**Python:**

1. Name at least three mutable and immutable built-in types.

Mutable:

* Lists
* Dictionaries
* Sets

Immutable:

* Integers, floats
* Strings
* Tuples

1. Jaký je rozdíl list a tuples

Hlavním rozdílem spočívá v jejich přeměnlivosti.

Seznamy jsou mutable, což znamená, že mohou být změněny po jejich vytvoření, zatímco tuples (n-tice) jsou immutable, což znamená, že jejich obsah nemůže být po vytvoření změněn.

Z hlediska databází dává smysl, že použití seznamů se využívá pro ukládání objektů, na které budeme později aplikovat změny nebo filtrovat podle určitých kritérií.

Z hlediska databází dává smysl použití tuples na ukládání neměnných hodnot.  
Například jména a oříjmení osoby.

1. Co se stane, pokud předáte argument do funkce a uvnitř ho změníte?  
   Změní se hodnota tohoto argumentu po zavolání dané funkce?
   * Ano, změní se hodnota argumentu uvnitř funkce a projeví se i po zavolání funkce.  
     V případě „extendlist“ budou výsledky následující:  
     [42, 'item']

[42]

[42, 'item']

V případě, že nezadáme nový list, který přepíše původně vytvořený list, který se založí při startu programu, tak přenáší hodnoty. Toho si můžeme všimnout u list1, který vypíše stejný obsah jako list3, na rozdíl od toho, kdy zadáme prázdný list v list2 a vypíše se pouze 42.

1. Jaký je rozdíl mezi shallow a deep kopií v Pythone?
   * Rozdíl je v rozdílné hloubce kopírování.  
     Zatímco „shallow“ copy vytváří nový objekt, který obsahuje odkazy na stejné objekty jako kopírovaný objekt, tak „deepcopy“ vytváří zcela nový objekt s vlastními kopiemi všech prvků. To znamená, že pokud se provede změna prvku v seznamu, neovlivní to jeho deep kopii.
   * A picture containing text, screenshot, font

     Description automatically generated
2. Jaké možnosti nabízí python pro implementování konkurentních výpočtů? Stručně je popište.
   1. Threading – používá vlákna (threads), což umožňuje paralelní zpracování úloh. Což může vest k neefektivnímu řešení využití vláken kvůli tzv. Global Interpreter Lock (GIL).
   2. Multiprocessing – používá procesy, což umožňuje paralelní zpracování úloh na vice jádrech CPU. V tomto případě GIL nehraje roli, protože každý process má vlastní proctor pro interpret.

**Design pattern, python idiomy a OOP:**

1. Stručně popište návrhový vzor singleton. Uveďte alespoň 1 příklad praktického využití.
   * Návrhový vzor singleton se zabývá vytvářením objektů. Hlavním účelem je zajistit, že bude existovat pouze jedna instance třídy v rámci aplikace a umožnit přístup k této instanci odkudkoliv v kódu.  
     Příkladem praktického využití může být například připojení k databázi.
2. Jak by jste implementovali iterator třídu (class)? Co by měla obsahovat (například jaké metody)?
   * Pro implementaci iteratoru ve třídě bychom mohli využít metody \_\_init\_\_ \_\_iter\_\_ a \_\_next\_\_
3. Co to je monkey patching?
   1. Monkey patching je technika v programování, kdy se přepisuje kód již definované funkce s cílem změnit podstatu/chování této funkce bez přímé úpravy původního zdrojového kódu.
   2. Tato technika je užitečná pro rychlou opravu chyb. Nicméně přepsání kódu může vést k narušení stability kódu a bezpečnosti.
4. Co to je dependency injection a jak byste ji implementovali?
   1. Jedná se o techniku, která odděluje závislosti komponent od implementace tím, že poskytuje objekty, které potřebují další objekty k fungování.
   2. Implementace by mohla být za pomocí knihoven jako je např. injector
5. Jaký je rozdíl mezi instanční, statickou a třídní metodou?
   1. Instanční:
      1. Je vázaná na instanci třídy, má přístup k atributům této instance přes klíčové slovo „self“. Každá instance může mít vlastní kopii atributů a hodnot, což znamená, že instanční metoda může pracovat s unikátním stavem každé instance.
   2. Statická:
      1. Nemá přístup k datům instance ani třídy a provádí pouze operaci na vstupních argumentech.
   3. Třídní:
      1. Má přístup k datům třídy, ale ne k datům instance. Třídní metody jsou obecně používány pro manipulaci s daty třídy jako celku a vytvoření nových instancí této třídy.
6. Co jsou to dunder/magické metody? Vyjmenujte a stručně popište pár z nich
   1. Dunder (double underscore) nebo magické metody jsou speciální metody definované v Pythonu, které umožňují vytvářet objekty se specifickým chováním. Tyto metody lze rozpoznat podle jejich názvu, který začíná a končí dvojicí podtržítek.
   2. Příklady:
      1. \_\_init\_\_(self, parametry)
         1. Tato metoda se volá při vytváření instance třídy a slouží k inicializaci jejích atributů.
         2. “self” je prvním parametrem této metody a reprezentuje nově vytvořenou instanci třídy. Ostatní parametry jsou závislé na konkrétní implementaci.
      2. \_\_len\_\_(self)
         1. Tato metoda slouží k získání délky objektu.
         2. “self” je jediným parametrem této metody a reprezentuje object,u kterého chceme zjistit délku.
      3. \_\_iter\_\_(self)
         1. Tato metoda slouží k iteraci přes object pomocí cyklu for.
         2. “self” je jediným parametrem této metody a reprezentuje object, přes který chceme iterovat.
      4. \_\_str\_\_(self)
         1. Tato metoda slouží k získání řetězci objektu.
         2. “self” je jediným parametrem této metody a reprezentuje object, jehož řetězec chceme získat.
      5. \_\_add\_\_(self, other)
         1. Tato metoda umožňuje definovat chování instance třídy při sčítání s jinou instancí třídy.
7. Co jsou a jak byste implementovali privátní, chráněné, a veřejné atributy třídy v Pythonu? Co jejich použití znamená pro třídní dědičnost?
   1. Veřejné atributy:
      1. Jsou standardně implementované pomocí obyčejných atributů. Tyto atributy jsou přístupné zvenčí třídy bez jakýchkoliv omezení.
      2. Základem je psát jména atributů malými písmeny a oddělovat slova podtržítkem.
   2. Chráněné atributy:
      1. Základem je psát jména atributů malými písmeny, slova oddělovat podrtržítkem.
      2. Značí se podtržítkem na začátku před prvním slovem.
      3. Jsou přístupné pouze z různých instancí tříd, které dědí od této třídy a nikoliv z vnějšího kódu.
   3. Privátní atributy:
      1. Značí se dvojitým podtržítkem před názvem atributu (první slovo).  
         To změní jméno atributu tak, aby k němu bylo možné přistupovat pouze uvnitř třídy.
      2. Nelze k němu přistupovat z metody třídy, které ji dědí ani z vnějšího kódu.
   4. Dědičnost:
      1. Veřejné atributy jsou standardně používány jako běžné atributy tříd a jsou přístupné zvenku třídy i potomkům.
      2. Chráněné jsou zděděny do potomků a potomci k nim mohou přistupovat.
      3. Privátní nejsou přístupné z vnější ani z potomků třídy.
8. Co jsou to context managers a jak byste je použili?
   1. Context managers jsou objekty, které umožňují definovat blok kódu, který se bude spouštět před a po výkonání určité operace.
   2. Například v kontextu správy souborů může být context manager použit k otevření souboru, provedení operací na souboru a následnému uzavření souboru.s
   3. Využití lze aplikovat i v kontextu navázání spojení na databázi, vykonání operace jako přidání sloupce, řádku databáze, uzavření databáze, atd..
9. Kdy byste implementovali vlastní výjimku?
   1. Lze aplikovat výjimku na to, pokud:
      1. Chceme zajistit jednotnou správu chyb v celé aplikaci.
      2. Chceme, aby se nám posílalo zpětné hlášení o chybě v programu, kde bude více popsaná.
      3. Chceme z analýzy logů vyvolat výjimku, která nám zaznamená podezřelou aktivitu. Kdy lze připojit k výjimce další atributy jako:
         1. Zdrojová ip adresa, typ útoku, časové razítko, popis incidentu.
10. Co jsou to dekorátory? Vymenujte alespoň jeden příklad jejich použití. Je možné aplikovat více dekorátorů najednou?
    1. Jedná se o funkce, které umožní změnit chování jiné funkce bez nutnosti ji přepisovat. Dekorátor vezmu původní funkci, přidá k ní novou funkcionalitu.
    2. Je možné aplikovat více dekorátorů najednou. V tomto případě by se aplikoval postupně od nejbližšího postupně k poslednímu, který obaluje celou funkci.
    3. Příklad může být pro logování.

**CODE SNIPPETS:**

* + - 1. Co by mělo být výstupem následujících ukazek kódu?
         1. První řádek vytiskne prázdný seznam, protože index 10 překračuje délku seznamu.
         2. Druhý řádek vytiskne řetězec „ ict\nd media „ protože se využívá prefix „r“ u řetězce, tudíž text zůstane beze změny.
      2. Co by mělo být výstupem následujících ukazek kódu?  
         - Vysvětlení: Otázka č. 3
         1. [42, 'item']
         2. [42]
         3. [42, 'item']

1. Implementujte funkci, která najde všechny lichá čísla v intervalu , a uloží je jako list.
   1. A picture containing text, screenshot, font, software

      Description automatically generated
2. Implementujte funkci, která generuje nekonečnou řadu lichých čísel.
   1. A picture containing text, screenshot, font, software

      Description automatically generated
3. Napište regulární výraz, který nalezne protokol, IPv4 adresu a port z řetězce níže. Na vstupu se může vyskytnout libovolný protokol, IPv4 adresa a libovolný port. Protokol a port jsou volitelné části a nemusí se vyskytnout. V řetězci níže musí regulární výraz najít skupiny “protocol=udp”, “ipv4=127.0.0.1”, “port=53”: “udp://127.0.0.1:53”
   1. (\w+):\/\/(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}):(\d+)

Obsah obrázku text, snímek obrazovky, Písmo

Popis byl vytvořen automatickyObsah obrázku text, Písmo, snímek obrazovky, číslo

Popis byl vytvořen automaticky

**Testování a kvalita kódu:**

1. Jaký je rozdíl mezi jednotkovým a integračním testováním?
   1. Jednotkové testy se zaměřují na testování izolovaných částí kódu, zatímco integrační testy se zabývají testováním interakcí mezi více moduly.
   2. Taktéž se jednotkové testy často spouštějí v průběhu vývoje, zatímco integrační testy se mohou spouštět až v pozdějších fází vývoje nebo během testování před uvedením do provozu.
2. Co je to mocking a jaké má benefity při testování?
   1. Spočívá v nahrazení reálných objektů vytvořených v rámci systému falešnými objekty, které slouží jako simulace reálných objektů.
   2. Hlavním benefitem je testování, které se neodvíjí od externích komponentů jako jsou například databáze, což zlepšuje spolehlivost testů, protože eliminuje chyby externích komponentů.
   3. Dalším benefitem je rychlost testování.
3. Co to je white box a black box testování?
   1. Black box se zaměřuje na testování softwaru z pohledu uživatele, tudíž tester nemá detailní znalosti o vnitřních mechanismech softwaru.
   2. White box se zaměřuje na testování z pohledu vývojáře. Tester má podrobné informace o vnitřní implementaci a může použít tyto znalosti k identifikaci problémů a chyb.
4. Co to je statická analýza kódu (nápověda: pep8)?
   1. Statická analýza kódu je proces, při kterém se zdrojový kód programu prochází a kontroluje s určitými pravidly a standardy.
   2. PEP8 je doporučení pro styl psaní kódu. Obsahuje pravidla pro formátování kódu, jména proměnných, konvence pro dokumentační řetězce, odsazení, atp…
5. Co znamenají zkratky CI a CD?
   1. CI označuje pojem Continuous Integration, což je proces častého automatického sloučení (merge) kódu z různých vývojových větví do hlavní vývojové větve projektu.
   2. CD označuje pojem Continuous Delivery. Jedná se o průběžné dodávání softwarových změn z vývojového týmu do produkčního prostředí. Obvykle zahrnuje automatizované testování a ověřování správnosti aplikace.

**Databáze:**

1. Jaké jsou rozdíly mezi SQL a NoSQL databázovými systémy? Vyjmenujte několik kandidátů z každé skupiny
   1. SQL
      1. Hlavní rozdíly jsou:
         1. Relační struktura (pevně stanovené schema)
         2. SQL k dotazování dat
         3. Vysoká konzistence dat
      2. Kandidáti:
         1. MySQL
         2. PostgreSQL
         3. Oracle Database
   2. NoSQL
      1. Hlavní rozdíly jsou:
         1. Nerelační struktura dat. (data jsou organizována v dokumentech, grafu, atd..)
         2. Často nemají pevné schema
         3. Často používají jiné dotazovací jazyky než SQL
      2. Kandidáti:
         1. MongoDB
         2. Amazon DynamoDB
         3. Cassandra
2. Co je to databázový index? Jaké typy indexů znáte? (Ideálně v PostgreSQL)
   1. Databázový index umožňuje rychlejší vyhledávání a filtrování dat v databázi.
   2. Indexy vytvářejí seřazenou kopii určitého sloupce nebo skupiny sloupců v tabulce.
   3. Typy indexů:
      1. B-tree index: jedná se o nejčastěji používaný typ indexu v PostgreSQL.
      2. Hash index: je vhodný pro rychlé vyhledávání hodnot pomocí hashovací funkce.
3. Co byste dělali, když byste měli optimalizovat nějaký SQL dotaz?
   1. Použití indexů
   2. Omezení množství načítaných dat za pomocí dodatečných příkazů, např: WHERE, GROUP BY nebo JOIN.
4. Co to je ELT a ETL (nápověda: extrakce, načítání, transformace)? Jaké jsou mezi nimi rozdíly?
   1. Jedná se o způsoby, jak převádět data z jednoho systému do druhého.
   2. ETT:
      1. Extrahování, transformace, načítání
      2. Jedná se o proces, který začíná extrakcí dat ze zdrojů, jako jsou databáze, soubory nebo webové služby. Data jsou pak transformována do požadovaného formátu a nakonec načtena do cílového systému.
   3. ETL:
      1. Extrahování, načítání, transformace
      2. Začíná procesem extrakcí dat. Data jsou následně přenesena do cílového systému bez přímé transformace. Namísto toho se transformace provádí v cílovém systému pomocí nástrojů pro zpracování dat, jako je například SQL.