

## **Matematyka dyskretna**

1. Reprezentacja liczb w komputerze, liczby całkowite bez znaku i ze znakiem, układ uzupełnieniowy U2.
2. Zasada szufladkowa Dirchleta. Zasada włączania-wyłączania.
3. Rekursja, definicje rekurencyjne funkcji, rekurencja liniowa.
4. Rekursywne typy danych, stosy i kolejki, algorytmy zdefiniowane rekurencyjnie, zasada dziel i rządź, sortowanie przez scalanie.
5. Grafy nieskierowane i skierowane, drzewa. Drzewa binarnych przeszukiwań, algorytmy wstawiania i wyszukiwania. Notacja polska beznawiasowa.
6. Spójność i acykliczność, drzewa rozpinające. Cykle i ścieżki Eulera i Hamiltona.
7. Reprezentacja drzew i grafów. Algorytmy grafowe, przeszukiwanie w głąb i wszerz, konstruowanie drzew rozpinających, w tym minimalnych, szukanie najkrótszej ścieżki.

## **Sieci komputerowe**

1. Model ISO-OSI stosu protokołów komunikacyjnych
2. Właściwości protokołu IP wersji 4 i 6
3. Routing statyczny i dynamiczny
4. Porównanie działania protokołów TCP i UDP
5. Charakterystyka modelu klient-serwer
6. Organizacja domen DNS w sieci Internet
7. Protokół HTTP - bezstanowość, metody, najważniejsze nagłówki, metody uwierzytelniania, HTTPS

## **Wstęp do technologii web**

1. Responsywność strony - jakie mamy możliwości?
2. Optymalizacja czasu ładowania strony
3. Na czym polega kaskadowość stylów i ich właściwości?
4. Pozycjonowanie - omów różnice pomiędzy wartościami static, relative, absolute i fixed
5. Różnica między wartościami grid i flex właściwości display, przykłady użycia. Czy są sytuacje, gdzie możemy użyć jednej z nich, a drugiej nie (i odwrotnie)?
6. CSS vs SCSS
7. Możliwości języka SCSS na uzyskanie clean code

## **Bazy danych I**

1. Model relacyjnych baz danych. Własności relacji.
2. Negatywne strony istnienia redundancji w bazie danych. Jakie są sposoby jej zwalczania? Przykłady.

3. Rodzaje związków w teorii relacyjnych baz danych. w jaki sposób są one realizowane w języku T-SQL?
4. Grupowanie danych w T-SQL. Zasada działania GROUP BY i HAVING.
5. Pojęcie widoku (perspektywa, ang. view); zastosowania widoków.
6. Sposoby łączenie tabel.

### **Matematyczne podstawy informatyki**

1. Omówić wyrażenia regularne i związek z deterministycznymi automatami skończonymi.
2. Zastosowania wyrażień regularnych.
3. Własności języków regularnych.
4. Omówić klasę języków bezkontekstowych oraz związek gramatyk bezkontekstowych i skończonych automatów ze stosem.
5. Omówić maszyny Turinga i klasę języków akceptowanych przez maszyny Turinga.

### **Algorytmy i struktury danych**

1. Kopce binarne: definicja, ilustracja przykładem, operacja wstawiania i operacja usuwania elementu maksymalnego – idea i złożoność czasowa.
2. Drzewa poszukiwań binarnych: definicja, ilustracja przykładem, operacje wstaw, szukaj, usuń – idea i złożoność czasowa.
3. Algorytmy sortowania Quick-sort i sortowanie kopcowe (Heap-sort): idea algorytmów i omówić ich złożoność czasową
4. Stosy, kolejki i kolejki priorytetowe: wyjaśnić typowe operacje wykonywane na tych strukturach danych (czyli Push i Pop dla stosów, wstawienie i pobranie elementu z kolejki dla kolejek, wstawienie i usunięcie elementu maksymalnego dla kolejek priorytetowych), podać przykładową implementację.
5. Tablice z haszowaniem: idea haszowania i sposoby rozwiązywania kolizji (czyli metoda łańcuchowa i adresowanie otwarte).

### **Języki programowania I**

1. Czy da się programować bez wykorzystania mechanizmu „zmiennych”? Odpowiedź uzasadnij.
2. Na czym polega rekurencja „ogonowa”? Podaj przykłady jej użycia.
3. Omów podobieństwa i różnice pomiędzy strukturami listowymi (List[T]) i tablicowymi (Array[T]).
4. Czym są funkcje „wyższego rzędu” i do czego mogą być przydatne? Odpowiedź zilustruj przykładami z użyciem języka Scala.
5. Wymień i scharakteryzuj znane Ci rodzaje kolekcji w języku Scala. Odpowiedź zilustruj przykładami.
6. Omów podstawowe pojęcia związane z programowaniem obiektowym w języku Scala: obiekt, klasa, cecha.

Zilustruj je przykładami.

7. Na czy polega model „aktorski” programowania równoległego? Czy aktor może wykonywać kilka akcji „jednocześnie”?

## **Bazy danych II**

1. Omów model grafowy oferowany przez bazę Neo4j oraz najważniejsze konstrukcje języka Cypher

2. Porównaj modele relacyjny oraz grafowy oraz podaj przykłady zastosowań, w których jeden z nich wykazuje przewagę nad drugim.

3. Omów najważniejsze cechy modelu dokumentowego bazy MongoDB.

4. Przedstaw koncepcję potoków i ich wykorzystania w procesie agregacji danych w języku zapytań bazy MongoDB.

## **Systemy operacyjne**

1. Cechy systemów rozproszonych

2. Różnica między kompilatorem i interpreterem

3. Systemy operacyjne czasu rzeczywistego

4. Algorytmy planowania przydziału procesora

5. Zagadnienia ochrony w systemie operacyjnym

6. Co to jest jądro systemu operacyjnego?

7. Co to jest maszyna wirtualna?

8. Warstwowy model systemu operacyjnego

## **Testowanie automatyczne**

1. Piramida testów i rola poszczególnych jej komponentów w zapewnianiu jakości oprogramowania

2. Pojęcie długu technologicznego: konsekwencje i sposoby zapobiegania

3. Test Driven Development w procesie rozwoju oprogramowania

4. Wyjaśnij pojęcia: fake, stub, mock i spy, w kontekście testowania oprogramowania

5. Koncepcje Behavioral Driven Development na przykładzie biblioteki behave języka Python

6. Tworzenie testów behawiouralnych w języku Gherkin

## **Zarządzanie projektem informatycznym**

1. Zdefiniuj zarządzanie projektem, programem, portfelem

2. Omów klasyfikacje projektów informatycznych

3. Wymień charakterystyczne cechy projektów informatycznych

4. Czynniki sukcesu i przyczyny porażek projektów informatycznych

5. Wymień kluczowe obszary zarządzania projektem
6. Wymień procesy zarządzania zakresem projektu i różnice pomiędzy zakresem projektu a zakresem produktu
7. Dokonaj klasyfikacji otoczenia projektu IT i podaj przykłady interesariuszy
8. Zdefiniuj ryzyko w projekcie i wymień techniki zarządzania ryzykiem

### **Inteligencja obliczeniowa**

1. Uczenie nadzorowane i nienadzorowane - definicje, przykłady
2. Dobór cech a odkrywanie cech w data science
3. Czego dotyczy i na czym polega algorytm propagacji wstecznej?
4. Regresja a regresja logistyczna.
5. Klasteryzacja hierarchiczna - do czego służy, rodzaje, korzyści ze stosowania
6. Krzywe ROC - co to są i do czego służą?

### **Programowanie obiektowo-funkcyjne**

1. Pojęcie klasy i obiektu w językach programowania na wybranych przykładach
2. Wyjaśnij pojęcie polimorfizmu na przykładach kodu w języku Java
3. Pojęcie interfejsu w języku Java, jego implementacja i znaczenie w zasadzie odwracania zależności (dependency inversion principle)
4. Lambda-wyrażenia w języku Java i ich zastosowanie w przetwarzaniu strumieni
5. Znaczenie zasady pojedynczej odpowiedzialności (single responsibility principle) i otwarte-zamknięte (open closed principle) w tworzeniu testowalnego kodu obiektowego
6. Na wybranych przykładach wyjaśnij zasadę segregacji interfejsów (interface segregation principle) w programowaniu obiektowym

### **Analiza i projektowanie systemów informatycznych**

1. Istota i sposoby dokumentowania wymagań systemowch
2. Sposoby odwzorowywania podstawowych pojęć obiektowości na diagramach klas UML
3. Diagramy czynności a diagramy sekwencji UML - zastosowanie, podobieństwa i różnice
4. Narzędzia CASE - zastosowanie i typowa funkcjonalność
5. Główne fazy procesu wytwórczego oprogramowania