Gymnázium a Jazyková škola s právem státní jazykové zkoušky Zlín

CVIČENÍ Z INFORMATIKY – ALGORITMIZACE

Maturitní témata – profilová část MZ 2027

1. Algoritmus, zápis a správnost

pojem algoritmu a jeho vlastnosti – konečnost, determinovanost, obecnost, rezultativnost, správnost, příklady z běžného života; způsoby zápisu – vývojový diagram, strukturogram, pseudokód, převod slovního zadání na diagram; ověřování správnosti – trasovací (krokovací) tabulka, invariant cyklu, zdůvodnění ukončení cyklu, testování na vstupech, ukázka v Pythonu

2. Datové typy, výrazy a desetinná čísla

datové typy a proměnné – int, float, str, bool, None, statické a dynamické typování, explicitní a implicitní konverze; operátory a výrazy – priorita, celočíselné dělení a modulo u záporných čísel, zkrácené vyhodnocování, bitové operace stručně; desetinná čísla v programu – projevy nepřesnosti typu float v kódu, zaokrouhlovací chyby, porovnání pomocí math.isclose, funkce round a bankovní zaokrouhlení; peníze v programu – modul decimal, počítání v celých haléřích, výpočet ceny s DPH, ukázka v Pythonu

3. Větvení a cykly

řízení toku programu – podmíněný příkaz if, else, elif, vnořené podmínky, příkaz match, logické výrazy v podmínkách; cyklus s pevným počtem opakování for a funkce range, cyklus s podmínkou while, příkazy break a continue, nekonečný cyklus a jeho záměrné použití, vnořené cykly, invariant cyklu, volba vhodného typu cyklu, ukázka v Pythonu

4. Podprogramy a dekompozice problému

podprogramy – definice a volání funkce, parametry a argumenty, výchozí a pojmenované parametry, návratová hodnota, předávání hodnotou a odkazem a chování měnitelných objektů v Pythonu, lokální a globální proměnné, rozsah platnosti; dekompozice problému – návrh shora dolů, rozhraní funkcí, jedna funkce jeden úkol, znovupoužitelnost, rozdělení práce v týmu, ukázka v Pythonu

5. Řízení kreslicího stroje: želví grafika jako simulace plotru

želva jako model kreslicího nebo řezacího stroje (plotr, laserová řezačka, CNC) – stav stroje (poloha, směr, pero nahoře/dole) a příkazy, které ho mění; modul turtle – plátno a souřadnice, příkazy forward, backward, left, right, penup, pendown, goto, color, fillcolor, begin_fill, end_fill; relativní pohyb a absolutní souřadnice; obrazce cyklem – pravidelný mnohoúhelník a vnější úhel 360 / n, hvězda, spirála; funkce s parametry, skládání vzorů a náhodné polohy; přejezd bez kreslení a délka dráhy stroje, stručně jazyky strojů (HPGL, G-kód); rekurzivní strom stručně, praktické kreslení vzoru v Pythonu

6. Vstup, validace, výjimky a práce s datem

čtení vstupu od uživatele, převod řetězce na číslo a datum, validace typu, formátu a rozsahu, opakované dotazování; výjimky – běžné výjimky, bloky try, except, else, finally, zachycení konkrétní výjimky, raise a vlastní výjimky; práce s datem a časem –

modul datetime, třídy date, datetime a timedelta, převod strptime a formátování strftime, rozdíl dat, věk z data narození, pracovní dny, časová pásma, ukázka v Pythonu

7. Řetězce a textové úlohy

řetězec jako posloupnost znaků, indexace a řezy, neměnnost řetězců, základní metody (hledání, nahrazování, rozdělení a spojení, změna velikosti písmen), formátování výstupu (f-řetězce), kódování znaků a funkce ord a chr, hledání vzoru v textu – jak pracuje find a naivní algoritmus porovnávání vzoru, typické textové úlohy – palindromy, anagramy, četnost znaků, Caesarova a Vigenèrova šifra a jejich prolomení, ukázka v Pythonu

8. Regulární výrazy v Pythonu

regulární výraz jako vzor pro hledání v textu, syntaxe – znakové třídy, kvantifikátory, kotvy ^ a $, skupiny a alternativa, hladové a líné kvantifikátory; modul re v Pythonu – match, search, fullmatch, findall, sub, surové řetězce; validace formátu vstupu (e-mail, PSČ, telefon, rodné číslo jen formálně), vytěžení dat z textu a logu, hromadné nahrazení se skupinami; limity regulárních výrazů a jejich ladění, ukázka v Pythonu

9. Textové a binární soubory, CSV a JSON

textové a binární soubory – rozdíl, otevření a režimy r, w, a, rb, wb, konstrukce with, parametr encoding při otevření textového souboru a chyba UnicodeDecodeError, typ bytes, zjištění velikosti souboru, kopírování po blocích, rozpoznání typu souboru podle prvních bajtů; strukturovaná data – formát CSV, oddělovač a hlavička, modul csv, formát JSON, serializace a deserializace modulem json; zpracování dat ze souboru – statistika, ošetření vadných řádků, ukázka v Pythonu

10. Automatizace práce se soubory a složkami

automatizace opakované práce se soubory, modul pathlib – Path, iterdir, glob a rglob, name, stem, suffix, stat (velikost, datum změny); moduly os a shutil – vytvoření složek, kopírování, přesun, přejmenování; bezpečná práce – režim dry run, kolize názvů, zálohy; procházení podsložek, třídění podle přípony nebo data, hromadné přejmenování, protokol do CSV; spuštění skriptu z příkazové řádky s argumentem, ukázka v Pythonu

11. Pole, seznamy a matice

pole a seznam – rozdíl v pojetí, indexování včetně záporných indexů, průchod seznamem, hledání minima a maxima, součty a průměr, posuny a rotace prvků, řezy; dvourozměrná pole a matice jako seznam seznamů, průchod po řádcích a po sloupcích, součty řádků a sloupců, transpozice, otočení matice, sousedé buňky a hra Život, časová složitost průchodu, ukázka v Pythonu

12. Slovník, množina a hašování

slovník jako dvojice klíč – hodnota, vkládání, vyhledávání, mazání a průchod, metody get a items, použití pro četnosti a indexy; množina a množinové operace, odstranění duplicit; hašování – hašovací funkce a její vlastnosti, hašovací tabulka, kolize a jejich řešení řetězením a otevřeným adresováním, faktor naplnění a zvětšení tabulky, průměrná složitost O(1), neměnnost klíčů, ukázka v Pythonu

13. Dynamické datové struktury: spojový seznam, zásobník a fronta

statická a dynamická datová struktura, uzel s daty a odkazem na další uzel, jednosměrný spojový seznam – hlava, průchod, hledání, vložení na začátek a konec,

odebrání prvku, obousměrný seznam; zásobník nad uzly – push, pop, peek, test prázdnosti, LIFO; fronta nad uzly – ukazatel na začátek a konec, FIFO; srovnání se seznamem v Pythonu a collections.deque, složitost operací, pomalé list.pop(0); využití – undo, kontrola závorek, fronta požadavků, implementace v Pythonu

14. Lineární a binární vyhledávání

sekvenční vyhledávání v poli a jeho varianty (zarážka, všechny výskyty), binární vyhledávání v seřazeném poli – iterativní a rekurzivní verze, podmínka seřazenosti, časová složitost O(n) a O(log n), typické chyby (výpočet středu, nekonečný cyklus, meze), volba metody podle počtu hledání, ukázka v Pythonu

15. Algoritmy třídění: od bubble sortu k rychlému řazení

princip a využití třídění v praxi – řazení tabulek, příprava pro vyhledávání; bubble sort – průchody, porovnání a výměny, nejlepší a nejhorší případ, složitost $O(n^2)$; optimalizace – ukončení po průchodu bez výměny, zkracování prohledávaného úseku, obousměrný shaker sort; stabilita a řazení na místě; selection a insertion sort v přehledu; rychlé řazení intuitivně – princip rozděl a panuj, quicksort a pivot, průměrně O(n log n), pomalý případ na seřazených datech a volba pivotu; vestavěné řazení v Pythonu – sorted, sort, parametry key a reverse, Timsort, ukázka v Pythonu

16. Řešení programátorské úlohy

práce se zadáním programátorské úlohy – rozbor zadání (vstup, výstup, limity), oddělení příběhu od pravidel, vstupní soubor s T případy, předloha pro čtení vstupu a zápis výstupu, řešení jednoho případu jako funkce, testování na ukázkovém vstupu a vlastních testech, hraniční případy; průchod polem se stavem; základní a plná sada – odhad počtu operací z limitů, hrubá síla × lepší nápad, převedení na už vyřešenou úlohu; ukázka v Pythonu

17. Časová a paměťová složitost

měření efektivity algoritmu, časová a paměťová složitost, O-notace a běžné třídy (O(1), O(log n), O(n), O(n log n), $O(n^2)$, $O(2^n)$), nejhorší, průměrný a nejlepší případ, odhad složitosti z kódu (cykly, vnořené cykly, půlení), kompromis čas × paměť, memoizace a cache, měření doby běhu a profilace, hledání úzkého hrdla, ukázka v Pythonu

18. Grafy: reprezentace, BFS, DFS a Dijkstra

graf – vrcholy a hrany, orientované a ohodnocené grafy, reprezentace maticí sousednosti a seznamem sousedů, prohledávání do šířky (BFS) s frontou a nejkratší cesta v neohodnoceném grafu, prohledávání do hloubky (DFS) rekurzí nebo zásobníkem, komponenty souvislosti a detekce cyklu, Dijkstrův algoritmus – princip a ruční průběh v tabulce (bez implementace), praktické použití (mapy, sítě, bludiště), ukázka v Pythonu

19. Vizualizace grafu a stromu

strom jako zvláštní případ grafu – souvislý graf bez cyklů, n vrcholů a n – 1 hran, kořen, rodič, potomek, list, úroveň vrcholu; stromy kolem nás – složky na disku, rodokmen, organizační schéma, turnajový pavouk; uložení grafu jako seznam hran v souboru a slovník sousedů, uložení stromu slovníkem rodičů; rozmístění vrcholů na kružnici pomocí cos a sin a kreslení vrcholů a hran želví grafikou; animace průchodu do šířky podle pořadí z hotové funkce; vykreslení stromu po úrovních, kdy úrovně spočítá jeden cyklus bez rekurze; rozpoznání, zda je graf stromem; praktická vizualizace datové struktury v Pythonu

20. Náhoda a simulace

skutečná náhoda a pseudonáhodnost, generátor pseudonáhodných čísel a semínko (seed), opakovatelnost výsledků, modul random – random, randint, choice, shuffle, sample; modul secrets pro hesla a tokeny; simulace náhodných pokusů – hod kostkou, četnost a relativní četnost, porovnání s teoretickou pravděpodobností, vliv počtu opakování na přesnost; metoda Monte Carlo – odhad čísla π a pravděpodobnosti, jednoduchá simulace fronty nebo epidemie, ukázka v Pythonu

21. Základy objektově orientovaného programování

principy objektově orientovaného programování nezávisle na jazyce – třída a objekt (instance), stav (atributy) a chování (metody), konstruktor a odkaz na aktuální objekt, zapouzdření a přístup k atributům, statické (třídní) členy, dědičnost a volání konstruktoru předka, přepisování metod a polymorfismus, abstraktní třída a rozhraní, kompozice × dědičnost, návrh tříd a skica diagramu tříd UML; ukázka v Pythonu, nebo v C# podle volby studenta – srovnání zápisu (self × this, __init__ × konstruktor, konvence _ × private, super() × base)

22. Grafické rozhraní a událostmi řízené programování

událostmi řízené programování – rozdíl oproti sekvenčnímu běhu, smyčka událostí, události (klik, klávesa, časovač) a jejich obsluha (callback); ovládací prvky a jejich rozvržení; validace vstupu v GUI a hlášení chyb, oddělení logiky od rozhraní; Python tkinter – Tk, Label, Entry, Button, grid a pack, mainloop, messagebox; C# WinForms ve Visual Studiu – návrhář formuláře, Form, TextBox, Button, Label, událost Click, TryParse; praktická úprava aplikace

23. Textová hra: hlavní smyčka, stav a stavový automat

návrh malé interaktivní konzolové aplikace na příkladu textové únikové hry – hlavní smyčka, načtení a ošetření příkazu hráče, ukončení; stav hry v proměnných a datových strukturách – aktuální místnost, inventář, svět hry jako slovník místností s popisem, východy a předměty; rozdělení programu do funkcí; hra jako stavový automat – místnosti jako stavy, příkazy jako přechody, diagram stavů; uložení a načtení rozehrané hry ve formátu JSON, ošetření chybějícího souboru; přívětivost konzole – srozumitelné výpisy a hlášky; ukázka v Pythonu

24. Chyby, testování a čitelnost kódu

druhy chyb – syntaktická, běhová a logická, čtení chybového hlášení a tracebacku, typy výjimek; hledání chyb – ladicí výpisy, modul logging a jeho úrovně; testování – testovací případ a očekávaný výsledek, hraniční hodnoty, assert, jednoduché unit testy v unittest nebo pytest; čitelnost kódu – pojmenování, funkce s jednou odpovědností, komentáře a docstringy, PEP 8, refaktoring beze změny chování ověřený testy, ukázka v Pythonu

25. Vývojové prostředí, ladění a audit kódu od AI

textový editor × IDE, VS Code – otevření složky jako pracovního prostoru, rozšíření Python a Pylance, volba interpretu a virtuálního prostředí venv, integrovaný terminál, spuštění programu; ladění v debuggeru – breakpoint, krokování, panel proměnných, sledování výrazu, zásobník volání; IntelliSense, přejmenování symbolu, formátování, panel Problémy a linter, zmínka o Gitu; AI asistent – zadání se specifikací a příklady, audit kódu od AI (čtení, testy, okrajové případy, vymyšlené funkce), obhajoba každého řádku, rizika, praktická ukázka

Zlín září 2026

Mgr. Michal Mikláš	Mgr. Přemysl Šil, MBA, BBA
garant předmětového týmu	ředitel školy