

За начинаещи C# разработчици:

1. Какво е наследяване в C# и как се използва?

- Отговор: Наследяването е основен принцип на обектно-ориентираното програмиране, който позволява на един клас да наследи свойства и методи от друг клас.

2. Какво представлява LINQ и за какво се използва?

- Отговор: LINQ (Language Integrated Query) е набор от функции за извличане на данни от различни източници в единен, декларативен начин.

3. Каква е разликата между абстрактен клас и интерфейс в C#?

- Отговор: Абстрактният клас може да съдържа реализация на методи, докато интерфейсът само декларира на методи без тяхната реализация.

4. Какво е делегат в C#?

- Отговор: Делегатът е тип, който представлява препратки към методи с определена сигнатура и връщан тип.

5. Как работи управлението на паметта в C#?

- Отговор: Управлението на паметта в C# се извършва автоматично чрез събирач на боклук (garbage collector), който освобождава паметта, заета от обекти, които вече не се използват.

6. Какво е изключение и как се управлява в C#?

- Отговор: Изключението е проблем, възникващ по време на изпълнението на програмата. Управлява се чрез блокове `try`, `catch` и `finally`.

7. Какво представлява Entity Framework?

- Отговор: Entity Framework е ORM (Object-Relational Mapping) инструмент, който позволява работа с бази данни чрез обекти.

8. Какво е MVC патернът и как се използва в C#?

- Отговор: MVC (Model-View-Controller) е дизайн патерн, който разделя приложението на три основни компонента: модел, изглед и контролер.

9. Как се справяте с многонишково програмиране в C#?

- Отговор: Многонишковото програмиране в C# се управлява чрез класове като `Thread` и `Task`, като се внимава за синхронизация и конкурентност.

10. Какво е Liskov Substitution Principle?

- Отговор: Това е принцип в обектно-ориентираното програмиране, който гласи, че обектите на базов клас трябва да могат да бъдат заменени с обекти на производни класове без да се наруши функционалността на програмата.

За начинаещи Java разработчици:

1. Каква е разликата между == и .equals() в Java?

- Отговор: Операторът == сравнява референции, докато .equals() сравнява съдържанието на обектите.

2. Какво е изключение и как се управлява в Java?

- Отговор: Изключението е проблем, възникващ по време на изпълнението на програмата. Управлява се чрез блокове try, catch и finally.

3. Каква е разликата между абстрактен клас и интерфейс в Java?

- Отговор: Абстрактният клас може да съдържа реализация на методи, докато интерфейсът само декларирации на методи без тяхната реализация.

4. Какво е полиморфизъм в Java и как се използва?

- Отговор: Полиморфизмът е способността на обект да приема много форми. В Java, това се постига чрез наследяване и интерфейси.

5. Каква е разликата между ArrayList и LinkedList в Java?

- Отговор: ArrayList е базиран на масив, докато LinkedList е базиран на свързан списък. Това води до разлики в производителността при различни операции.

6. Как работи събирачът на боклук в Java?

- Отговор: Събирачът на боклук автоматично управлява паметта, като освобождава паметта, заета от обекти, които вече не се използват.

7. Какво е Java Virtual Machine (JVM) и защо е важна?

- Отговор: JVM е среда, която позволява изпълнението на Java код на различни платформи без промяна на кода.

8. Какво представлява Spring Framework?

- Отговор: Spring е мощен Java фреймуърк, който предоставя обширна поддръжка за разработка на уеб приложения и микросервиси.

9. Как се справяте с многонишково програмиране в Java?

- Отговор: Многонишковото програмиране в Java се управлява чрез класове като Thread и Runnable, като се внимава за синхронизация и конкурентност.

10. Какво е JPA и Hibernate?

- Отговор: JPA (Java Persistence API) е спецификация за управление на релационни данни в Java, а Hibernate е популярна имплементация на JPA.

-

За начинаещи JavaScript разработчици:

1. **Какво представлява затварянето (closure) в JavaScript и за какво се използва?**

- Отговор: Затварянето е функция, която запазва достъп до променливите от обхвата, в който е била създадена, дори след като е изпълнена.

2. **Каква е разликата между null и undefined в JavaScript?**

- Отговор: `null` е присвоена стойност, показваща липса на стойност, докато `undefined` показва, че променлива не е била инициализирана.

3. **Как работи наследяването в JavaScript?**

- Отговор: Наследяването в JavaScript се осъществява чрез прототипи, където обекти могат да наследяват свойства и методи от други обекти.

4. **Какво е обещание (promise) в JavaScript и как се използва?**

- Отговор: Обещанието е обект, който представлява крайния резултат от асинхронна операция, като предоставя методи за обработка на успешни и неуспешни резултати.

5. **Каква е разликата между let, const и var?**

- Отговор: `let` и `const` са блок-обхватни декларации, докато `var` е функционален обхват. `const` се използва за деклариране на константи.

6. **Как се справяте с асинхронно програмиране в JavaScript?**

- Отговор: Асинхронното програмиране може да се управлява чрез обещания, `async/await`, и обратни извиквания (`callbacks`).

7. **Как работи събитийен цикъл (event loop) в JavaScript?**

- Отговор: Събитийен цикъл управлява изпълнението на асинхронен код, като позволява неблокиращо изпълнение.

8. **Каква е разликата между DOM и BOM?**

- Отговор: DOM (Document Object Model) е API за HTML и XML документи, докато BOM (Browser Object Model) се отнася до обекти, специфични за браузъра.

9. **Как се справяте с крос-браузърна съвместимост?**

- Отговор: Крос-браузърната съвместимост се постига чрез използване на стандартизирани уеб технологии и тестване на различни браузъри.

10. **Какво е AJAX и как се използва в JavaScript?**

- Отговор: AJAX (Asynchronous JavaScript and XML) е техника за създаване на бързи и динамични уеб страници, като се извършват асинхронни заявки към сървъра.

За начинаещи Python разработчици:

- Въпрос:** Какво е list comprehension и дайте пример за неговото използване. **Отговор:** List comprehension е синтактично съкращение за създаване на списъци. Пример: `[x**2 for x in range(10)]` създава списък с квадратите на числата от 0 до 9.
- Въпрос:** Обяснете разликата между mutable и immutable типове данни в Python. **Отговор:** Mutable типове данни (като списъци и речници) могат да бъдат променени, докато immutable типове (като низове и кортежи) не могат.
- Въпрос:** Как работи garbage collection в Python? **Отговор:** Python използва събиране на боклук, базирано на брояч на референции и поколенията на обектите, за автоматично освобождаване на паметта.
- Въпрос:** Какво представлява GIL в Python и какво е неговото влияние? **Отговор:** GIL (Global Interpreter Lock) е механизъм, който ограничава изпълнението на множество нишки в един Python процес. Това може да бъде ограничение при многонишково програмиране.
- Въпрос:** Как се справяте с управлението на зависимости в Python проекти? **Отговор:** Използвам инструменти като pip и virtualenv за управление на зависимости и виртуални среди.
- Въпрос:** Обяснете как работи наследяването в Python. **Отговор:** Python поддържа множествено наследяване и използва специален метод, наречен `super()`, за достъп до родителските класове.
- Въпрос:** Какво е lambda функция в Python и дайте пример. **Отговор:** Lambda функция е анонимна функция, дефинирана с ключовата дума `lambda`. Пример: `lambda x: x * 2` е функция, която удвоява стойността на x.
- Въпрос:** Как се справяте с изключения в Python? **Отговор:** Използвам блокове `try` и `except` за улавяне и обработка на изключения.
- Въпрос:** Какво представлява PEP 8 и защо е важно? **Отговор:** PEP 8 е стандарт за стил на кода в Python, който помага за поддържането на чист и консистентен код.
- Въпрос:** Обяснете разликата между deep и shallow copy. **Отговор:** Shallow copy създава нов обект, но не копира вложените обекти, докато deep copy създава напълно независимо копие на целия обект, включително вложените обекти.

За начинаещи QA инженери:

- Въпрос:** Какви видове тестове познавате и кога се използват? **Отговор:** Познавам различни видове тестове като unit тестове, интеграционни тестове, системни тестове и приемни тестове, които се използват в различни етапи на разработката.
- Въпрос:** Какво представлява TDD (Test-Driven Development)? **Отговор:** TDD е методология, при която първо се пишат тестовете, а след това кодът, който да мине тези тестове.
- Въпрос:** Обяснете разликата между регресионно и санитарно тестване. **Отговор:** Регресионното тестване проверява дали новият код не е счупил съществуващите функционалности, докато санитарното тестване е бърза проверка на основните функционалности след промяна.
- Въпрос:** Как се използва Selenium за автоматизиране на тестове? **Отговор:** Selenium е инструмент за автоматизиране на уеб браузъри, който позволява записване, редактиране и повторение на тестове.
- Въпрос:** Какво е API тестване и как се извършва? **Отговор:** API тестването проверява интерфейсите за програмиране на приложения за тяхната функционалност, надеждност и сигурност.
- Въпрос:** Как се определят приоритетите на тестовете? **Отговор:** Приоритетите се определят въз основа на важността на функционалността, риска от дефект и честотата на използване.
- Въпрос:** Какво представлява натоварване (load) тестване? **Отговор:** Натоварването тестване проверява как системата се справя при високо натоварване или голям брой потребители.
- Въпрос:** Как се документират намерените грешки? **Отговор:** Грешките се документират в система за управление на дефекти с детайлно описание, стъпки за възпроизвеждане и съответната сериозност.
- Въпрос:** Какво е мобилно тестване и какви са неговите особености? **Отговор:** Мобилното тестване включва тестване на приложения за мобилни устройства и се фокусира върху специфични аспекти като съвместимост, производителност и сигурност.
- Въпрос:** Как се извършва тестване на уеб приложения? **Отговор:** Тестването на уеб приложения включва проверка на функционалността, интерфейса, сигурността, съвместимостта с различни браузъри и устройства.

За начинаещи системни администратори:

- Въпрос:** Как поддържате сигурността на сървъра? **Отговор:** Редовно актуализирам софтуера, използвам силни пароли, конфигурирам фаеруол и следя за сигурностни заплахи.
- Въпрос:** Какви са основните задачи на системен администратор? **Отговор:** Основните задачи включват поддръжка на сървъри и мрежова инфраструктура, инсталиране и конфигуриране на софтуер, решаване на технически проблеми.
- Въпрос:** Как се справяте със сървърен проблем под високо натоварване? **Отговор:** Анализирам ресурсите на сървъра, проверявам дневниците за грешки и прилагам подходящи оптимизации.
- Въпрос:** Какви са вашите опити с Linux/Windows сървърни операционни системи? **Отговор:** Имам опит с инсталиране, конфигуриране и поддръжка на [Linux/Windows] сървъри, включително управление на потребители и права.
- Въпрос:** Как се справяте с аварийни ситуации и възстановяване след срив? **Отговор:** Имам план за аварийно възстановяване, редовно правя резервни копия и бързо реагирам при сринове.
- Въпрос:** Как мониторирате състоянието на сървърите и мрежата? **Отговор:** Използвам инструменти като Nagios, Zabbix или подобни за наблюдение на сървърите и мрежата в реално време.
- Въпрос:** Как се справяте с мрежови проблеми? **Отговор:** Използвам диагностични инструменти като ping, traceroute, и netstat за идентифициране и решаване на мрежови проблеми.
- Въпрос:** Как поддържате и актуализирате софтуера на сървъра? **Отговор:** Редовно проверявам за актуализации и ги прилагам, като същевременно следя за съвместимост и стабилност.
- Въпрос:** Как управлявате потребителски акаунти и права? **Отговор:** Създавам, управлявам и ограничавам потребителски акаунти и права според нуждите на организацията.
- Въпрос:** Как се справяте с резервното копиране и възстановяване на данни? **Отговор:** Използвам надеждни решения за резервно копиране и редовно тествам процеса на възстановяване за гарантиране на надеждността.