

MATURITNÍ OPAKOVACÍ TÉMATA: Informační technologie 18-20-M/01

INFORMAČNÍ SYSTÉMY

Teoretické základy algoritmizace a řešení problémů

- Definice algoritmu a vysvětlení jeho základních vlastností (konečnost, determinismus, elementárnost).
- Metody zápisu algoritmů (vývojové diagramy, pseudokód, strukturogramy).
- Princip dekompozice složitého problému na podproblémy (top-down přístup).
- Fáze návrhu a tvorby programu (od analýzy problému po testování).

Kódování, komprese a šifrování dat

- Objasnění principů ukládání textových dat (ASCII, Unicode, UTF-8).
- Vysvětlení rozdílu mezi bezztrátovou a ztrátovou kompresí dat vč. příkladů formátů.
- Charakteristika Huffmanova kódování jako příkladu redukce redundance.
- Princip symetrického a asymetrického šifrování (veřejný vs. soukromý klíč) a hashování.

Datové typy, proměnné a paměťová reprezentace

- Reprezentace celých a desetinných čísel ve dvojkové soustavě (dvojkový doplněk, standard IEEE 754).
- Rozdělení datových typů a rozdíl mezi hodnotovými (primitivními) a referenčními typy z hlediska alokace paměti (Zásobník (Stack) vs. Halda (Heap)).
- Problematika omezeného rozsahu hodnot a přetečení (overflow).
- Porovnání statického a dynamického typování, implicitní a explicitní konverze (přetypování).

Základní řídicí struktury a zpracování chyb

- Vysvětlení a demonstrace řídicích struktur: sekvence, větvení (if, switch) a cykly (for, while, do-while).

- Využití logických operátorů při tvorbě složených podmínek a zamezení vzniku nekonečných cyklů.
- Identifikace a ošetřování chyb za běhu programu (Runtime Errors).
- Princip a syntaxe zpracování výjimek (bloky try, catch, finally) a propagace výjimky.

Abstraktní datové struktury

- Charakteristika pole a řetězce (string) jako základních strukturovaných typů.
- Vysvětlení principu a použití lineárních struktur: zásobník (LIFO) a fronta (FIFO).
- Charakteristika spojových seznamů a jejich výhody oproti statickým polím.
- Využití slovníků/map (struktury typu klíč-hodnota).

Algoritmy vyhledávání a třídění, asymptotická složitost

- Vysvětlení pojmu časová a paměťová složitost algoritmu (asymptotická Big-O notace).
- Srovnání lineárního a binárního vyhledávání a stanovení podmínek pro jejich použití.
- Popis principu jednoduchých třídících algoritmů (např. Bubble sort, Insertion sort) a jejich limitů.
- Popis efektivních třídících algoritmů (např. Quicksort, Mergesort) a principu rozdělení a panování.

Pokročilé algoritmické principy a rekurze

- Vysvětlení pojmu rekurze a principu volání funkce sama sebe.
- Nutnost definice ukončovací (bázové) podmínky a rizika přetečení zásobníku (Stack Overflow).
- Porovnání rekurzivního a iterativního řešení problému (čitelnost vs. výkon).
- Příklady typického nasazení rekurze (faktoriál, procházení stromů či složek).

Základy objektově orientovaného programování

- Vysvětlení pojmů třída (šablona) a objekt (instance) vč. jejich vztahu.
- Definice atributů (stavu) a metod (chování) objektu.
- Význam konstruktoru a destrukturu v životním cyklu objektu.

- Objasnění principu zapouzdření (encapsulation) a využití modifikátorů přístupu (private, public).

Pokročilé principy OOP

- Vysvětlení principu dědičnosti a znovupoužitelnosti kódu.
- Přepisování metod (override) a volání metod předka.
- Charakteristika polymorfismu (stejné rozhraní, různé chování).
- Rozdíl mezi abstraktní třídou a rozhraním (interface) a řešení vícenásobné dědičnosti.

Principy čistého kódu a návrhové vzory

- Pravidla pro pojmenovávání proměnných, metod a tříd; význam formátování kódu a komentářů (Clean Code).
- Vysvětlení principu DRY (Don't Repeat Yourself) a rozčlenění kódu do logických celků (Single Responsibility).
- Účel a kategorizace softwarových návrhových vzorů (Design Patterns).
- Demonstrace vybraného návrhového vzoru (např. Singleton, Factory nebo Observer).

Práce se soubory a datové formáty

- Práce se souborovým systémem z pohledu aplikace (cesty, práva čtení/zápisu).
- Způsoby práce se streamy (proudy) při čtení a zápisu textových a binárních souborů.
- Vysvětlení pojmu serializace a deserializace dat.
- Charakteristika a syntaxe formátů JSON a XML pro uchovávání a přenos dat.

Návrh a implementace grafického rozhraní (GUI)

- Princip event-driven programování (událostně řízené programování) a naslouchače událostí (event listeners).
- Základní grafické komponenty (tlačítka, textová pole, seznamy) a jejich validace.
- Účel správců rozvržení (Layout Managers) při navrhování responzivních oken.
- Architektonické oddělení uživatelského rozhraní od aplikační logiky (např. vzor MVVM nebo MVC).

Relační databáze a datové modelování

- Vysvětlení pojmů entita, atribut a vztahy (1:1, 1:N, M:N) v rámci ER diagramu.
- Úloha primárního a cizího klíče pro zajištění vztahů mezi tabulkami.
- Způsob realizace vztahu M:N pomocí vazební tabulky.
- Důvody a postup normalizace databáze (odstranění redundance a anomálií).

Dotazovací jazyk SQL

- Rozdělení jazyka SQL na DDL, DML a DCL a příklady příkazů (CREATE, SELECT, INSERT, GRANT).
- Filtrování a řazení záznamů (WHERE, ORDER BY) a použití poddotazů.
- Agregační funkce a seskupování dat (GROUP BY, HAVING).
- Způsoby spojování tabulek pomocí JOIN (INNER, LEFT, RIGHT).

Bezpečnost dat a integrita v databázích

- Zajištění doménové, entitní a referenční integrity (omezení NOT NULL, UNIQUE, CHECK).
- Vysvětlení pojmu databázová transakce a vlastností ACID.
- Zásady zálohování a obnovy (recovery) databázových systémů.
- Princip útoku SQL injection a metody obrany (parametrizované dotazy).

Architektura informačních systémů

- Definice a účel informačního systému; role uživatelů a správa jejich oprávnění.
- Vysvětlení vícevrstvé architektury (datová, aplikační, prezentační vrstva).
- Charakteristika modelu klient-server a jeho výhody oproti monolitickým aplikacím.
- Fáze životního cyklu softwaru a nutnost testování a údržby.

Tvorba statických webových stránek a UI

- Struktura HTML5 dokumentu a význam sémantických značek (header, nav, section, footer).
- Formuláře, jejich prvky a validace na straně klienta.
- Kaskádové styly (CSS) a princip box modelu, selektory a specifická.
- Zajištění responzivity webu pomocí media queries, Flexboxu nebo CSS Gridu.

Klientské skriptování a dynamický web

- Začlenění a role jazyka JavaScript na straně klienta.
- Manipulace se strukturou HTML stránky prostřednictvím modelu DOM.
- Rozdíl mezi synchronním a asynchronním během programu; princip Callbacks a Promises (async/await).
- Komunikace se serverem na pozadí (Fetch API / AJAX) bez nutnosti obnovení stránky.

Serverové programování a frameworky

- Princip fungování webového serveru a zpracování HTTP požadavku (směrování (routing)).
- Význam MVC (Model-View-Controller) architektury u serverových aplikací.
- Zabezpečení přístupu k databázi ze serveru (Object-Relational Mapping - ORM).
- Výhody využití backendových frameworků a šablonovacích systémů.

Síťová komunikace a webové služby

- Charakteristika bezstavového protokolu HTTP a role stavových kódů (200, 404, 500...).
- Základní HTTP metody (GET, POST, PUT, DELETE) a jejich sémantika.
- Definice REST API a princip poskytování služeb (resources).
- Práce s bezstavovým protokolem pomocí Cookies a správa stavu (Session).

Zabezpečení webových aplikací

- Rozdíl mezi autentizací a autorizací uživatele.
- Moderní formy udržování relace (např. JSON Web Tokens - JWT).
- Rizika a ochrana proti zranitelnosti Cross-Site Scripting (XSS).
- Rizika a ochrana proti zranitelnosti Cross-Site Request Forgery (CSRF).

Správa verzí a týmová spolupráce (Git)

- Význam verzovacího systému pro individuální i týmový vývoj.
- Základní operace lokálního repozitáře (commit, historie změn).
- Práce se vzdáleným repozitářem (push, pull, clone) a platformy (GitHub, GitLab).

- Strategie větvení (branching), slučování kódu (merge) a řešení případných konfliktů.

Testování softwaru

- Rozdíl mezi manuálním a automatizovaným testováním aplikací.
- Kategorizace testů: jednotkové (Unit), integrační a systémové.
- Techniky Black-box a White-box testování; význam testování hraničních hodnot.
- Regresní testování a sledování chyb (sledování chyb (bug tracking)).

Kontejnerizace a nasazování aplikací

- Rozdíl mezi během aplikace v kontejneru a klasickým virtuálním strojem.
- Základní principy fungování Dockeru (Image, Container, Dockerfile).
- Přínosy kontejnerizace pro přenositelnost kódu ("u mě to funguje" problém).
- Úvod do procesů kontinuální integrace a nasazování (CI/CD pipelines).

Využití umělé inteligence ve vývoji softwaru

- Princip fungování velkých jazykových modelů (LLM) a jejich omezení (např. halucinace).
- Pravidla pro efektivní zadávání pokynů modelu (Prompt Engineering) a poskytování kontextu.
- Začlenění AI asistentů (Copilot) do vývojového prostředí (generování, doplňování a refaktorizace kódu).
- Rozdíl mezi běžným chatbotem a autonomním AI agentem (využití Tool Calling).