

Zagadnienia na egzamin magisterski na kierunku Informatyka (dla studentów studiów stacjonarnych i niestacjonarnych II stopnia)

Przygotowanie i publikowanie artykułów naukowych

1. Schemat IMRaD.
2. Identyfikator ORCID.
3. Rodzaje artykułów naukowych.
4. Wskaźniki bibliometryczne. Lista Filadelfijska.
5. Bazy danych bibliograficznych. Baza Scopus.

Zaawansowana eksploracja danych

1. Metody identyfikacji obserwacji odstających.
2. Metody estymacji gęstości rozkładu prawdopodobieństwa.
3. ANOVA, MANOVA.
4. Modele regresji.
5. Metody redukcji wymiaru i liczności próby.
6. Metody analizy skupień.
7. Filtry splotowe. Sieci konwolucyjne.
8. Metody uczenia maszynowego.
9. Ocena jakości modeli klasyfikacyjnych.
10. Ocena jakości modeli regresyjnych i prognozujących.

Metody wnioskowania wielokryterialnego

1. Jakie są główne zadania normalizacji wartości analizowanych kryteriów optymalizacji.
2. Na czym polega metoda leksykograficzna.
3. W jaki sposób wyznaczane są wagi ważności kryteriów w metodzie AHP.
4. Opisz warianty należące do zbioru wariantów optymalnych w sensie Pareto.
5. Scharakteryzuj metodę Blina.

Programowanie aplikacji w języku PL/SQL

1. Omówić ogólną budowę bloku PL/SQL.
2. Czym są kursory i do czego się je wykorzystuje?
3. Jakie działania realizuje pętla kursora?
4. Czym są wyjątki i jak się nimi posługiwać w języku PL/SQL?
5. Opisać zasady definiowania i posługiwania się funkcjami i procedurami PL/SQL.
6. Jak się definiuje procedury i funkcje składowane w bazie danych? Do czego się je wykorzystuje?
7. Jakie są rodzaje kolekcji? Jak się je definiuje i do czego można je wykorzystać?
8. Co to są wyzwalacze bazy danych? Jak się je definiuje i do czego wykorzystuje?
9. Czym są pakiety definiowane z użyciem języka PL/SQL? Jak się je tworzy, co zawierają i jak się je wykorzystuje?
10. Czym charakteryzuje się dynamiczny SQL i PL/SQL? Podać przykład użycia dynamicznego SQL-a.

Projektowanie i implementacja systemów business intelligence

1. Wskazać różnice między hurtownią danych i data mart.
2. Opisać modele danych stosowane w hurtowniach danych.
3. Opisać możliwe architektury hurtowni danych.
4. Wymienić różnice między hurtownią danych i systemem klasy ERP.
5. Wyjaśnić pojęcia "Fakt", "Wymiar" oraz "Miara". Dla każdego z nich podać adekwatne przykłady.
6. Opisać procesy zachodzące w ramach działań ETL.
7. Do czego wykorzystuje się wymiary wolnozmiennie, jakie są ich rodzaje i czym one się charakteryzują?
8. Przedstawić architekturę systemu business intelligence.
9. Opisać operacje realizowane w ramach ROLAP.
10. Wymienić wiodące narzędzia business intelligence i opisać ich główne funkcjonalności.

Bezpieczeństwo aplikacji internetowych

1. Czym jest atak SQL Injection? Omów pojęcie i podaj przykłady.
2. Techniki obrony przed atakami SQL Injection.
3. Czym jest atak XSS? Omów pojęcie i podaj przykłady.
4. Porównaj metody uwierzytelniania użytkowników.
5. Omów powody rejestrowania aktywności użytkowników w systemie informatycznym.
6. Podaj przykład sposobu przechowywania uprawnień użytkowników w bazie danych.
7. Jaki jest cel testowania bezpieczeństwa aplikacji internetowej?
8. W jaki sposób serwer może rozpoznać tożsamość zalogowanego użytkownika aplikacji internetowej?

Programowanie aplikacji internetowych w JEE

1. Scharakteryzuj model aplikacji wielowarstwowej Java/Jakarta EE i podstawowe komponenty webowe.
2. Omów podstawowe mechanizmy zaimplementowane w Spring (IoC, DI, AOP).
3. Co to jest stereotyp w Spring? Wymień i scharakteryzuj najważniejsze z nich.
4. Jaka jest rola kontrolera w aplikacjach Spring Web MVC? Wymień i scharakteryzuj podstawowe adnotacje stosowane w klasie kontrolera.
5. Wymień i scharakteryzuj niezbędne adnotacje stosowane w klasie encji w Spring.
6. Jak wykorzystać i jakie metody udostępnia interfejs JpaRepository?
7. Jakie znasz metody dostępu do baz danych w aplikacjach internetowych na przykładzie Spring? Do czego służą wzorce DAO i DTO?
8. Na czym polega technologia odwzorowania obiektowo-relacyjnego (ORM)? Jakie są adnotacje dostępne w Spring stosowane do zdefiniowania zależności pomiędzy klasami encji?
9. Jak można Implementować mechanizmy bezpieczeństwa w Spring Boot?
10. Czym jest wzorzec REST? Wymień jego cechy i podaj przykład implementacji w Spring Boot.

Programowanie full-stack w chmurze obliczeniowej

1. Poziomy chmur komputerowych wykorzystywane w programowaniu aplikacji.
2. Programowanie full-stack w chmurze komputerowej, zasady i rozwiązania.
3. Usługi chmur komputerowych dostępne dla programistów full-stack.
4. Szkielety programistyczne wykorzystywane do programowania full-stack.
5. Programowanie full-stack aplikacji stanowych (ang. statefull) w chmurze komputerowej.
6. Programowanie full-stack aplikacji bezstanowych (ang. stateless) w chmurze komputerowej.
7. System Kubernetes w programowaniu full-stack.
8. Powiązanie aplikacji z chmurową bazą danych, sposoby i rodzaje wtyczek.
9. Zarządzanie użytkownikami i bezpieczeństwem w aplikacjach opracowywanych w chmurze komputerowej.
10. Integracja innych serwisów z aplikacją chmurową.

Zaawansowane programowanie w Swift

1. Omów wzorzec projektowy Model-View-ViewModel (MVVM).
2. Omów typy generyczne w Swift na wybranym przykładzie.
3. Omów zmienne funkcyjne w Swift. Podaj przykład.
4. Typ wyliczeniowy w Swift. Omów na wybranym przykładzie.
5. Rola i składnia protokołów w Swift.
6. Wyjaśnij pojęcie opcjonalności w Swift i przedstaw, jak mechanizmy rozpakowywania zmiennych (unwrapping) wpływają na bezpieczeństwo i stabilność kodu.
7. Omów znaczenie rozszerzeń (extensions) w Swift i przedstaw przykłady ich zastosowania w powiększaniu funkcjonalności istniejących typów.
8. Wyjaśnij, jak działają właściwości obserwowane (property observers) w Swift i w jaki sposób mogą być używane do monitorowania zmian wartości.
9. Wyjaśnij różnicę pomiędzy deklaracją zmiennej poprzedzoną dyrektywą @StateObject, a @ObservedObject.
10. Omów różnice między strukturami (struct) a klasami (class) w Swift oraz przedstaw przykłady zastosowań dla każdej z tych konstrukcji.

Zaawansowane programowanie aplikacji mobilnych na platformę Android

1. Wyjaśnij na czym polega mechanizm końcowej lambdy (ang. trailing lambda) w języku Kotlin. Jakie korzyści daje zastosowanie tego rozwiązania?
2. Omów rozwiązania stosowane w języku Kotlin mające na celu zmniejszenie ilości tworzonego kodu (klasy danych, funkcje rozszerzające, parametry nazwane i domyślne).
3. Czym są funkcje z adnotacją @Composable w Jetpack Compose? Wymień i omów ich cechy.
4. Omów właściwości procesu rekompozycji interfejsu użytkownika w Jetpack Compose.
5. Omów cykl życia elementów interfejsu użytkownika w kompozycji. Jakie czynniki mają wpływ na zachowanie elementów w kompozycji?
6. Na czym polega mechanizm wyciągania stanu (ang. state hoisting) w Jetpack Compose? Cemu on służy?
7. Stan jakiego rodzaju przechowuje się za pomocą funkcji remember(), osobnej klasy i obiektu ViewModel w Jetpack Compose? Omów zalety i wady tych rozwiązań.
8. Wymień i omów zastosowanie podstawowych komponentów w Material Design.
9. Na czym polegają podstawowe zasady tworzenia aplikacji w Jetpack Compose (zasada pojedynczego źródła prawdy i zasada jednokierunkowego przepływu danych).
10. Przedstaw i omów zalecaną architekturę współczesnej aplikacji mobilnej dla systemu Android.
11. Wymień i krótko omów najlepsze praktyki tworzenia aplikacji mobilnych dla systemu Android.
12. Wymień i krótko omów cechy właściciela stanu logiki biznesowej w aplikacji dla systemu Android?
13. Wymień i krótko omów cechy właściciela stanu logiki interfejsu użytkownika w aplikacji dla systemu Android?
14. Omów budowę warstwy danych w aplikacji dla systemu Android.
15. Wymień i krótko omów cechy warstwy domeny w aplikacji dla systemu Android.

Zaawansowane programowanie w języku Python

1. Co to są i do czego służą dekoratory funkcji?
2. Co to są metaklasy i jakie mają zastosowania w zaawansowanych aplikacjach?
3. Opisz mechanizm obsługi wyjątków w Pythonie- „try-except”.
4. Co to są i do czego służą klasy abstrakcyjne w języku Python?
5. W jakim celu w programowaniu w języku Python wykorzystuje się metody specjalne? Omów jedną z nich.
6. Czym są funkcje lambda w języku Python i jakie mają zastosowania?
7. Co to jest funkcja mapująca w języku Python?
8. Czym różnią się *args i **kwargs i do czego służą? Wyjaśnij ich działanie i podaj przykład funkcji, która wykorzystuje oba typy argumentów.
9. Jak działa list comprehension (składnia list składanych) w Pythonie?
10. Jak działa metoda __init__ w klasie Python? Wyjaśnij jej rolę w inicjalizacji obiektów.

Internet Rzeczy

1. Krótko opisz zagadnienie Internetu Rzeczy.
2. Co to jest magistrala, interfejs, protokół? Scharakteryzuj i opisz różnice.
3. Scharakteryzuj pojęcie mikrokontrolera i mikroprocesora. Podaj różnice między tymi pojęciami.
4. Wyjaśnij, co oznacza skrót CISC. Krótko opisz ten termin.
5. Wyjaśnij, co oznacza skrót RISC. Krótko opisz to pojęcie.
6. Opisz i podaj różnice pomiędzy UART a USRT.
7. Czym jest i co powoduje pojemność pasożytnicza?
8. Jakie cechy powinien posiadać sensor inteligentny?
9. Czym są i do czego służą aktuatory?
10. Opisz zasadę działania modulacji szerokości impulsów PWM (ang. pulse width modulation).

Wzorce projektowe i techniki pisania czystego kodu

1. Wymień i krótko scharakteryzuj konstrukcyjne wzorce projektowe.
2. Wymień i krótko scharakteryzuj strukturalne wzorce projektowe.
3. Wymień i krótko scharakteryzuj behawioralne wzorce projektowe.
4. Czym różni się wzorzec projektowy Proxy od wzorca Facade?
5. Czym różni się wzorzec projektowy Sample Factory od wzorca Factory Method?
6. Podaj zastosowania wzorca projektowego Flyweight.
7. Podaj przykłady zastosowania wzorca Chain of Responsibility.
8. Podaj techniki, które umożliwiają zaimplementowanie wzorca Singleton w Javie.
9. Co oznacza skrót SOLID i na czym polegają techniki kryjące się pod poszczególnymi literami skrótu?
10. Co mówią techniki pisania czystego kodu o komentarzach w kodzie źródłowym?
11. Co to jest refactoring i jak można go przeprowadzić?
12. Co to jest programowanie aspektowe i jak można je zrealizować w Javie?
13. Co to jest programowanie funkcyjne i jak można je zrealizować w Javie?
14. Jakie znasz zasady pisania czystego kodu w stosunku do nazw elementów programu, długości kodu oraz argumentów funkcji?
15. Co daje stosowanie abstrakcji w kodzie programu w kontekście jego utrzymania i rozwoju?

Bariery w przestrzeni cyfrowej

1. Projektowanie uniwersalne – idea, przepisy prawne, zasady.
2. Ergonomia interfejsów oprogramowania – definicja, obszary, typy i przykłady.
3. Użyteczność i dostępność interfejsu oprogramowania.
4. Technologie wspomagające osoby z niepełnosprawnościami.
5. Wytyczne dostępności treści internetowych WCAG 2.1 – zasady, poziomy, weryfikacja.
6. Metody oceny jakości interfejsu – klasyfikacja, typy metod.
7. Techniki oceny jakości interfejsów z udziałem i bez udziału użytkowników.
8. Metodyka SUS.
9. Ocena heurystyczna – heurystyki Nielsena-Molicha.
10. Okulografia – idea, istota, urządzenia, eksperyment, rezultaty.