

LISTA PYTAŃ NA EGZAMIN DYPLOMOWY - KIERUNEK - INFORMATYKA

Studia I-go stopnia

FIZYKA

1. Wyjaśnij pojęcie ładunku elektrycznego, opisz jakimi charakteryzuje się właściwościami.
2. Podstawowe pojęcia obwodów elektrycznych: prąd, potencjał, napięcie, energia, moc, rezystancja, impedancja.
3. Pole magnetyczne, pole elektryczne, pole elektromagnetyczne – opisz właściwości i zastosowanie w architekturze komputera.
4. Cechy i zakresy fal elektromagnetycznych – zastosowanie w przesyłaniu sygnału.

PODSTAWY ELEKTROTECHNIKI I ELEKTRONIKI

1. Porównaj budowę i zasadę działania diody półprzewodnikowej i diody Zenera.
2. Porównaj budowę i zasadę działania tranzystora bipolarnego i tranzystora polowego.
3. Przedstaw zasady działania podstawowych układów elektronicznych: wzmacniaczy, generatorów, filtrów.
4. Przedstaw typy i zasady: modulacji i demodulacji.
5. Podaj przykłady zastosowania przetworników A/C i C/A.
6. Scharakteryzuj typy i przedstaw zasady działania bramek logicznych.
7. Porównaj: układ kombinacyjny i układ sekwencyjny.
8. Scharakteryzuj zasadę superpozycji w odniesieniu do obliczeń rozptyłu prądów w obwodzie.
9. Przedstaw zastosowanie prawa Ohma i praw Kirchhoffa – w kontekście obliczania prądów i spadków napięć w obwodzie.
10. Podaj przykład wykorzystania twierdzenia o mocy maksymalnej w obwodzie.

ALGORYTMY I STRUKTURY DANYCH

1. Czym powinien cechować się dobry algorytm.
2. Scharakteryzuj podstawowe struktury algorytmów (algorytmy liniowe, algorytmy z rozgałęzieniami, algorytmy z powtórzeniami).
3. Podaj przykład oszacowania złożoności obliczeniowej algorytmu.
4. Metody projektowania algorytmów (rekurencja, „dziel i zwyciężaj”, metoda zachłanna, programowanie dynamiczne, algorytmy z powrotami) – działanie, zastosowania.
5. Porównaj podstawowe struktury danych (tablice, listy, kolejki, stosy, mapy, grafy, drzewa) – cechy, zastosowania.

ARCHITEKTURA KOMPUTERÓW

1. Porównaj konstrukcje modelu programowego procesora w podejściu CISC i RISC.
2. Omów podstawy realizacji systemu pamięci podręcznej uwzględniając jej poziomowość.
3. Zdefiniuj budowę modelu programowego jednostki centralnej – omów niezbędne rejestry, tryby adresowania, listę instrukcji oraz model operacji warunkowych.
4. Omów architekturę przykładowego mikrokontrolera. Czym różni się mikrokontroler od mikroprocesora?
5. Przedstaw klasyfikację układów programowalnych. Krótko scharakteryzuj każdą z klas.

6. Na czym polega minimalizacja funkcji logicznych.
7. Omów podstawowe rodzaje przerzutników. Wyjaśnij zasadę działania przerzutnika typu: D, T, JK.

BAZY DANYCH

1. Jakie są różnice pomiędzy SQL, MySQL oraz SQL Server?
2. Charakterystyka procesu projektowania bazy danych.
3. Przetwarzanie informacji w bazach danych. Porównaj metody oraz zastosowanie.
4. Porównaj typy bazy danych i system zarządzania bazą danych.
5. Jakie są typy bazy danych, jaki wybrałbyś do np. małego sklepu internetowego, dużego portalu społecznościowego.
6. Do czego wykorzystujemy relacyjne bazy danych, a do czego noSQL?
7. Optymalizacja zapytań do baz danych – podaj podstawowe metody na przykładzie.
8. W jakim celu stosowane są wyzwalacze w bazach danych – podaj przykłady.
9. Współczesne bazy danych w odniesieniu do modelu relacyjnego.
10. Agregacja danych – gdzie ją lepiej zrobić po stronie serwera, czy użytkownika końcowego, podaj przykład

INŻYNIERIA OPROGRAMOWANIA

1. Wskaż w jakich fazach procesu twórczego oprogramowania przeprowadza się walidację i weryfikację statyczną, a w jakich walidację i weryfikację dynamiczną. Z jakimi dokładnie produktami jest to związane?
2. Przedstaw uwarunkowania trójkąta inżynierii oprogramowania oraz wskaż w jaki sposób możemy w projekcie minimalizować ryzyka (w postaci zagrożeń) związane z jego atrybutami.
3. Posługując się praktycznym przykładem wyjaśnij na czym polega planowanie adaptacyjne. Jaki model cyklu życia oprogramowania wspiera planowanie tego typu?
4. Czy w przypadku zastosowania planowania predykcyjnego możemy zastosować kaskadowy model zarządzania cyklem wytwarzania oprogramowania? Uzasadnij swoją odpowiedź posługując się praktycznym przykładem.
5. Jakie uwarunkowania projektu powinna obejmować analiza na poziomie wymagań biznesowych, a jakie na poziomie wymagań użytkowników i poziomie wymagań funkcjonalnych?
6. Na co dokładnie mają wpływ w projekcie programistycznym wymagania pozafunkcyjne opisujące usługi lub charakterystyki wydajnościowe produktu programistycznego?
7. Czym się różnią wymagania systemowe od wymagań funkcjonalnych i wymagań determinowanych przez interfejsy zewnętrzne dla projektowanego oprogramowania?
8. Przedstaw istotne różnice pomiędzy strategią pozyskiwania wymagań zorientowaną na użycie a strategią pozyskiwania wymagań zorientowaną na produkt.
9. Wymień co najmniej trzy główne aktywności zarządzania wymaganiami dla produktu programistycznego, które nie przyczynią się do powstania konieczności wprowadzania zmian w projekcie.
10. Jaka istotna cecha modelu spiralnego wyróżnia go w odniesieniu do każdego innego modelu zarządzania cyklem wytwarzania oprogramowania?
11. Jaka jest największa wada zastosowania modelu ewolucyjnego w procesie wytwarzania oprogramowania?

12. Które perspektywy modelu „4+1” opisują wewnętrzną strukturę tworzonego oprogramowania na różnych poziomach abstrakcji i szczegółowości?
13. Omów jakie atrybuty zewnętrzne oraz jakie atrybuty wewnętrzne powinny zostać zawsze zdefiniowane metrycznie (określone ilościowo) w przypadku specyfikacji, jakie w przypadku kodu źródłowego, a jakie dla danych testowych.
14. W jaki sposób możemy poprawnie oszacować opłacalność projektu informatycznego?
15. Jaka jest relacja pomiędzy kosztem a opłacalnością projektu programistycznego?
16. Co należy uwzględnić w procesie estymacji nakładu pracy dla danego przedsięwzięcia programistycznego? Na co rzutuje ten atrybut procesu w projekcie?
17. W związku z tym, że w każdym projekcie programistycznym estymacji podlega wiele atrybutów procesu i produktów rozwoju oprogramowania, wyjaśnij w jaki sposób możemy ocenić sprawdzalność wybranych metod estymacji i procesu i produktu.
18. Wyjaśnij w jaki sposób możemy dokonać oceny ryzyka dla realizowanego przedsięwzięcia programistycznego.
19. Jakie możliwe działania można podjąć odnośnie ryzyka typu szansa, a jakie odnośnie ryzyka typu zagrożenie w projekcie programistycznym?
20. Jaka jeszcze inna perspektywa oprócz perspektywy specyfikacyjnej i implementacyjnej dotyczy modelowania rozwiązania z wykorzystaniem języka UML? Na czym ona polega?

SZTUCZNA INTELIGENCJA

1. Podaj rzeczywiste przykłady zastosowania metod sztucznej inteligencji
2. Omów budowę sztucznych sieci neuronowych. Czym się różni sieć neuronowa w architekturze płytkiej od głębokiej.
3. Czy jest predykcja. W jakim celu się ja stosuje?
4. Gdzie stosujemy uczenie nadzorowane, a gdzie nienadzorowane – opisz oraz wymień typy znanego oprogramowania
5. Jaka jest różnica między klasyfikacją a regresją – oraz gdzie są wykorzystywane, podaj znane ci rozwiązania.
6. Podaj wady i zalety rozwiązań opartych o sztuczną inteligencję w znanych ci rozwiązaniach.
7. Co to jest walidacja? Jakie znasz metody walidacji?
8. Podaj rzeczywiste przykłady zastosowania metod sztucznej inteligencji
9. Przygotowujesz oprogramowanie, którego jednym celów będzie rozpoznawanie twarzy użytkownika. Zaproponuj poszczególne kroki budowy takiego modułu.

BEZPIECZEŃSTWO INFORMACJI

1. Wymień i porównaj szyfry podstawieniowe i przestawieniowe.
2. Wymień i porównaj szyfry symetryczne i asymetryczne.
3. Wykonujesz testy penetracyjne – przedstaw charakterystykę poszczególnych etapów.
4. Ataki Dos/DDoS – scharakteryzuj rodzaje i metody przeprowadzania ataku.
5. Omów zasadę przeprowadzania ataków Man in the middle (MITM), jak się przed tym chronić.
6. Wymień i scharakteryzuj wybrane narzędzia pentestera.
7. Omów na przykładach wybrane ataki socjotechniczne.
8. Omów zasadę działania oraz porównaj systemy IDS oraz IPS.
9. Czy jest możliwość bycia anonimowym w Internecie?

PROGRAMOWANIE

1. W jaki sposób identyfikujemy miejsce przekazania sterowania w inne miejsce w programie w językach interpretowanych, a w jaki w językach kompilowanych?
2. Wskaż różnice pomiędzy programowaniem imperatywnym a programowaniem funkcyjnym. Które podejście związane jest ze zmianą stanu?
3. Czym jest postać strukturalna kodu źródłowego i co należy zrobić aby można było ją otrzymać w przypadku kiedy mamy do czynienia nawet z programowaniem funkcyjnym?
4. Wskaż sprawdzalną metodę zapewniania skalowalności kodu źródłowego tworzonego z zastosowaniem paradygmatu programowania obiektowego.
5. Które składowe klasy są najważniejsze i dlaczego?
6. Który mechanizm paradygmatu programowania obiektowego jest najważniejszy i dlaczego?
7. Jaka jest istotna różnica pomiędzy programowaniem klasowym a programowaniem prototypowym?
8. Co jeszcze oprócz funkcji i metod możemy określić jako abstrakcję przy tworzeniu oprogramowania z wykorzystaniem programowania zorientowanego obiektowo?
9. Omów mechanizm polimorfizmu w zakresie możliwości wykorzystania szablonów.
10. Wyjaśnij na czym polega polimorfizm w zakresie możliwości wykorzystania mechanizmu przeciążania operatorów.
11. Jaka jest znacząca różnica pomiędzy metaprogramowaniem a tworzeniem kodu źródłowego z wykorzystaniem podejścia generycznego?
12. Omów struktury kontrolne przy pomocy których możliwe jest zbudowanie dowolnego programu. Podaj praktyczne przykłady każdego wskazanego typu struktury.
13. Co w praktyce oznacza, że kod źródłowy programu spełnia warunek deterministyczności?
14. Omów typy i przypadki zastosowania znanych Ci modyfikatorów parametrów metod języków obiektowych takich jak C++, Java i C#.
15. W jaki sposób możemy zdefiniować metodę synchroniczną w kodzie źródłowym i na czym ona polega?
16. Podaj praktyczny przykład obsługi wyjątku kontrolowanego i wyjątku niekontrolowanego. Kiedy możemy stosować wyjątki kontrolowane?
17. W jaki sposób możemy sterować procesem serializacji obiektów w tworzonym oprogramowaniu?
18. Omów uwarunkowania dostępu do składowych prywatnych obiektu przy zastosowaniu mechanizmu odzwierciedleń (refleksji).
19. Odwołując się do znanego Ci języka programowania obiektowego wskaż problemy związane z wykorzystywaniem kolekcji przechowujących obiekty.
20. W jaki sposób w kodzie źródłowym należy odzwierciedlić brak modyfikatora metod, który jest widoczny w diagramie klas UML?

TECHNOLOGIE INTERNETOWE I MOBILNE

1. Opisz jak przebiega komunikacja komputera z serwerem HTTP podczas próby odczytu pliku *.php 2. Omów budowę i zasadę działania formularza HTML5 oraz przedstaw przynajmniej trzy typy pól.

2. Omów różnice między metodami przesyłania danych GET i POST. Podaj przykłady zastosowań obu metod.
3. Omów na przykładach budowę i sposoby wykorzystania kaskadowych arkuszy stylów CSS.
4. Technologia Ajax – czym jest i w jakich rozwiązaniach jest wykorzystywana
5. Omów zasadę komunikacji aplikacji klienckiej z serwerem bazodanowym na przykładzie PHP i MySQL
6. Przedstaw sposób wykorzystania mechanizmu plików cookies.
7. Przedstaw sposób wykorzystania mechanizmu zmiennych sesyjnych
8. Omów podstawowe konstrukcje i znaczenie języka XML
9. Scharakteryzuj możliwości języka HTML5 umożliwiające wygodne tworzenie aplikacji graficznych.
10. Wymień po jednym przykładzie stosowanego na stronach WWW skryptowego języka programowania, wykonywanego a) po stronie klienta b) po stronie serwera.

INTERNET RZECZY, SIECI KOMPUTEROWE

1. W jaki sposób zbudować prostą sieć do małego biura.
2. W jakich sytuacjach wskazane byłoby zastosowanie medium: kable miedziane/światłowód/transmisja bezprzewodowa.
3. Jakie problemy pojawiają się z adresacją IPv4.
4. Co jest potrzebne do wdrożenia IPv6 w sieci LAN.
5. Jakie problemy występują w użytkowaniu i zarządzaniu sieciami Wi-Fi.
6. Jak zabezpieczyć urządzenia sieciowe przed zagrożeniami.
7. Jak wykorzystać sieci VLAN w sieciach korporacyjnych.
8. W jaki sposób zapewnić wysoką niezawodność sieci LAN.
9. Jak ograniczyć ataki na sieci LAN.
10. Kiedy routing statyczny ma przewagę nad routingiem dynamicznym.
11. Jakie narzędzia stosuje się do rozwiązywania problemów w sieci.
12. Jakie znaczenie ma zgodność rozwiązania ze standardem.
13. W jaki sposób dobrać technologię dostępu do Internetu.
14. Jak zapewnić właściwy przeszły danych głosowych w sieci.
15. W jaki sposób ograniczyć dostęp do wybranych hostów lub usług.
16. Czym różni się Arduino od Raspberry Pi.
17. Jakie języki programowania używane są w technologiach IoT.
18. Jakie komponenty będą potrzebne do zbudowania systemu inteligentnego domu.
19. W jakim celu stosuje się przetwarzanie w Chmurze.
20. Jakie rozwiązania stosowane są do wymiany informacji w systemach IoT.
21. W jaki sposób systemy IoT minimalizują zapotrzebowanie na zasilanie.

GRAFIKA KOMPUTEROWA W GRACH

1. Scharakteryzuj dwie wybrane metody prototypowania graficznego. Podaj ich mocne i słabe strony.
2. Omów narzędzia modelowania i prototypowania graficznego, podaj ich mocne i słabe strony.
3. Wyjaśnij na czym polega przygotowanie obiektów do animacji.
4. Na czym polega przygotowanie wirtualnych przestrzeni gier.
5. Porównaj dwa wybrane narzędzia do przygotowania graficznej prezentacji gry.
6. Jakie czynniki mają wpływ na opracowanie koncepcji, analizy założeń i wymagań projektowych gry.

7. Przedstaw kolejne kroki procesu projektowania gry komputerowej.
8. Porównaj projektowanie gry jednoplatformowej z wieloplatformową.
9. Podaj różnice pomiędzy silnikami gier, a platformami gier.
10. Na czym polega dostosowanie projektu gry do różnych platform
11. Dokonaj porównania sprzętu (konsol, urządzeń mobilnych itp.) na których w rzeczywistych warunkach tworzy się gry.