

Вопросы по дифференцированной зачет по МДК.01.03 Разработке мобильных приложений

1. Расскажите, как в Dart организована точка входа в программу. В чём разница между выполнением Dart-скрипта через CLI и работой Dart-кода внутри мобильного приложения (Flutter) с точки зрения функции `main()`?

2. Что такое null-безопасность и зачем она нужна? Как работает null-безопасность в Dart? Объясните разницу между типами `String`, `String?` и оператором `!` (bang operator).

3. Чем отличаются модификаторы `const`, `final` и `late`? Приведите пример, когда использование `late` может привести к ошибке времени выполнения, а `const` — нет.

4. Как осуществляется приведение типов в Dart? В чём разница между операторами `as`, `is` и `is!`? Когда приведение типов может выбросить исключение?

5. Расскажите про механизм работы циклов в Dart. В чём разница между `for`, `while` и `do-while`? Как правильно прервать или пропустить итерацию (`break`, `continue`)?

6. Что такое функции высшего порядка? Приведите примеры `map`, `where`, `fold` на коллекциях. Чем функция высшего порядка отличается от обычной функции, возвращающей функцию?

7. Что такое функция как объект первого класса? Приведите пример, где функция передаётся как аргумент в другую функцию. Чем это отличается от вызова метода класса?

8. Что такое замыкание (closure)? Как оно захватывает переменные из внешней области видимости? Приведите пример замыкания, которое «запоминает» состояние между вызовами (например, счётчик).

9. Чем отличаются обязательные параметры функции от опциональных позиционных и именованных? В каком порядке они должны объявляться? Что делает ключевое слово `required`?

10. Расскажите про инкапсуляцию в Dart. Как сделать поле или метод приватным? Чем отличается вызов `super()` (конструктор) от `super.метод()`?

11. Что такое геттеры и сеттеры? Когда их использование обязательно, а когда можно обойтись обычным полем? Приведите пример вычисляемого геттера.

12. Расскажите про наследование в Dart. Чем отличается переопределение метода (`@override`)? Всегда ли нужно переопределять метод родительского класса?

13. Что такое абстрактные классы и абстрактные методы? Можно ли создать экземпляр абстрактного класса? Зачем нужны абстрактные классы, если есть интерфейсы?

14. Чем отличается реализация интерфейсов (`implements`) от наследования (`extends`)? Что произойдет, если класс реализует интерфейс, но не реализует все его методы?

15. Что такое миксины (`mixin`)? Как они решают проблему множественного наследования? В чём отличие использования `with` от `implements` и `extends`?

16. Расскажите про обобщения (`generics`) в Dart. Зачем нужны параметры типа `<T>`? Как ограничить тип параметра (`extends`)? Приведите пример generic-класса.

17. Что такое фабричный конструктор (`factory`)? Чем он отличается от обычного конструктора и от константного (`const`)? Приведите пример использования `factory` для возврата объекта из кеша.

18. Расскажите про перечисления (`enum`). Какие виды конструкторов вы знаете? Что такое композиция в Dart?

19. Что такое переопределение операторов? Какие операторы можно переопределить, а какие нельзя? Приведите пример переопределения оператора `+` для класса `Vector`.

20. Что такое расширения классов? Зачем они нужны, если можно просто написать функцию? Приведите пример расширения для типа `String` (например, метод `toCamelCase`).

21. Что такое каскадный оператор `..`? Приведите пример, где он упрощает код. Чем отличается каскад от вызова методов через точку?

22. Расскажите про коллекцию List. В чём разница между List.fixedLength и обычным списком? Как удалить элемент по индексу и по значению?

23. Чем отличаются Set и Map? Когда использовать Set, а когда — List? Как проверить наличие элемента в Set и в List? Для чего используется Map?

24. Что такое Iterable? Чем он отличается от List? Почему Iterable может быть lazy?

25. Что такое кортежи в Dart? Чем record отличается от List и от Map? Как получить доступ к полям кортежа (позиционные и именованные)? Что такое возврат нескольких значений из функции с помощью record?

26. Расскажите про обработку исключений. В чём разница между try-catch и try-catch-finally? Что произойдёт, если исключение не перехвачено?

27. Как создать свой тип исключения? Что означает символ _ в Dart? Что такое динамическая типизация?

28. Что такое асинхронность? Что такое Future и для чего он нужен? Чем отличается синхронный код от асинхронного на примере чтения файла?

29. Расскажите про Event Loop в Dart. Что такое микрозадачи и чем они отличаются от очереди событий?

30. Как работают ключевые слова async и await? Что произойдёт, если вызвать асинхронную функцию без await? Как правильно обработать ошибку внутри async-функции?

31. Что такое изоляты? Чем изолят отличается от потока в других языках? Как организовать обмен данными между изолятами?

32. Расскажите про работу с файлами в Dart. В чём разница между синхронным и асинхронным чтением?

33. Как сохранить данные объекта в JSON-файл и восстановить их обратно? Что такое сериализация и десериализация?