Informatyka

12.Pytania na egzamin ustny

Opracował: Maciej Penar

Spis treści

[1. Ogólna zasada egzaminu pogadanki 3](#_Toc496977974)

[2. O co mogę pytać 3](#_Toc496977975)

# Ogólna zasada ~~egzaminu~~ pogadanki

Ogólna zasada:

* Na podstawie magicznych zapisków które sporządzam po każdych zajęciach wytypowane zostaną „point of weakness” – tam gdzie została zwęszona niewiedza na zajęciach, zostanie zadane pytanie
* Im bardziej złośliwe i nieprecyzyjne pytanie, tym większe prawdopodobieństwo jego wystąpienia

# O co mogę pytać

Kodowanie liczb:

* Jakie są rodzaje kodowań? Podać ogólną zasadę.
* Jak wykonać konwersję liczby dziesiętnej na liczbę o znanej podstawie – i odwrotnie
* Czy liczby pozycyjne mogą mieć podstawę mniejszą od 0?
* Czy liczby pozycyjne mogą mieć podstawę będącą ułamkiem?
* Co to jest kodowanie Znak Moduł. Czy działa tylko dla liczb binarnych?
* Co to jest kodowanie U1. Czy działa tylko dla liczb binarnych?
* Co to jest kodowanie U2 Czy działa tylko dla liczb binarnych?
* Co to jest kodowanie BIAS. Czy działa tylko dla liczb binarnych?
* Czego dot. format IEEE 754? Podać ogólną zasadę działania.
* Podać zakresy wartości dla kodowań przy zadanej liczbie pozycji.
* Ogólna zasada dodawania liczb pozycyjnych
* Jak wykonał(a)byś odejmowanie liczb pozycyjnych
* Ogólna zasada mnożenia liczba pozycyjnych
* Jak wykonał(a)byś dzielenie liczb pozycyjnych
* Podaj liczbę X w kodowaniu Y.

Schematy blokowe:

* Z czego składa się schemat blokowy
* Jakie jest zastosowanie schematów blokowych
* Jaki jest zestaw operacji które możemy użyć wewnątrz bloku operacji
* Czy blok warunkowy może działać jak **switch**
* Ile punktów wejścia może mieć schemat blokowy
* Ile punktów wyjścia może mieć schemat blokowy – ustosunkuj się do tzw. programowania strukturalnego
* Chcę zwrócić dzielenie liczby a przez b. W jaki sposób obsłużyć błąd np. dzielenie przez 0 na schemacie blokowym. Czy zaproponowany sposób obsługi błędu nie narusza spójności zwracanych typów. (Czy blok wyjścia mogę traktować jako **return**)
* Podać schemat blokowy dla algorytmu X
* Podać „złożoność obliczeniową” wynikającą ze podanego schematu blokowego, dla podanych parametrów wejściowych

C/C++

* Jakie paradygmaty możemy wyróżnić w C/C++
* Co to jest wyrażenie, zmienna, instrukcja, literał.
* Co to jest procedura, funkcja, metoda.
* Do czego służy funkcja main()
* Do czego służy printf() / scanf(). Czy istnieją skutki uboczne takiego kodu: printf(„%s %s %s”);
* Co oznacza słowo kluczowe void.
* Jakie typy proste są dostępne w C/C++. Jaka jest szerokość typów (w bajtach). Do czego służy sizeof().
* Jaka jest zaleta obiektowości
* Jakie są wady obiektowości
* Sterta / stos – co to jest? Z czym związane?
* Co oznacza #include/#define
* Wymienić etapy kompilacji programów w C
* Operacje binarne / logiczne (AND, OR, NOT, XOR) – tabele prawdy. Przesunięcie bitowe.
* Jaka jest różnica pomiędzy klasą a strukturą
* Wymienić i omówić modyfikatory dostępu.
* Omówić zaprzyjaźnianie.
* Omówić dziedziczenie / polimorfizm.
* Omówić przeciążanie operatorów.
* Omówić konstruktory. Do czego, ile, jakie, kiedy są wywoływane?
* Omówić destruktory. Do czego, ile, jakie, kiedy są wywoływane?
* Omówić zasięg zmiennych. Jakie mamy do dyspozycji – czy możemy modyfikować.
* Co oznacza / do czego służy / w jakim kontekście może być użyte dane słowo kluczowe np. **static, const, typedef, class, struct**
* Stała, wskaźnik na stałą, stały wskaźnik na stałą – przypadki użycia
* Dynamiczna alokacja pamięci – jak realizowana w C/C++
* Do czego służy new/delete/new[]/delete[]. Czy można używać wymiennie.
* Co oznacza w specyfikacji C/C++ że wynik operacji jest niezdefiniowany. Jakie ma to reperkusje dla programisty.
* Co to jest „lukier składniowy” (syntactic sugar)? Podać przykład.
* Co to jest algorytm. Jakie ma cechy. Podać siostrzane pojęcie.
* Złożoność obliczeniowa prostych problemów np. sortowanie bąbelkowe (wyszukane)
* Intuicyjne rozumienie złożoności
* Czy C++ wspiera adnotacje

Ultra trudne pytania z C/C++:

* Wzorzec RAII
* Czy C jest typizowane statyczne czy dynamicznie. Jak sytuacja wygląda w C++? (Rozważ słowo kluczowe auto).

Sieci komputerowe:

* Co to jest IP? Jaki jest format adresu w wersji IPv4?
* Adresacja w sieciach: IP oraz maski
* Adres rozgłoszeniowy, adres bramy domyślnej – jak liczyć?
* Szerokość maski, a liczba adresowalnych urządzeń
* Ipconfig / tracert / nslookup / ping /nc
* Plik hosts – zastosowanie
* DNS – ogólna idea. Podać adresy DNS Google’a. (8.8.4.4, 8.8.8.8)
* URI/URL
* Do czego służy przeglądarka internetowa. Co się dzieje po wpisaniu adresu?
* Operacje http. Różnica pomiędzy GET/POST.
* Co to jest gniazdo internetowe?
* Co jednoznacznie identyfikuje punkt końcowy w internecie.
* Czy socket i websocket to to samo?
* Co to jest nagłówek http – wymienić parę. Jak korzystać?
* Co to jest ciało (body) http – omówić zastosowanie.
* Kodowanie BASE64 – do czego służy
* URL Encoding – do czego służy

Bazy danych:

* Wyjaśnij co to jest Baza Danych, z czego się składa?
* Co oznacza **pasywność** Bazy Danych?
* Wyjaśnij co to jest schemat bazy danych.
* Co to jest relacyjna baza danych?
* Co to jest relacja/tabela?
* Co to jest związek? W tym związek jeden-do-jednego, jeden-do-wielu oraz wiele-do-wielu. Podaj przykłady.
* Co to jest klucz, klucz obcy?
* ~~Co to jest indeks?~~
* Co to jest transakcja? Rozwinąć oznacza ACID w kontekście transakcji.
* Co to oznacza CRUD?
* ~~Co oznacza stwierdzenie „Schemat Bazy Danych w 1 postaci normalnej”. Ile postaci znasz. Co oznaczają.~~
* **Jak wyrazić związek wiele-do-wielu w Relacyjnych Bazach Danych?**
* OLTP / OLAP
* ORM – co to? Po co to komu?
* **Omówić SQL (np. projekcja/selekcja/złączenie)**
* ~~Jak zamodelować drzewo w SQL’u?~~
* Na przykładzie wybranej bazy danych omówić sposób obsługi brakujących (niepełnych) danych
* ~~Poziomy izolacji transakcji – do czego służą? Czy można wybrać poziom izolacji? Jaki jest domyślny?~~

Systemy operacyjne:

* Omówić szybkość: procesor / dysk / pamięć RAM / IO sieci (rzędy wielkości, jaka jest relacja)
* Wątek/ Proces – omówić

Co ślina na język przyniesie

Lista nie jest wyczerpująca: