

Maturitní témata

IKT, školní rok 2017/18

1 Struktura osobního počítače

- Von Neumannova architektura: zakreslete, vysvětlete její smysl a popište, jakým způsobem se od ní běžné počítače odchyľují.
- Osobní počítač: popište strukturu hardware, vysvětlete funkci a popište parametry jednotlivých komponent a periferních zařízení.
- Procesor: vysvětlete funkci procesoru a jakou úlohu hraje instrukční sada. Popište přehledově jazyk Assembler.

2 Operační systém

- Princip: vysvětlete obecně účel a princip fungování operačního systému. Srovnajte, na kterých zařízeních operační systém přináší užitek a kde by byl zbytečný nebo nevhodný.
- Pojmy: vysvětlete pojmy proces, multitasking, uživatel, fyzická a virtuální paměť.
- Příklady: popište důležité a běžné používané operační systémy pro osobní počítače a chytré telefony.

3 Uživatelský software

- Aplikace: popište některé kancelářské a odborné úlohy, a zmiňte se o software, který je řeší.
- Licence: vyjmenujte a charakterizujte některé běžně se vyskytující softwarové licence. Popište některé metody ochrany software a poskytování software jako služby. Vysvětlete, v čem spočívá cloud computing a jak se projevuje z hlediska uživatele.
- Formáty souborů: srovnajte významné formáty souborů pro textové dokumenty. Srovnajte významné formáty souborů pro bitmapovou a vektorovou grafiku a pro video. Přehledově zmiňte, které z těchto formátů jsou (mohou být) komprimované a jakým způsobem.
- Aplikace pro telefon: popište jejich smysl a způsob, jakým jsou distribuované. Zmiňte se o několika aplikacích nebo o několika úlohách, kterým můžou aplikace sloužit.

4 Počítačové síť

- Pojmy: popište čtyři vrstvy architektury TCP/IP. Vysvětlete pojmy MAC adresa, IP adresa.
- Síť: popište, k čemu síť slouží, jak funguje celkově a jak na úrovni operačního systému.
- Směrování: vysvětlete podstatu směrování paketu, a jakou úlohu v něm hraje podsíť.

- Sít'ové prvky: popište některé běžně používané sít'ové prvky a zmiňte se, ve které sít'ové vrstvě pracují. Popište a zdůvodněte, jak a z jakých prvků byste vytvořili lokální sít' pro malý podnik.

5 Internet

- Internet: popište tuto sít', zmiňte se o jejím společenském kontextu a historii.
- Připojení: popište způsoby připojení k Internetu, význam technologie NAT (network address translation) a protokolu IP verze 6 pro běžného uživatele.
- Služby: popište službu zajišťující získání IP adresy pro nově připojený počítač a službu pro překlad doménového jména na IP adresu. Popište, jak funguje e-mail.

6 Web

- HTML: popište možné uplatnění jazyka HTML a jeho základní syntaktické prvky. Srovnajte jej s jinými způsoby formátování textu. Zmiňte rozdíly mezi HTML 5 a staršími verzemi.
- CSS: vysvětlete obecně účel kaskádových stylů a popište způsob jejich použití.
- Worldwide Web: vysvětlete fungování této služby. Zmiňte se o dalších souvisejících službách v Internetu (tj. přímo na webu i mimo něj).

7 Číselné soustavy

- Zápis čísel: popište obecný zápis celého čísla, porovnejte zápis čísel v různých číselných soustavách (desítková, dvojková, osmičková, šestnáctková). Na libovolném příkladu vysvětlete převod mezi číselnými soustavami.
- Číselné datové typy: popište číselné typy dostupné v některých programovacích jazycích.
- Celá čísla: popište, jak jsou v procesoru reprezentována celá čísla, včetně záporných.
- Floating-point: popište stručně zápis čísla s plovoucí řádovou čárkou. Vysvětlete jeho výhody, nevýhody a alternativy pro různé účely.

8 Algoritmizace

- Algoritmus: vysvětlete pojmy algoritmus, algoritmizace na příkladu Euklidova algoritmu pro nalezení největšího společného dělitele dvou čísel.
- Asymptotická složitost: na příkladu Euklidova algoritmu vysvětlete, jakým způsobem lze určit časovou a paměťovou složitost, a co nám v praxi říká o rychlosti výpočtu.
- Vývojový diagram: vysvětlete základní symboly vývojových diagramů na příkladu Euklidova algoritmu.

9 Programovací jazyky

- **Kompilace:** vysvětlete úlohu kompilátoru a rozdíl mezi kompilací a interpretací kódu. Vysvětlete podstatu a účel kompilace just-in-time.
- **Správa paměti:** popište rozdíl mezi proměnnými alokovanými na zásobníku (stack) a dynamicky (heap). Vysvětlete, co znamená alokování paměti z pohledu operačního systému. Srovnajte, jak s proměnnými pracují jazyky, které používají garbage collector.
- **Procedurální a neprocedurální programování:** popište členění zdrojového kódu v libovolném procedurálním jazyce. Popište, v čem spočívá neprocedurální programování a na jaké úlohy je vhodné.
- **Vývojové prostředí:** uveďte moderní nástroje pro vývoj software a spolupráci vývojářského týmu.

10 Řídicí struktury

- **Podmíněný příkaz:** uveďte příkazy pro větvení programu v některých programovacích jazycích, a další řídicí struktury, které lze využít obdobně.
- **Cykly:** uveďte různé formy cyklu v některých programovacích jazycích. Podrobněji vysvětlete, v čem se liší a za jakých okolností jsou navzájem záměnné.
- **Rekurze:** vysvětlete tento pojem na příkladu funkce faktoriál. Popište, jaké výhody a nevýhody přináší rekurze při psaní kódu, a čím ji lze nahradit.

11 Datové typy

- **Pojmy:** vysvětlete pojmy proměnná, rozsah proměnné (scope), reference.
- **Jednoduché datové typy:** uveďte příklady v některých programovacích jazycích.
- **Operátory:** v libovolném programovacím jazyce uveďte některé operátory nad jednoduchými datovými typy a zmiňte se o pořadí jejich vyhodnocování.
- **Složené datové typy:** vysvětlete význam polí a složených datových typů v programovacích jazycích, které tyto konstrukty používají.

12 Objektově orientované programování

- **Pojmy:** vysvětlete pojmy třída, objekt, zapouzdření, konstruktor, dědičnost, polymorfismus. Naznačte implementaci těchto pojmů v libovolném programovacím jazyce.
- **Motivace:** vysvětlete obecně, k čemu může sloužit objektově orientované programování, a jaké jsou jeho nejdůležitější zásady.
- **Operátory:** vysvětlete, jakou úlohu má přetěžování operátorů v jazycích, které ho umožňují.

13 Lineární datové struktury

- Fronta a zásobník: charakterizujte tyto datové struktury a zdůrazněte rozdíly mezi nimi. Naznačte jejich implementaci s polem pevně dané velikosti a omezeným počtem prvků.
- Amortizovaná složitost: vysvětlete, jak lze efektivně alokovat paměť pro pole proměnlivé velikosti při postupném přidávání prvků.
- Spojový seznam: popište jednosměrný a obousměrný spojový seznam, a operace vkládání a odebírání prvku na libovolné pozici.

14 Halda

- Halda: popište obecně, jaké operace tato datová struktura umožňuje a k čemu je užitečná.
- Binární halda: podrobně vysvětlete, jak lze implementovat binární haldu pomocí pole proměnlivé velikosti, včetně jejích dvou základních operací.
- Heapsort: popište algoritmus třídění haldou a vyjádřete se obecně o jeho časové složitosti.

15 Šifrování a ověřování

- Symetrické šifry: popište, k čemu v praxi může sloužit symetrická šifra. Rozeberte její výhody a nevýhody oproti jiným způsobům zabezpečení.
- Asymetrické šifry: vysvětlete obecně, jakým způsobem se používá asymetrická šifra a digitální podpis. Uveďte příklady, jak lze tyto technologie použít v praxi.
- Hashovací funkce: popište některé uplatnění kryptografických hashovacích funkcí, a rozdíl mezi kryptografickou hashovací funkcí a kontrolním součtem.

16 Backtracking

- Princip: vysvětlete podstatu backtrackingu na úloze obchodního cestujícího (úkolem je najít nejkratší cestu, která prochází všemi vrcholy grafu). Obecně naznačte programovou realizaci algoritmu využívajícího backtracking.
- Efektivita: vysvětlete, jaká je obvykle časová a paměťová složitost algoritmů využívajících backtracking, a jakým způsobem lze takové programy v praxi zrychlovat.

17 Třídění

- Třídění: zdefinujte úlohu třídění čísel a třídění obecných hodnot. Podrobněji vysvětlete, co znamená třídění in-place (bez kopírování).
- Přímé třídění: popište a srovnajte algoritmy třídění přímým vkládáním a přímým výběrem. Určete asymptotickou časovou složitost těchto algoritmů.
- Quicksort: obecně popište algoritmus a jeho časovou složitost v nejhorším a v průměrném případě.

- Mergesort: obecně popište algoritmus třídění sléváním a operaci slévání, vysvětlete jejich časovou a paměťovou složitost.

18 Teorie grafů

- Graf: vysvětlete pojmy vrchol, hrana, cesta, cyklus, strom, orientovaná hrana.
- Implementace: popište některé způsoby implementace grafu v programu, přehledově porovnejte jejich efektivitu pro různé úlohy.
- Minimální kostra: na libovolném příkladu porovnejte alespoň dva efektivní algoritmy.

19 Cesty v grafu

- Průchod grafem: na libovolném příkladu vysvětlete průchod grafem do šířky a do hloubky.
- Dijkstrův algoritmus: na libovolném příkladu popište Dijkstrův algoritmus hledání nejkratší cesty. Zmiňte se o potřebných datových strukturách a časové složitosti algoritmu.
- Strom: popište reprezentaci zakořeněného stromu v programu a operace přidání a odebrání prvku.

20 Databázové systémy

- Relační databáze: vysvětlete, jak je strukturovaná relační databáze, a jak v ní lze reprezentovat relace one-one, one-many a many-many.
- Software a hardware: popište obvyklé použití databáze, zmiňte se o některých databázových aplikacích.
- Dotazy: popište základní typy dotazů v jazyce SQL (Standard Query Language).
- Příklad: popište uplatnění databáze pro sklad. Navrhněte strukturu této databáze.

21 Výpočetní modely

- Regulární výrazy: popište některé základní syntaktické prvky. Zmiňte se o možných způsobech použití regulárního výrazu.
- Konečný automat: obecně popište tento pojem a vysvětlete jeho souvislost s regulárními výrazy.
- Turingův stroj: obecně popište průběh výpočtu Turingova stroje, vysvětlete jeho souvislost s běžnými počítači a stručně zmiňte, jaké úlohy je schopen řešit.