### \*\*Python\*\*

1. \*\*Типы данных, операторы, переменные, константы\*\*

int, float, str, bool, list, tuple, dict. Операторы: `+ - \* / % // \*\*`. Переменные — имена для хранения данных. Константы — переменные, значение которых не меняется (по соглашению — заглавные буквы).

2. \*\*Операторы ветвления, выбора, цикла\*\*

`if`, `elif`, `else`; циклы: `for`, `while`.

3. \*\*Массивы одномерные и многомерные\*\*

Одномерный — список `[1, 2, 3]`, многомерный — `[[1,2], [3,4]]`.

4. \*\*Строки. Методы\*\*

Методы: `.lower()`, `.upper()`, `.strip()`, `.find()`, `.replace()`, `.split()`, `.join()`.

5. \*\*Функции\*\*

Определяются через `def`, принимают аргументы и могут возвращать значение через `return`.

6. \*\*Файловый ввод/вывод\*\*

Открытие файла: `open()`, чтение: `.read()`, `.readlines()`, запись: `.write()`.

7. \*\*Класс как абстрактный тип\*\*

`class` — шаблон объекта. Позволяет описать свойства и методы.

8. \*\*Использование объектов и классов\*\*

Создание: `obj = MyClass()`, доступ к методам: `obj.method()`.

9. \*\*Списки, кортежи, словари\*\*

Список — изменяемый `[1,2]`, кортеж — неизменяемый `(1,2)`, словарь — пары ключ-значение `{‘a’: 1}`.

10. \*\*Компоненты для работы с текстом\*\*

`str` и методы, модули `re`, `string`.

11. \*\*Модули\*\*

Импорт: `import module`, свои: `import my\_module`.

12. \*\*Ввод с клавиатуры\*\*

`input("Введите: ")`

13. \*\*Проверка алфавитно-цифровых символов\*\*

`"abc123".isalnum()`

14. \*\*Список vs Кортеж\*\*

Список — изменяемый, кортеж — нет.

15. \*\*Функция range\*\*

Возвращает последовательность чисел: `range(start, stop, step)`.

16. \*\*Локальные и глобальные переменные\*\*

Локальные — внутри функции, глобальные — во всей программе.

17. \*\*Разница между func и func()\*\*

`func` — ссылка на функцию, `func()` — вызов.

18. \*\*Функция map\*\*

Применяет функцию ко всем элементам: `map(str, [1,2,3])`.

19. \*\*Передача переменных\*\*

Объекты передаются по ссылке (но immutable объекты не изменяются).

20. \*\*Разворот списка\*\*

`list[::-1]`

21. \*\*self\*\*

Ссылка на текущий экземпляр класса.

22. \*\*Объединение списков\*\*

`list1 + list2` или `list1.extend(list2)`

23. \*\*Список vs Массив\*\*

Списки — универсальны, массивы — из модуля `array`, однородные.

24. \*\*Изменяемые и неизменяемые объекты\*\*

Изменяемые: list, dict. Неизменяемые: int, str, tuple.

25. \*\*Округление до 3 знаков\*\*

`round(num, 3)`

26. \*\*Где быстрее поиск\*\*

В словаре (O(1) vs O(n))

27. \*\*Модуль vs Пакет\*\*

Модуль — файл .py, пакет — папка с `\_\_init\_\_.py`.

28. \*\*Инкремент/декремент\*\*

`x += 1`, `x -= 1`

29. \*\*Проверка значения в списке\*\*

`x in list`

30. \*\*append vs extend\*\*

`append` — добавляет один элемент, `extend` — несколько.

31. \*\*pass, continue, break\*\*

`pass` — ничего не делает, `continue` — перейти к следующей итерации, `break` — выйти из цикла.

32. \*\*Типы данных (5)\*\*

int, float, str, list, bool.

33. \*\*Обработка исключений\*\*

`try: ... except: ...`

34. \*\*Tkinter: окно, кнопки\*\*

`Tk()`, `Button()`, `.pack()` или `.grid()`.

35. \*\*Tkinter: Label, Entry\*\*

Метка: `Label()`, поле ввода: `Entry()`.

36. \*\*Работа с SQLite\*\*

Модуль `sqlite3`, команды: `connect()`, `cursor()`, `execute()`.

37. \*\*ООП в Python\*\*

Да, полностью: классы, наследование, полиморфизм.

38. \*\*Ключ в словаре\*\*

Только неизменяемый объект: str, int, tuple.

39. \*\*init\*\*

Метод-конструктор: `def \_\_init\_\_(self):`

40. \*\*Синхронный vs Асинхронный код\*\*

Синхронный — пошаговый, асинхронный — может делать паузы и не блокирует поток.

41. \*\*Генератор vs Итератор\*\*

Генератор создаётся через `yield`, сам итератор.

---

### \*\*Java\*\*

42. \*\*Назначение и среды\*\*

Язык для кроссплатформенных приложений. Среда — IntelliJ IDEA, Eclipse.

43. \*\*Типы и операции\*\*

int, double, boolean, char, String. Операции: `+ - \* / %`.

44. \*\*ООП в Java\*\*

Классы, объекты, наследование, инкапсуляция, полиморфизм.

45. \*\*Работа со строками\*\*

Класс `String`, методы: `.length()`, `.substring()`, `.equals()`, `.concat()`.

46. \*\*Геттеры/сеттеры\*\*

Методы доступа: `getName()`, `setName(String n)`.

---

### \*\*Android Studio\*\*

47. \*\*Activity и жизненный цикл\*\*

Activity — один экран. Жизненный цикл: `onCreate`, `onStart`, `onResume`, `onPause`, `onStop`, `onDestroy`.

48. \*\*AndroidManifest\*\*

XML-файл с настройками приложения: разрешения, компоненты.

49. \*\*Intent\*\*

Объект для запуска активностей и обмена данными.

50. \*\*Диалоговые окна\*\*

AlertDialog, DatePickerDialog, ProgressDialog и др.

51. \*\*Layout\*\*

LinearLayout, RelativeLayout, ConstraintLayout и др.

52. \*\*Исполняемые файлы\*\*

`.apk` — Android Package, результат сборки.

53. \*\*View и ViewGroup\*\*

View — элемент UI (Button, TextView). ViewGroup — контейнер (LinearLayout, RelativeLayout).

54. \*\*Структура проекта\*\*

Папки: `java/`, `res/`, `manifests/`.

55. \*\*SDK Manager и AVD Manager\*\*

SDK — для загрузки инструментов, AVD — виртуальные устройства.

56. \*\*Простое меню\*\*

Создание через XML (`menu.xml`), подключение в `onCreateOptionsMenu()`.

57. \*\*res/values/\*\*

Хранят ресурсы: строки, цвета, стили (`strings.xml`, `colors.xml`).

58. \*\*Среды разработки Android\*\*

Android Studio, Eclipse с ADT, Visual Studio + Xamarin.

59. \*\*Адаптеры и списки\*\*

ListView/RecyclerView + Adapter для отображения данных.

60. \*\*Работа с графикой\*\*

`Canvas`, `Bitmap`, `ImageView`, анимации, `Drawable`.