

## Zagadnienia na egzamin dyplomowy

### INFORMATYKA I stopień

#### SPECJALNOŚĆ: INŻYNIERIA OPROGRAMOWANIA

##### Zaawansowane programowanie obiektowe (C++)

- Przegląd pojęć i właściwości programowania zorientowanego obiektowo (klasy, obiekty, hermetyzacja, dziedziczenie, polimorfizm) oraz sposoby ich realizacji w języku C++.
- Wielokrotne wykorzystanie kodu w C++ (klasy ze składowymi w postaci obiektów, dziedziczenie prywatne, dziedziczenie wielokrotne, szablony funkcji i klas). Polimorfizm dynamiczny i statyczny.
- Przegląd bibliotek C++ i ich możliwości wykorzystania. Metaprogramowanie w C++ (szablony, klasy cech i wytycznych).
- Programowanie generyczne z wykorzystaniem biblioteki STL (kontenery, iteratory, funktory i algorytmy). Kontenery specjalne i numeryczne.
- Programowanie generyczne z wykorzystaniem bibliotek Boost.
- Programowanie generyczne w QT.
- Obsługa WE/WY. Obsługa błędów i wyjątków.
- Klasa string i elementy bibliotek STL i Boost. Wyrażenia regularne.
- Zarządzanie pamięcią. Zapobieganie wyciekom zasobów - technika RAII. Inteligentne wskaźniki.
- Wyrażenia i funkcje lambda. Krotki w C++.
- Testowanie w C++ (CPPUnit).

##### Programowanie aplikacji internetowych

- Architektura aplikacji internetowej współpracującej z bazą danych. Metody protokołu HTTP. Nagłówki żądania i odpowiedzi HTTP. PHP jako przykład języka skryptowego działającego po stronie serwera. Podstawowe elementy języka. Metody pobierania danych z formularzy HTML.
- Typy danych w języku PHP. Tablice asocjacyjne. Operacje na plikach. Ciągi i wyrażenia regularne.
- Funkcje i szablony w PHP. Interakcja aplikacji internetowej z systemem plików na serwerze WWW.
- Programowanie obiektowe w PHP. Klasy i obiekty. Definiowanie składowych klasy. Metody i typy argumentów w metodach. Zarządzanie dostępem do klasy. Dziedziczenie. Metody i składowe statyczne. Klasy abstrakcyjne i interfejsy. Obsługa wyjątków.
- Współpraca aplikacji internetowej z bazami danych. Podstawowe zapytania SQL. Serwer MySQL i narzędzie administrowania bazami danych PHPMYAdmin. PHP i MySQL - interfejs mysqli i PDO.

- Zagrożenia transakcji internetowej. Szyfrowanie i certyfikaty cyfrowe. HTTP jako protokół bezstanowy. Pojęcie, zasada działania i funkcje sesji. Uwierzytelnianie w kontroli sesji.
- Strategia tworzenia bezpiecznych aplikacji internetowych. Rodzaje ataków sieciowych. Metody zabezpieczania aplikacji ich implementacja w PHP.
- Model modułowej aplikacji internetowej na przykładzie strony fikcyjnej firmy.
- Projektowanie obiektowe. Wprowadzenie do wzorców projektowych. Elementy języka UML. Aplikacje wielowarstwowe i wzorec projektowy MVC. Modele, widoki, kontrolery i ich implementacja w PHP.
- Transmisja danych z serwera w trybie asynchronicznym – dynamiczna interakcja z użytkownikiem za pomocą obiektu Ajax, jQuery i PHP.
- Przegląd aktualnych platform programistycznych do tworzenia aplikacji internetowych. Wstęp do mapowania obiektowo relacyjnego (ORM).

### **Zaawansowana inżynieria oprogramowania**

- Obiektowość w informatyce. Analiza i modelowanie obiektowe systemów informatycznych. Perspektywy modelu.
- Inżynieria i analiza wymagań. Obiektowa specyfikacja wymagań do systemu informatycznego. Persony i historyjki użytkownika. Scenariusze przypadków użycia.
- Język UML – historia i stan obecny. Ogólne zasady języka UML. Istota i typologia diagramów, związki na diagramach. Stereotypy.
- Perspektywa zewnętrzna. Model przypadków użycia. Podstawowe diagramy sekwencji.
- Perspektywa strukturalna. Modelowanie klas i ich związków. Interfejsy. Metoda CRC.
- Modelowanie dynamiki systemu – perspektywa zachowań. Diagramy czynności, stanów, sekwencji i współpracy.
- Modelowanie implementacji SI – diagram komponentów i diagramy wdrożeniowe.
- Profilowanie UMLa. Język OCL.
- Narzędzia CASE. Generowanie kodu. Inżynieria odwrotna oprogramowania – cele, metody, narzędzia.
- Wzorce projektowe: pojęcia podstawowe, zastosowania, klasyfikacja, przykłady. Wzorce w zarządzaniu zasobami i aplikacjach internetowych. Wzorce interfejsu.
- Techniki obiektowe w projektowaniu baz danych. ORM.
- Podejście komponentowe wytwarzania oprogramowania. FDD i TDD. Inżynieria oprogramowania zorientowana na serwisy. Inżynieria oprogramowania systemów czasu rzeczywistego. Inżynieria niezawodności i bezpieczeństwa systemów informatycznych.

### **Programowanie w języku SWIFT**

- Zmienne, stałe, typy liczbowe i literały.
- Napisy i literały znakowe.
- Instrukcje sterujące: warunkowe i iteracyjne.
- Definicje i deklaracje tablic, wyliczenia.
- Złożone typy danych: struktury, zbiory, słowniki.
- Funkcje: definicja, parametry, typ i wywołanie.

- Paradygmaty programowania zorientowanego obiektowo.
- Definicja klasy i jej składowe, dziedziczenie klas.
- Użycie protokołów.
- Posługiwanie się złożonymi typami danych i możliwości obiektowe.
- Korzystanie z kontrolek, obsługa akcji i delegacji.
- Tworzenie komentarzy i pisanie testów jednostkowych.

### **Zarządzanie bazami SQL i NoSQL**

- Wprowadzenie do zarządzania bazami danych. Relacyjny model bazy danych.
- Modelowanie struktury relacyjnego modelu danych.
- Zarządzanie i administracja bazami danych SQL na przykładzie wybranych Systemów
- Zarządzania Bazami Danych (SZBD).
- Narzędzia do zarządzania. Mechanizmy bezpieczeństwa w bazach danych. Zarządzanie użytkownikami i rolami. Import i eksport danych, zarządzanie kopiami zapasowymi.
- Wprowadzenie do nierelacyjnych baz danych. Podstawowe pojęcia. Typy baz danych NoSQL.
- Omówienie rodzajów nierelacyjnych baz danych na przykładach. Analiza porównawcza. Zarządzanie i administracja bazami danych NoSQL.
- Mechanizmy bezpieczeństwa w bazach danych. Zarządzanie użytkownikami i rolami. Import i eksport danych, zarządzanie kopiami zapasowymi.
- Język zapytań dla baz SQL.
- Języki zapytań dla baz NoSQL.

### **Integracja systemów**

- Podstawowe pojęcia i zasady integracji systemów.
- Style integracji aplikacji – transfery danych, współdzielenie danych i funkcjonalności.
- Rodzaje architektur integracyjnych.
- Wzorce integracji: informacji, aplikacji, danych, usług.
- Architektura sterowana zdarzeniami, przetwarzanie sieciowe i chmurowe.
- Podstawowe technologie integracyjne. Narzędzia stosowane w integracji systemów.
- Standardy zapisu i wymiany dokumentów.
- Systemy wymiany wiadomości w systemach rozproszonych - MQ.
- Usługi sieciowe SOAP i REST, standardy tworzenia serwisów.
- Integracja aplikacji chmurowych, usługi chmurowe wykorzystywane do integracji systemów.
- Integracja systemów wykorzystujących elementy Internetu Rzeczy.

### **Szkielety programistyczne w aplikacjach internetowych**

- Webowe szkielety programistyczne – koncepcje, architektury, technologie.
- Języki programowania aplikacji internetowych – standardy, rozszerzenia,
- paradygmaty.
- Laravel - szkielet programistyczny do budowy warstwy usług aplikacji internetowych.
- Warstwa danych – interakcja z bazami danych (operacje CRUD).

- Szkielet programistyczny Angulardo budowy warstwy prezentacji aplikacji internetowych.
- Biblioteka React do budowy warstwy prezentacji aplikacji internetowych.
- Interfejs programistyczny aplikacji internetowych (API) typu REST.
- Zarządzanie użytkownikami i bezpieczeństwo.
- Diagnostyka błędów i testowanie.
- Wdrażanie, wydajność i optymalizacja.

### **Programowanie aplikacji w chmurze obliczeniowej**

- Pojęcie chmury, rodzaje chmur, koncepcja usług w chmurze.
- Wykorzystanie technik wirtualizacji w chmurach komputerowych.
- Obsługa pamięci masowych w chmurze komputerowej.
- Rozwiązania chmurowe w systemach Linux.
- Paradygmat programowania równoległego MapReduce, system Hadoop jako przykład programowania w chmurze.
- Elementy aplikacji w chmurze komputerowej.
- Wykorzystanie języka Java do programowania w chmurze komputerowej.
- Usługi Pig i Hive jako elementy chmury komputerowej.
- Chmura komputerowa Amazon.
- Rozwiązania chmurowe firm Google oraz Microsoft.

### **Programowanie aplikacji mobilnych na platformę iOS**

- Wprowadzenie do programowania mobilnego na platformie iOS.
- Architektura systemu iOS.
- Wzorzec Model-Widok-Kontroler.
- Wytyczne dotyczące interfejsu użytkownika Apple iOS (SwiftUI vs. UIKit).
- Środowisko Xcode, Storyboard.
- Omówienie kontrolera i widoku.
- Emulatory – zastosowanie i ograniczenia.
- Zastosowanie systemu kontroli wersji Git.
- Wytwarzanie prostych aplikacji iOS.
- Wytwarzanie aplikacji opartych o widok tabeli.
- Wytwarzanie aplikacji opartych o mapy i lokalizacje.
- Wytwarzanie aplikacji z obsługą gestów.
- Wytwarzanie aplikacji z obsługą sieci.
- Lekka baza danych. Zarządzanie danymi.
- Testowanie i debugowanie aplikacji.
- Zarządzanie pamięcią. Dystrybucja aplikacji.

### **Programowanie aplikacji mobilnych na platformę Android**

- Wprowadzenie do platformy Android. Obsługiwane typy urządzeń.
- Środowisko programisty aplikacji urządzeń mobilnych. Dostępne narzędzia i ich możliwości.
- Wstęp do tworzenia aplikacji mobilnych – elementy projektu aplikacji dla platformy Android.

- Tworzenie graficznego interfejsu użytkownika. Sposoby przekazywania danych pomiędzy elementami aplikacji.
- Sposoby trwałego przechowywania danych na potrzeby aplikacji mobilnych.
- Wielozadaniowość na platformie Android. Komunikacja składników aplikacji.
- Podstawy komunikacji sieciowej na platformie Android.
- Wykorzystanie usług Google Play.
- Modyfikacja standardowego wyglądu komponentów graficznych.
- Wstęp do multimediiów na platformie Android.
- Wstęp do grafiki na platformie Android.
- Różnice pomiędzy aplikacjami dla różnych typów urządzeń.

### **Interakcja człowiek-komputer**

- Wstęp do koncepcji interakcji człowiek-komputer i pojęcia interfejsu. Elementy kognitywistyki. Wysiłek poznawczy i zapamiętywalność. Teoretyczne modele: zachowań i ukierunkowanych działań człowieka, Rasmussen, ICS, Fitsa, KLM, Hicka. Implikacje wynikające z modeli. Ergonomia interfejsu.
- Elementy interfejsu oprogramowania. Typy i obiekty interfejsu, mechanizmy interakcji, style dialogu i interakcji (TUI, GUI, CLI, WIMP, MUI, NUI). Środowisko fizyczne i fizyczny interfejs. Złote zasady Schneidermana.
- Projektowanie interfejsu. Profile użytkownika. Analiza potrzeb: analiza zadań i aktywności użytkownika. Modelowanie pracy użytkownika i projektowanie UX. Metody projektowania (szkice, storyboards, mockupy, metody formalne, prototypy). Narzędzia wspomagające projektowanie interfejsu oprogramowania.
- Jakość interfejsu (kryteria jakości, rekomendacje). Użyteczność i dostępność. Ocena jakości interfejsów oprogramowania. Testowanie jakości interfejsów. Ocena z udziałem użytkowników (metody, organizacja eksperymentów, eye-tracking, klik-tracking). Metody oceny bez udziału użytkownika (ocena heurystyczna Nielsena-Molicha, LUT i WUP), wykorzystanie rekomendacji (WCAG, W3C), metoda ekspercka). Narzędzia wspomagające ocenę jakości interfejsu oprogramowania.

### **Zaawansowane programowanie w Javie**

- Wykorzystanie plików XML.
- Polimorfizm.
- Internacjonalizacja.
- Wybrane wzorce projektowe.
- Bezpieczeństwo.
- Biblioteka Lombok.
- Wprowadzenie do biblioteki Spring.
- Wstrzykiwanie zależności.
- Wybrane moduły biblioteki Spring.
- Wzorzec MVC.
- Biblioteka Hibernate.
- Biblioteka JSoup.
- Biblioteka JUnit.
- Metody macierzyste.

- Programowanie usług.

### **Architektura i programowanie w .NET**

- Wprowadzanie do języka C# - paradygmat obiektowy, funkcyjny i zdarzeniowy. Tworzenie klas, metod i funkcji lambda. Typy danych, właściwości i główne różnice pomiędzy językiem C# a innymi językami obiektowymi. Mechanizm dostępu do danych LINQ.
- Platforma .NET i architektura aplikacji platformy. Aplikacje okienkowe, mobilne, wbudowane i internetowe przy wykorzystaniu platformy .NET.
- Wprowadzenie do architektury i problematyki aplikacji internetowych. Składniki podstawowej architektury aplikacji internetowych. Architektura Model-Widok-Kontroler. Platforma ASP.NET.
- Generowanie widoków aplikacji oraz silnik widoków Razor. Wykorzystanie modeli pośrednich i obiektów DTO.
- Tworzenie aplikacji korzystających z silnika Razor Pages.
- Wstrzykiwanie zależności oraz wykorzystanie dodatkowych bibliotek platformy .NET. Wprowadzenie do problematyki mapowania relacyjno-obiektowego (ORM). Połączenia z systemami bazodanowymi na platformie .NET. Wykorzystanie mechanizmu sekretów do oddzielenia danych konfiguracyjnych aplikacji.
- Architektura zorientowana na usługi na przykładzie REST. Tworzenie aplikacji typu API w architekturze REST. Mechanizmy uwierzytelniania z wykorzystaniem nazw użytkowników i haseł oraz nagłówek uwierzytelniających. Pamięć podręczna oraz własne generowanie odpowiedzi aplikacji.
- Testy jednostkowe oraz integracyjne aplikacji platformy ASP.NET.
- Wdrażanie aplikacji przy wykorzystaniu konteneryzacji.

### **Hurtownie i eksploracja danych**

- Cele i struktura systemów Online Analytical Processing (OLAP). Systemy OLAP i systemy transakcyjne: cechy wspólne i różnice.
- Logiczny i fizyczny model danych w bazach wykorzystywanych w hurtowniach danych.
- Metodologia budowy hurtowni danych w modelu OLAP. Metadane w modelu OLAP. Narzędzia wykorzystywane w budowie hurtowni danych.
- Eksploracja danych – Data Mining, organizacja danych na potrzeby ich eksploracji.
- Algorytmy i metody eksploracji danych –klasyfikacja, regresja.
- Algorytmy i metody eksploracji danych – grupowanie, sekwencjonowanie.
- Algorytmy i metody eksploracji danych – analiza przebiegów czasowych.
- Algorytmy i metody eksploracji danych – asocjacje i charakterystyki.
- Algorytmy i metody eksploracji danych – drzewa decyzyjne i ich indukcja.
- Proces eksploracji danych.
- Przykłady zastosowań eksploracji danych – eksploracja danych tekstowych (TextMining).
- Przykłady zastosowań eksploracji danych – eksploracja danych internetowych (Web Mining).
- Narzędzia eksploracji danych.

## **Procesy wytwarzania oprogramowania**

- Wprowadzenie do systemów kontroli wersji.
- Rozproszone systemy kontroli wersji.
- Dobre praktyki w programowaniu obiektowym.
- Refaktoryzacja kodu oprogramowania.
- Metodyka Test-Driven Development w procesie wytwarzania oprogramowania.
- Testy jednostkowe.
- Testy integracyjne.
- Testy funkcjonalności.
- Zapewnianie jakości oprogramowania i ciągła integracja.

## **Zarządzanie projektami wytwarzania oprogramowania**

- Zarządzanie projektami. Procesy realizacji projektów i zarządzanie nimi. Cecha SMART. Podejście metodyczne. Klasyczne zarządzanie projektami. Role uczestników projektów. Metodyki PMI i PRINCE2.
- Wytwarzanie oprogramowania. Podejścia klasyczne i zwinne. Programowanie Ekstremalne. Podejścia: FDD, TDD, LSD i DSDM. Wybrane zasady i techniki zapewnienia wysokiej jakości kodu. Refaktoryzacja.
- Metodyki zwinne. Manifest zwinności. Podejście zwinne i projekty wytwarzania oprogramowania. Metodyki SCRUM, Kanban, APM. Ekstremalne zarządzanie projektami. Certyfikacja kompetencji. Wdrażanie praktyk zwinnych. Agile Coaching.
- Szacowanie i planowanie działań w metodykach zwinnych. Metryki estymacji wielkości projektów. Techniki planowania wydań i iteracji. Metody szacowania projektów zwinnych. Modyfikacja oszacowań. Budżetowanie projektów wytwarzania oprogramowania.

## **Utrzymanie i rozwój oprogramowania**

- Procesy związane z utrzymaniem oprogramowania. Typy oprogramowania i ich wpływ na procesy jego utrzymania. Problemy kosztów posiadania oprogramowania (TCO) i współczesne tendencje ich obniżenia. Modele ASP. Centra przetwarzania i chmury. Organizacja wspomagania użytkownika – modele, przykłady. Problemy bezpieczeństwa w modelach ASP.
- Umowy utrzymania oprogramowania i infrastruktury. Katalogi usług i ich grupy. Poziomy usług. SLA i parametry w niej definiowane.
- Analiza wydajności i kosztów hostingu oprogramowania. Metryki jakości i sposoby ich pomiaru.
- Zarządzanie zasobami i zadaniami w systemach utrzymania oprogramowania. Optymalizacja procesów – definiowanie kryteriów, pomiar, metody wielokryterialnej optymalizacji dynamicznej. Modelowanie i symulacja zarządzania zadaniami w centrum przetwarzania. Oprogramowanie do symulacji systemów dyskretnych. Metodyka budowy modeli, symulacji i analizy wyników.
- Rozwój oprogramowania – problemy, zadania, strategie, metody i zagrożenia. Zarządzanie modernizacją i reorganizacją systemów informatycznych. Problemy logistyczne i wersjonowania oprogramowania. Analiza i optymalizacja kosztów modyfikacji – modele.

- Integracja systemów informatycznych. Pojęcie integracja. Metody i techniki integracji istniejących systemów informatycznych. Architektura korporacyjna. Interoperacyjność oprogramowania i metody jej zapewnienia. Prawne i organizacyjne aspekty interoperacyjności. Ocena dojrzałości interoperacyjności – modele i metody.
- Metodyki zarządzania infrastrukturą informatyczną. Biblioteka ITIL: fazy, procesy i funkcje.

### **Informatyzacja przedsiębiorstw**

- Dane, informacje, systemy informacyjne a informatyczne w przedsiębiorstwie. Istota i elementy SIZ.
- Typologia i zakres zastosowań SIZ. Klasy systemów informatycznych.
- Cykle życia SIZ. Strategie informatyzacji przedsiębiorstw.
- Organizacja procesu kompleksowej informatyzacji przedsiębiorstwa.
- Metody i narzędzia analizy przedwdrożeniowej
- Metodyka wyboru powielarnego SIZ z pośród ofert rynkowych
- Zarządzanie wdrażaniem systemu informatycznego
- Eksploatacja systemów informatycznych w przedsiębiorstwie
- Outsourcing jako alternatywa informatyzacji
- Problemy informatyzacji przedsiębiorstw

### **Badania operacyjne**

- Wprowadzenie do badań operacyjnych, własności zbiorów i funkcji wypukłych.
- Programowanie liniowe – wybór asortymentu, problem mieszanki, problem diety.
- Metoda SIMPLEX.
- Klasyczne metody minimalizacji funkcji.
- Programowanie sieciowe – metoda CPM, PERT, GERT.
- Algorytmy genetyczne, pojęcia podstawowe, działanie metody.
- Algorytmy genetyczne, operacje krzyżowania, mutacji, funkcja przystosowania.
- Wprowadzenie do metod ewolucyjnych, pojęcia podstawowe, właściwości, zastosowanie, przykładowe rozwiązania.
- Algorytmy inteligencji stadnej, Particle Swarm Optimization (PSO).
- Metody optymalizacji wielokryterialnej (OW), pojęcia podstawowe, metody z funkcją dystansową.
- Metody OW – metody skalaryzacji i metody leksykograficzne.
- Metody OW bazujące na relacjach przewyższania, pojęcia podstawowe, metoda ELECTRE I, ELECTRE II, ORESTE.
- Gry dwuosobowe o sumie zerowej oraz gry z naturą.
- Zagadnienie kolejek, pojedyncze i wielokrotne kanały obsługi. Wprowadzenie do zmiennych lingwistycznych. Zaliczenie.

### **Komponentowe podejście do wytwarzania aplikacji**

- Zagadnienia konstrukcji oprogramowania komponentowego. Miejsce technologii komponentowej w cyklu życia oprogramowania. Zalety i wady. Przegląd środowisk programowania komponentowego (CCM, EJB, COM+, .Net, ...). Architektura wielowarstwowa. Serwery aplikacji. Systemy dedykowane.

- Salesforce. Wprowadzenie. Przegląd architektury, systemu i produktów.
- Praca w Force.com. Model przetwarzania (MVC, sandboxes, typy aplikacji). Cykl życia oprogramowania. Autentykacja. Platforma Trailhead.
- Tworzenie bazy danych w Force.com. Typy obiektów i danych. Pola. Powiązania.
- Język programowania Apex. Podstawowe konstrukcje. Zapytania do bazy danych. Wyzwalacze. Procesy asynchroniczne i wsadowe. Ograniczenia. Kolejność wykonania zadań.
- Interfejs użytkownika. Typy interfejsu: klasyczny i błyskawiczny. Narzędzia Visualforce.
- Logika biznesowa. Narzędzia. Warstwa bazy danych – reguły poprawności, formuły. Wzorce. Procesy pracy: podstawy, tworzenie, programowanie.
- Zarządzanie danymi. Identyfikator rekordu. Operacje bazodanowe. Ładowanie danych. Połączenie z serwerem. Wspomaganie manipulacji danymi.
- Analityka i raportowanie. Wyświetlanie tabel i wykresów. Analiza danych z wykorzystaniem Einstein Analytics and Einstein Discovery.
- Administrowanie aplikacjami. Model bezpieczeństwa na poziomach obiektów i pól. Kontrola dostępu. Licencjonowanie. Lokalizacja.