

Témata

1. Operační systém, Data a informace, HTML a CSS

- Definujte operační systém. Vysvětlete jeho funkci a strukturu. Popište druhy rozhraní a celkovou filozofii operačního systému. Popište princip spuštění počítače.
- Co je základní jednotka informace, jaké jsou její násobky? Jak se provádí převod dat do digitální podoby (A/D převodník). Co znamená kódování informací, záznam (úložiště) a přenos informací (přenosová cesta). Co znamená bezztrátová a ztrátová komprese dat? Jaké formáty souborů využívají kompresi dat, určete typ komprese u těchto souborů.
- Charakterizujte, jakým způsobem vytváříme HTML stránky, co je to párová a nepárová značka. Co znamená CSS a k čemu lze kaskádové styly použít. Rozdělte CSS a na jednotlivých příkladech vysvětlete použití.

2. Textové editory, Algoritmizace, OS Linux vs. Windows

- Objašněte postup při formátování textu a použití stylů v dokumentu. Objašněte typografická pravidla pro psaní textových souborů.
- Co je algoritmus, jaké jsou jeho vlastnosti? Objašněte schematické vyjádření algoritmu vývojovým diagramem, symboly vývojového diagramu, zápis algoritmu v programovacím jazyku. Uveďte jednoduchý příklad = nakreslete vývojový diagram a proveďte zápis téhož algoritmu v programovacím jazyku Python.
- Porovnejte OS Linux s operačním systémem Windows a naznačte jejich výhody a nevýhody.

3. Tabulkové procesory, Programovací jazyky, Grafické formáty

- Vysvětlete pojem tabulkový procesor. Popište, z jakých částí se skládá sešit, co vše můžeme zapsat do buňky. Ukažte efektivní práci s tabulkovým procesorem.
- Vysvětlete kritérium rozdělení programovacích jazyků na kompilované a interpretované, co to je kompilátor, co je to interpret. Uveďte příklad kompilovaného jazyka a kompilátoru, uveďte příklad interpretovaného jazyka a interpretu.
- Popište, jaký je rozdíl mezi rastrovým a vektorovým formátem. Popište základní rozdíl v definici a tím i výhody/nevýhody obou typů 2D grafiky. Vysvětlete formáty GIF, JPG, JPEG, BMP, PNG, PDF. Objašněte pojmy pixel, DPI.

4. Databáze, Python - datové typy a datové struktury, HTML

- Popište databázi MS Access, vysvětlete její použití v informačních systémech. Popište postup vytvoření tabulky, vlastnosti datových typů, jejich význam a práce s tabulkou.
- Na příkladech ukažte použití základních datových typů jazyka Python (integer, float, boolean, string), popište a ukažte použití pokročilých datových struktur jazyka Python (tuple, list, dictionary, set). Popište vlastnosti pokročilých datových struktur. (Viz příloha

- 1)
- Popište základní strukturu HTML dokumentu. Jakým způsobem vytváříme nadpisy, odstavce, seznamy a výčty, vodorovné čáry, obrázky, hypertextové odkazy a tabulky.
5. **Prezentace, Strukturované programování, DB dotazy v MS Access**
 - Vyjmenujte základní pravidla pro tvorbu prezentace. Vysvětlete strategii vytváření prezentace. Srozumitelně předvedte práci s jednotlivými snímky, jejich návrh a zpracování. (text, odrážky, obrázky, grafy, tabulky, animace.)
 - Vývojovým diagramem znázorněte základní struktury strukturovaného programování (příkaz, podmíněný příkaz, cyklus for a cyklus while) a přepište je do jazyku Python se správnou syntaxí. Vysvětlete použití funkcí na příkladu v jazyku Python.
 - Popište základní typy dotazů v databázi MS Access, strukturu dotazu, třídění informací, podmínky zobrazení informací.
6. **Internet, Objektové programování, Barevné systémy**
 - Vysvětlete pojem Internet, dále objasněte pojmy World Wide Web (WWW), Internet Service Provider (ISP), Uniform Resource Locator (URL).
 - Popište základní charakteristiky objektového programování, objasněte výhody konceptu OOP, na příkladu objektu (class Geometricky_utvar (Viz příloha 2)) v jazyku Python ukažte, co je to vlastnost objektu a co je to metoda objektu.
 - Na jakém principu zobrazuje barvy RGB systém, popište jeho vlastnosti. Popište další způsob zobrazení barev CMY(K), základní rozdíly a kde se používá.
7. **Tvorba WWW stránek, IDE, Hromadná korespondence**
 - Objasněte podmínky pro používání značkovacího jazyka a popište základní strukturu HTML dokumentu. Vysvětlete pojem META informace a DTD - Dokument Type Definition.
 - Vysvětlete pojem IDE (Integrated development environment). Na příkladu Pycharm ukažte okno struktury projektu, okno textového editoru, způsob spuštění programu, konzolový výstup, instalaci knihoven.
 - Na příkladu předvedte použití „Hromadné korespondence“ v MS Word. Vysvětlete přípravu dokumentu k tisku, použití záhlaví a zápatí, práci s tabulkou a grafické prostředí.
8. **Základy počítačové grafiky, Databáze SQL, start PC**
 - Vysvětlete pojem rastrová grafika a princip jejího definování, uveďte její výhody a nevýhody, možnosti získání, zpracování a příklady použití bitmapové/rastrové grafiky v každodenní praxi, uveďte příklady souborových formátů digitální grafiky, jejich rozdíly, výhody/nevýhody a užití.
 - Co to je SQL, popište strukturu SQL databáze (tabulky, relace). Ukažte jaké SQL příkazy obsahují jednotlivé podkategorie DDL,

- DML, DCL. Vysvětlete syntaxi základních příkazů (CREATE TABLE, INSERT, UPDATE, SELECT, COMMIT).
- popište kontrolu technického vybavení počítače (POST) při zapnutí počítače.
9. **Relační databáze MS Access, Uživatelské rozhraní, Vektorová grafika**
- Hlavní objekty databáze MS Access a jejich účel.
 - Popište základní typy uživatelských rozhraní (CLI, GUI, webové). Popište nástroje používané v jazyku Python pro tvorbu rozhraní (click, Tkinter, Qt, Flask, Django).
 - Vysvětlete pojem vektorová grafika a princip jejího definování, uveďte její výhody a nevýhody. Uveďte možnosti získání, zpracování a příklady použití vektorové grafiky v každodenní praxi.
10. **Internet věcí, Cykly jazyku Python, Grafy v MS Excel**
- uveďte, jaké licencované pásmo využívají sítě Lora v Evropě, provozatele LoRa sítí v České republice. Popište MQTT protokol a jeho využití. Uveďte příklady nasazení platform ESP a Rpi v IoT. Popište nástroje pro automatizaci v průmyslu 4.0
 - Popište typy cyklů používaných v jazyku Python. Vysvětlete, kdy který cyklus použijete při tvorbě programu, uveďte příklad použití cyklu FOR pro procházení seznamu, uveďte příklad použití cyklu WHILE při ošetření uživatelského vstupu. Příklady znázorněte vývojovým diagramem a запиšte v jazyku Python.
 - Na příkladu objasněte vytvoření grafu podle tabulky v MS Excel a úpravu vzhledu stránky před tiskem (záhlaví a zápatí).
11. **Operační systém Linux, Vývojové fáze aplikace, internet věcí**
- Popište historický vývoj OS Linux. Vysvětlete pojem distribuce OS Linux, uveďte příklady distribucí. Popište strukturu OS Linux (například slupkovým modelem). Uveďte kde se OS Linux používá (set-top-box, ..., server). Vysvětlete co je to SHELL. Uveďte příklady desktopových grafických prostředí (Xfce, ...).
 - Popište jednotlivé vývojové fáze aplikace (Plánování a příprava, analýza a návrh aplikace ...). Uveďte jak je v procesu zapojen programátor-analytik, kódér, správce systému, tester, zákazník, obchodník. Uveďte jaké typy testů a kdy se ve vývojovém cyklu aplikace používají.
 - Internet věcí - popište oblasti využití internetu věcí a jejich přínosy. Jaké jsou možnosti bezdrátového přenosu dat v IoT? Vysvětlete pojmy LoRa, LoRaWAN a SigFox.
12. **Prezentace, Framework click, Virtuální prostředí v jazyku Python**
- Názorně ukažte práci s přechody snímků v prezentaci, závěrečnou úpravu prezentace a způsoby uložení.
 - Vysvětlete co to je framework, jakým způsobem šetří programátorovi práci. Vysvětlete k čemu slouží framework click v programovacím jazyku Python a popište čím se programátor nemusí již zabývat, když

- tento framework využije.
- Popište důvody použití virtuálního prostředí v jazyku Python. Ukažte jakými způsoby ho lze vytvořit (IDE Pycharm, terminal), jak se aktivuje, jakými způsoby do něj instalujeme moduly (IDE Pycharm, terminal). Popište k čemu slouží příkaz freeze instalátoru pip, soubor requirements.txt a jak nainstalujeme moduly do virtuálního prostředí z requirements.txt. Ukažte jak spustíme program v jazyku Python ve virtuálním prostředí.
13. **Virtualizace, práce s textem, Správa OS Linux**
- Objasněte pojem virtualizace, co znamená host a hostitel. Uveďte nástroje určené pro profesionální virtualizaci serverů. Ukažte prakticky vytvoření virtuálního počítače (nastavení procesoru, paměti a úložiště) v programu VirtualBox, popište další kroky, které je potřeba učinit pro nainstalování OS na virtuálním počítači. Popište možnosti sdíleného úložiště s hostitelem a nastavení síťového rozhraní virtuálního počítače, klonování virtuálního počítače.
 - Vysvětlete na příkladu jednoduchého (MS Word) a profesionálního softwaru (Adobe InDesign) možnosti práce s textem, obrázky, fotografiemi, barvami a dalšími grafickými prvky na počítači.
 - Vysvětlete vytvoření uživatele, přiřazení uživatele do skupiny. Uživatelská oprávnění, nastavení práv k souborům a adresářům.
14. **Docker / orchestrace kontejnerů, převody jednotlivých typů a druhů grafiky mezi sebou, NoSQL databáze**
- Vysvětlete pojem lehké virtualizace a porovnejte její výhody a nevýhody oproti plně virtualizaci. Popište základní pojmy lehké virtualizace s využitím Dockeru - Docker image, Dockerfile, Docker kontejner. Vysvětlete k čemu slouží nástroje pro orchestraci kontejnerů (Kubernetes)
 - Vysvětlete převody jednotlivých typů a druhů 2D a 3D vektorové i rastrové grafiky mezi sebou navzájem – možnosti, výhody, omezení, důvody
 - Vysvětlete základní rozdíl mezi SQL a NoSQL databázemi. Zdůvodněte proč je využití SQL databází častější. Uveďte a stručně popište používané datové modely NoSQL databází (klíč-hodnota, sloupcové úložiště, úložiště typu graf). Uveďte zástupce NoSQL databází. Charakterizujte databázi TinyDB a popište její datový model, způsob vkládání dat a kladení dotazů.
15. **AI, Git, 3D Grafika na PC – základní principy - software**
- Vysvětlete pojem umělé inteligence. Vysvětlete aplikaci umělé inteligence ve formě LLM (large language model). Uveďte strukturu neuronové sítě používanou pro LLM a historii jejího vzniku. Uveďte zástupce SW umělé inteligence a oblast jejich použití (ChatGPT-jazykový model, Dall.e - tvorba obrazu, ...). Na příkladu ukažte využití AI v oblasti programování (provedte slovní zadání a posuďte získaný kód).
 - Co znamená VCS? Na příkladu nejpoužívanějšího VCS Git popište

- způsob verzování. Objasněte pojmy staging area, commit, repozitář, branch, merge, konflikt při merge-ování.
 - 3D Grafika na PC – vysvětlete základní principy konstrukce a simulace, uveďte používaný software, druhy a výhody/nevýhody.
16. **Github, ARM architektura mikrokontrolery, zásadní rozdíly v přípravě grafiky pro tisk anebo webové platformy/mobilní aplikace**
- Vysvětlete k čemu slouží Github. Objasněte pojmy Pull from Github, Push to Github, Github branch, Github fork, Github send pull request
 - Vysvětlete co to je mikrokontroler. Vysvětlete co znamená ARM platforma. Uveďte příklady konkrétních zařízení (ESP, Arduino, Rpi), možnosti programování, použité programovací jazyky (C/C++, Python ...), způsoby komunikace (seriová, SPI, I2C, ssh, ...). Vysvětlete k čemu slouží I/O piny, porovnejte jednotlivá zařízení z hlediska napěťových úrovní signálů. Porovnejte jednotlivé platformy z hlediska vhodnosti pro různé typy projektů, zejména s ohledem na I/O periférie.
 - Popište zásadní rozdíly v přípravě grafiky pro tisk anebo webové platformy/mobilní aplikace (barevné modely, rozlišení, práce s textem)
17. **Webové stránky, Kybernetická bezpečnost, MS Excel**
- Co znamená zkratka PHP. Na jakém principu pracují dynamické stránky. Vysvětlete filosofii webové aplikace a vysvětlete funkci na příkladech. Popište SEO optimalizaci webu.
 - Jakými prostředky chrání systém správce sítě? Objasněte, jak uživatel pozná bezpečné internetové stránky. Popište, jaké nebezpečí hrozí používáním internetu z hlediska počítačových virů – spyware, malware a další rizika. Jak se chráníte proti těmto rizikům na různých OS jako uživatel?
 - Jakým způsobem adresujeme buňku (relativní, absolutní adresace) a jak dokážeme pojmenovat buňku a oblast buněk? Co může být obsahem buněk v tabulkovém procesoru MS Excel (hodnota, vzorec)? Vše ukažte na praktických příkladech.
18. **Formát JSON, Licence SW, AI – umělá inteligence, definování obrazové složky slovním popisem**
- Popište formát JSON. Jaký modul a jaké pokročilé datové struktury v jazyku Python umožňují přímé zpracování tohoto formátu? Na příkladu url = “https://api.spacexdata.com/v4/rockets” ukažte praktické využití formátu JSON pro přenos dat z API. Vysvětlete pojem API.
 - Co to je SW Licence? Co to je autorský zákon? Vysvětlete pojem open source, historii vzniku myšlenky open source a uveďte jemu odpovídající licence. Vysvětlete co to je EULA.
 - AI – umělá inteligence, generování obrazové složky slovním popisem. Výhody, omezení, rozlišení, výhody/nevýhody, použití (retuš, manipulace, možnosti zneužití, autorské právo)

19. Framework Flask, správa OS, Elektronická komunikace

- Vysvětlete co to je framework, jakým způsobem šetří programátorovi práci. Vysvětlete k čemu slouží framework Flask v programovacím jazyku Python a popište čím se programátor nemusí již zabývat, když tento framework využije.
- Popište, jak vytváříme uživatelské účty a proč. Popište správa počítače, nástroje pro správu.
- V čem spočívá elektronická komunikace? Vysvětlete základní pojmy, jako jsou e-mail, SMS, VoIP, chat.

20. OOP dědičnost, Linux-terminál, SQL databáze

- Popište OOP obecně a zdůvodněte proč tento koncept využíváme v programování. Popište základní charakteristiky OOP. Na příkladu objektů Geometricky_utvar -> Ctverec (Viz příloha 2) vysvětlete pojem dědičnost aplikovanou v jazyku Python.
- Co byl původně terminál v OS Unix? Proč v OS Linux používáme emulátor terminálu (terminál)? Co to je shell? Uveď různé shelly v OS Linux a obecně popiš v čem se mohou lišit. Prakticky předved' práci v terminálu na příkladu příkazu ls a popiš parametry příkazu (-help, -l, -a, -la).
- Popište stručně databáze SQLite a mySQL. Vysvětlete základní rozdíl při využití SQLite a mySQL v aplikaci (jakou databázi v jakém typu aplikace použijeme).

21. OOP enkapsulace, Instalace aplikačního SW v Unix like systémech, GPIO Raspberry Pi 4

- Popište OOP obecně a zdůvodněte proč tento koncept využíváme v programování. Popište základní charakteristiky OOP. Na příkladu v jazyku Python s objektem Geometricky_utvar (Viz příloha 2) vysvětlete pojem enkapsulace.
- Popište způsoby instalace uživatelského SW na OS Linux (v terminálu, s grafickým rozhraním) a ostatních Unix like systémech. Objašněte pojmy balíčkovací systémy OS Linux, App Store - iOS, Google Play - Android, Snap - Linux, Flatpack - Linux. Uveďte příklady příkazů balíčkovacího systému apt.
- Vysvětlete pojem GPIO u Raspberry Pi 4, na webu najdete obrázek pro rozložení pinů a popište jej. Vysvětlete problém připojení periferie určené pro ARDUINO na GPIO Raspberry Pi.

22. OOP polymorfismus, HTML formát, HTTPS protokol

- Popište OOP obecně a zdůvodněte proč tento koncept využíváme v programování. Popište základní charakteristiky OOP. Na příkladu v jazyku Python s objekty Geometrick_utvar_ -> Ctverec, Obdelnik, Kruh (Viz příloha 2) vysvětlete pojem polymorfismus.
- Popište formát HTML (účel, historii vývoje), uveďte k čemu slouží jednotlivé tagy (html, head, body, div, span ...) a pomocí nástrojů pro vývojáře v prohlížeči ukažte na libovolné webové stránce jejich použití.
- Co to je HTTPS protokol. Uveďte možnosti předání požadavku

na server pomocí HTTPS protokolu (pomocí webového prohlížeče, z terminálu pomocí příkazu curl, z Pythonu pomocí requests.get() - první 2 možnosti předvedte prakticky na příkladu <https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45>, co to je URL, jak uvádíme parametry předávané pomocí metody GET)

23. CSS, VOIP, Linux - Kernel

- Vysvětlete co to je CSS. Popište stylování pomocí CSS. Ukažte jak fungují jednotlivá pravidla a pomocí nástrojů pro vývojáře v prohlížeči ukažte na libovolné webové stránce jejich použití.
- Vysvětlete pojem VOIP. Vysvětlete jak funguje přeměna analogového signálu na digitální, jak se přenáší “hlas” po internetu a co to je VOIP brána. Vysvětlete jakým způsobem se dovolá VOIP telefon na telefon pevné linky (nakreslete jednoduché schéma). Objasněte výhody a nevýhody HW a SW VOIP ústředn.
- Vysvětlete co znamená kernel v OS Linux a pohovořte o historickém vývoji. Z jakých částí se kernel skládá, jak je zabezpečen přístup vyšších vrstev OS k jádru (SYSCALL)

24. Formát JSON vs. XML, Linux přístupová práva, Python - operátory

- Popište obecně formát JSON, historii a důvod jeho vzniku. Na URL <https://api.sunrise-sunset.org/json?lat=36.900&lng=-4.45> načtěte webovým prohlížečem data ve formátu JSON a porovnejte, jak by vypadal stejný obsah dat ve formátu XML. Zdůvodněte proč je JSON vhodnější pro přenos než XML.
- Vysvětlete jak jsou zabezpečena přístupová práva uživatelů v OS Linux. Vysvětlete co je owner, group, others. Ukažte na praktickém příkladu jak jsou práva zobrazena ve výpisu souboru. Vysvětlete co znamenají symboly -rwxrwxrwx u souboru a drwxrwxrwx u adresáře. Vytvořte spustitelný soubor obsahující pouze příkaz ls -al a nastavte mu práva tak, abyste jej mohli spouštět, ale aby nikdo jiný jej nemohl ani přeciť. Co znamenají příkazy chmod 600 soubor.txt, chmod u-w soubor.txt, chmod ug=rwx soubor.txt?
- Co jsou to operátory v jazyku Python? Rozdělte operátory do skupin podle způsobu použití (aritmetické,) a uveďte, které operátory do které skupiny patří.

25. JS, OS Linux - procesy, HTTP vs. HTTPS

- Co to je JavaScript? Pohovořte o vzniku JS, na které straně (server - uživatel) lze JS použít (uvažujte i NodeJS). Za jakým účelem je JS používán nejčastěji a jaký je důvod jeho použití?
- Co to je v OS Linux proces, v jakém stavu se proces může nacházet? Objasněte co to je rodičovský proces, systemd, daemon. Objasněte příkazy pro správu procesů (ps, bg, fg, top, kill, nice)
- Obecně popište HTTP protokol, vysvětlete jaký je rozdíl mezi HTTP a HTTPS protokolem.

Přílohy

Příloha 1

```
i = -1
f = 300.2
m = 'Hello world\n'

t = (12.4, 45.6, 0, 0)
l = [1, 2, 3, 4, 5, 5]
s = {'apple', 'banana', 'orange', 'raspberry', 'orange'}
d = {
    'mark': 'Ford',
    'type': 'Mustang',
    'year': 1966,
    'year': 1967,
    'colors': ['wimbledon white', 'rangoon red', 'poppy red', 'guardsman blue', 'prairie bro
```

Příloha 2

```
import math

class Geometricky_utvar:
    def __init__(self, nazev=None):
        self.nazev = nazev
        self.obvod = self.vypocti_obvod()
        self.obsah = self.vypocti_obsah()
        self.params = self.params()

    def vypocti_obvod(self):
        return None

    def vypocti_obsah(self):
        return None

    def params(self):
        return None

class Ctverec(Geometricky_utvar):
    def __init__(self, strana):
        self.strana = strana
        super().__init__('Ctverec')

    def vypocti_obvod(self):
```



```

        return 4 * self.strana

    def vypocti_obsah(self):
        return self.strana ** 2

    def params(self):
        return 'a = ' + str(self.strana)

class Obdelnik(Geometricky_utvar):
    def __init__(self, strana_a, strana_b):
        self.strana_a = strana_a
        self.strana_b = strana_b
        super().__init__('Obdelnik')

    def vypocti_obvod(self):
        return 2 * self.strana_a + 2 * self.strana_b

    def vypocti_obsah(self):
        return self.strana_a * self.strana_b

    def params(self):
        return 'a = ' + str(self.strana_a) + ', b = ' + str(self.strana_b)

class Kruh(Geometricky_utvar):
    def __init__(self, polomer):
        self.polomer = polomer
        super().__init__('Kruh')

    def vypocti_obvod(self):
        return 2 * math.pi * self.polomer

    def vypocti_obsah(self):
        return math.pi * (self.polomer ** 2)

    def params(self):
        return 'r = ' + str(self.polomer)

def main():
    utvary = [Ctverec(5), Kruh(4), Obdelnik(5,2)]
    for u in utvary:
        print(u.nazev, 's parametry', u.params, 'ma obvod', u.obvod, 'a obsah', u.obsah, 'a

```

```
if __name__ == '__main__':  
    main()
```