzie/pamięci

Nazwa zajęć: **Architektura internetu**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie działanie sieci komputerowych w oparciu o warstwowe modele architektury Internetu i znajomość protokołów komunikacyjnych.
2. zna i rozumie najważniejsze instytucje odpowiedzialne za standardy sieciowe i charakter udostępnianych przez nie dokumentów.

w zakresie umiejętności:

1. projektuje i buduje sieci lokalne z dostępem do Internetu w standardzie Ethernet – przewodowe i bezprzewodowe.
2. konfiguruje stacje robocze i urządzenia sieciowe oraz analizuje ruch sieciowy.
3. instaluje i konfiguruje podstawowe serwisy internetowe, takie jak: routing, DNS, DHCP, FTP, HTTP i objaśnia zasady ich działania.
4. korzysta z różnych źródeł: systemowych, literaturowych i internetowych, w tym także anglojęzycznych, do samodzielnego rozwiązywania problemów konfiguracji administracji lokalnych sieci komputerowych.

Treści programowe dla zajęć:

Modele warstwowe sieci komputerowych (OSI, TCP/IP) Geneza Internetu i standardy jego architektury. Instytucje odpowiedzialne za standardy internetowe i ich zasoby.

Architektura i elementy sieci lokalnej opartej na standardzie Ethernet.

Metody transmisji danych – analogowej i cyfrowej. Media transmisyjne. Sieci przewodowe i bezprzewodowe. Rodzaje okablowania.

Fizyczna i logiczna struktura sieci. Topologie, metody dostępu do nośnika. Urządzenia sieciowe i łączenie sieci.

Podstawowe protokoły komunikacyjne warstwy internetowej i ich współdziałanie.

Zasady adresacji IPv4 i IPv6. Mapowanie adresów fizycznych i logicznych. Trasowanie.

Sterowanie transmisją – protokoły: TCP i UDP. Transmisja grupowa (strumieniowa).

Podstawowe usługi sieciowe – DNS, DHCP, TELNET (SSH), FTP, SMTP, MIME, HTTP.

Zarządzanie siecią i bezpieczeństwo sieciowe.

Nazwa zajęć: **Kwantowe uczenie maszynowe**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna formalizm matematyczny i pojęciowy do opisu wybranych algorytmów i modeli klasycznego uczenia maszynowego.
2. zna formalizm matematyczny i pojęciowy do opisu i wyjaśnienia wybranych algorytmów kwantowych stosowanych w uczeniu maszynowym.
3. zna formalizm matematyczny i pojęciowy do opisu i wyjaśnienia kodowania informacji jako różnego typu parametrów układów kwantowych.
4. zna pojęcia złożoności obliczeniowej i ilości zasobów potrzebnych w kwantowym uczeniu maszynowym.
5. wie w jaki sposób programuje się komputer kwantowy do zastosowań w uczeniu maszynowym.
6. zna zasadę działania, cechy i możliwości popularnych komercyjnie dostępnych komputerów kwantowych.

w zakresie umiejętności:

1. potrafi stosować formalizm matematyczny i pojęciowy do opisu wybranych algorytmów i modeli klasycznego uczenia maszynowego.
2. potrafi stosować formalizm matematyczny i pojęciowy do opisu i wyjaśnienia wybranych algorytmów kwantowych stosowanych w uczeniu maszynowym.
3. potrafi stosować formalizm matematyczny i pojęciowy do opisu i wyjaśnienia kodowania informacji jako różnego typu parametrów układów kwantowych.
4. potrafi posługiwać się pojęciami złożoności obliczeniowej i ilości zasobów potrzebnych w kwantowym uczeniu maszynowym.
5. potrafi zaimplementować wybrane algorytmy kwantowego uczenia maszynowego.
6. potrafi wymienić, opisać zasadę działania, cechy i możliwości popularnych komercyjnie dostępnych komputerów kwantowych.

Treści programowe dla zajęć:

Elementy klasycznego uczenia maszynowego: sposoby w jakie komputery klasyfikują dane, model deterministyczny i stochastyczny, regresja i klasyfikacja, ilościowy opis jakości modelu, wnioskowanie bayesowskie, procesy gaussowskie i markowskie, sztuczne sieci neuronowe, maszyny Boltzmanna, metoda jąder, metoda wektorów nośnych

Elementy informatyki kwantowej: strategie kodowania informacji, wyżarzanie kwantowe, wzór Suzukiego-Trottera, bramki kwantowe, algorytm Deutscha-Jozsy, algorytm Grovera, algorytm estymacji fazy, kwantowy algorytm mnożenia i odwracania macierzy

Złożoność obliczeniowa klasycznego i kwantowego uczenia maszynowego: rodzaje asymptotycznego przyspieszenia kwantowego, konieczna ilość próbek (sample complexity), PAC (Probably Approximately Correct), wpływ szumu, złożoność modelu

Kodowanie informacji kwantowej: kodowanie w bazie obliczeniowej (interwał Walda, interwał Wilsona), kodowanie amplitudowe (przygotowanie superpozycji w liniowym czasie, ekonomiczne superpozycji kwantowych), kodowanie Q_{sample} (łączenie rozkładów, rozkłady brzegowe, próbkowanie przez odrzucanie), kodowanie w parametrach hamiltonianu (symulacja hamiltonianu w czasie wielomianowym, ekonomiczna symulacja hamiltonianów, potęgowanie macierzy gęstości)

Zastosowanie obliczeń kwantowych do wnioskowania: modele liniowe (iloczyn wewnętrzny, obwód: repeat-until-success), metoda jąder (kwantowe jądra i odwzorowania cech, twierdzenie reprezentanta, klasyfikacja, macierze Grama), modele probabilistyczne (Q_{samples} , przybliżenie pola średniego)

Optymalizacja modeli kwantowych na komputerach kwantowych: przyspieszenia obliczeń algebraicznych (algorytm Harrowa-Hassadima-Lloyda (HHL), model Hopfielda, reguła uczenia Hebbiana), optymalizacja oparta na wyszukiwaniu Grovera lub wzmacnianiu amplitud (algorytm Durra-Hoyera, trik Ventury-Martinez, wzmacnianie amplitud perceptronów kwantowych), podejście hybrydowe (obliczenia kwantowo-klasyczne, metoda wariacyjna, hamiltonian Boltzmanna, stan Gibbsa, przybliżona termalizacja kwantowa, algorytm Nelder-Mead, metoda gradientów), adiatyczne kwantowe uczenie maszynowe (wyżarzanie), programowanie komputera kwantowego

Modele kwantowe na komputerach klasycznych: maszyny Boltzmanna (kwantowy model Isinga), tomografia kwantowa, rozkład macierzy unitarnych, kwantowe łańcuchy Markova, kwantowy model Hopfielda, algorytm Qboost

Perspektywy rozwoju i zastosowania kwantowego uczenia maszynowego: wyspecjalizowane (nieuniwersalne) realistycznie komputery kwantowe, budowa komputera kwantowego opartego na transmonach

Nazwa zajęć: Analiza matematyczna z zastosowaniami 2

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie pojęcia ciągów i szeregów funkcyjnych oraz ich rolę w informatyce.
2. zna i rozumie podstawy teorii szeregów potęgowych i Fouriera. Zna ich przykłady zastosowań w informatyce.
3. zna podstawy transformaty Fouriera, jej wersji dyskretnej i ich zastosowania w informatyce.
4. zna podstawy rachunku różniczkowego funkcji wielu zmiennych. Rozumie jego rolę w wielu działach informatyki.
5. zna i rozumie podstawy optymalizacji w zastosowaniach informatycznych.
6. zna i rozumie podstawy rachunku całkowego funkcji wielu zmiennych i jego rolę w matematycznych podstawach informatyki.
7. zna i rozumie podstawy teorii interpolacji i jej zastosowania w informatyce.
8. zna i rozumie podstawy równań różniczkowych oraz ich metod numerycznych.

w zakresie umiejętności:

1. potrafi zbadać zbieżności ciągów i szeregów funkcyjnych. Potrafi określić rolę aproksymacji w zagadnieniach informatyki, w tym aproksymacji wielomianowej.
2. potrafi badać przybliżenia funkcji wielomianami i wielomianami trygonometrycznymi. Potrafi określić błąd przybliżenia.
3. potrafi operować transformatą Fouriera i dyskretną transformatą Fouriera w zagadnieniach informatyki.
4. potrafi wykonać obliczenia z zastosowaniem rachunku różniczkowego funkcji wielu zmiennych. Potrafi wskazać jego zastosowania w informatyce.
5. potrafi zastosować rachunek różniczkowy w zagadnieniach optymalizacyjnych.
6. potrafi operować aparatem całkowym dla funkcji wielu zmiennych i wskazać jego zastosowania w informatyce.

7. potrafi operować podstawowymi metodami dyskretyzacji równań różniczkowych i metod numerycznych przydatnych przy ich rozwiązywaniu.

w zakresie kompetencji społecznych:

1. jest gotów do analizy problemów informatycznych pod kątem doboru odpowiedniego aparatu matematycznego.

Treści programowe dla zajęć:

Ciągi i szeregi funkcyjne. Pojęcie ciągu i szeregu funkcyjnego. Zbieżność punktowa. Zbieżność jednostajna ciągu i szeregu funkcyjnego. Przypomnienie informacji o metrykach. Metryki na zbiorach funkcji. Wybrane metody badania zbieżności. Kryterium Weierstrassa zbieżności jednostajnej szeregów. Twierdzenie o zmianie kolejności przejść granicznych. Ciągłość, różniczkowalność i całkowalność szeregów funkcyjnych.

Szeregi potęgowe i szeregi Taylora. Wzór Taylora. Klasa funkcji analitycznych. Promień zbieżności szeregu potęgowego, wzór Cauchy'ego-Hadamarda. Różniczkowanie szeregu potęgowego wyraz po wyrazie. Zachowanie się sumy szeregu potęgowego na końcach przedziału zbieżności (tw. Abela). Funkcje e^z , $\sin z$ i $\cos z$ argumentu zespolonego. Wzory Eulera.

Szereg Fouriera i transformata Fouriera. Funkcje okresowe. Szereg trygonometryczny w postaci rzeczywistej i zespolonej. Wzory Eulera-Fouriera. Problem zbieżności szeregu trygonometrycznego. Szereg Fouriera funkcji ciągłej i całkowalnej. Różniczkowanie szeregu Fouriera. Iloczyn ortogonalny funkcji. Układy ortogonalne w informatyce. Własność minimum współczynników Fouriera. Nierówność Bessela i tożsamość Parsewala. Zbieżność szeregu Fouriera funkcji kawałkami różniczkowalnej. Transformata Fouriera (ciągła) i jej podstawowe własności. Podstawowe zastosowania transformaty Fouriera. Dyskretna transformata Fouriera. Szybka transformata Fouriera (FFT). Zastosowania do kompresji danych (np. MP3, JPEG).

Rachunek różniczkowy funkcji wielu zmiennych. Przykłady informatyczne. Długość wektora, norma. Odwzorowania liniowe. Różne metody reprezentowania funkcji wielu zmiennych. Ciągłość. Definicja pochodnej odwzorowania, elementarne własności. Pochodne cząstkowe i ich związek z pochodną odwzorowania, macierz Jacobiego i jakobian. Twierdzenie o różniczkowaniu funkcji złożonej, twierdzenie o wartości średniej. Reguła łańcuchowa, gradient i pochodna kierunkowa.

Podstawy matematyczne metod gradientowych w informatyce. Wybrane zastosowania gradientu w informatyce. Pochodne cząstkowe wyższych rzędów, twierdzenie Schwarz'a. Wzór Taylora dla funkcji wielu zmiennych, aproksymacja wielomianami Taylora. Ekstrema funkcji wielu zmiennych.

Ekstrema warunkowe i ich zastosowania w informatyce. Metoda mnożników Lagrange'a. Metody optymalizacyjne. Metody gradientowe i bezgradientowe.

Interpolacja i jej zastosowania w informatyce. Przegląd podstawowych metod i algorytmów. Interpolacja wielomianowa i błąd interpolacyjny. Obliczanie wartości wielomianu interpolacyjnego w punkcie z wykorzystaniem uogólnionego algorytmu Hornera. Ilustracja, za pomocą wielomianu Wilkinsona, wrażliwości wyznaczania w arytmetyce zmiennopozycyjnej zer wielomianów wysokich stopni na zaburzenia współczynników. Interpolacja funkcjami sklejanymi przykładem zastosowania wielomianów niskich stopni. Całkowanie numeryczne.

Rachunek całkowy funkcji wielu zmiennych. Obszary normalne i regularne. Zbiory miary zero i całkowalność. Całki na prostokącie i kostce wielowymiarowej. Całka wielokrotna Riemanna i jej podstawowe własności. Zamiana całek wielokrotnych na iterowane. Całki wielokrotne w informatyce. Podstawy równań różniczkowych. Metody różnicowe równań różniczkowych. Metody numeryczne rozwiązywania równań różniczkowych.

Nazwa zajęć: **Ochrona własności intelektualnej**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Wyjaśnia pojęcie, istotę i funkcje prawa własności intelektualnej.
2. Opisuje strukturę prawa własności intelektualnej.
3. Wyróżnia organy państwowe działające w sferze prawa własności intelektualnej.
4. Wyróżnia wolności, prawa i obowiązki obywatela w sferze prawa własności intelektualnej.

w zakresie umiejętności:

1. Korzysta ze źródeł prawa własności intelektualnej i proponuje rozwiązania konkretnych problemów.
2. Stosuje odpowiednie przepisy związane z prawem własności intelektualnej.

w zakresie kompetencji społecznych:

1. W sposób zrozumiały, jasny i precyzyjny dla laików omawia prawo własności intelektualnej.
2. Wyjaśnia aspekty etyczne i prawne funkcjonowania prawa własności intelektualnej.
3. Uczestniczy w przygotowaniu i realizacji projektów w poszanowaniu praw własności intelektualnej.
4. Korzysta z utworów w sposób zgodny z prawem

Treści programowe dla zajęć:

Wprowadzenie do prawa własności intelektualnej: źródła prawa, przedmiot prawa własności intelektualnej, podstawowe zasady prawa własności intelektualnej.

Wprowadzenie do prawa autorskiego: podmiot, przedmiot prawa autorskiego, utwór, rodzaju utworów. Osobiste i majątkowe prawa autorskie; dozwolony użytek, odpowiedzialność za naruszenie praw autorskich, przepisy szczególne dotyczące programów komputerowych.

Umowy w prawie autorskim ze szczególnym uwzględnieniem umów w branży IT.

Wprowadzenie do prawa własności przemysłowej: zakres ochrony.

Prawo własności przemysłowej: znaki towarowe, patenty inne przedmioty prawa własności przemysłowej.

Nazwa zajęć: Platforma chmurowa Microsoft Azure

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna podstawy obliczeń w chmurze

w zakresie umiejętności:

1. potrafi skonfigurować zasoby dyskowe w chmurze
2. potrafi wykorzystać interfejs webowy do konfiguracji zasobów chmury
3. potrafi skonfigurować podstawowe zasoby obliczeniowe w chmurze

w zakresie kompetencji społecznych:

1. jest gotowy do rozumienia roli i znaczenia wykorzystania chmury Microsoft Azure w zadaniach przetwarzania danych, modeli wykorzystania sztucznej inteligencji i potrzeb dalszego rozwijania się
2. jest gotowy do rozumienia roli i znaczenia wykorzystania chmury Microsoft Azure w kontekście zapewnienia uczciwości intelektualnej w działaniach własnych i innych osób; postępuje etycznie

Treści programowe dla zajęć:

Globalna infrastruktura: definicja przetwarzania w chmurze, zalety i wady chmury obliczeniowej, porównanie wiodących dostawców

Zalety przetwarzania w chmurze, samodzielne obliczanie i optymalizacja kosztów w Azure, scenariusz środowiska w chmurze, wpływa poszczególnych aspektów na opłaty w Azure

Architektura i usługi platformy Azure

Magazyn na platformie Azure, różne typy magazynów, infrastruktura rozproszona

Usługa Azure SQL

Automatyzowanie zadań platformy Azure przy użyciu skryptów za pomocą programu PowerShell

Usługa Azure Machine Learning

Azure OpenAI

Nazwa zajęć: Podstawy fizyki kwantowej

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna doświadczenia, które nie dają się wyjaśnić na podstawie praw fizyki klasycznej.
2. zna i rozumie postulaty i podstawowe prawa fizyki kwantowej i wynikające z nich konsekwencje.

w zakresie umiejętności:

1. potrafi tłumaczyć zjawiska z zakresu podstaw fizyki kwantowej odwołując się do postulatów i praw fizyki kwantowej oraz używając stosownych modeli matematycznych.
2. potrafi rozwiązywać proste problemy z zakresu fizyki kwantowej (na poziomie podręcznika N. Zetilli: Quantum Mechanics – concepts and applications, w zakresie rozdziałów 1-7).
3. potrafi planować i przeprowadzać proste eksperymenty obejmujące podstawy fizyki kwantowej oraz analizować i przedstawiać ich wyniki.

Treści programowe dla zajęć:

Wczesna fizyka kwantowa a. promieniowanie ciała doskonale czarnego b. zjawisko fotoelektryczne c. zjawisko Comptona d. doświadczenie Davissona-Germera e. hipoteza de Broglie f. model Bohra Eksperyment dwuszczykowy i efekt Aharonova-Bohma

Matematyczne narzędzia fizyki kwantowej a. fale i analiza Fouriera b. operator rzutowania c. rozkład spektralny macierzy d. funkcje macierzy

Postulaty Fizyki Kwantowej a. wektory stanu b. operatory c. wyniki pomiarów i prawdopodobieństwa ich uzyskania d. równanie Schrödingera

Spin a. doświadczenie Einsteina–deHaasa b. doświadczenie Sterna – Gerlacha c. matematyczny opis spinu – spinowe macierze Pauliego d. transformacje spinorów przy obrotach układu współrzędnych

Problemy jednowymiarowe: baza ciągła

a. cząstka swobodna b. nieskończenie głęboka studnia potencjału c. oscylator harmoniczny e. skok potencjału f. bariera potencjału g. reprezentacja położeniowa i pędowa
Ewolucja czasowa układu

a. cząsteczka amoniaku w polu elektrycznym b. cząstka o spinie $1/2$ w polu magnetycznym c. oscylacje neutrin

Zasada nieoznaczoności

Moment pędu

a. obroty b. moment pędu c. dodawanie momentu pędu

Symetrie i zasady zachowania

a. operatory unitarne b. przesunięcia w przestrzeni c. przesunięcia w czasie

Nazwa zajęć: **Bezpieczeństwo danych - Zero Trust**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna rozwiązania dla strategii "zero trust".
2. zna rozwiązania do wykrywania i reagowania na zagrożenia.
3. zna zagadnienie zarządzania tożsamością i uprzywilejowanym dostępem.
4. zna zagadnienie scentralizowanego zarządzania tożsamością i dostępem pracowników i konsumentów (IAM).
5. zna rozwiązania (PAM).
6. zna metody ochrony punktów końcowych i danych.
7. zna metody zarządzania i ochrony urządzeń mobilnych.
8. zna rozwiązania w zakresie bezpieczeństwa danych.
9. zna rozwiązania z zakresu bezpieczeństwa dostępu (SASE).
10. zna metodologię Zero Trust w środowiskach chmurowych.
11. zna zaufane środowiska wykonawcze (TEE).
12. zna sprzętowe moduły bezpieczeństwa (HSM).
13. rozumie rolę kryptografii bezpiecznej kwantowo i zwinności krypto.
14. zna ideę i narzędzia "Confidential computing".

Treści programowe dla zajęć:

Rozwiązania dla strategii zero trust

Rozwiązania do wykrywania i reagowania na zagrożenia

Zarządzanie tożsamością i uprzywilejowanym dostępem

Scentralizowane zarządzanie tożsamością i dostępem pracowników i konsumentów (IAM)

Rozwiązania (PAM)

Ochrona punktów końcowych i danych

Zarządzanie i ochrona mobilnych urządzeń

Rozwiązania w zakresie bezpieczeństwa danych

Rozwiązania z zakresu bezpieczeństwa dostępu (SASE)

Zero Trust w środowiskach chmurowych

Zaufane środowiska wykonawcze (TEE)

Sprzętowe moduły bezpieczeństwa (HSM)

Kryptografia bezpieczna kwantowo i zwinność krypto

Confidential computing – idea i narzędzia

Nazwa zajęć: **Termodynamika i podstawy fizyki statystycznej**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. ma wiedzę w zakresie podstawowych pojęć, zjawisk i formalizmu termodynamiki fenomenologicznej, praw termodynamiki oraz teoretycznych modeli wybranych układów termodynamicznych.
2. zna formalizm matematyczny potrzebny do opisu oraz analizy praw termodynamiki.
3. zna podstawowe rozkłady: mikrokanoniczny, kanoniczny i wielki kanoniczny i potrafi podać przykłady ich zastosowań.

w zakresie umiejętności:

1. potrafi sformułować słownie podstawowe prawa termodynamiki fenomenologicznej i statystycznej oraz je zinterpretować, a także zapisać je w formalizmie matematycznym.

2. umie obliczyć właściwości termodynamiczne (równanie stanu, zmiany entropii i innych funkcji stanu, itp.) prostych substancji w typowych procesach.
3. umie podać parametry określające stan termodynamiczny układu oraz zdefiniować funkcje stanu oraz potrafi podać i opisać różne formy energii oraz jej przekazu.
4. rozumie i potrafi opisać podejście fenomenologiczne i statystyczne do zjawisk termodynamicznych, rozumie pojęcie mikrostanu i makrostanu oraz potrafi określić prawdopodobieństwo ich występowania.

w zakresie kompetencji społecznych:

1. potrafi wytłumaczyć rolę termodynamiki w fizyce oraz ewolucję rozumienia procesów termodynamicznych osobom spoza dziedziny.

Treści programowe dla zajęć:

Opis układu termodynamicznego. Pojęcie równowagi termodynamicznej. Parametry stanu. Zerowa zasada termodynamiki. Skale temperatury. Równania stanu. Diagramy fazowe. Metody pomiaru temperatury i ciśnienia.

Elementy rachunku różniczkowego funkcji wielu zmiennych. Różniczki zupełne i niezupełne, formy różniczkowe.

Energia w układach termodynamicznych. Pojęcia pracy i ciepła. Doświadczenia Joule'a. I zasada termodynamiki. Ciepła molowe i ciepła przemian fazowych. Entalpia. Przekaz energii w postaci ciepła: przewodnictwo cieplne, konwekcja, promieniowanie.

Model gazu doskonałego: idealizacja w modelu, przemiany gazu doskonałego. Entropia gazu doskonałego. Modele gazu rzeczywistego: równanie van der Waalsa.

Entropia. II zasada termodynamiki. Procesy odwracalne i nieodwracalne, strzałka czasu. Cykl Carnota. Twierdzenie Clausiusa. Warunki równowagi termodynamicznej. Entropia mieszania i paradoks Gibbsa. Maszyny ciepłe i ich sprawności. Silnik Stirlinga. Chłodziarka i pompa cieplna. Proces Joule'a-Thomsona i skraplanie gazów. Maszyny parowe.

Termodynamika w zastosowaniu. Potencjały termodynamiczne. Energia swobodna i entalpia swobodna. Zasady pracy minimalnej. Związki Maxwella.

Przemiany fazowe czystych substancji (układy wielofazowe). Układy otwarte. Przemiany fazowe mieszanin. Roztwory rozcieńczone. Równowaga chemiczna. Warunek równowagi faz. Równania Gibbsa-Duhema.

III zasada termodynamiki: temperatura zera bezwzględnego.

Kinetyczna teoria gazów: związek ciśnienia i temperatury ze średnią prędkością kwadratową cząstek, rozkład prędkości Maxwella, średnia droga cząstek. Zjawiska transportu (lepkość, przewodnictwo cieplne, dyfuzja).

Układy bardzo wielu cząstek. Modele układów termodynamicznych. Prawdopodobieństwo, mikrostan a makrostan, suma statystyczna, prawdopodobieństwo a entropia. Rozkład mikrokanoniczny i kanoniczny. Układy otwarte i wielki rozkład kanoniczny.

Statystyczna definicja entropii i temperatury. II zasada termodynamiki w obrazie statystycznym.

Statystyka Maxwella-Boltzmann. Suma statystyczna. Obliczanie wartości średnich. Twierdzenie o ekwipartycji energii. Przykład gazu doskonałego. Rozkład Maxwella.

Elementy przejść fazowych: fluktuacje, diagram fazowy, przejścia fazowe pierwszego rodzaju oraz ciągle.

Wprowadzenie do statystyk kwantowych. Bozony i fermiony. Zdegenerowany gaz Fermiego.

Nazwa zajęć: **Statystyka**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna podstawy języka programowania R, w szczególności konstrukcje programistyczne oraz składnię tego języka.
2. zna i rozumie metody statystyki opisowej.
3. zna i rozumie metody estymacji parametrów modelu statystycznego.
4. zna i rozumie metody weryfikacji hipotez statystycznych.
5. zna i rozumie metody regresji.
6. zna i rozumie metody statystyki wielowymiarowej.

w zakresie umiejętności:

1. potrafi programować w języku R.
2. potrafi przygotować dane do analizy statystycznej w języku R.

3. potrafi wykorzystać metody statystyki opisowej do przedstawienia rozkładu empirycznego próby z użyciem języka R oraz zinterpretować otrzymane wyniki.
4. potrafi dokonać estymacji nieznanymi parametrów modelu statystycznego wspomagając się obliczeniami wykonanymi w języku R.
5. potrafi dokonać badania istotności różnic za pomocą odpowiednich testów statystycznych i języka R.
6. potrafi wykonać analizę regresji w języku R oraz zinterpretować jej wyniki.
7. potrafi wykorzystać metody statystycznej analizy wielowymiarowej korzystając z języka R.

Treści programowe dla zajęć:

Podstawy języka programowania R. Podstawowe konstrukcje programistyczne (przypisanie, instrukcje sterujące, pętle, wywoływanie podprogramów i przekazywanie parametrów) oraz składnia i struktury danych tego języka.

Opis rozkładu empirycznego badanej cechy za pomocą odpowiednich tabel, wykresów oraz statystyk opisowych, np. szereg rozdzielczy, histogram, wykres słupkowy, wykres kołowy, średnia, mediana, wariancja, odchylenie standardowe, współczynnik zmienności.

Model statystyczny (model normalny, wykładniczy, dwumianowy, Poissona, jednostajny). Estymacja punktowa i przedziałowa parametrów modelu.

Testy statystyczne, hipotezy statystyczne, obszar krytyczny, błędy pierwszego i drugiego rodzaju, poziom istotności testu, p-wartość, test ilorazu wiarygodności. Testy t Studenta, testy chi-kwadrat Pearsona.

Analiza wariancji: założenia, hipoteza ogólna, testy post hoc

Podstawowe modele regresji: regresja liniowa prosta, regresja liniowa wielokrotna, regresja nieliniowa, regresja logistyczna. Metody szacowania parametrów. Dobór modelu oraz jego założenia.

Wybrane procedury statystyki wielowymiarowej: analiza składowych głównych, analiza skupień, klasyfikacja.

Nazwa zajęć: **Seminarium dyplomowe - stacjonarne**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie umiejętności:

1. Potrafi przedstawić ustnie kilkudziesięciominutową prezentację na zadany temat na odpowiednim poziomie merytorycznym
2. Potrafi zredagować szczegółowy konspekt prezentacji oraz przygotowującej pracy inżynierskiej.
3. Potrafi wyszukiwać materiały w bazach danych i zasobach bibliotecznych niezbędne do przygotowania prezentacji oraz pracy inżynierskiej.
4. Potrafi dokonać właściwej i krytycznej oceny oraz selekcji materiału zebranego do prezentacji i pracy inżynierskiej

w zakresie kompetencji społecznych:

1. Rozumie konieczność systematycznej pracy, stałego uzupełniania i aktualizowania posiadanej wiedzy
2. Potrafi formułować i objaśniać najważniejsze pojęcia i twierdzenia z działów informatyki i fizyki bezpośrednio związanych z tematyką seminarium oraz pracą inżynierską
3. Potrafi sprawnie posługiwać się językiem technicznym

Treści programowe dla zajęć:

Kształtowanie umiejętności prezentowania informacji na temat różnych aspektów inżynierii oprogramowania

Raportowanie i prezentowanie wykonania prac związanych z tematyką pracy inżynierskiej oraz dostarczenie końcowej wersji samodzielnie przygotowanej pracy inżynierskiej o charakterze systematyzującym i projektowym lub aplikacyjnym.

Nazwa zajęć: **Algebra liniowa z zastosowaniami 1**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna pojęcie wektora i macierzy o elementach z różnych ciał oraz podstawowe działania na nich
2. zna pojęcia macierzy układu liniowych równań algebraicznych, zbioru rozwiązań oraz rozumie metody rozwiązywania układów równań
3. rozumie potrzebę stosowania metod numerycznych do rozwiązywania problemów matematycznych oraz ograniczenia z tego wynikające
4. zna i rozumie twierdzenia algebry liniowej (m.in. tw. Kroneckera-Capellego, Laplace'a, Cauchy'ego)
5. zna pojęcia przestrzeni liniowej wektorów, liniowej niezależności wektorów i bazy wektorów
6. zna pojęcie wyznacznika, sposoby jego obliczania oraz własności wyznacznika

7. zna zaprezentowane algorytmy numerycznej algebry liniowej
8. zna pojęcie macierzy odwrotnej oraz metody obliczania odwrotności macierzy nieosobliwych

w zakresie umiejętności:

1. potrafi wykonać działania na wektorach i macierzach oraz opisać wektor/macierz o wybranych własnościach
2. rozwiązuje układy liniowych równań algebraicznych opisane macierzą niskiego stopnia i umie wyjaśnić metody rozwiązywania układów liniowych równań algebraicznych w przypadku macierzy wyższych stopni
3. stosuje poznane na zajęciach metody numeryczne do rozwiązywania zagadnień sformułowanych w języku algebry liniowej
4. oblicza wyznacznik i rząd macierzy niskiego stopnia, zna metody obliczania wyznacznika i rzędu macierzy wyższych stopni, wyciąga wnioski o własnościach macierzy na podstawie jej wyznacznika lub rzędu
5. potrafi zdefiniować macierz odwrotną, odwracalną, osobliwą, nieosobliwą oraz obliczyć odwrotność macierzy nieosobliwej niskiego stopnia; wskazuje metody obliczania odwrotności macierzy nieosobliwej wyższych stopni; potrafi podać własności macierzy odwracalnych
6. umie podać przykłady przestrzeni i podprzestrzeni liniowych; potrafi badać własności liniowych kombinacji wektorów
7. potrafi rozwiązywać zadania posługując się pojęciem bazy
8. potrafi podać przykłady przekształceń liniowych między różnymi przestrzeniami liniowymi; umie określić, czy dane przekształcenie jest przekształceniem liniowym; umie wyznaczyć macierz przekształcenia liniowego

Treści programowe dla zajęć:

Punkty i wektory w przestrzeni. Przekształcenie liniowe – intuicyjne wprowadzenie działań na wektorach i macierzach. Macierz kwadratowa, prostokątna, macierz transponowana, hermitowsko-sprzężona, jednostkowa (identycznościowa), zerowa, trójkątna górna, trójkątna dolna, pasmowa. Rachunek macierzowy. Iloczyn skalarny wektorów o współrzędnych rzeczywistych (w tym przedstawienie zasadności skalowania współrzędnych wektora przy obliczeniach iloczynu skalarnego na komputerze).

Układy liniowych równań liniowych – zapis macierzowy. Rozwiązywanie układów liniowych równań algebraicznych za pomocą operacji elementarnych wykonywanych na macierzy rozszerzonej układu. Postać zredukowana i całkowicie zredukowana macierzy, rząd macierzy. Tw. Kroneckera-Capellego. Macierz odwrotna, odwracalna, osobliwa, nieosobliwa. Odwrotność macierzy uzyskana za pomocą operacji elementarnych.

Rozwiązywanie układów liniowych równań algebraicznych o elementach z ciała liczb rzeczywistych metodą eliminacji Gaussa bez wyboru elementu głównego. Przyczyny niestabilności numerycznej tego algorytmu. Metoda eliminacji Gaussa z częściowym i pełnym wyborem elementu głównego. Zapis kroków metody eliminacji Gaussa bez wyboru elementu głównego za pomocą macierzy przekształceń elementarnych. Przykłady zastosowania metody eliminacji Gaussa dla układów liniowych równań algebraicznych z macierzami o elementach z ciał skończonych.

Przestrzeń i podprzestrzeń liniowa. Przykłady podprzestrzeni przestrzeni $R^{n \times n}$. Układ wektorów, kombinacja liniowa układu wektorów. Wymiar podprzestrzeni, a rząd macierzy. Liniowa zależność i niezależność układu wektorów wraz z przykładami i wymaganymi twierdzeniami.

Przekształcenie liniowe – formalna definicja. Przykłady przekształceń liniowych. Macierz przekształcenia liniowego. Sprawdzanie, czy funkcja jest przekształceniem liniowym. Sprawdzanie, czy istnieje przekształcenie liniowe zadane przez konkretne przyporządkowanie wektorów.

Baza i wymiar przestrzeni liniowej; przykłady baz przestrzeni liniowych skończonego wymiaru; sprawdzanie, czy dany układ wektorów tworzy bazę danej przestrzeni liniowej. Baza a wyznacznik i rząd macierzy. Współrzędne wektora względem bazy. Znajdowanie bazy wśród układu wektorów rozpinających daną przestrzeń; znajdowanie bazy danej przestrzeni liniowej; obliczanie wymiaru danej przestrzeni liniowej.

Wyznacznik macierzy kwadratowej i jego własności. Twierdzenie Laplace'a. Schemat Sarrusa. Definicja wyznacznika w oparciu o permutacje. Twierdzenie Cauchy'ego. Minory (główne, wiodące). Wykorzystanie wyznaczników do obliczania rzędu macierzy (metoda minorów obejmujących) i rozwiązywania układów liniowych równań algebraicznych (wzory Cramera). Porównanie metody rozwiązywania układów liniowych równań algebraicznych wykorzystującej wzory Cramera z metodą eliminacji Gaussa w kontekście obliczeń na dużych układach równań.

Rozkład LU uzyskany algorytmem eliminacji Gaussa bez wyboru elementu głównego i metodą Doolittle'a. Numeryczne obliczanie wyznacznika macierzy. Macierz dołączona i obliczanie odwrotności macierzy w oparciu o macierz dołączoną. Numeryczne obliczanie odwrotności macierzy - zastosowanie

rozkładu LU do rozwiązywania układów liniowych równań algebraicznych o stałej macierzy układu i zmieniającym się wektorze wyrazów wolnych.

Nazwa zajęć: Paradygmaty programowania

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna i rozumie definicje paradygmatów programowania, ich rodzaje oraz cel ich powstania
2. zna i rozumie zasady doboru danego paradygmatu programowania do konkretnego zadania programistycznego w celu jego najlepszego rozwiązania

w zakresie umiejętności:

1. potrafi rozróżniać paradygmaty programowania i jest w stanie dobrać odpowiedni język programowania do postawionego zadania
2. potrafi korzystać ze zintegrowanych środowisk programistycznych, które ułatwiają pisanie oprogramowania w językach programowania wymienionych w celach kształcenia
3. potrafi napisać podstawowe programy w językach programowania wymienionych w celach kształcenia
4. potrafi odnaleźć odpowiednie informacje w dokumentacji technicznej wybranego języka programowania w języku angielskim

Treści programowe dla zajęć:

Czym są paradygmaty programowania – definicja, rodzaje, cele ich powstania. Główne różnice pomiędzy paradygmatami programowania na przykładzie konkretnych języków programowania.

Programowanie imperatywne proceduralne (ang. imperative procedural) na przykładzie języków C/C++, Python.

Programowanie imperatywne zorientowane obiektowo (ang. imperative object-oriented) na przykładzie języków Java, C# (krótko o SmallTalk).

Programowanie deklaratywne ogólnie (ang. declarative) na przykładzie języków SQL, HTML i programowanie w logice.

Programowanie deklaratywne funkcyjne (ang. functional) na przykładzie języków Lisp, Haskell, Scheme, Clojure, Elixir, Java.

Programowanie sterowane danymi (ang. data-driven) na przykładzie języków AWK, JQ, XSLT.

Programowanie sterowane zdarzeniami (ang. event-driven) na przykładzie JavaScript

Programowanie współbieżne i w modelu SIMD na przykładzie OpenMP, MPI

Języki wbudowane (ang. embedded) na przykładzie języka Lua

Główne języki programowania ich cechy charakterystyczne, różne podziały języków (zarządzenia pamięcią, typowanie, dynamiczność). Języki interpretowane i kompilowane.

Współczesne trendy w programowaniu: asyncio w Python, automatyczne zarządzanie pamięcią (garbage collector) w Java, transpilacja w TypeScript, lekka wielowątkowość z wykorzystaniem coroutines.

Nazwa zajęć: Implementacja API w oparciu o zasady REST

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie umiejętności:

1. Potrafi zaimplementować REST API oraz korzystać z bibliotek JavaScript ułatwiających jego tworzenie i testowanie.

Treści programowe dla zajęć:

Wprowadzenie do zasad REST.

Metody HTTP, Routing, ćwiczenia w Postman.

Kody odpowiedzi HTTP, niestandardowe odpowiedzi.

Format przekazywanych danych – JSON, XML.

Przykłady bibliotek – NEST JS.

Sposoby autoryzacji.

Dokumentacja REST API – Swagger.

Testowanie REST API.

Tworzenie klienta, Postman, cURL, Wget.

Alternatywne koncepcje tworzenia API.

Przydatne platformy – Heroku.

Nazwa zajęć: Rachunek prawdopodobieństwa

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna pojęcie przestrzeni probabilistycznej oraz zdarzenia losowego, a także podstawowe twierdzenia i własności związane z przestrzeniami probabilistycznymi.
2. zna pojęcie zmiennej losowej, rozkładu prawdopodobieństwa, wartości oczekiwanej, wariancji i innych parametrów rozkładu prawdopodobieństwa, oraz pojęcie niezależności zmiennych losowych; zna podstawowe rozkłady dyskretne i ciągłe.
3. zna pojęcie łańcucha Markowa oraz jego podstawowe własności.

w zakresie umiejętności:

1. potrafi konstruować przestrzenie probabilistyczne dla zadanych problemów rachunku prawdopodobieństwa; potrafi wyrazić podstawowe problemy w języku zdarzeń losowych; umie zastosować poznane twierdzenia i własności przestrzeni probabilistycznych do rozwiązywania prostych zagadnień rachunku prawdopodobieństwa.
2. potrafi wyznaczyć rozkład prawdopodobieństwa zmiennej losowej; potrafi obliczyć wartość oczekiwaną, wariancję, współczynnik korelacji zmiennych losowych; potrafi określić, czy podane zmienne losowe są niezależne.
3. potrafi zbudować łańcuch Markowa dla danego problemu oraz przeprowadzić symulację tego łańcucha; potrafi wyznaczyć rozkład stacjonarny danego łańcucha Markowa i określić jego podstawowe własności.
4. potrafi oszacować prawdopodobieństwo zdarzenia losowego z wykorzystaniem nierówności Markowa, nierówności Czebyszewa oraz Centralnego Twierdzenia Granicznego.
5. potrafi rozwiązywać proste problemy rachunku prawdopodobieństwa przy użyciu środowiska R; potrafi stworzyć program w języku Python do rozwiązania bardziej skomplikowanych problemów rachunku prawdopodobieństwa, w tym do symulacji łańcuchów Markowa oraz do modelowania zmiennych losowych o podanym rozkładzie.

Treści programowe dla zajęć:

Eksperymenty losowe. Klasyczna definicja prawdopodobieństwa. Aksjomatyczna definicja przestrzeni probabilistycznej. Podstawowe własności prawdopodobieństwa.

Prawdopodobieństwo geometryczne. Niezależność zdarzeń. Prawdopodobieństwo na iloczynach kartezjańskich.

Prawdopodobieństwo warunkowe. Wzór na prawdopodobieństwo całkowite, twierdzenie Bayesa i wzór łańcuchowy.

Jednowymiarowe zmienne losowe. Dystrybuanta i jej własności. Przegląd podstawowych rozkładów dyskretnych (dwumianowy, Poissona, geometryczny, Pascala, hipergeometryczny).

Zmienne losowe wielowymiarowe. Rozkłady brzegowe. Niezależność zmiennych losowych. Wartość oczekiwana, wariancja, kowariancja. Funkcje zmiennej losowej.

Łańcuchy Markowa i ich własności. Rozkłady stacjonarne.

Zmienne losowe ciągłe. Gęstość i dystrybuanta rozkładu ciągłego. Przegląd podstawowych rozkładów ciągłych (jednostajny, wykładniczy, normalny).

Odchylenie od średniej. Nierówność Markowa, nierówność Czebyszewa.

Prawo Wielkich Liczb. Centralne Twierdzenie Graniczne.

Nazwa zajęć: Optyka z fotoniką

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna i rozumie równania Maxwella w postaci całkowitej lub różniczkowej oraz wyprowadza równanie falowe dla ośrodka jednorodnego.
2. zna i rozumie oraz wyprowadza podstawowe zasady optyki geometrycznej oraz zna i rozumie ich wykorzystanie do opisu podstawowych zjawisk w optyce: odbicia, załamania fal na granicy między dwoma ośrodkami.
3. zna i rozumie oraz objaśnia formowanie struktury pasmowej w kryształach fonicznych i możliwości modelowania propagacją fal elektromagnetycznych przy jej wykorzystaniu.
4. zna i rozumie oraz wyprowadza relację dyspersji dla fal elektromagnetycznych, polarytonów plazmonowych oraz zna i rozumie różnice i podobieństwa z falami elektromagnetycznymi z zakresu podczerwieni i światła widzialnego.
5. zna i rozumie działanie przykładowych urządzeń stosowanych w fotonice i optyce.

w zakresie umiejętności:

1. potrafi wywieść podstawowe właściwości fal elektromagnetycznych z równań Maxwella.
2. potrafi posługiwać się operatorami wektorowymi do opisu zjawisk falowych.

3. potrafi czytać artykuły naukowe ze zrozumieniem oraz potrafi przedstawiać na piśmie wyniki, streszczenia i znaczenia badań.

Treści programowe dla zajęć:

Równania Maxwella i równanie falowe (wprowadzenie do rachunku wektorowego i tensorowego, równania w próżni, ośrodku dielektrycznym i niejednorodnym, warunki brzegowe).

Fale elektromagnetyczne (widmo fal elektromagnetycznych, fale płaskie, fale monochromatyczne, wektor Poytinga, prędkość fazowa i grupowa fali, paczki falowe, optyka geometryczna, prawo Snella, prawo Fresnela).

Kryształy fotoniczne (fale elektromagnetyczne w ośrodku periodycznym, fotoniczna struktura pasmowa, metody obliczeniowe struktury pasmowej, zastosowania kryształów fotonicznych w optoelektronice, meta-materiały).

Plazmony i polarytony – sprzężenie fali elektromagnetycznej z drganiami gazu elektronowego i sieci jonowej (gaz elektronów swobodnych, model Drudego, perspektywy zastosowania plazmoniki w optoelektronice i fotonice).

Podstawowe urządzenia stosowane w optyce i fotonice (budowa i zasada działania: anteny, lasery, falowody, izolatory, modulatory i detektory).

Nazwa zajęć: **Technologie internetowe**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Zna podstawowe pojęcia i protokoły związane z Internetem
2. Posiada wiedzę na temat bezpieczeństwa i prywatności usług w Internecie
3. Zna architekturę klient-serwer
4. Rozumie pojęcie chmury obliczeniowej oraz model Platform as a Service (PaaS)

w zakresie umiejętności:

1. Potrafi wykorzystać i skonfigurować podstawowe narzędzia przydatne w pracy z Internetem
2. Potrafi korzystać z podpisów elektronicznych oraz szyfrowania wiadomości e-mail
3. Potrafi stworzyć i opublikować statyczną stronę internetową
4. Potrafi stworzyć dynamiczną stronę internetową
5. Potrafi zainstalować i skonfigurować System Zarządzania Treścią (CMS)
6. Potrafi korzystać z narzędzi zwiększających prywatność w Internecie
7. Potrafi operować danymi w formacie JSON
8. Potrafi stworzyć prostą aplikację internetową w chmurze obliczeniowej
9. Potrafi skonfigurować oprogramowanie typu firewall

Treści programowe dla zajęć:

Pojęcia podstawowe: Internet, domena, DNS, serwis, protokół, dostawca usług internetowych.

- konfiguracja serwera DNS
- Whois
- lokalizacja po IP
- traceroute

Poczta internetowa: protokoły e-mailowe, własności poczty internetowej, operacje, bezpieczeństwo, szyfrowanie i podpisywanie wiadomości (PGP), dostawcy usług, następcy email.

- wysyłanie (SMTP) i odbieranie (POP3) wiadomości przez Telnet
- Konfiguracja szyfrowania i podpisów PGP/GPG

Przesyłanie danych w Internecie: architektura klient-serwer, formaty danych.

- omówienie pojęć JSON, RPC
- obsługa narzędzi cURL oraz jq

Prywatność i bezpieczeństwo w Internecie, VPN

- Omówienie ciasteczek, webstorage, trybu incognito, AdBlocka, konfiguracja proxy, firewall, SSH, sieci Tor oraz UFW.
- Konfiguracja maszyny wirtualnej niezbędnej do dalszej pracy na zajęciach.

Witryny internetowe

- typy witryn: statyczne i dynamiczne,
- elementy witryny,
- zasada działania protokołu HTTP.
- Tworzenie stron internetowych: HTML i CSS.
- Wdrażanie stron WWW, bezpieczeństwo, SEO, aspekty komercyjne,
- Google Analytics,
- HTTPS.

Dynamiczne witryny internetowe

- Programowanie witryn internetowych z wykorzystaniem języka Python oraz bibliotek Pyramid i Jinja2

Systemy Zarządzania Treścią (CMS)

- Wordpress - instalacja, konfiguracja, bezpieczeństwo,
- komunikacja z bazą danych pgAdmin/phpMyAdmin

Witryny typu Single Page Application (SPA)

- wprowadzenie do JavaScript i TypeScript,
- stworzenie prostej aplikacji przy pomocy platformy programistycznej Angular

Chmura obliczeniowa: model Platform as a Service (PaaS)

- stworzenie prostej aplikacji w chmurze Heroku

Nazwa zajęć: **Inżynierski projekt zespołowy 2**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. Potrafi uruchomić procesy prowadzące do pozyskania systemu informatycznego o wysokiej jakości
2. Potrafi zarządzać harmonogramem, zakresem i kryteriami akceptacji projektu
3. Potrafi wdrożyć/przygotować do wdrożenia system informatyczny
4. Potrafi uczestniczyć w projekcie zespołowym prowadzonym metodami zwinnymi
5. Potrafi optymalizować (refaktoryzować) strukturę kodu źródłowego
6. Potrafi zaprojektować użyteczny system informatyczny
7. Potrafi implementować fragmenty systemu informatycznego w celu realizacji wymagań projektowych
8. Potrafi dokonać publicznej prezentacji i demonstracji systemu informatycznego
9. Potrafi korzystać z narzędzi wspierających projekty informatyczne

w zakresie kompetencji społecznych:

1. Jest gotów/gotowa do dyskusji w środowisku zawodowym o projekcie informatycznym stosując przy tym terminologię specjalistyczną

Treści programowe dla zajęć:

Refaktoryzacja.
Implementacja i testy przygotowanego rozwiązania.
Wdrożenie produktów projektu.
Przygotowanie do publicznej prezentacji projektu.

Nazwa zajęć: **Języki programowania JavaScript**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. opisuje wybrane koncepcje i struktury języka JavaScript/TypeScript
2. identyfikuje wybrane zastosowania technologii JavaScript/TypeScript
3. wskazuje metody projektowania, wdrażania i testowania aplikacji JavaScript/TypeScript

w zakresie umiejętności:

1. tworzy proste aplikacje w technologii JavaScript/TypeScript z wykorzystaniem wybranych bibliotek i platform programistycznych
2. rozwiązuje problemy i zadania jako uczestnik zespołu
3. analizuje dostępne materiały i źródła wiedzy w celu rozwiązywania problemów programistycznych

w zakresie kompetencji społecznych:

1. jest gotowy/a do prezentacji i omówienia wypracowanych rozwiązań

Treści programowe dla zajęć:

Wprowadzenie do JavaScript/TypeScript. Podstawowe zastosowania i struktury językowe.
Architektura aplikacji klient - serwer w kontekście programowania JavaScript/TypeScript.
Wprowadzenie do tworzenia prostych aplikacji klienckich JavaScript/TypeScript. Przykładowa architektura i podstawowe komponenty aplikacji klienckiej JavaScript/TypeScript.
Korzystanie z API REST z poziomu aplikacji klienckiej JavaScript/TypeScript.
Routing w aplikacjach klienckich JavaScript/TypeScript.
Debugowanie, testowanie i obsługa błędów w aplikacjach klienckich JavaScript/TypeScript.
Wprowadzenie do tworzenia prostych aplikacji serwerowych JavaScript/TypeScript. Przykładowa architektura i podstawowe komponenty aplikacji serwerowej JavaScript/TypeScript.
Debugowanie, testowanie i obsługa błędów w aplikacjach serwerowych JavaScript/TypeScript.
Podsumowanie rozwiązań problemów wypracowanych w trakcie semestru.

Nazwa zajęć: **Platforma chmurowa Google Cloud Platform**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Zna i rozumie miejsce chmury obliczeniowej w ekosystemie narzędzi i technologii informatycznych stosowanych w praktyce, oraz zna podstawy jej działania.
2. Zna i stosuje interfejsy API udostępniane w ramach platformy Google Cloud.
3. Rozumie koncepcję przetwarzania informacji w chmurach obliczeniowych, ze szczególnym uwzględnieniem chmury Google.

w zakresie umiejętności:

1. Analizuje dokumentację udostępnianą przez firmę Google w celu osiągnięcia określonego stanu konfiguracji.
2. Umie uruchomić i skonfigurować maszynę wirtualną w usłudze chmurowej Google.
3. Umie wykorzystać chmurę obliczeniową Google do świadczenia usług z zakresu przetwarzania danych.
4. Umie korzystać z narzędzi Google w sposób zapewniający bezpieczeństwo uruchamianych aplikacji.

Treści programowe dla zajęć:

Pojęcie i zasady działania chmury obliczeniowej. Cechy chmury. Rodzaje chmur.

Wprowadzenie do Google Cloud Platform. Sposoby korzystania z Google Cloud oraz różne opcje obliczeniowe.

Wdrażanie modeli pamięci masowej w GCP.

Opcje usług zarządzania aplikacjami w GCP.

Administrowanie bezpieczeństwem w Google Cloud.

Sposoby budowania bezpiecznych sieci w GCP.

Narzędzia do automatyzacji i zarządzania chmurą.

Zarządzane usługi big data.

Uczenie maszynowe w GCP.

Nazwa zajęć: **Bazy danych**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie podstawowe cechy i zadania systemu zarządzania relacyjną bazą danych
2. zna rolę systemu baz danych w systemie informatycznym i jego przykładowe architektury
3. zna i rozumie istotę oraz składowe relacyjnego modelu danych oraz jego podstawę teoretyczną
4. zna i rozumie podstawowe pojęcia i własności nierelacyjnych baz danych
5. rozumie potrzebę normalizacji schematu, nakładania ograniczeń integralnościowych
6. zna metody optymalizacji wykonywania zapytań, w tym budowę i rodzaje indeksów i fizyczną strukturę zapisu danych w bazie
7. zna pojęcie i własności transakcji w bazie danych; rozumie trudności wynikające ze współbieżnego wykonywania transakcji; zna koncepcję blokad oraz poziomów izolacji transakcji

w zakresie umiejętności:

1. potrafi zaprojektować relacyjną bazę danych w modelu koncepcyjnym oraz ocenić istniejący schemat
2. potrafi dobrać odpowiednie rozwiązanie bazodanowe do rzeczywistego problemu
3. wykonuje podstawowe i zaawansowane operacje na bazie danych z wykorzystaniem języka SQL
4. programuje serwer bazodanowy
5. potrafi znormalizować i zdenormalizować schemat
6. stosuje odpowiedni poziom izolacji transakcji
7. potrafi zaprojektować i zaimplementować prosty system bazodanowy
8. korzysta z odpowiednich metod optymalizacji zapytań

Treści programowe dla zajęć:

Historia baz danych; Modele baz danych; Podstawowe pojęcia relacyjnego modelu danych: relacja, atrybut, krotka, klucz podstawowy, klucz obcy, inne ograniczenia integralnościowe; algebra relacji i rachunek relacji; standardy SQL

Modelowanie konceptualne bazy danych na diagramie ER; związki 1:1, 1:N i N:M, ograniczenia udziału, encje słabe, związki cykliczne; zasady transformacji diagramu ER do modelu relacyjnego

Pojęcie zależności funkcyjnej, klucz kandydujący, klucz główny i nadklucz; postaci normalne; omówienie zasad i algorytmów normalizacji oraz anomalii wynikających z braku normalizacji.

Język SQL – polecenie SELECT – filtrowanie, projekcja, sortowanie; podzapytania; złączenia wewnętrzne i zewnętrzne, samo-złączenia oraz anty-złączenia; funkcje agregujące oraz grupowanie; operacje na zbiorach

Język SQL – polecenia DDL oraz DML; tworzenie obiektów bazodanowych; typy danych; metadane

Język SQL – widoki, wyrażenia tablicowe CTE, tabele i zmienne tablicowe; elementy programowania bazy danych: funkcje i procedury użytkownika, skrypty, operator APPLY; procedury wyzwalane. Optymalizacja zapytań - optymalizacja regułowa i kosztowa; analiza planu wykonania zapytania; statystyki; rola i budowa indeksów; B+-drzewa; indeksy klastrujące i nieklastrujące; fizyczna organizacja danych w bazie danych, strony, zakresy, wielkość strony, podział stron.

Transakcje w bazach danych - pojęcie transakcji, własności ACID; uszeregowalność transakcji; pojęcie blokady, algorytm 2PL; poziomy izolacji transakcji oraz anomalie; dziennik logu transakcji

Podstawy tworzenia aplikacji bazodanowych

Bazy rozproszone, partycjonowanie, bazy NoSQL

Nazwa zajęć: **Przedsiębiorczość**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Zna rodzaje własności intelektualnej oraz wybrane akty prawne dotyczące prawa nowych technologii
2. Zna zapisy ustawy o prawie autorskim i prawach pokrewnych
3. Zna jednolite prawo patentowe UE, rozumie pojęcie patentu europejskiego, zna procedurę udzielania patentów europejskich
4. Zna zagadnienia związane z przedsiębiorczością akademicką; rozumie istotę firm spin-out i spin-off, parków technologicznych, inkubatorów przedsiębiorczości, uczelnianych centrów transferu technologii; zna uwarunkowania legislacyjne oraz instrumenty finansowe
5. Rozumie pojęcia: dozwolony użytek, plagiat, prawo cytatu; potrafi stosować prawo autorskie w pracach naukowych i dyplomowych; rozumie istotę prawa autorskiego w Internecie
6. Rozumie pojęcie własności przemysłowej i jej ochrony, zna zapisy ustawy prawo własności przemysłowej; Rozumie pojęcia: wynalazki i ochrona patentowa; zna procedurę patentową, zna procedurę korzystania z patentowych baz danych
7. Rozumie pojęcia działalności gospodarczej, znaczenie prawa działalności gospodarczej, formy działalności gospodarczej, pojęcie przedsiębiorcy, kategorii przedsiębiorców, umowy w działalności gospodarczej, procedurę rejestrację działalności gospodarczej; biznes-planu

w zakresie kompetencji społecznych:

1. rozumie i docenia znaczenie uczciwości intelektualnej w działaniach własnych i innych osób; ma świadomość problemów etycznych w kontekście rzetelności badawczej (plagiat czy też auto-plagiat)

Treści programowe dla zajęć:

Własność intelektualna, rodzaje własności intelektualnej, kapitał intelektualny, innowacje, know-how, gospodarka oparta na wiedzy; zarządzanie własnością intelektualną w przedsiębiorstwie

Prawo nowych technologii - ustawy: o prawie autorskim i prawach pokrewnych; o ochronie danych osobowych; o ochronie baz danych; o dostępie do informacji publicznej; prawo własności przemysłowej

Własność intelektualna i jej ochrona, prawa wyłączne, autorskie prawa osobiste i autorskie prawa majątkowe, przedmiot i podmiot prawa autorskiego; WIPO i konwencje międzynarodowe

Dozwolony użytek, plagiat, prawo cytatu, prawo autorskie w pracach naukowych i dyplomowych; prawo autorskie w Internecie, licencje na oprogramowanie

Własność przemysłowa i jej ochrona, prawo własności przemysłowej; wynalazki, wzory użytkowe i wzory przemysłowe

Ochrona patentowa, procedura patentowa, patentowe bazy danych; prawo ochronne na wzory użytkowe, prawo z rejestracji wzorów przemysłowych

Jednolite prawo patentowe UE, Europejska Organizacja Patentowa, patent europejski, procedura udzielania patentów europejskich

Przedsiębiorczość akademicka, firmy spin-out i spin-off, parki technologiczne, inkubatory przedsiębiorczości, uczelniane centra transferu technologii, uwarunkowania legislacyjne, zasoby ludzkie i instrumenty finansowe

Działalność gospodarcza, prawo działalności gospodarczej, formy działalności gospodarczej, pojęcie przedsiębiorcy, kategorie przedsiębiorców, umowy w działalności gospodarczej, rejestracja działalności gospodarczej; biznes-plan

Nazwa zajęć: **Komputery kwantowe - fizyka operacji na transmonach**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie pojęcie transmonu oraz podstawy elektrodynamiki obwodów kwantowych.
2. zna i rozumie metody charakteryzacji i kalibracji kubitów transmonowych i ich oddziaływań, oparte na sterowaniu pulsami mikrofalowymi.
3. zna i rozumie metody charakteryzacji fizycznej architektury procesora kwantowego.

4. rozumie znaczenie kwantowej korekcji i łagodzenia błędów i zna przykłady takich rozwiązań.

w zakresie umiejętności:

1. potrafi rozwiązywać problemy rachunkowe dotyczące transmonów i ich oddziaływań.
2. potrafi programować eksperymenty charakteryzujące i kalibrujące kubity i ich oddziaływania.
3. potrafi programować eksperymenty charakteryzujące możliwości procesora kwantowego.

Treści programowe dla zajęć:

Implementacja kubitu i podstawowych bramek opartych na transmonach, podstawy elektrodynamiki obwodów kwantowych

Kalibracja bramek jednokubitowych na rzeczywistym procesorze kwantowym - środowisko Qiskit Pulse
Charakteryzacja wydajności procesora kwantowego: objętość kwantowa, randomized benchmarking, CLOPS - środowisko Qiskit Experiments

Macierz gęstości, tomografia stanów kwantowych - środowisko Qiskit Experiments

Pomiar i kalibracja parametrów transmonu, wyższe poziomy energetyczne transmonu, pomiar czasów T1, T2 - środowisko Qiskit Pulse

Hamiltonian Jaynesa-Cummingsa - środowisko Qiskit Pulse

Pomiar przesunięcia Starka - środowisko Qiskit Pulse

Tomografia hamiltonianu układu oddziałujących transmonów - środowisko Qiskit Pulse

Metody łagodzenia błędów przetwarzania informacji kwantowej - ZNE, PEC, Dynamical Decoupling, łagodzenie błędów odczytu - środowisko Qiskit Experiments, Qiskit Runtime

Nazwa zajęć: Teoria informacji i kodowanie

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna i rozumie fundamentalne zasady i koncepcje teorii informacji i kodowania; zna standardowe narzędzia teorii informacji i kodowania dla gromadzenia, analizy i prezentacji danych; rozumie własności, charakterystyki i funkcje informacji oraz entropii;
2. zna standardowe metody kodowania danych informatycznych

w zakresie umiejętności:

1. potrafi obliczyć ilość informacji, entropię źródła informacji oraz redundancję metody kodowania; stosuje standardowe metody kodowania danych informatycznych
2. potrafi oszacować efektywność metod kodowania przy realizacji systemu informatycznego

w zakresie kompetencji społecznych:

1. jest gotów/gotowa pozyskiwać informacje o nowoczesnych sposobach kodowania informacji z literatury i innych źródeł, integrować je, dokonywać ich interpretacji, wyciągać logiczne wnioski i formułować opinie
2. jest gotów/gotowa uzupełniać i doskonalić nabytą wiedzę i umiejętności; jest przygotowany/przygotowana do samodzielnego zdobywania i doskonalenia wiedzy oraz umiejętności profesjonalnych i badawczych; rozumie podstawy teorii informacji i kodowania i jest gotów/gotowa funkcjonować w społeczeństwie informacyjnym

Treści programowe dla zajęć:

Notacja

Podstawy kodowania, klasyfikacja kodów

Teoria informacji, entropia, ilość informacji

Optymalne kodowanie, kodowanie Huffmana i Shannona-Fano

Kodowanie słownikowe

Kodowanie arytmetyczne

Nazwa zajęć: Matematyka dyskretna

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna i rozumie różne metody dowodzenia twierdzeń.
2. zna i rozumie podstawowe zasady i prawa przeliczania.
3. zna podstawowe typy rekurencji i sposoby ich rozwiązywania. Zna interpretacje i zależności rekurencyjne dla podstawowych liczb kombinatorycznych, w szczególności: współczynników wielomianowych, Fibonacciego, Catalana, Bella, Stirlinga.
4. zna i rozumie podstawowe pojęcia teorii grafów.
5. zna przykłady klasycznych zastosowań teorii grafów.

w zakresie umiejętności:

1. Potrafi wykorzystywać różne metody dowodzenia implikacji. Potrafi stosować zasadę indukcji matematycznej.

2. Potrafi stosować podstawowe zasady i prawa przeliczania. Umie wykorzystywać zasadę szufladkową. Umie przeprowadzić dowody prostych tożsamości kombinatorycznych.
3. potrafi układać i identyfikować wybrane zależności rekurencyjne oraz rozwiązywać je różnymi metodami.
4. potrafi się posługiwać podstawowymi pojęciami teorii grafów.
5. potrafi posługiwać się klasycznymi algorytmami teorii grafów; rozumie znaczenie praktyczne teorii grafów - umie podać przykłady, w których stosuje się poznane zagadnienia i fakty teorii grafów w praktyce.

w zakresie kompetencji społecznych:

1. jest gotów/gotowa do przedstawienia laikowi przykładów, w których stosuje się poznane zagadnienia i fakty z matematyki dyskretnej w praktyce.

Treści programowe dla zajęć:

Metody dowodzenia implikacji. Zasada szufladkowa. Twierdzenie o indukcji matematycznej.
Podstawowe zasady i prawa przeliczania - zasada bijekcji, prawa dodawania i mnożenia.
Schematy wyboru. Zasada włączania i wyłączania. Współczynniki wielomianowe. Tożsamości kombinatoryczne.
Zależności rekurencyjne. Układanie i rozwiązywanie prostych i liniowych równań rekurencyjnych.
Złożone zależności rekurencyjne. Liczby: Fibonacciego, Catalana, Bella, Stirlinga.
Podstawowe pojęcia teorii grafów.
Klasyczne problemy i algorytmy grafowe – problemy: najkrótszych ścieżek, optymalnego drzewa rozpiętego, chińskiego listonosza, wędrującego komiwojażera, przydziału zadań, kolorowania grafów i map.

Nazwa zajęć: Podstawy programowania

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Zna logiczną zasadę funkcjonowania komputera oraz działania programu (maszyna RAM).
2. Zna podstawowe konstrukcje programistyczne (instrukcje sterujące, wywoływanie procedur i funkcji oraz różne typy przekazywania parametrów).
3. Zna podstawowe wbudowane typy danych oraz zasady tworzenia złożonych typów danych.
4. Zna zasady dynamicznego zarządzania pamięcią
5. Zna i rozumie paradygmaty programowania obiektowego (abstrakcja, enkapsulacja, dziedziczenie i polimorfizm)

w zakresie umiejętności:

1. Potrafi stosować podstawowe konstrukcje programistyczne (instrukcje sterujące, wywoływanie procedur i funkcji oraz różne typy przekazywania parametrów).
2. Potrafi czytać i analizować kod "proceduralny" napisany w językach C oraz Python i rozumie istotę i efekty wykonywanych operacji.
3. Potrafi czytać i analizować kod obiektowy napisany w językach C++, Python oraz Java.
4. Potrafi pisać programy w stylu obiektowym z wykorzystaniem wszystkich paradygmatów programowania obiektowego.

Treści programowe dla zajęć:

Logiczny zasada działania komputera i programu (maszyna RAM). Tworzenie oprogramowania.
Algorytm, procesor, program, język programowania. Kompilacja, interpretacja i konsolidacja programu.
Dane i kod.
Sterowanie przebiegiem programu
Typ danych, typy podstawowe, reprezentacja danych. Stałe i zmienne.
Złożone typy danych i zarządzanie pamięcią.
Podprogramy. Przekazywanie parametrów. Przeciążanie, funkcje biblioteczne.
Obiekty i klasy, pola, metody. Enkapsulacja, dziedziczenie i polimorfizm, klasy abstrakcyjne, interfejsy, tworzenie obiektów oraz korzystanie z ich pól i metod.
Graficzny interfejs użytkownika.

Nazwa zajęć: Platformy programistyczne Angular i React

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Zna platformy programistyczne aplikacji webowych React i Angular oraz związane z nimi dobre praktyki i wzorce projektowe.

w zakresie umiejętności:

1. Potrafi korzystać z platform programistycznych React i Angular w praktyce projektowej.

Treści programowe dla zajęć:

Wprowadzenie do React.
React - Routing.
Wprowadzenie do Redux.
React - testowanie.
React - wzorce i reużywalne komponenty.
Wprowadzenie do Angular.
Angular vs React.
Angular - Change Detection, Zone.js, Chrome DevTools.
Angular - Directives and pipes.
Angular - Dependency Injection.
Wprowadzenie do RxJS, Angular Elements, Angular PWA.
Projekt Angular/React.

Nazwa zajęć: **Teoria informacji kwantowej**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie pojęcia kwantowej teorii informacji.
2. zna i rozumie takie elementy teorii kwantowej jak splątanie, ewolucja, pomiar, szum, kanał kwantowy.
3. zna i rozumie właściwości prototypowych protokołów kwantowych, tj. dystrybucji splątania (współdzielona losowość), gęstego kodowania, teleportacji.
4. zna i rozumie miary odległości, kwantowa i klasyczna informacja i entropia.
5. zna i rozumie kwantowe pojęcie kompresji.
6. zna i rozumie zagadnienie koncentracji splątania.
7. zna i rozumie zagadnienie komunikacji z wykorzystaniem splątania kwantowego.
8. zna i rozumie pojęcie pojemności kanałów kwantowych.

w zakresie umiejętności:

1. potrafi stosować pojęcia kwantowej teorii informacji.
2. potrafi stosować takie elementy teorii kwantowej jak splątanie, ewolucja, pomiar, szum, kanał kwantowy.
3. potrafi stosować właściwości prototypowych protokołów kwantowych w analizie bardziej złożonych przypadków.

Treści programowe dla zajęć:

Qubit
Teleportacja kwantowa
Gęste kodowanie
Kanały kwantowe
Kwantowa korekcja błędów
Elementy klasycznej teorii informacji
Entropia von Neumanna
Kwantowa kompresja danych
Koncentracja splątania
Mieszane stany splątane
Dostępna informacja
Nierówność odsprzęgająca
Pojemności kanałów kwantowych
Protokoły "matka" i "ojciec"

Nazwa zajęć: **Platforma chmurowa IBM Cloud**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie dostępne usługi w infrastrukturze IBM Cloud.

w zakresie umiejętności:

1. potrafi posługiwać się interfejsem graficznym, narzędziami wiersza poleceń i interfejsem API podczas dostępu do usług IBM Cloud.
2. potrafi wybrać odpowiednią usługę IBM Cloud, sposób jej wdrożenia i go zastosować w celu dostarczenia programistom i zespołom produkcyjnym określonej funkcjonalności.
3. potrafi wdrożyć aplikację w infrastrukturze IBM Cloud.

Treści programowe dla zajęć:

Wprowadzenie do IBM Cloud: lokalizacje, regiony, strefy; typy kont i typy wsparcia technicznego; rozliczenia, zużycie zasobów oraz ich oszacowania; zarządzanie identyfikacją i dostępem użytkowników.

Infrastruktura związana z obliczeniami, siecią i przestrzenią dyskową: serwery wirtualne i dedykowane; magazyn danych blokowych, plików i obiektów; usługi sieciowe; wirtualna chmura prywatna; VMWare. Wdrażanie aplikacji: kontenery i Kubernetes; OpenShift i środowisko bezserwerowe.

Usługi IBM Cloud: bazy danych, integracja środowiska, sztuczna inteligencja, analityka danych, DevOps.

Nazwa zajęć: Warsztat programisty

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Zna i rozumie systemy i narzędzia systemowe będące podstawą warsztatu programistycznego informatyka.
2. Zna i rozumie języki skryptowe wykorzystywane w systemowych interpreterach poleceń.

w zakresie umiejętności:

1. Potrafi korzystać z poleceń powłoki Bash, w tym zarządzać systemem operacyjnym i pisać skrypty.
2. Potrafi zainstalować, uruchomić i korzystać z maszyny wirtualnej oraz z oprogramowania do konteneryzacji.
3. Potrafi korzystać z rozproszonego systemów kontroli wersji.
4. Potrafi korzystać z poleceń powłoki PowerShell.

Treści programowe dla zajęć:

Linux – graficzny interfejs użytkownika GUI; interfejs wiersza poleceń CLI; podstawowe komendy powłoki; dostęp zdalny przez SSH

Linux – edytory tekstu

Linux – strumienie plików i potoki; zmienne środowiskowe; przetwarzanie tekstu; wyrażenia regularne

Linux – operacje na plikach; informacje o systemie; dowiązania twarde i symboliczne; wyszukiwanie aplikacji i plików; środowisko użytkownika; uprawnienia plików

Linux – zarządzanie procesami

Linux – skrypty w Bashu

Linux – przeglądarki internetowe i programy do pobierania z sieci; transfer danych przez sieć

Linux – podstawowa konfiguracja sieci i rozwiązywanie problemów; narzędzia DNS

Linux – administracja systemem i podnoszenie uprawnień; instalacja i aktualizacja oprogramowania; konta, użytkownicy i grupy; planowanie przyszłych procesów; systemy plików

Linux – kompilacja kodu; biblioteki systemowe; biblioteki języków programowania

Maszyny wirtualne, Windows Subsystem for Linux oraz konteneryzacja

Systemy rozproszonej kontroli wersji; Git

Windows – PowerShell

Nazwa zajęć: Bezpieczeństwo, ryzyko i zgodność w chmurze

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. rozumie znaczenie śledzenia zmian w środowisku chmurowym.
2. zna metody i rozumie potrzebę nadążania za aktualizacjami przepisów i najlepszymi praktykami.
3. zna metody zapewnienia ciągłej zgodności.
4. zna wymagania dotyczące zgodności z przepisami dla sprzedawców.
5. zna metody radzenia sobie z rosnącymi zagrożeniami cybernetycznymi.
6. zna metody zdobywania i utrzymywania obserwowalności.
7. rozumie zagadnienie zgodności systemów kubernetes.
8. rozumie zagadnienie obserwowalności w systemach kubernetes.
9. zna metody zabezpieczenia systemu kubernetes.
10. zna narzędzia do zapewnienia zgodności w chmurze.

Treści programowe dla zajęć:

Śledzenie zmian

Nadążanie za aktualizacjami przepisów i najlepszymi praktykami

Zapewnienie ciągłej zgodności:

Zgodność z przepisami dla sprzedawców

Radzenie sobie z rosnącymi zagrożeniami cybernetycznymi

Zdobywanie i utrzymywanie obserwowalności

Zgodność systemów kubernetes

Obserwowalność w systemach kubernetes
Zabezpieczenie systemu kubernetes
Narzędzia do zapewnienia zgodności w chmurze

Nazwa zajęć: **Mechanika ogólna**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna prawa kinematyki i potrafi je sformułować w języku wektorowym.
2. zna i rozumie prawa dynamiki Newtona.
3. zna i rozumie zasadę zachowania energii.
4. zna i rozumie zasady dynamiki ruchu obrotowego i zasadę zachowania momentu pędu.
5. zna prawo grawitacji i rozumie jego konsekwencje.
6. zna i rozumie transformację Galileusza i Lorentza. Zna założenia STW i rozumie wynikające z nich konsekwencje.

w zakresie umiejętności:

1. potrafi opisać ruch w 1, 2 i 3 wymiarach.
2. potrafi zastosować prawa Newtona do rozwiązywania prostych problemów.
3. potrafi zastosować zasadę zachowania energii w przypadku występowania sił zachowawczych.
4. potrafi zastosować zasadę zachowania pędu w układach wielu ciał.
5. potrafi zastosować zasady dynamiki i zasadę zachowania momentu pędu do badania ruchu obrotowego.
6. potrafi przetransformować czterowektory z jednego układu inercjalnego do innego układu inercjalnego.

Treści programowe dla zajęć:

Kinematyka1. Wielkości średnie i chwilowe2. Ruch ze stałym przyspieszeniem3. Ruch w wyższych wymiarach4. Wektory w ruchu dwuwymiarowym5. Wektor położenia i jego pochodne6. Ruch po okręgu
Prawa Newtona1. Zasada bezwładności2. Drugie prawo Newtona3. Trzecie prawo Newtona4. Ciężenie powszechne5. Tarcie statyczne i kinetyczne
Zasada zachowania energii1. Twierdzenie o pracy i energii2. Zasada zachowania energii3. Tarcie a zachowanie energii4. Zasada zachowania energii w dwu wymiarach5. Praca jako iloczyn skalarny6. Siły zachowawcze i niezachowawcze7. Zastosowanie zasady zachowania energii do grawitacyjnej energii potencjalnej
Dynamika układów wielu ciał1. Środek masy i ciężkości2. Zasada zachowania pędu3. Zderzenia sprężyste i niesprężyste
Dynamika ruchu obrotowego bryły sztywnej1. Pojęcie bryły sztywnej2. Przyspieszenie kątowe3. Bezwładność, pęd i energia w ruchu obrotowym4. Moment obrotowy i twierdzenie o pracy i energii5. Obliczanie momentu bezwładności6. Twierdzenie o osiach równoległych7. Energia kinetyczna w ruchu obrotowym8. Zachowanie momentu pędu i energii w ruchu obrotowym9. Żyroskop
Szczególna Teoria Względności1. Postulaty STW2. Transformacja Lorentza3. Prawo transformacji prędkości4. Względność równoczesności5. Dylatacja czasu6. Skrócenie długości7. Przeszłość, teraźniejszość i przyszłość w teorii względności8. Geometria czasoprzestrzeni9. Czas właściwy10. Czterowektory i ich transformacje

Nazwa zajęć: **Matematyka elementarna**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna definicję pochodnej funkcji w punkcie oraz jej interpretację geometryczną.
2. zna własności pochodnych oraz podstawowe wzory pochodnych najważniejszych funkcji elementarnych.
3. zna interpretację geometryczną całki oznaczonej.
4. zna własności całek oraz podstawowe techniki całkowania.
5. zna pojęcie liczb zespolonych.
6. zna wzór na rozwinięcie funkcji w szereg Taylora.
7. zna pojęcie równania różniczkowego.
8. zna pojęcie podstawowe pojęcia algebry liniowej.

w zakresie umiejętności:

1. potrafi obliczać pochodną funkcji.
2. potrafi znajdować ekstrema funkcji jednej zmiennej.
3. potrafi obliczać całkę nieoznaczoną i oznaczoną funkcji jednej zmiennej.
4. potrafi wykonywać podstawowe działania na liczbach zespolonych.

5. potrafi przedstawić liczbę zespoloną w postaci trygonometrycznej i wykładniczej.
6. potrafi przybliżyć funkcje paroma wyrazami rozwinięcia w szereg Taylora.
7. potrafi znaleźć rozwiązanie równania różniczkowego pierwszego rzędu o rozdzielonych zmiennych.
8. potrafi znaleźć rozwiązanie równania różniczkowego liniowego rzędu pierwszego.
9. potrafi znaleźć rozwiązanie równania różniczkowego liniowego jednorodnego rzędu drugiego o stałych współczynnikach.
10. potrafi wykonywać podstawowe działania na macierzach.
11. potrafi wyznaczyć wartości własne macierzy oraz jej wyznacznik.

Treści programowe dla zajęć:

Rachunek różniczkowy
Liczby zespolone
Całki nieoznaczone i oznaczone
Równania różniczkowe
Algebra liniowa

Nazwa zajęć: **Praktyczne umiejętności projektowe**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. Zna i rozumie aspekty bezpieczeństwa danych, zabezpieczenia aplikacji, wykrywania błędów w aplikacji.
2. Zna i rozumie modele tworzenia aplikacji, strategie budowania aplikacji dla różnych zastosowań praktycznych.
3. Zna i rozumie proces zbierania wymagań i zarządzania projektem informatycznym

w zakresie umiejętności:

1. Potrafi wskazać swoją rolę w projekcie i określić związane z nią zadania i odpowiedzialności
2. Potrafi powiązać przedstawione studium przypadku i określić relacje w stosunku do prowadzonego zespołowego projektu inżynierskiego

w zakresie kompetencji społecznych:

1. Jest gotów/gotowa zadawać pytania i konsultować się z praktykami na temat własnego zadania projektowego.

Treści programowe dla zajęć:

Zbieranie wymagań; Priorytetyzacja zakresu; Grooming backlogu i zarządzanie projektem, w tym narzędzia zarządzania projektem

Testy z użytkownikiem; Testy użyteczności; Scenariusze testowe, wywiad, określenie potrzeb użytkowników

Testy automatyczne; Testy automatyczne - narzędzia; Testy jednostkowe, pokrycie testami, narzędzia kontroli jakości kodu; Testy integracyjne: narzędzia, projektowanie i implementacja

Zarządzanie procesem wytwórczym, SCRUM, Kanban, Waterfall, rola Scrum Mastera

Repozytorium, kodu, przechowywanie dużych plików; aspekty prawne zdobywania i przechowywania danych. Przechowywanie haseł i ustawień. Code review - proces i zasady, narzędzia

Proces ciągłej integracji, narzędzia ciągłej integracji. Wersjonowanie projektu, tagowanie, wydawanie wersji

Deploy projektu, serwery aplikacji, ustawienia domeny, https, load balancery, proxy

Rozwiązania chmurowe, koszty utrzymania aplikacji

Projektowanie bazy danych, nierelacyjne, relacyjne, czasowe bazy danych. Wydajność bazy danych, cacheowanie danych

Architektura monolitu; Architektura mikroserwisowa, routing w architekturze mikroserwisowej;

Popularne stacki technologiczne i połączenia technologii

Skalowalność aplikacji (horyzontalne i wertykalne), wydajność, testowanie wydajności, optymalizacja kodu

Bezpieczeństwo, zabezpieczenie aplikacji: Frontend i Backend, podstawowe podatności i niebezpieczeństwa

Prezentacje zastosowań w projektach inżynierskich

Nazwa zajęć: **Obliczenia kwantowe w biznesie**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna motywacje do wykorzystywania obliczeń kwantowych w biznesie.

2. zna przykłady zastosowań obliczeń kwantowych w biznesie w sektorach: usług lotniczych, rynków finansowych i bankowości, chemii i ropy naftowej, elektroniki, opieki zdrowotnej, nauk przyrodniczych, logistyki, bezpieczeństwa.

w zakresie umiejętności:

1. potrafi zastosować obliczenia kwantowe dla wybranych przypadków w biznesie w sektorach: usług lotniczych, rynków finansowych i bankowości, chemii i ropy naftowej, elektroniki, opieki zdrowotnej, nauk przyrodniczych, logistyki, bezpieczeństwa.

Treści programowe dla zajęć:

Motywacja do wykorzystania obliczeń kwantowych w biznesie.

Przykłady zastosowań:

- usługi lotnicze
- bankowość i rynki finansowe
- chemia i ropa naftowa
- elektronika
- opieka zdrowotna
- nauki przyrodnicze
- logistyka
- bezpieczeństwo

Nazwa zajęć: **Mechanika kwantowa**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie postulaty fizyki kwantowej.
2. rozumie ewolucję czasową układu w obrazie Schroedingera i Heisenberga.
3. zna teorię momentu pędu.
4. rozumie znaczenie symetrii w fizyce kwantowej.
5. zna podstawowe metody obliczeń przybliżonych stosowanych w mechanice kwantowej.

w zakresie umiejętności:

1. potrafi przeprowadzać obliczenia w reprezentacjach" pędowej, położeniowej i energetycznej.
2. potrafi rozwiązać równania Schroedingera i Heisenberga dla prostych przypadków.
3. potrafi znaleźć funkcje własne operatora momentu pędu i dodawać momenty pędu.
4. potrafi zastosować twierdzenie Noether w prostych przypadkach.
5. potrafi zastosować rachunek zależny i niezależny od czasu w prostych przypadkach.

Treści programowe dla zajęć:

Podstawy mechaniki kwantowej

1. Wektory stanu, operatory, notacja bra-ket
2. Macierzowa reprezentacja operatorów i zmiana bazy
3. Operator pędu jako generator przesunięć
4. Funkcja falowa w reprezentacji położeniowej i pędowej

Dynamika kwantowa

1. Ewolucja czasowa w obrazie Schroedingera i Heisenberga
2. Oscylator harmoniczny
3. Propagator i całki po trajektoriach

Moment pędu

1. Obroty, definicja operatora momentu pędu i relacje komutacyjne jego składowych
2. Spin
3. SU(2) i SO(3)
4. Wartości własne i wektory własne operatora momentu pędu
5. Orbitalny moment pędu
6. Dodawanie momentów pędu
7. Operatory tensorowe

Symetrie

1. Symetrie dyskretne
2. Symetrie ciągłe
3. Symetria odwrócenia czasu

Metody przybliżone

1. Rachunek zaburzeń niezależny od czasu
2. Rachunek zaburzeń zależny od czasu

Nazwa zajęć: **Zapewnianie jakości w systemach informatycznych**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna i rozumie podstawowe pojęcia i terminy związane z zapewnianiem jakości w systemach informatycznych
2. zna i rozumie elementy zarządzania zespołem informatycznym
3. zna i rozumie sposoby wykonywania testów i ich znaczenie w zapewnianiu jakości w systemach informatycznych

w zakresie umiejętności:

1. potrafi wykorzystać narzędzia automatyzujące proces testowania, a także przeprowadzić testy zapewniające dostępność dla osób z niepełnosprawnością
2. potrafi przeprowadzić testy bezpieczeństwa
3. potrafi wykorzystać różne techniki testowania zapewniające wysoką jakość oprogramowania

Treści programowe dla zajęć:

Wprowadzenie do zapewnianie jakości w systemach informatycznych
Zasady skutecznego działania oraz zarządzanie zespołem (komunikacja i przywództwo)
Modele zapewniania jakości w wybranych metodykach wytwarzania oprogramowania
Automatyzacja wykonywania testów
Testy niefunkcjonalne (bezpieczeństwo, wydajność, dostępność)
Zarządzanie ryzykiem
Zarządzanie grupami projektów (programami)
Automatyzacja testów (GUI, API)
Testy dostępności wg standardu WCAG
Wprowadzenie do testów bezpieczeństwa; metody zabezpieczania aplikacji webowych przed atakami hackerskimi
Przygotowanie na podstawie danych statystycznych tzw. "Persony" określającej modelowego użytkownika aplikacji, jego potrzeb i oczekiwań
Testowanie białoskrzynkowe, statyczna analiza kodu, testy mutacyjne
Testowanie czarnoskrzynkowe; analiza przypadków testowych; przygotowanie przypadków testowych na podstawie wymagań
Testy wydajności
Teoria testowania i jej techniki

Nazwa zajęć: Analiza i projektowanie obiektowe

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna rolę analizy i projektowania obiektowego w cyklu życia oprogramowania
2. zna podstawowe techniki i metody analizy i projektowania obiektowego

w zakresie umiejętności:

1. potrafi wykorzystać UML w analizie i projektowaniu obiektowym
2. potrafi zebrać wymagania w postaci przypadków użycia
3. potrafi zbudować model wiedzy dziedzinowej w oparciu o wymagania zapisane w postaci przypadków użycia
4. potrafi wykorzystać diagramy interakcji w procesie projektowania obiektowego
5. umie przypisać obiektom odpowiedzialności z wykorzystaniem odpowiednich technik i metod
6. potrafi zbudować projektowy diagram klas
7. potrafi wykorzystać wzorce projektowe w procesie projektowania obiektowego

w zakresie kompetencji społecznych:

1. jest gotów/gotowa do budowy oprogramowania o wysokiej jakości

Treści programowe dla zajęć:

Analiza i projektowanie obiektowe w cyklu życia oprogramowania
Programowanie obiektowe, a analiza i projektowanie obiektowe
Określanie wymagań przy pomocy przypadków użycia
Analiza obiektowa - model wiedzy dziedzinowej
Analiza obiektowa - wprowadzanie asocjacji i atrybutów
Diagramy interakcji w procesie projektowania obiektowego
Projektowanie obiektowe - przypisywanie obiektom odpowiedzialności
Projektowy diagram klas

Wzorce projektowe

Nazwa zajęć: **Zaawansowana inżynieria oprogramowania**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. rozumie problem rzetelności i bezpieczeństwa systemów informatycznych
2. rozumie zagadnienie powtórnego użycia kodu i inżynierii opartej na komponentach
3. rozumie zagadnienie inżynierii oprogramowania rozproszonego i zorientowanego na usługi
4. rozumie zagadnienie inżynierii złożonych systemów
5. rozumie zagadnienie inżynierii oprogramowania czasu rzeczywistego
6. rozumie zagadnienie zarządzania projektami
7. rozumie problem zarządzania jakością i konfiguracją

w zakresie umiejętności:

1. potrafi wykorzystać odpowiednie metryki do określania wymagań i mierzalnych kryteriów dostępności, niezawodności, bezpieczeństwa, zabezpieczeń oraz odporności oprogramowania
2. potrafi wykorzystać i projektować oprogramowanie/komponenty, pod kątem ich ponownego użycia do budowy nowego systemu informatycznego
3. potrafi zastosować odpowiednie wzorce architektoniczne i projektowe do budowy oprogramowania rozproszonego i zorientowanego na usługi
4. potrafi zaprojektować oraz zaimplementować złożone usługi sieciowe zgodne z ideą REST
5. potrafi opracować koncepcję realizacji złożonego systemu socjotechnicznego
6. potrafi zastosować odpowiednie wzorce projektowe do budowy prostego systemu czasu rzeczywistego
7. potrafi dokonać analizy ryzyka dla projektu informatycznego
8. potrafi przygotować plan realizacji dla złożonego przedsięwzięcia informatycznego
9. potrafi wykorzystać narzędzia informatyczne w celu zapewnienia wysokiej jakości oprogramowania

Treści programowe dla zajęć:

- dlaczego niezawodność i bezpieczeństwo są ważnymi atrybutami dla wszystkich systemów oprogramowania;
- pięć wymiarów rzetelności: dostępność, niezawodność, bezpieczeństwo, zabezpieczenia i odporność;
- pojęcie systemów socjotechnicznych i dlaczego konieczne jest traktowanie systemów jako całość, a nie tylko systemy oprogramowania;
- redundancja i różnorodność jako podstawowe pojęcia wykorzystywane w osiągnięciu niezawodnych systemów i procesów;
- korzyści i problemy związane z ponownym wykorzystaniem oprogramowania podczas opracowywania nowych systemów;
- koncepcja platformy programistycznej oraz sposób jej wykorzystania w rozwoju aplikacji;
- pojęcie linii produktów oprogramowania, tworzonej z wykorzystaniem wspólnej architektury i komponentów wielokrotnego użytku, które są konfigurowane dla każdej wersji produktu;
- tworzenie i rozwój systemów poprzez konfigurację gotowego oprogramowania
- czym jest komponent oprogramowania, który może być zawarty w programie jako element wykonywalny;
- kluczowe elementy modeli komponentów i ich wsparcie przez oprogramowanie middleware;
- procesy oparte na komponentach (CBSE) do ponownego wykorzystania oraz z ponownym wykorzystaniem;
- różne rodzaje kompozycji komponentów i ich wady oraz zalety;
- kluczowe zagadnienia, które należy wziąć pod uwagę przy projektowaniu i wdrażaniu rozproszonych systemów oprogramowania;
- model obliczeniowy klient-serwer i warstwową architekturę systemów;
- powszechnie używane wzorce dla architektur systemów rozproszonych, typy systemów, dla których ma zastosowanie dany wzorzec architektoniczny;
- pojęcie oprogramowania jako usługi (SaaS), zapewniającej internetowy dostęp do zdalnie wdrażanych systemów aplikacyjnych;
- podstawowe pojęcia usługi sieciowej, standardów usług sieciowych i architektury zorientowanej na usługi;
- styl architektoniczny REST I i istotne różnice między usługami typu REST i SOAP;
- proces inżynierii usług, którego celem jest tworzenie usług sieciowych wielokrotnego użytku;

- system socjotechniczny i dlaczego kwestie ludzkie, społeczne i organizacyjne wpływają na wymagania i projektowanie systemów informatycznych;
- projektowani koncepcyjne i dlaczego jest to niezbędny pierwszy etap w procesie inżynierii systemów;
- dlaczego różne procesy zamówień systemowych są stosowane dla różnych typów systemów;
- kluczowe procesy rozwoju inżynierii systemów i ich relacjach;
- systemy systemów i czym różnią się on od systemu indywidualnego;
- systemy klasyfikacji systemów i różnice między różnymi typami systemów;
- dlaczego konwencjonalne metody inżynierii oprogramowania oparte na redukcjonizmie są nieadekwatne do tworzenia systemów;
- procesy inżynierii systemów i wzorców architektonicznych dla systemów.
- pojęcie oprogramowania wbudowanego, które służy do sterowania systemami reagującymi na zdarzenia zewnętrzne w ich otoczeniu;
- procesy projektowania systemów czasu rzeczywistego, w których systemy oprogramowania są zorganizowane jako zbiór współpracujących ze sobą procesów;
- wzorce architektoniczne, które są powszechnie stosowane w projektowaniu wbudowanych systemów czasu rzeczywistego;
- organizacja systemów operacyjnych czasu rzeczywistego i rola, jaką odgrywają we wbudowanym systemie czasu rzeczywistego;
- podstawowe zadania kierowania projektami informatycznymi;
- zarządzanie ryzykami, które mogą wystąpić w projektach informatycznych;
- czynniki wpływające na motywację osobistą i ich znaczenie dla kierowania projektami;
- kluczowe kwestie wpływające na pracę zespołową, takie jak skład zespołu, organizacja i komunikacja.
- podstawy kalkulacji kosztów oprogramowania i czynniki wpływające na cenę systemu oprogramowania;
- co powinno się znaleźć w planie projektu tworzonym w ramach procesu rozwoju typu plan-driven;
- na czym polega planowanie projektów oraz wykorzystanie wykresów do przedstawienia harmonogramu projektu;
- zwinne metody planowania projektów;
- procesy zarządzania jakością i wiedzą, dlaczego planowanie jakości jest ważne;
- normy w zarządzaniu jakością, w jaki sposób normy są stosowane w zapewnianiu jakości;
- przeglądy i inspekcje wykorzystywane jako mechanizm zapewniania jakości oprogramowania;
- zarządzanie jakością w metodach zwinnych opiera się na rozwoju kultury jakości zespołu;
- podstawowe funkcje, które powinien zapewniać system kontroli wersji oraz sposób, w jaki jest to realizowane w systemach scentralizowanych i rozproszonych;
- wyzwania związane z budowaniem systemu oraz korzyści płynące z ciągłej integracji;
- dlaczego zarządzanie zmianami w oprogramowaniu jest ważne oraz podstawowe czynności w procesie zarządzania zmianami;
- podstawy zarządzania wersjami oprogramowania;

Nazwa zajęć: Kryptografia kwantowa i postkwantowa

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka

w zakresie wiedzy:

1. zna protokoły dystrybucji kluczy kwantowych oraz prawa fizyki, na których są oparte.
2. zna algorytmy i kierunki rozwoju kryptografii postkwantowej (quantum secure) oraz założenia obliczeniowe, na których są oparte.
3. zna algorytmy kwantowe i klasyczne stosowane w kryptoanalizie kodów klasycznych i rozumie założenia obliczeniowe, na których są oparte

w zakresie umiejętności:

1. potrafi przeanalizować i ocenić przydatność kwantowych protokołów kryptograficznych w obecności szumu.
2. potrafi przeanalizować i ocenić przydatność kwantowych algorytmów kryptoanalitycznych w odniesieniu do ograniczeń sprzętowych.
3. potrafi modelować lub implementować protokoły kryptograficzne kwantowe i postkwantowe (quantum-secure)

w zakresie kompetencji społecznych:

1. jest gotów/gotowa wytłumaczyć potrzebę rozwoju technik kryptografii kwantowej i postkwantowej w kontekście rozwoju informatyki kwantowej, publikowania nowych standardów zabezpieczeń cyfrowych oraz zaleceń komisji eksperckich.

Treści programowe dla zajęć:

Wstęp do kryptografii kwantowej:1. Zastosowanie zakazu klonowania do bezpiecznego przesyłania informacji.2. Protokół BB84 dystrybucji kluczy kwantowych.3. Implementacje optyczne protokołu BB84.4. Protokół Wiesnera pieniędzy kwantowych.5. Optyczna implementacja pieniędzy kwantowych.6. Protokoły Shora i Aaronsona pieniędzy kwantowych.

Protokoły dystrybucji kluczy kwantowych:1. Protokół BB84 - krótkie przypomnienie,2. Protokół Ekerta E91 z wykorzystaniem stanów splątanych.3. Protokół Bennetta B92 z wykorzystaniem interferometrów Macha-Zehndera.4. Protokół Renesa R04.5. Implementacje protokołów BB84 i E91 z wykorzystaniem satelity kwantowego.

Zalecane długości klucza publicznego. Wyzwania i nagrody RSA.

Klasyczne algorytmy faktoryzacji liczb:1. Sito Eratostenesa,2. Metoda Monte Carlo,3. Metoda Fermata,4. Uogólniona metoda Fermata,5. Metoda Legendre'a ułamków łańcuchowych,6. Metoda sita kwadratowego,7. Porównanie efektywności algorytmów faktoryzacji liczb.

Podstawowe algorytmy kwantowe w kryptoanalizie kodów klasycznych:1. Algorytm Grovera wyszukiwania klucza w szyfrach symetrycznych.2. Kwantowa transformata Fouriera.3. Algorytm Simona.4. Algorytm Shora faktoryzacji i jego zastosowanie do łamania kryptosystemów RSA i Rabina.5. Algorytm Shora liczenia logarytmów dyskretnych i jego zastosowanie do łamania kryptosystemów ElGamala.

Niekonwencjonalne algorytmy kwantowe i DNA w kryptoanalizie kodów klasycznych:1. Algorytm faktoryzacji metodą sum Gaussa i kociąt Schroedingera.2. Implementacja tego algorytmu z wykorzystaniem spektroskopii NMR.3. Kwantowe wyżarzanie (ang. quantum annealing).4. Implementacja kwantowego wyżarzania z wykorzystaniem nadprzewodzących kubitów.5. Algorytm Adlemana oparty na obliczeniach DNA i technikach biochemicznych.

Liczby pierwsze:1. Liczby pierwsze Mersenne'a.2. Great Internet Mersenne Prime Search (GIMPS).3. Test Lucasa-Lehmera liczb Mersenne'a.4. Spirala Ulama - spirala liczb pierwszych.

Hipoteza Riemanna i liczby pierwsze:1. Funkcja Z Eulera.2. Funkcja zeta Riemanna.3. Problemy milenijne (ang. Millennium Problems).4. Zera funkcji zeta Riemanna i wartości własne hamiltonianowi.5. PT-symetryczna mechanika kwantowa Bendera.6. Problem Riemanna i komunikacja nadświetlna.

Złożoność obliczeniowa problemów w kryptografii:1. Deterministyczna maszyna Turinga i problemy typu P (algorytmy wielomianowe).2. Niedeterministyczna maszyna Turinga i problemy typu NP (niedeterministyczny alg. czasu wielomianowego).3. Problemy typu NTIME, NP, NEXPTIME, NSPACE, NPSPACE i NEXPSPACE.4. Problemy NP trudne (ang. NP hard).5. Problemy zupełne w klasie NP (problemy NP-zupełne, ang. NP complete).6. Hipoteza $P = NP$.7. Uniwersalna maszyna Turinga.8. Kwantowa maszyna Turinga = uniwersalny komputer kwantowy.9. Problemy typu BQP (ang. Bounded-error Quantum Polynomial-time).

Problemy NP-trudne w kryptografii:1. Kryptosystem McEliece'a.2. NTRUEncrypt.3. Kryptosystem Merkle'a-Hellmana.4. Złożoność obliczeniowa algorytmów plecakowych.5. Czy faktoryzacja liczb jest problemem NP-zupełnym?

Kryptografia postkwantowa, tj. kryptografia klasyczna odporna na kryptoanalizę kwantową metodą Shora.

Podstawowe kierunki rozwoju kryptografii postkwantowej:1. Szyfry kratowe (ang. lattice-based ciphers)2. Szyfry wielowartościowe (ang. Multivariate ciphers)3. Szyfry oparte na funkcjach skrótu (ang. hash-based ciphers)4. Szyfry oparte na kodach (ang. code-based ciphers, np. kryptosystem McEliece'a)5. Szyfry na krzywych eliptycznych (np. supersingular elliptic curve isogeny ciphers)6. Szyfry symetryczne.

Standardy kryptografii postkwantowej. Technologie kwantowe I i II generacji. Przyszłość kryptografii kwantowej.

Nazwa zajęć: **User Experience i budowanie użytecznych interfejsów**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna zagadnienia User Experience i zasady tworzenia interfejsów.

w zakresie umiejętności:

1. potrafi zaprojektować wygodny interfejs użytkownika ze szczególnym uwzględnieniem aplikacji internetowych.

2. potrafi rozwiązywać problemy grafiki komputerowej i komunikacji człowiek-komputer.

Treści programowe dla zajęć:

Podstawy UX
Podstawy UI
Omówienie narzędzia Figma
Figma - komponenty
Dobre praktyki UI
Ćwiczenia z narzędzia Figma
Warsztaty projektowania UX
Tworzenie projektów UI w wersji Web
Tworzenie projektów UI w wersji Mobile

Nazwa zajęć: **Muzyka algorytmiczna**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. Zna współczesną terminologię muzyczną.
2. Zna protokoły służące do obsługi dźwięków.
3. Zna podstawowe pojęcia z akustyki.

w zakresie umiejętności:

1. Potrafi wykorzystać w implementacji istniejące biblioteki muzyczne.
2. Opanował podstawowe techniki współczesnej kompozycji.
3. Stosuje elementarne twierdzenia matematyczne w procesie generowania muzyki.
4. Rozwija własne zainteresowania muzyczne.

w zakresie kompetencji społecznych:

1. Potrafi pracować w zespole.
2. Opanował podstawowe zasady zachowania scenicznego.

Treści programowe dla zajęć:

Wprowadzenie do języka Ruby.
Muzyka tonalna i atonalna.
Struktura rytmiczna w dziele muzycznym.
Minimalizm, Serializm i forma w muzyce.
Koncepcja skali muzycznej i algorytmiczna interpretacja.
Kompozycja z wykorzystaniem przekształceń liniowych.
Wykorzystanie Łańcuchów Markowa w kompozycji algorytmicznej.
Synteza i brzmienie dźwięku.
Protokół MIDI.
SoundDesign.
Kompozycja zespołowa.

Nazwa zajęć: **Inżynierski projekt zespołowy 1**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. Potrafi przeprowadzić proces formowania zespołu projektowego
2. Potrafi kontaktować się z klientem/grupą docelową w celu określenia i weryfikacji zakresu projektu informatycznego
3. Potrafi zdefiniować kryteria akceptacji dla projektu informatycznego
4. Potrafi wizualizować system informatyczny za pomocą makiety/prototypu
5. Potrafi dokonać publicznej prezentacji i demonstracji systemu informatycznego
6. Potrafi uczestniczyć w projekcie zespołowym prowadzonym metodami zwinnymi
7. Potrafi zaprojektować użyteczny system informatyczny
8. Potrafi implementować fragmenty systemu informatycznego w celu realizacji wymagań projektowych
9. Potrafi korzystać z narzędzi wspierających projekty informatyczne

w zakresie kompetencji społecznych:

1. Jest gotów/gotowa do podejmowania krytycznych decyzji w zakresie wykorzystania danych technologii w projekcie informatycznym
2. Jest gotów/gotowa do dyskusji w środowisku zawodowym o projekcie informatycznym stosując przy tym terminologię specjalistyczną

Treści programowe dla zajęć:

Formowanie zespołu projektowego.
Wizja projektu.
Prototyp produktu projektu.
Zakres projektu.

Wybór architektury systemu, narzędzi i metodyki pracy.
Implementacja projektu.
Przygotowanie projektu do publicznej prezentacji.

Nazwa zajęć: **Elektronika**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie wiedzy:

1. zna budowę i działanie tranzystorów bipolarnych i unipolarnych
2. zna wady i zalety różnych wzmacniaczy
3. zna układy pomiarowe w elektronice
4. zna podstawowe układy cyfrowe
5. zna działanie przetworników cyfrowo-analogowych i analogowo-cyfrowych

Treści programowe dla zajęć:

Tranzystory bipolarne i unipolarne: typy, konfiguracje

Budowa i zastosowania lamp elektronowych

Układy scalone analogowe

Układy cyfrowe: bramki logiczne, dekodery, liczniki, synteza układów kombinacyjnych, prawa de Morgana

Działanie, parametry i zastosowania przetworników A/D & D/A

Nazwa zajęć: **Język angielski A2**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. porozumiewać się w rutynowych, prostych sytuacjach komunikacyjnych, wymagających jedynie bezpośredniej wymiany zdań na tematy znane i typowe. Potrafi w prosty sposób opisywać swoje pochodzenie i otoczenie, w którym żyje, a także poruszać sprawy związane z najważniejszymi potrzebami życia codziennego.
2. czytać ze zrozumieniem krótsze teksty w języku angielskim o charakterze ogólnym.
3. zrozumieć prosty oryginalny materiał audio lub wideo z życia codziennego, kulturalnego i społecznego, na poziomie ogólnym jak i wychwycić niezbędne szczegóły.

Treści programowe dla zajęć:

Czasy gramatyczne: Present Simple and Present Continuous, Past Simple and Past Continuous, Present Perfect and Present Perfect Continuous, Past Perfect oraz czasy przyszłe dla poziomu A2.

Inne struktury gramatyczne potrzebne do wyrażania różnorodnych treści i opinii (np. czasowniki modalne, przymiotniki, strona bierna, zdania warunkowe, mowa zależna) dla poziomu A2.

Słownictwo dotyczące życia codziennego oraz związane z bezpośrednim środowiskiem studenta (jedzenie, osobowość, podróże, zainteresowania, edukacja, zakupy, pieniądze, technologia, rodzina, studia, praca, technologia, podstawowe słownictwo związane z kierunkiem studiów).

Strategie efektywnego czytania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanymi słów.

Strategie efektywnego słuchania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanymi słów.

Wyrażanie różnorodnych funkcji językowych np. prośby, opisy, wyrażanie opinii, wyrażanie zgody, brak zgody, pytania o pozwolenie, skargi, itp.

Nazwa zajęć: **Język angielski B1**

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. tworzyć ustne wypowiedzi na przygotowane tematy, prezentować i argumentować swoje stanowisko oraz innych osób w zakresie problematyki związanej ze swoim otoczeniem jak i w zakresie tematyki ogólno-akademickiej.
2. czytać ze zrozumieniem teksty w języku angielskim o charakterze ogólnym jak i akademickim oraz analizować ich treść i wybierać niezbędne informacje.
3. zrozumieć dostosowany do poziomu oryginalny materiał audio lub wideo na poziomie ogólnym oraz wychwytywać niezbędne szczegóły.

Treści programowe dla zajęć:

Czasy gramatyczne: Present Simple and Present Continuous, Narrative Tenses, Present Perfect and Present Perfect Continuous, Future Perfect and Future Continuous.

Inne struktury gramatyczne potrzebne do wyrażania różnorodnych treści i opinii: mowa zależna oraz pytania w mowie zależnej, formy przymiotnikowe i przysłówkowe.

Słownictwo dotyczące życia codziennego oraz jak i ogólno-akademickie w zakresie następujących tematów: praca, rozmowa kwalifikacyjna o pracę, służba zdrowia, podróżowanie, moda oraz dress code, środowisko naturalne, zmiany klimatyczne.

Strategie efektywnego czytania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanych słów w zakresie bloków tematycznych określonych w treści 3.

Strategie efektywnego słuchania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanych słów w zakresie bloków tematycznych określonych w treści 3.

Udzielanie odpowiedzi, udział w dyskusji oraz wyrażanie różnorodnych funkcji językowych w zakresie: przeprowadzania oraz udziału w rozmowie kwalifikacyjnej o pracę, przedstawiania problemów, moderowania dyskusji oraz wyrażania opinii na tematy zawarte w treści 3.

Nazwa zajęć: Język angielski B21

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. tworzyć ustne wypowiedzi na przygotowane tematy, prezentować i argumentować swoje stanowisko oraz innych osób na tematy związane ze swoim otoczeniem jak i na tematy ogólno-akademickie.
2. czytać ze zrozumieniem teksty w języku angielskim charakterze ogólnym jak i akademickim, związane z kierunkiem studiów, oraz analizować ich treść i wybierać niezbędne informacje.
3. zrozumieć oryginalny materiał audio lub wideo na większość tematów dotyczących życia codziennego, kulturalnego i społecznego, na poziomie ogólnym jak i wychwycić niezbędne szczegóły.
4. przygotować i wygłosić prezentację na wybrany temat.
5. opracować teksty oraz wypowiedzi dotyczące życia społecznego, uniwersyteckiego i zawodowego.
6. redagować wybrane teksty w stylu formalnym.
7. uzupełniać i doskonalić nabytą wiedzę i umiejętności.

Treści programowe dla zajęć:

Przegląd i utrwalenie umiejętności w zakresie posługiwania się formami i funkcjami czasów gramatycznych odpowiednich dla poziomu B2.

Inne struktury gramatyczne potrzebne do wyrażania różnorodnych treści i opinii: okresy warunkowe typ 1,2,3 oraz mieszane; struktury gramatyczne 'wish,'get used to/used to, past modals, formy bezokolicznikowe i imiesłowowe.

Słownictwo dotyczące problematyki współczesnego świata w zakresie następujących tematów: ekstremalne sytuacje, refleksja na temat planów życiowych, terapeutyczna funkcja muzyki, higiena snu, komunikacja niewerbalna oraz wybrane słownictwo akademickie i specjalistyczne związane z kierunkiem studiów.

Strategie efektywnego czytania w celu zrozumienia ogólnego sensu wypowiedzi w tekstach popularno-naukowych oraz specjalistycznych; domyślanie się znaczenia nieznanych słów w zakresie bloków tematycznych określonych w treści 3.

Strategie efektywnego słuchania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanych słów w zakresie bloków tematycznych określonych w treści 3.

Udzielanie odpowiedzi, udział w dyskusji oraz wyrażanie różnorodnych funkcji językowych w zakresie tematyki określonej w treści 3.

Redagowanie wybranych typów tekstów formalnych.

Nazwa zajęć: Język angielski B22

Po zakończeniu zajęć i potwierdzeniu osiągnięcia efektów uczenia się student/ka w zakresie umiejętności:

1. tworzyć ustne wypowiedzi na przygotowane tematy, prezentować i argumentować swoje stanowisko oraz innych osób na tematy związane ze swoim otoczeniem jak i na tematy ogólno-akademickie;
2. czytać ze zrozumieniem teksty w języku angielskim o charakterze ogólnym jak i akademickim, związane z kierunkiem studiów, oraz analizować ich treść i wybierać niezbędne informacje;
3. zrozumieć oryginalny materiał audio lub wideo na większość tematów dotyczących życia codziennego, kulturalnego i społecznego, na poziomie ogólnym jak i wychwycić niezbędne szczegóły;
4. przygotować i wygłosić prezentację na wybrany temat;
5. opracować teksty oraz wypowiedzi dotyczące życia społecznego, uniwersyteckiego i zawodowego;
6. redagować wybrane teksty w stylu formalnym;
7. uzupełniać i doskonalić nabytą wiedzę i umiejętności.

Treści programowe dla zajęć:

Przegląd i utrwalenie umiejętności w zakresie posługiwania się formami i funkcjami czasów gramatycznych odpowiednich dla poziomu B2.

Inne struktury gramatyczne potrzebne do wyrażania różnorodnych treści i opinii: strona bierna, następstwo czasów, zdania celu, porównania, rzeczowniki policzalne i niepoliczalne, przedimki.

Słownictwo dotyczące problematyki współczesnego świata w zakresie następujących tematów: system sprawiedliwości, przestępstwa internetowe, świat mediów i e-mediów, problematyka biznesu i ekonomii, reklamy, nowoczesne miasta, wystąpienia publiczne, problemy współczesnej nauki, tematyka science-fiction oraz wybrane słownictwo akademickie i specjalistyczne związane z kierunkiem studiów.

Strategie efektywnego czytania w celu zrozumienia ogólnego sensu wypowiedzi w tekstach popularno-naukowych oraz specjalistycznych; domyślanie się znaczenia nieznanymi słów w zakresie bloków tematycznych określonych w treści 3.

Strategie efektywnego słuchania w celu zrozumienia ogólnego sensu wypowiedzi; domyślanie się znaczenia nieznanymi słów w zakresie bloków tematycznych określonych w treści 3.

Udzielanie odpowiedzi, udział w dyskusji oraz wyrażanie różnorodnych funkcji językowych w zakresie tematyki określonej w treści 3.

Redagowanie wybranych typów tekstów formalnych.