

Типові питання на технічній співбесіді

Перелік напрямів:

1. *Python Developer*
2. *C#.Net Developer*
3. *FrontEnd Developer*
4. *Javascript Developer*
5. *Java Developer*
6. *Node.Js Developer*
7. *QA Engineer*

Python Developer

Запитання для Junior

Від джуніорів вимагають знати та вміти застосовувати базові структури даних і модулі, найпростіші запити до баз даних, знати найпростіші команди системи контролю версій тощо.

Загальне з Computer Science та Web Development:

1. Що таке інженерія і процес розробки загалом?
2. Які знаєте принципи програмування?
3. Чим відрізняються процедурна та об'єкто-орієнтована парадигми програмування?
4. Які основні принципи ООП (наслідування, інкапсуляція, поліморфізм)?
5. Що таке множинне наслідування?
6. Які є шість етапів розробки продукту в Software Development lifecycle і яка різниця між Agile і Kanban?
7. Які є методи HTTP-запитів та яка між ними різниця?
8. Як виглядають HTTP-request/response?
9. Що таке авторизація і як вона працює?
10. Що таке cookies?
11. Що таке вебвразливість?
12. Які знаєте класичні бази даних?
13. Як читати специфікацію в конкретній мові (наприклад, PEP8 у Python)?
14. Як відбувається взаємодія клієнта і сервера?
15. Які є підходи до проектування API?
16. Як використовувати патерни програмування?

17. Що таке Acceptance Testing і навіщо його використовують?
18. Що таке модульні та інтеграційні тести, API-тести?
19. Як писати unit-тести?
20. Які є best practices у написанні автотестів?
21. Які базові команди системи контролю версій?
22. Як використовувати Git?
23. У чому різниця між хешуванням і шифруванням?

Python:

24. Python — мова, що інтерпретується чи компілюється?
25. Які є змінні та незмінні типи даних?
26. Що таке область видимості змінних?
27. Що таке introspection?
28. Різниця між `is` і `==`?
29. Різниця між `__init__()` і `__new__()`?
30. У чому різниця між потоками та процесами?
31. Які є види імпорту?
32. Що таке клас, ітератор, генератор?
33. Що таке метаклас, змінна циклу?
34. У чому різниця між ітераторами та генераторами?
35. У чому різниця між `staticmethod` та `classmethod`?
36. Як працюють декоратори, контекстні менеджери?
37. Як працюють `dict comprehension`, `list comprehension` і `set comprehension`?
38. Чи можна використовувати кілька декораторів для однієї функції?
39. Чи можна створити декоратор з класу?
40. Які є основні популярні пакети (`requests`, `pytest`, etc)?
41. Що таке `lambda`-функції?
42. Що означає `*args`, `**kwargs` та як вони використовуються?
43. Що таке exceptions, `<try-except>`?
44. Що таке PEP (Python Enhancement Proposal), які з них знаєте (PEP 8, PEP 484)?
45. Напишіть `hello-world` сервіс, використовуючи один із фреймворків.
46. Які є типи даних і яка різниця між `list` і `tuple`, навіщо вони?
47. Як використовувати вбудовані колекції (`list`, `set`, `dictionary`)?
48. У чому полягає складність доступу до елементів `dict`?
49. Як створюється об'єкт в Python, для чого `__new__`, навіщо `__init__`?
50. Що знаєте з модуля `collections`, якими ще `built-in` модулями користувались?
51. Що таке шаблонізатор та як у ньому виконувати базові операції (об'єднувати ділянки шаблону, виводити дату, виводити дані з серверного боку)?
52. Як Python працює з HTTP-сервером?
53. Що відбувається, коли створюється віртуальне середовище?

Бази даних:

54. Які є базові методи роботи з SQL- базою даних у Python?
55. Що таке SQL-транзакція?

- 56. Як зробити вибірку із SQL-бази з простою агрегацією?
- 57. Який вигляд має запит, який виконує JOIN між таблицями й до самих себе?
- 58. Як відправляти запити у SQL-базу даних без ORM?

Алгоритми:

- 59. Що таке алгоритми (наприклад, Big-O notation)?
- 60. Які є базові алгоритми сортування?
- 61. Що таке Bubble Sort і як це працює?
- 62. Що таке лінійна складність сортування?

Junior C#.Net Developer

Загальне

- 1. Назвіть основні принципи ООП.
- 2. Що таке спадкування, інкапсуляція, абстракція, поліморфізм: наведіть приклади (бажано з власного досвіду). Від якого класу неявно успадковуються всі класи в .NET? Чи дозволено множинне успадкування в C#?
- 3. Що таке рекурсія?
- 4. Що таке лямбда-вираз?
- 5. Що таке паралельне програмування (багатопотоковість) і його призначення? Які класи використовуються?
- 6. Що таке JSON?
- 7. Як ви розумієте REST?
- 8. Розкажіть про SPA concept.
- 9. Які GoF-патерни використовували?
- 10. Яка різниця між GET і POST HTTP методами?
- 11. Яку проблему вирішує Docker? Які його плюси та мінуси?
- 12. Чим принципово відрізняються unit-тести від інтеграційних тестів?

Обробка винятків

- 13. Що таке Exception?
- 14. Для чого служать try, catch, finally? В якому випадку може не виконатись блок finally?
- 15. Що таке call stack? Які ключові слова ви знаєте?

Платформа .NET

- 16. Що таке ASP.NET?
- 17. Які існують типи Action filters?
- 18. Що таке Web Service?

- 19.Що таке CLR?
- 20.Що таке збирач сміття (Garbage Collector) на базовому рівні?
- 21.Що таке делегат?
- 22.Чи відрізняється Delegate від Action?
- 23.Що таке LINQ і для чого використовується? Наведіть кілька прикладів застосування LINQ.
- 24.Що таке простір імен (namespace) і навіщо це потрібно?

Типи даних, колекції та структури даних

- 25.Які типи даних ви знаєте?
- 26.Які примітивні типи знаєте?
- 27.Що таке Nullable-тип?
- 28.Що таке тип значення, а що таке тип посилання? Що з цього class, а що struct? У якій ділянці пам'яті вони зберігаються?
- 29.Чим відрізняються value від reference type? String — це reference чи value?
- 30.В чому відмінність між string builder і string?
- 31.Що таке дженерики? Які проблеми вони вирішують?
- 32.Що таке boxing/unboxing?
- 33.Що таке Array, List, HashSet, Dictionary? Наведіть приклади використання цих структур даних. Яка складність операцій з ними (пошук, вставка, видалення)?
- 34.Які знаєте колекції?
- 35.Що робить оператор yield?

Класи, структури та інтерфейси

- 36.Що таке клас?
- 37.Чим відрізняється клас від абстрактного класу?
- 38.Чим відрізняється абстрактний клас від інтерфейсу? Для чого потрібні інтерфейси і які завдання вони виконують?
- 39.Які ви знаєте модифікатори доступу?
- 40.У чому різниця між звичайним класом і статичним?
- 41.У чому різниця перевизначення методу між ключовими словами new та override?
- 42.Яка відмінність між const та read only?
- 43.Різниця між структурою і класом. Наведіть приклади структур.
- 44.Чи може екземпляр структури зберігатися у купі (heap)? Як це зробити?

Асинхронність

- 45.Що таке асинхронність і чим вона відрізняється від багатопотоковості?
- 46.Які є ключові слова для використання асинхронності в коді?
- 47.Що означають ключові слова async/await?

Бази даних

- 48.Різниця між реляційними та нереляційними базами, плюси та мінуси використання обох варіантів.
- 49.Що таке індекси в RDBMS?
- 50.Які типи JOIN існують у SQL?

Тестування

- 51.Для чого потрібні unit-тести?
- 52.Які переваги та недоліки використання unit-тестів?
- 53.З яких трьох логічних блоків складається unit-тест?

Практичні завдання

Junior only

- 1.Опишіть, як би ви реалізували калькулятор. Якщо знадобиться додати підтримку для римської системи числення, що будете робити?
 - 2.Напишіть програму, щоб перевірити, чи є число простим, чи ні.
 - 3.Знайдіть найменший елемент у масиві.
 - 4.Впорядкуйте структури папок у файловій системі.
 - 5.Напишіть програму, що симулює fizz-buzz.
 - 6.Розкажіть, що робить цей код, і запропонуйте, як його поліпшити.
- а)

```
public bool IsArrayEmpty(string[] array)
{
    if (array.Length > 0)
        return false;
    else
        return true;
}
```

б)

```
protected string GetClass(object url)
{
    string result = string.Empty;

    if (SiteMap.CurrentNode != null && SiteMap.CurrentNode.Url == url.ToString())
        result = "class=\"active\"";

    return result;
}
```

Junior Front-end Developer

Загальні питання

1. Чи стежите ви за новинами у світі фронтенду? Як саме?
2. Що відрізняє хороший код від поганого?
3. Які HTTP-методи та статуси знаєте?
4. У чому небезпека виконання коду, що надходить від користувача?
5. Що таке ООП?
6. Яка різниця між класом і абстрактним класом?

Основи HTML/CSS

7. Що таке DOCTYPE і навіщо його використовують?
8. Які основні теги структури HTML-сторінки?
9. Що таке семантичні теги та навіщо вони потрібні? Наведіть приклади таких тегів.
10. Що можете сказати про тег `
`?
11. Які теги з найсвіжіших оновлень ви знаєте?
12. У чому різниця між `<div>` і ``.
13. Що таке data-атрибути, в яких випадках їх використовуєте?
14. Як використовують атрибути `class` та `id`? Яка різниця між класами та ідентифікаторами?
15. Що таке внутрішні та зовнішні гіперпосилання і які вони мають атрибути?
16. Що ви знаєте про SVG? Які є варіанти додавання SVG на сторінки сайту? Чим вони відрізняються?
17. Як створити форму в HTML і забезпечити її валідацію?
18. Які є типи списків у HTML? Що таке `ul` / `ol` / `dl`?
19. Що таке `iframe` і для чого його використовують?
20. Які методи відправки даних форми ви знаєте?
21. Які основні відмінності між блоковими та вбудованими елементами в HTML? Як вони впливають на структуру сторінки?
22. Які CSS-методології ви знаєте та які використовуєте самі? Обґрунтуйте вибір нинішньої методології.
23. Що таке CSS-селектори? Наведіть приклади.
24. Як працює наслідування в CSS?
25. Що таке «модель коробки» в CSS і як її можна змінити?
26. У чому різниця між `margin` і `padding`?
27. Що таке Flexbox? Як реалізувати верстку за допомогою Flexbox чи Grid і який підхід найбільш доречний в якому випадку?
28. Що таке медіазапити та як їх використовувати?

29. Як можна приховати елемент за допомогою CSS? Чи однаково будуть працювати `opacity: 0`; `display: none`; `visibility: hidden`?
30. Як можна зробити лого посередині екрана?
31. Як розмістити хрестик для закриття справа зверху елемента?
32. Як зробити останній елемент списку червоним, незалежно від довжини списку без JS?
33. Які є можливості задати колір?
34. Що таке `z-index`? Чи кожен елемент його має?
35. Які переваги і недоліки використання CSS-препроцесорів?
36. Які методи центрування елемента в CSS?
37. Які методи вертикального центрування в CSS ви знаєте?
38. Для чого існують різні рівні заголовків, якщо розмір можна встановити стилями?
39. Які є варіанти позиціонування елемента на сторінці?
40. Які бувають види `display`? Який використовуєте найчастіше? У чому особливість `inline-block`?
41. У чому відмінність між `position: relative` та `position: absolute`?
42. Які одиниці розмірів є в CSS?
43. Які ризики використання від'ємних значень для відступів?
44. Що таке псевдокласи та псевдоелементи? В чому між ними різниця? Як ви використовуєте псевдокласи та псевдоелементи в CSS?
45. Що таке адаптивний дизайн?
46. Яка різниця між картинками, доданими через HTML-тег ``, і CSS-властивістю `background-image`?
47. Як ви вирішуєте проблеми з кросбраузерною сумісністю в розробці вебсайтів, зокрема в контексті різних версій браузерів?
48. Як із врахуванням HTML і CSS забезпечити доступність сайту?
49. Як реалізувати `responsive design`, використовуючи CSS?
50. Поясніть концепцію CSS-змінних.
51. Які властивості CSS впливають на швидкодію?
52. Що зазвичай найбільше знижує показники швидкодії сайту?
53. Як зробити дві колонки на CSS так, щоб одна розтягувалася, а інша — ні?
54. Які плюси та мінуси CSS-анімацій порівняно з JS-анімаціями?
55. Як можна перебити `important` або досягти схожого ефекту без `important`?
56. Чи валідний цей HTML? Виправте всі помилки. Розкажіть про валідність HTML загалом.

``

`Some Link in Menu`

`Some Link in Menu`

57. `` Що таке `mobile-first` підхід, які його переваги та недоліки?
58. Поясніть, як обчислюється специфічність у CSS-селекторах.
59. Для чого використовують пре-/постпроцесори?
60. Назвіть типи інпутів у формах. Як відправити форму на сервер (без JS)?

Web Performance

- 61. Що ви знаєте про Core Web Vitals?
- 62. Чим відрізняються лабораторні та польові заміри перформансу?
- 63. Які формати картинок знаєте, чим вони відрізняються?
- 64. Ваш сайт довго завантажується на мобільному пристрої. Розкажіть крок за кроком, як будете шукати причину.
- 65. Які інструменти використовуєте для оптимізації швидкості завантаження вебсторінки у роботі з HTML і CSS?
- 66. Як ви оптимізуєте роботу зображень на вебсторінці для покращення продуктивності?
- 67. Які методи оптимізації вебсторінок ви знаєте?

Основи JavaScript

- 68. Перерахуйте типи даних у JS.
- 69. Чим відрізняється масив та об'єкт? У яких випадках що використовуєте?
- 70. Events в JS. Розкажіть про event propagation, bubbling, delegation, preventing тощо.
- 71. У чому різниця між var, let і const. У яких випадках що потрібно використовувати?
- 72. Чи можливо перевизначити const?
- 73. Чи відомі принципи DRY, KISS, YAGNI ?
- 74. У чому відмінності LocalStorage від SessionStorage? Як ви використовуєте LocalStorage та SessionStorage в JavaScript? У яких випадках вважаєте їхнє використання доцільним?
- 75. Що таке замикання (closure) в JavaScript і як воно працює?
- 76. Що таке область видимості змінної?
- 77. Що таке анонімна функція?
- 78. Що таке стрілкова функція і як вона відрізняється від звичайної функції?
- 79. Які методи перебору масивів ви знаєте?
- 80. Що таке JSON і як працювати з ним у JavaScript?
- 81. Як виконується асинхронний код у JavaScript?
- 82. Як ви працюєте з асинхронним кодом у JavaScript без використання async/await? Які інші підходи ви розглядаєте для роботи з асинхронністю?
- 83. Різниця між == і ===.
- 84. Що таке прототип у JavaScript?
- 85. У чому різниця між null та undefined?
- 86. Що таке контекст? Як можна змінювати контекст виконання функції?
- 87. Що таке Promise? У яких випадках ви ним користуєтесь?
- 88. Які особливості об'єктоорієнтованого програмування у JavaScript?
- 89. Як з JS-коду звернутись до HTML-елемента та змінити його текст?
- 90. Що таке деструктуризація?
- 91. Що буде, якщо звернутись до елемента, якого немає за індексом (const arr = ['a', 'b']; console.log(arr[20])?
- 92. У чому різниця між for..of та for...in?

TypeScript

- 93. Які основні відмінності між TypeScript і JavaScript?
- 94. Які переваги та недоліки використання TypeScript?
- 95. Що таке типи даних у TypeScript і як їх використовують?
- 96. Розкажіть про інтерфейси в TypeScript і наведіть приклади їх використання.
- 97. Як працюють Generics у TypeScript і чому вони корисні?
- 98. Як можна використовувати Enums у TypeScript і які їхні переваги?
- 99. Як TypeScript покращує безпеку типів проти JavaScript?
- 100. Які можливості для рефакторингу та підтримки коду надає TypeScript?
- 101. Що таке Decorators у TypeScript і в яких випадках їх використовують?
- 102. Як у TypeScript використовують простори імен і модулі?
- 103. Які стратегії міграції з JavaScript на TypeScript ви знаєте?

API браузера

- 104. Що таке DOM?
- 105. Як JavaScript взаємодіє з DOM?
- 106. Що таке AJAX?
- 107. Як виконати HTTP-запит за допомогою JavaScript?
- 108. Що таке JSONP і як його використовують?
- 109. Які є способи зберігання даних у браузері?
- 110. Що таке обробка подій у JavaScript?
- 111. Що таке event bubbling та event capturing у контексті обробки подій у DOM?
Як використовуєте ці концепції для ефективного обробки подій?
- 112. Що таке делегування подій у DOM, як ви використовуєте цей підхід для оптимізації обробки подій, особливо на великих сторінках?

Фреймворки та бібліотеки

- 113. Навіщо використовують фреймворки, які їхні переваги та недоліки?
- 114. Чим варто керуватися у виборі фреймворку?
- 115. Що таке компонентно-орієнтований підхід?
- 116. У чому проблема керування станом і які є підходи до її вирішення?
- 117. Що таке віртуальний DOM?
- 118. Які методи життєвого циклу компонентів ви знаєте?
- 119. Як оптимізувати продуктивність рішень з використанням фреймворків?
- 120. Що таке рендеринг на стороні сервера? Навіщо він потрібен, які бувають варіанти реалізації?
- 121. Як взаємодіяти з API в межах бібліотек/фреймворків?
- 122. Які особливості управління даними та маршрутизації?
- 123. Які основні відмінності між Angular і React?
- 124. Як ви розрізняєте класові та функціональні компоненти в React? В яких випадках вважаєте за доцільне використовувати той або інший тип компонента?

- 125. Як можна оптимізувати продуктивність у React-застосунках? Які підходи використовуєте для уникнення надмірного рендерингу та для оптимізації часу завантаження компонентів?
- 126. Як реалізуєте управління станом у React-застосунках? Які підходи використовуєте для підтримки читабельності та ефективного управління більш складним/глобальним станом?
- 127. Як вирішуєте питання управління формами в React? Які підходи використовуєте для валідації та обробки форм?
- 128. Як виконати код функції лише при першому рендері компонента?
- 129. Як обробляєте події у React?
- 130. Що таке React-хуки, які вбудовані хуки використовуєте?
- 131. Яким умовам має відповідати функція, щоб бути хуком?
- 132. Як ви робите серверний рендеринг у Next.js? Які переваги цього підходу бачите, у яких випадках вважаєте його найбільш доцільним?
- 133. Що таке компоненти у Vue.js?
- 134. Як працює директива ngIf в Angular?
- 135. Є завдання порахувати коефіцієнт, який буде потрібен у компоненті і який залежить тільки від декількох констант у проекті. Де його найкраще порахувати?

Основи мережі та протоколи

- 136. Що таке HTTP і HTTPS, яка між ними різниця?
- 137. Які основні методи HTTP-запитів (GET, POST, PUT, DELETE) і їхні відмінності?
- 138. Що таке статус-коди HTTP, наведіть приклади найбільш поширених.
- 139. Що таке REST API і які його основні принципи?
- 140. Які засоби безпеки використовують під час роботи з вебзапитами?
- 141. Що таке JSON і XML і чим вони відрізняються у використанні?
- 142. Як працює механізм CORS (Cross-Origin Resource Sharing)?
- 143. Що таке WebSocket і в чому його особливості порівняно з традиційними HTTP-запитами?
- 144. Як функціонує кешування вебресурсів?
- 145. Що таке SSL/TLS і як це використовують для реалізації безпеки?

Робота з бекендом

- 146. Що таке API і як фронтенд взаємодіє з ним?
- 147. Які є формати обміну даними між фронтендом і бекендом?
- 148. Що таке CRUD-операції і як вони реалізуються на практиці?
- 149. Як використовувати вебсокети для реалізації реального часу комунікацій?
- 150. Які засоби аутентифікації та авторизації використовують у сучасних вебзастосунках?
- 151. Які основні принципи безпечного зберігання та передачі користувацьких даних?
- 152. Що таке GraphQL і в чому його переваги та недоліки порівняно з REST?
- 153. Які стратегії кешування даних можуть бути використані у фронтенді?

- 154. Як можна обробляти великі обсяги даних без негативного впливу на продуктивність фронтенду?
- 155. Які інструменти і підходи використовують для моніторингу та відлагодження запитів до сервера?

Алгоритми та структури даних

- 156. Що таке алгоритмічна складність і як її оцінювати?
- 157. Розкажіть про основні структури даних, такі як масиви, списки, стеки та черги.
- 158. Які алгоритми сортування ви знаєте і як вони працюють?
- 159. Як реалізувати пошук у ширину та пошук у глибину на графі й де це потрібно на фронтенді?
- 160. Що таке рекурсія, наведіть приклади її використання.
- 161. Як можна оптимізувати алгоритми за допомогою мемоізації?
- 162. Які алгоритми пошуку ви знаєте і як вони відрізняються один від одного?
- 163. Як працює хеш-таблиця та які її переваги та недоліки?
- 164. Розкажіть про алгоритми роботи з деревами, такі як двійкове дерево пошуку.
- 165. Які алгоритми можна використовувати для пошуку найкоротшого шляху в графі?

Git і системи контролю версій

- 166. Що таке система контролю версій і навіщо її використовують?
- 167. Які основні команди Git ви знаєте і для чого їх використовують?
- 168. Яка різниця між git merge та git rebase?
- 169. Що таке commit у Git і як його правильно формувати?
- 170. Як виправити помилки в попередніх комітах?
- 171. Що таке гілки в Git і як їх ефективно використовувати?
- 172. Як створити нову гілку в Git?
- 173. Які стратегії вибору віток використовують у різних розробницьких процесах (наприклад, Git Flow)?
- 174. Як використовувати теги в Git?
- 175. Що таке конфлікт злиття і як його вирішити?
- 176. Які найкращі практики роботи з віддаленими репозиторіями?
- 177. Як використовувати .gitignore файл?

Тестування

- 178. Числою відповідальністю є баг на продакшні — розробника чи тестувальника?
- 179. Як ви тестуєте свій код? Чи писали код, який не можна було перевірити до деплою?
- 180. Що таке AAA в тестуванні?

Accessibility

181. Що таке вебдоступність?
182. Як тестувати вебдоступність?
183. Назвіть базові техніки створення доступних вебінтерфейсів.
184. В яких випадках треба використовувати посилання, а в яких — кнопки?
185. Який текст має міститися в атрибуті alt?

Junior Javascript Developer

Загальні:

1. Які методи HTTP-запитів ви знаєте?
2. Які версії HTTP-протоколу вам відомі?
3. Які знаєте коди відповіді (стану) HTTP?
4. Що таке Cross-Origin Resource Sharing? Як усунути проблеми з CORS?
5. Що таке cookie?
6. Який максимальний розмір cookie?
7. Що означає директива use strict?
8. Чим JS відрізняється під час роботи на front-end і back-end?
9. Що таке статична і динамічна типізація?
10. Як клієнт взаємодіє із сервером?
11. Що таке REST?
12. Поясніть поняття мутабельність / імутабельність? Які типи є мутабельними й навпаки?
13. Як шукати помилки в коді? Чи використовуєте дебагер?
14. Яких відомих людей зі світу JS знаєте?

JS Core

15. Які існують типи даних у JS?
16. Як перевірити, чи об'єкт є масивом?
17. Як перевірити, чи число є скінченним?
18. Як перевірити, що змінна рівна NaN?
19. Чим відрізняється поведінка isNaN() та Number.isNaN()?
20. Порівняйте ключові слова var, let, const.
21. Що таке область видимості?
22. Що таке деструктуризація?
23. Для чого призначені методи setTimeout і setInterval?
24. Порівняйте підходи роботи з асинхронним кодом: callbacks vs promises vs async/await.
25. Чи можна записувати нові властивості / функції в прототипи стандартних класів

- (Array, Object тощо)? Чому ні? У яких випадках це робити можна? Як убезпечити себе, якщо потрібно розширити прототип?
26. Назвіть методи масивів, які пам'ятає, і скажіть, для чого вони потрібні.
27. Які перебираючі методи масиву знаєте? У чому їхня відмінність?
28. Як працюють оператори присвоєння / порівняння / рядкові / арифметичні / бітові тощо?
29. Опишіть призначення і принципи роботи з колекціями Map і Set.
30. Що означає глибока (deep) та поверхнева (shallow) копія об'єкта? Як зробити кожен з них?

Функції:

31. Яка різниця між декларацією функції (function declaration) та функціональним виразом (function expression)?
32. Що таке анонімна функція?
33. Розкажіть про стрілкові функції (arrow function). В чому полягають відмінності стрілкових функцій від звичайних?
34. Що таке і для чого використовують IIFE (Immediately Invoked Function Expression)?
35. Що таке hoisting, як він працює для змінних і функцій?
36. Що таке замикання (closure) і які сценарії його використання?
37. Як ви розумієте замикання? Що буде виведено в консолі в цьому випадку?

```
var f = function() {  
  console.log(1);  
}  
  
var execute = function(f) {  
  setTimeout(f, 1000);  
}  
  
execute(f); // что выведет в консоль и почему  
  
f = function() {  
  console.log(2);  
}
```

38. Що таке рекурсія?
39. Що означає ключове слово this?
40. Що таке втрата контексту, коли відбувається і як їй запобігти?
41. Методи функцій bind / call / apply — навіщо і в чому різниця?

Front-end

42. Що таке DOM?
43. Порівняйте атрибути підключення скрипту async і defer в HTML-документі.
44. Яка різниця між властивостями HTML-елементів innerHTML і innerText?
45. Опишіть процес спливання (bubbling) подій у DOM.
46. Як зупинити спливання (bubbling) події?
47. Як зупинити дефолтну обробку події?

- 48. Чому дорівнює `this` в обробнику подій (event handler)?
- 49. Що таке `LocalStorage` і `SessionStorage`? Який максимальний розмір `LocalStorage`?
- 50. Як отримати висоту блоку? Його положення щодо меж документа?
- 51. Що таке `webpack`?
- 52. Чим відрізняється `dev`-збірка від `prod`?

Верстка

- 53. Що таке блокова модель `CSS`?
- 54. Які способи центрування блокового контенту по горизонталі та вертикалі знаєте?
- 55. Які підходи у верстці вам відомі (`float`, `flex`, `grid`, etc)?
- 56. Як зробити додаток `responsive`?
- 57. Які є принципи семантичної верстки?
- 58. Навіщо потрібні префікси для деяких `CSS`-властивостей (`-webkit`-, `-moz`- тощо)?
- 59. Як спростити написання кросбраузерних стилів?
- 60. Практичне завдання: прокоментувати та виправити приклад поганого `CSS` або `HTML`.
- 61. Що таке `CSS`-препроцесори? З якими працювали? Що нового вони приносять у стандартний `CSS`?

Angular

- 62. Перерахуйте основні компоненти фреймворку (модуль, роут, директива тощо).
- 63. У чому різниця між компонентом і директивою?
- 64. Розкажіть про життєвий цикл компонента.
- 65. Перерахуйте часто використовувані хуки життєвого циклу компонента та розкажіть, для чого вони потрібні?
- 66. У чому різниця між конструктором і `ngOnInit`-хуком?
- 67. Як захистити роут від несанкціонованого доступу? Які механізми надає для цього фреймворк?
- 68. Що таке `Lazy loading`, як і для чого використовується?
- 69. Яке призначення `RouterOutlet`?
- 70. Як компоненти можуть взаємодіяти один з одним?
- 71. Як створити `two-way binding` властивість для компонента?
- 72. Які типи форм є у фреймворку? У яких випадках і що краще використовувати?
- 73. Які стани є у форми і як це можна застосувати?
- 74. Навіщо потрібні сервіси? Як з ними працювати?
- 75. Що таке `singleton`-сервіси? Яке їхнє призначення? Спосіб створення?
- 76. Які є способи оголошення сервісів?
- 77. Для чого потрібні модулі? Скільки їх має бути в проєкті?
- 78. Навіщо потрібні загальні модулі (`shared`)?
- 79. Які переваги типізації в `TypeScript`?
- 80. Які можливості `TypeScript` можна використовувати для типізації (тут мають на увазі інтерфейси, типи, `enum` тощо)?
- 81. Яка різниця між інтерфейсом і класом?
- 82. У чому різниця між інтерфейсом і абстрактним класом?

- 83. Яка різниця між інтерфейсом і типом?
- 84. Що таке RxJS? Як він використовується у фреймворку? Які компоненти фреймворку тісно пов'язані з ним?
- 85. Чим відрізняються Observable і Promise?
- 86. Для чого потрібні Subjects? Які типи Subjects існують?
- 87. Як зробити кілька послідовних запитів до API за допомогою HTTP-сервісу і RxJS?
- 88. Яка різниця між switchMap, concatMap, mergeMap?
- 89. Як можна конфігурувати Angular-застосунок?
- 90. Навіщо потрібні environment-файли? Коли їх краще не використовувати?
- 91. У чому різниця між «розумним» (smart) і «дурним» (dumb) компонентами? У яких випадках застосовується кожен з них?
- 92. У чому різниця між NgForm, FormGroup і FormControl і як їх застосовують для побудови форм?
- 93. Навіщо потрібен і як працює async pipe?
- 94. Як стежити за розвитком фреймворку? Яких відомих людей, пов'язаних з Angular, знаєте / читаєте?

React

- 95. Чи працювали з класовими компонентами? У чому їхня особливість?
- 96. Які дані краще зберігати в стані компонента, а які передавати через пропси? Наведіть приклад.
- 97. Чи ознайомлені з хуками? У чому їхні переваги? Чи доводилося робити свої і з якою метою?
- 98. Чи ознайомлені з фрагментами та порталами? Навіщо вони потрібні?
- 99. Коли й для чого використовують рефи?
- 100. Які ви знаєте методи життєвого циклу компонента?
- 101. В якому методі життєвого циклу компонента краще робити запити на сервер? Чому?
- 102. В якому методі життєвого циклу компонента краще робити підписку і відписку від лістенера? Чому? Навіщо відписуватися?
- 103. Чи був досвід роботи з контекстом? Коли його варто використовувати?
- 104. У чому особливість PureComponent?
- 105. Чи працював з мемоізованими селекторами (memoized selectors)? Для чого їх використовують і який принцип роботи?
- 106. У чому бачите переваги бібліотеки React?
- 107. Чому бібліотека React швидка? Що таке Virtual DOM і Shadow DOM?
- 108. Навіщо в списках ключі? Чи можна робити ключами індекси елементів масиву? Коли це виправдано?
- 109. В чому основна ідея Redux?
- 110. Робота зі стилями в React.
- 111. React — це бібліотека чи фреймворк? Яка різниця між цими двома поняттями.
- 112. Чи можна використовувати jQuery разом з React? Чому так / ні?
- 113. Що таке codemod?
- 114. Чи доводилося налаштовувати проєкт React з нуля? За допомогою яких інструментів ви це робили?

- 115.Перерахуйте всі бібліотеки, які використовували у зв'язці з React.
116.Що найскладніше доводилося реалізовувати за допомогою React?

Back-end

- 117.Що таке REPL?
118.Що таке streams в Node.js?
119.Що таке middleware?
120.Для чого використовують функцію setImmediate?
121.Навіщо потрібен app.param() в express?
122.Що таке token based authentication?

Бази даних

- 123.Напишіть простий запит для обчислення трьох авторів, у яких найбільше книг.
124.Напишіть запит, який вибирає останні три коментарі для конкретного користувача для двох таблиць: коментарі та користувачі.
125.Спроектуйте просту схему бази даних для бібліотеки.
126.Для чого використовують SQL-оператор HAVING?
127.Навіщо використовують SQL-оператор LEFT JOIN?
128.Чим відрізняється embed- від reference-зв'язку в MongoDB?
129.В одному проєкті програмісти зберігають дані в MongoDB-колекції коментарів, використовуючи такі типи даних (дивіться нижче). Що поганого в цьому рішенні?

```
id: ObjectId
text: string
author_id: string
created_at: Date
```

- 130.У проєкті знадобилося внести зміни в структуру таблиць, додати кілька полів і індекси. Як програмісти будуть робити це на продакшені?

Інструменти

- 131.Щоразу, коли ви робите pull, чомусь трапляється конфлікт в останньому рядку в усіх файлах, які ви редагували. Що відбувається?
132.Що робить команда git fetch?
133.Які git hygiene підходи ви знаєте?
134.Що таке CI/CD? Для чого це потрібно?

Практичні завдання

- 135.Розкажіть, які є способи копіювання простого об'єкта типу obj = {a: 1, b: 2, c: 3}
136.Напишіть deep clone для об'єкта.
137.Назвіть різні способи, як поміняти місцями значення двох змінних.
138.Менеджер попросив у задачі поміняти статуси з «active, inactive» на «active,

removed», але в коді фігурують тільки цифри й незрозуміло, який статус відповідає якій цифрі. Як допомогти майбутнім програмістам не лізти в документацію за кодом? Питання ставлять на конкретному прикладі з кодом.

139. Необхідно зробити мініпроект — список користувачів з формою створення/редагування користувача:

1. Для зберігання користувачів використовуйте Firebase (це безкоштовно).
2. Для стилізації використовуйте Bootstrap.
3. Мінімальний набір полів користувача:

ім'я;
прізвище;
електронна пошта;
телефон (у форматі +380 (XX) XXX-XX-XX);
дата народження;
буде плюсом — додавання аватара та можливість сгор-картинки.

4. Список користувачів повинен мати можливість фільтрації та пагінацію.
5. Проект має містити README-файл з кроками для запуску.

Junior Java Developer

Загальні

1. Які знаєте шаблони проектування? Розкажіть про два шаблони, які використовували в роботі.

Java Core

2. Які є типи даних в Java?
3. Чим відрізняється об'єкт від примітивних типів даних?
4. У чому різниця передачі параметрів за посиланням і за значенням?
5. Що таке JVM, JDK, JRE?
6. Навіщо використовують JVM?
7. Що таке bytecode?
8. Які ознаки JavaBean?
9. Що таке OutOfMemoryError?
10. Що таке стектрейс? Як його отримати?
11. Назвіть всі методи класу object.
12. У чому відмінність між try-with-resources і try-catch-finally при роботі з ресурсами?
13. Що таке конструктори? Які типи знаєте?
14. Що таке побітові операції?
15. Об'єкти яких стандартних класів immutable в Java?

16. Дайте коротку характеристику immutable object. Навіщо вони потрібні?
17. Як створити immutable object?
18. Які переваги immutable object перед звичайними об'єктами?

ООП

19. Що таке ООП? Назвіть принципи з прикладами.
20. У чому переваги ООП перед процедурним програмуванням?
21. У чому полягає головна особливість ООП?
22. Розкажіть, які переваги ми отримуємо із застосуванням ООП?
23. Розкажіть, які недоліки в ООП?
24. Розкажіть про принцип успадкування в ООП? Навіщо він потрібний?
25. Дайте визначення принципу поліморфізм в ООП? Як працює поліморфізм?
26. Що таке статичний і динамічний поліморфізм?
27. Дайте визначення принципу абстракції в ООП.
28. Які елементи мови відповідають за інкапсуляцію?
29. Які елементи мови відповідають за наслідування?
30. Які елементи мови відповідають за поліморфізм?
31. Що таке SOLID? Наведіть приклади.
32. Що таке перевантаження (overloading) методу?
33. Що таке перевизначення (override) методу?
34. Що таке клас, об'єкт, інтерфейс?
35. Що таке клас POJO? Наведіть приклад такого класу.
36. Які елементи може містити клас?
37. Дайте визначення об'єкта?
38. Розкажіть про наслідування в Java. Які особливості використання ключового слова super?
39. Що таке сигнатура методу? Наведіть приклади правильних і неправильних сигнатур.
40. Чи можна в конструкторі застосовувати return?
41. Чи можна в конструкторі виконати виняток (exception)?
42. З яких елементів складається заголовок класу? Напишіть приклад.
43. З яких елементів складається заголовок методу? Напишіть приклад.
44. Створіть в об'єкті-спадкоємці конструктор за замовчуванням, якщо в базовому класі він не визначений (але визначений інший конструктор).
45. Коли застосовується ключове слово this?
46. Що таке ініціалізатор?
47. Для успадкування класу public class Child extends Parent напишіть порядок ініціалізації об'єкта.
48. Які знаєте відносини між класами (об'єктами)?
49. Які асоціативні зв'язки між об'єктами ви знаєте?
50. Що таке модифікатори доступу в Java? Назвіть їх. Навіщо використовуються?
51. Назвіть основну особливість статичних змінних і методів.
52. Які основні обмеження діють на статичні змінні й методи?
53. Що означає ключове слово static? Чи може статичний метод бути перевизначеним або перевантаженим?

- 54. Чи може бути метод статичним і абстрактним одночасно?
- 55. Чи можна використовувати статичні методи всередині звичайних? Навпаки? Чому?
- 56. Що означає ключове слово `final`?
- 57. Що таке `abstract`? Абстрактний клас? Абстрактний метод?
- 58. Що таке `interface`? Чи може бути `final interface`?
- 59. У чому різниця між абстрактним класом та інтерфейсом в Java?
- 60. Де можна ініціалізувати статичні поля?
- 61. Що таке анонімні класи?
- 62. Що таке примітивні класи?
- 63. Що таке клас «обгортка» (`wrapper`)?
- 64. Що таке `Nested class`? Коли використовується?
- 65. Які модифікатори доступу можуть бути у класу?

Бібліотеки та стандарти

- 67. Що таке `Hibernate`? У чому різниця між `JPA` і `Hibernate`?
- 68. Що таке каскадність? Як вона використовується в `Hibernate`?
- 69. Чи може `entity`-клас бути абстрактним класом?
- 70. Що таке `entity manager`? За що відповідає?
- 71. Що таке клас `Assert`? Навіщо і як його використовувати?

String

- 72. Дайте характеристику `String` у Java.
- 73. Які є способи створення об'єкта `String`? Де він створюється?
- 74. Як порівняти дві стрічки в Java і/або відсортувати їх?
- 75. Запропонуйте алгоритм перетворення стрічки на символ. Напишіть відповідний код.
- 76. Як перетворити стрічку в масив байтів і назад? Напишіть відповідний код.
- 77. Що таке стрічковий пул і навіщо він потрібен?
- 78. Які `GOF`-шаблони застосовуються в стрічковому пулі?
- 79. Як розділити стрічку на частини? Напишіть відповідний код.
- 80. Чому масив символів кращий від стрічки для зберігання пароля?
- 81. Яка різниця між `String`, `StringBuffer` та `StringBuilder`?

Enum

- 82. Дайте коротку характеристику `Enum` у Java.
- 83. Чи може `Enum` реалізовувати (`implement`) інтерфейс?
- 84. Чи може `Enum` розширити (`extends`) клас?
- 85. Чи можна створити `Enum` без екземплярів об'єктів?
- 86. Чи можемо ми перевизначити метод `toString ()` для `Enum`?
- 87. Що буде, якщо не будемо перевизначати метод `toString ()` для `Enum`?
- 88. Чи можемо ми вказати конструктор всередині `Enum`?
- 89. У чому різниця між `==` і `equals ()`?
- 90. Що робить метод `ordinal ()` в `Enum`?

- 91. Чи можна використовувати Enum з TreeSet або TreeMap в Java?
- 92. Як пов'язані методи ordinal () і compareTo () в Enum?
- 93. Напишіть приклад Enum.
- 94. Чи можна використовувати Enum у switch case?
- 95. Як отримати всі наявні значення в екземплярі Enum?

Stream API

- 96. Що таке Stream у Java?
- 97. Назвіть основні властивості транзакцій.
- 98. Які є рівні ізоляції транзакцій?
- 99. Яка різниця між Statement і PreparedStatement?

Collections

- 100. Розкажіть про ітератори та їх застосування.
- 101. Яка ієрархія колекцій у Java Collection Framework?
- 102. Яка внутрішня будова ArrayList?
- 103. Яка внутрішня будова LinkedList?
- 104. Яка внутрішня будова HashMap?
- 105. Чим відрізняється ArrayList від LinkedList?
- 106. Чим відрізняється ArrayList від HashSet?
- 107. Навіщо в Java таке різноманіття імплементації динамічного масиву?
- 108. Навіщо в Java таке різноманіття імплементації key-value storage?
- 109. Як відсортувати колекцію елементів?

Class Object. Equals and hashCode

- 110. Дайте коротку характеристику class object у Java.
- 111. Для чого використовують Equals and hashCode в Java?
- 112. Розкажіть про контракт між Equals and hashCode в Java?
- 113. Які умови висуваються щодо перевизначення угоди при перевизначенні Equals?
- 114. Що буде, якщо не перевизначити Equals and hashCode?
- 115. Які значення ми отримаємо, якщо у нас не перерозподілені Equals and hashCode?
- 116. Чому симетричність виконується, тільки якщо x.equals (y) повертає значення true?
- 117. Що таке колізія в hashCode? Як з нею боротися?
- 118. Що буде, якщо елемент, який бере участь в контракті із hashCode, змінює своє значення?
- 119. Напишіть методи Equals and hashCode для класу Student, який складається з полів String name та int age.
- 120. У чому різниця застосування if (obj instanceof Student) та if (getClass () == obj.getClass ())?
- 121. Дайте коротку характеристику методу clone ().
- 122. У чому полягає особливість роботи методу clone () з полями об'єкта типу-посилання?

Exceptions

- 123. Дайте визначення поняттю exception (виняткова ситуація).
- 124. Які знаєте особливості використання оператора try...catch?
- 125. У чому різниця між error і exception?
- 126. Яка різниця між checked і unchecked, exception, throw, throws.
- 127. Якою є ієрархія винятків?
- 128. Що таке checked і unchecked exception?
- 129. Чи потрібно перевіряти checked exception?
- 130. Про що говорить і як використовувати ключове слово throws?
- 131. Які можливі способи обробки винятків ви знаєте?
- 132. Напишіть приклад перехоплення та обробки винятку в блоці try-catch методу.
- 133. Напишіть приклад перехоплення та обробки винятку в секції throws-методу і передачі викликаючому методу.
- 134. Напишіть приклад перехоплення та обробки винятку з використанням власних винятків.
- 135. Які є правила для перевіряння винятків під час наслідування?
- 136. Чи могли б ви написати код, якщо блок finally не буде виконано?
- 137. Напишіть приклад обробки декількох винятків в одному блоці catch
- 138. Який оператор дозволяє примусово викинути виняток? Напишіть приклад.
- 139. Чи може метод main викинути throws-виняток? Якщо так — куди передасть?
- 140. Напишіть приклад try with resources.

Багатопотоковість

- 141. Які засоби для роботи з багатопотоковістю знаєте?
- 142. Що таке процес і потік? Чим відрізняється процес від потоку?
- 143. Розкажіть про синхронізацію між потоками. Для чого використовують методи wait(), notify() - notifyAll(), join()?
- 144. Як зупинити потік?
- 145. Як між потоками обмінюватися даними?
- 146. У чому полягає відмінність класу Thread від інтерфейсу Runnable?
- 147. Є потоки T1, T2 і T3. Як реалізувати їхнє послідовне виконання?

Практичні завдання

- 148. [Matrix Diagonal Sum](#) (задача з Leetcode).
- 149. [Move Zeroes](#) (задача з Leetcode).
- 150. Given List<String> names. Видалить першу літеру з кожного імені та поверніть відсортований список.
- 151. Перевернути масив.
- 152. Перевірити, чи є стрічка паліндромом.
- 153. Написати простий алгоритм сортування (Bubble, Selection або Shuttle). Як його можна покращити?

154. Напишіть алгоритм (послідовність дій) складання літералу типу `int` і літералу типу `byte`. Поясніть, що відбувається з пам'яттю.

Junior Node.Js Developer

Node.js

1. Що таке Node.js?
2. Які основні переваги та недоліки використання Node.js?
3. Для яких завдань Node.js не підходить?
4. Які в Node.js головні компоненти?
5. Яким чином Node.js сервер здатен обробляти одночасно багато паралельних запитів від клієнтів, маючи лише один thread?
6. Чи можливо використовувати кілька потоків (threads)? За допомогою яких модулів це реалізовано?
7. Node.js інтерпретує чи компілює код програми?
8. Як зчитувати великі файли за допомогою Node.js?
9. Що таке libuv і v8? Яке їхнє призначення?
10. Яка різниця між microtasks і macrotasks? Наведіть приклади таких завдань.
11. Що таке стрим (stream)?
12. Які види стримів ви знаєте?
13. Що таке event loop? З яких компонентів складається і як працює?
14. Що таке логування і моніторинг?
15. Чим відрізняється моноліт від мікросервісу?
16. В чому різниця між такими ключовими словами мови, як `string` і `String`?

Express.js

17. Для чого використовують middleware?
18. Як переходити з однієї middleware в іншу?
19. Як пріоритизувати middleware?
20. Як організувати error handler?

JavaScript

21. Що таке асинхронність і асинхронний код?
22. Яка відмінність між `var`, `let` і `const`? Чому варто використовувати `const`, якщо змінна не буде змінюватися далі в коді?
23. Як відкласти виконання функції на конкретний час?
24. Які ви знаєте способи оголошення функції?

- 25.Що таке анонімна функція?
- 26.Наведіть приклади функції, що самовикликається.
- 27.У чому різниця між function expression і function declaration?
- 28.Як з JS масиву чисел отримати інший масив, де залишаться тільки числа понад 10?
Яку функцію масиву для цього використовувати?
- 29.Як видалити елемент масиву та об'єкта?
- 30.Для чого призначений тип void?
- 31.Де і для чого використовують *super()*?
- 32.Для чого потрібен *this* і в яких випадках його використовувати?
- 33.Що таке *NaN* і як його використати?
- 34.Що таке NPM? Які аналоги ви знаєте?
- 35.В чому переваги і недоліки NPM проти Yarn/PNPM?
- 36.Які методи Promise API ви знаєте? Яка різниця між ними?
- 37.Наведіть структуру HTTP request/response.
- 38.Що таке new Set() і new Map()?
- 39.Що таке логічний оператор && та || і чим відрізняються ці оператори від логічного оператора «??».

Бази даних

- 40.Для чого потрібні бази даних у застосунках?
- 41.Що таке ORM і для чого її використовують?
- 42.Що таке міграція даних? Для чого вона потрібна?
- 43.Що таке транзакція?
- 44.Як оновити значення колонки в таблиці?
- 45.За допомогою чого можна відфільтрувати таблицю Users за параметром віку?

WEB

- 46.Що таке Cross-Origin Resource Sharing (CORS)? Де трапляється?
- 47.Як отримати помилку CORS у консолі розробника?
- 48.Назвіть основні HTTP-методи RESTful або CRUD застосунків.
- 49.Що таке DNS?

HTML + CSS

- 50.Які способи центрування елемента по горизонталі і вертикалі ви знаєте?
- 51.Як звернутися до всіх посилань, у яких в атрибуті в кінці посилання href є «.com»?
Як приховати такі посилання?
- 52.Чи можна через HTML + CSS зробити фіксацію кліку? Наведіть приклад.
- 53.Що таке «схлопування»?
- 54.Якою буде відстань між цими двома елементами:

```
<div style="display: block">
```

```
<div style="margin-bottom: 25px">AAA</div>
```

```
<div style="margin-top: 21px">BBB</div>
```

```
</div>
```

Чи зміниться відповідь, якщо змінити display: block на display: flex?

55. Чому розробники рекомендують записувати під'єднання перед закриттям body?

Що буде, якщо прописати JS-код в head?

56. Для чого використовують атрибут defer і async у тегу script?

57. Що таке атрибут tabindex? Де його використовують?

Практичні завдання

58. Наведіть приклад коду для визначення, чи в змінній myNumber число буде кратне 11 (true, якщо так; false, якщо ні)

59. Як скопіювати масив у нову змінну без прив'язки за ключем?

60. Що буде виведено в консоль?

```
const trees = ["xyz", "xxx", "test", "ryan", "apple"];
```

```
delete trees[3];
```

```
console.log(trees.length);
```

```
console.log(trees[3]);
```

61. Вам дано: const source = [1, 2, 3, 5, 1, 5, 9, 1, 2, 8]. Створіть масив, в якому не буде дублікатів, і масив, в якому будуть тільки дані, які дублювалися.

62. Дано:

```
const obj = {
```

```
  name: "Vova",
```

```
  secondName: "Simba",
```

```
  age: 15,
```

```
  parents: {
```

```
    mother: {
```

```
      name: "Natali",
```

```
      secondName: "Simba",
```

```
      age: 38
```

```
    },
```



```

    father: {
      name: "Oleg",
      secondName: "Simba",
      age: 45,
      job: "driver"
    }
  }
};

```

Задача: методом деструктуризації оголосіть змінні `motherName`, `motherSecondName`, `motherAge`, `motherJob`, `fatherInitials` (об'єкт з ім'ям і прізвищем батька, без віку та посади). Якщо немає роботи (`job`), вкажіть значення `'unemployed'`.

63. Чи є відмінність запису створення функцій? Якщо є, то яка?

```
function makeFunc() {}
```

```
let myFunc = function();
```

64. Що таке опціональний ланцюжок в JS (?.)

```

let array;

array?.slice(1,3)

if (array?.length) {

  array.slice(1,3)

}

```

65. Напишіть асинхронний мапінг `«.map()»` об'єкта з використанням `callback`. Зверніть увагу на те, що функція `callback` асинхронна, `asyncObjectMap` має повернути об'єкт, а не `promise`.

```

function asyncObjectMap(obj, callback) {

  // код

}

await asyncObjectMap({ qwe: 123, fds: 32 }, async (x) => x * 2)

// => { qwe: 246, fds: 64 }

```

Junior QA Engineer

Теорія тестування

1. Що таке тестування?
2. Навіщо тестувати ПЗ?
3. Які є етапи тестування?
4. Які типи тестування можете назвати?
5. Які рівні тестування знаєте?
6. Які техніки тест-дизайну знаєте?
7. Що таке техніка аналізу класів еквівалентності?
8. Що таке техніка аналізу граничних значень? У чому цінність цієї техніки?
9. Що таке Regression і Confirmation тестування, яка між ними різниця?
10. Як часто варто проводити регресійне тестування продукту?
11. Які бувають види інтеграційного тестування?
12. Що таке Configuration testing?
13. Що таке Exploratory testing?
14. Які існують UI-стандарти?
15. Що таке Black/Grey/White Box Testing?
16. Що таке Performance Testing?
17. Що таке Smoke та Sanity тестування і яка між ними різниця?
18. Що таке Traceability Matrix?
19. Що таке Sanity Testing?
20. Що таке End-to-End тест?
21. Що таке тестування безпеки?
22. Що таке випробування на основі ризиків?
23. Що таке динамічне тестування?
24. Що таке «парадокс пестициду»?
25. Опишіть основні фази STLC? Дайте визначення Entry та Exit Criteria.
26. Що таке Bug, Error, Failure, Fault?
27. Які є атрибути баг-репорту? Які основні поля для заповнення?
28. Яка різниця між пріоритетом і серйозністю?
29. Наведіть приклади серйозного, але не пріоритетного багу.
30. У чому різниця між валідацією та верифікацією?
31. Навіщо потрібна тестова документація? Які її види?
32. Що таке тест-план? Які елементи він має?
33. Яку обов'язкову інформацію має містити тест-план? Як правильно його використовувати, підтримувати та чи взагалі він потрібний для більшості проєктів?
34. Яка різниця між чеклістом і тест-кейсами?
35. Наведіть приклад хорошого тест-кейсу.

AQA

Програмування

- 36.Що таке ООП? Назвіть його принципи з прикладами?
- 37.Що таке інтерфейс? Що таке абстрактний клас? Чим вони відрізняються?
- 38.Що таке SOLID? Наведіть приклади.
- 39.Що таке DRY, KISS, YAGNI?
- 40.Які патерни GOF вам відомі? Наведіть приклади їхнього використання.
- 41.Що таке PageObject і PageFactory?
- 42.Яка ієрархія Collections?
- 43.Яка різниця між Thread class та Runnable interface?
- 44.Яка різниця між String, StringBuffer та StringBuilder?
- 45.Різниця між final, finally і finalize?

Selenium

- 46.Що таке Selenium і для чого його використовують?
- 47.Що таке драйвер браузера?
- 48.Які види локаторів сторінки існують? Які їхні переваги й недоліки?
- 49.Що таке Selenium Waits? Які є і чим відрізняються?
- 50.Які exceptions може кинути Selenium? Що вони означають і як їх обробляти?
- 51.Для чого використовують JavaScriptExecutor? Наведіть приклади.
- 52.Що таке Selenium Grid?
- 53.Які існують способи click та send keys Selenium?
- 54.Як ви запускаєте паралельне виконання тестів? Що таке ThreadLocal?
- 55.Яка різниця між Action та Actions?
- 56.Як написати метод isElementPresent?
- 57.Як вичитати дані з динамічної вебтаблиці?
- 58.Чи можете ви назвати 10 інтерфейсів у Selenium?
- 59.Назвіть два способи, що дають змогу автоматизувати капчу.
- 60.Згадайте типи навігаційних команд Selenium.
- 61.Як знайти пошкоджені посилання в Selenium WebDriver?
- 62.Яку техніку варто розглянути, використовуючи весь сценарій, якщо «немає ні frame id, ні frame name»?

TestNG/JUnit

- 63.Для чого потрібні TestNG/JUnit?
- 64.Які анотації використовують у TestNG/JUnit?
- 65.Які assertions є в TestNG/JUnit?
- 66.Як виконувати тести паралельно в TestNG/JUnit?

Git

- 67. Для чого використовують системи контролю версій?
- 68. Що таке Git? Який принцип його роботи?
- 69. Що таке commits, branches у Git?
- 70. Для чого потрібні GitHub, GitLab та інші, що базуються на Git, вебхостинги проєктів?

CI

- 71. Що таке CI?
- 72. Як автоматичне тестування інтегрується в CI?
- 73. Як налаштувати Job чи Pipeline на знайомому вам CI-інструменті?
- 74. Які інструменти для генерації репорту після виконання автоматичних тестів ви знаєте?
- 75. Яку інформацію має містити звіт про виконання автоматичних тестів?

Web

- 76. Що таке клієнт-серверна архітектура?
- 77. Що може виступати у ролі клієнта?
- 78. Що таке REST API, SOAP? У чому різниця?
- 79. Які протоколи передачі даних знаєте?
- 80. Які методи взаємодії з API існують? У чому різниця між ними?
- 81. Як можна протестувати API, що там потрібно перевіряти?
- 82. Як розшифровується CRUD?
- 83. Чим відрізняється GET від POST?
- 84. Які відмінності між XML та JSON?
- 85. Які знаєте формати передачі даних?
- 86. Як відбувається шифрування?
- 87. Які бувають види баз даних?
- 88. Охарактеризуйте кожен клас status code (1xx; 2xx; 3xx; 4xx; 5xx).
- 89. Які є HTTP-методи?
- 90. Які знаєте Web elements?
- 91. Які браузерери знаєте? У чому їхня відмінність?
- 92. Для чого необхідні інструменти розробника в браузері (Chrome DevTools) і як вони допомагають у тестуванні.
- 93. Що таке кеш?
- 94. Що таке сесія?
- 95. Навіщо потрібні cookies?
- 96. Що таке фрейм?
- 97. Що таке HTML/CSS/JavaScript?
- 98. Яка структура має вебсторінка?
- 99. Навіщо чистити кеш?
- 100. Які види тестування можна застосувати лише до Web?
- 101. Для чого у вебсторінках використовують JavaScript?

102.Що таке REST?

103.Що таке AJAX?

Mobile

104.Які існують мобільні платформи?

105.Які версії Android та iOS зараз використовуються на ринку (мінімальні та максимальні)?

106.Які версії Android треба тестувати, якщо замовник сказав підтримувати з версії 5.0?

107.Назвіть типи мобільних програм.

108.Який формат файлів збірок програм для Android та iOS?

109.Що таке ADB?

110.Як зняти логи з пристрою AOS/IOS?

111.Що потрібно перевіряти під час використання сканера відбитка/Face ID?

112.Як я можу запускати тести Android без Appium?

113.Поясніть концепцію дизайну Appium.

Практичні завдання

114.Написати чекліст для функціонала кошика в інтернет-магазині.

115.Написати тестові набори даних для поля введення дати, яке відсіює користувачів віком до 18 років.

116.Написати чекліст тестування форми введення даних платіжної картки.

117.Протестувати «предмет» щодо різних видів тестування. (Предмет — ліфт, олівець, калькулятор тощо.)

118.Є Input поле, яке приймає цілі значення від 18 до 99 включно. Потрібно протестувати за допомогою техніки тест-дизайну Boundary Values Analysis і Equivalence Partitioning.

119.Є вебсторінка з полями: e-mail, password та кнопкою submit. Необхідно навести приклади негативних тест-кейсів, якими можна перевірити цю сторінку.

120.Навести приклади тест-кейсів для функціонала, який є на декількох сторінках проекту (наприклад, поле пошуку).

121.Як протестувати процес оплати в інтернет-магазині?

122.Як протестувати зламаний тостер?

123.Поясніть для 7-річної дитини, що таке база даних.

124.Визначте необхідну кількість функціональних тест-кейсів, щоб перевірити Log in форму.

125.Є форма реєстрації в вебзастосунку з полями (first name, last name, username, password, repeat password) та кнопкою Register. Які перевірки треба провести?

126.Поле username має бути обов'язковим, але воно не є обов'язковим. Наведіть приклад баг-репорту, створеного на цю помилку.

127.Як би ви провели smoke-testing для застосунку типу Telegram?

128.Як буде виглядати баг-репорт, якщо, наприклад, не працює електричний чайник?

129.Є таблиця books з полями: name, price, page_count. Потрібно вибрати всі імена книг, у яких price понад 10 одиниць і кількість сторінок від 20 до 100.

130.У вас є функціонал калькулятора, який доступний через веббраузер [за посиланням](#). Він має тільки функцію ділити, так би мовити, MVP-версію. Діапазони для вписування в чисельник і дільник від 0,1 до 99,9. Виведення значення відбувається автоматично, тому що front-end реалізований на React JS. Як ви тестуватимете цей функціонал? Які види тестування застосуєте? Які техніки тест-дизайну використаєте?

Калькулятор с функцией деление

0,1 - 99,9

/

0,1 - 99,9

=

131.Завдання на роботу з SQL.

Витягнути номер телефону та адресу користувача Muzik.

Витягнути дані про користувачів, які мають суму замовлення понад 2000 грн.

Підрахувати кількість замовлень у таблиці та загальну суму зроблених замовлень.

USER				ORDER			
id	name	phone	Delivery address	id	user_id	from_user	sum
0	Kolya	111	Kyiv	3434	0	Kolya	1000
1	Petya	222	Odesa	5454	1	Petya	20000
2	Sasha	333	Lviv	5741	2	Sasha	2500
3	Muzik	444	Chernigiv	5689	3	Muzik	4520

132.Ваша компанія розробляє програмне забезпечення для медичних систем, і ви тестуєте компонент, який керує дефібрилятором серця. Ви помітили, що одне рішення в тестовому модулі складається з 34 незалежних атомарних умов. Який метод тестування білого ящика слід вибрати для цього сценарію і чому?

133.Оздоровча програма для співробітників поєднана з оплатою медичного страхування і має такі правила:

Співробітники, які споживають 17 одиниць або менше алкоголю на тиждень, отримують \$28 знижки на оплату.

Для співробітників, які заповняють «Оцінку ризику для здоров'я» оплата зменшується на \$23.

Співробітники, які беруть участь у щорічному контролі за станом здоров'я в компанії: отримають знижку на \$50 за те, що мають індекс маси тіла (ІМТ) 25,5 або менше, і \$19 знижки при ІМТ нижче 30. Некурці отримують додаткову знижку на \$46. Курці, які приєдналися до курсу відмови від куріння, отримують знижку в \$24. Курці, які не приєдналися до курсу відмови від куріння, оплачують додатково \$75.

Використовуючи техніку класів еквівалентності, скільки тестів потрібно написати, щоб покрити вищезгадані умови на 100%?

134.Яка мінімальна кількість тестів необхідна для покриття таких умов автоцивілки:

Особи до 18 років не страхуються.

Для чоловіків на червоному авто додається +15% до вартості поліса.

Для жінок від 18 до 64 років страхова премія 1000 грн.

Для чоловіків від 18 до 64 років страхова премія 1200 грн.

Для осіб від 64 років страхова премія 1800 грн.

141.Напишіть сценарії автоматичного тестування для сортування за ціною і додавання товару у кошик [на сайті](#). До ваших тестів додайте документацію з налаштуваннями та розмістіть ваше рішення на GitHub.