



## Программирование на языке Python.dor\_БАК(2/2)\_24-172-Б\_241112

- 1 Основной принцип объектно-ориентированного программирования: «...»
- 2 В Python внутри методов класса первый параметр – всегда ...
- 3 В Python механизм, позволяющий создать новый класс на основе существующего, унаследовав все его атрибуты и методы, – это ...
- 4 Объект, являющийся атрибутом другого объекта, – это ...
- 5 Механизм для изменения поведения операторов для пользовательских классов – это ... операторов
- 6 Метод, предоставляющий строковое представление объекта для пользователя, – это метод \_\_...\_\_
- 7 При множественном наследовании Python использует для поиска методов алгоритм:
- 8 Способность объекта использовать методы и атрибуты различных классов – это ...
- 9 Соотнесите методы перегрузки операторов в Python с их функциями:
- 10 Расположите фрагменты кода, определяющие иерархию классов, в том порядке, в котором Python будет искать метод в классе X (порядок разрешения методов (MRO) для класса X):
- 11 Различие между классами Tk и Toplevel в Tkinter заключается в том, что Tk ...
- 12 Модуль для работы с Tkinter – это ...
- 13 Метод для создания главного цикла обработки событий в Tkinter – это ...
- 14 Виджет для выбора элемента из выпадающего списка в модуле ttk – это ...
- 15 Модуль Tkinter, добавляющий улучшенные виджеты и стили, – это модуль ...
- 16 Класс, предназначенный для создания и управления группами виджетов в Tkinter, – это класс ...





- 17 Менеджер геометрии для размещения виджетов в конкретной позиции в родительском виджете:
- 18 Метод для привязки функции к событию нажатия клавиши в Tkinter – это ...
- 19 Упорядочьте шаги для создания простого окна в Tkinter, от начального к завершающему:
- 20 Соотнесите виджеты Tkinter с их функциональным назначением:
- 21 Преимущество Django перед Flask – ...
- 22 Стандартный метод для запуска сервера разработки в Django – ...
- 23 Команда для создания нового приложения в Django – ...
- 24 Основная задача веб-фреймворков в Python – ...
- 25 Тип веб-фреймворков как набора инструментов в Python, который позволяет разработчику создать полноценное веб-приложение от начала до конца, – это ... веб-фреймворки
- 26 в Python метод функции `path()`, который указывается для подключения маршрутов уровня приложения, – это ...
- 27 Класс модели Django, используемый для работы с таблицами базы данных, – это ...
- 28 Компонент Django, отвечающий за взаимодействие с базой данных, – это ...
- 29 Расположите в правильном порядке части URL-адреса, начиная с первой:
- 30 Соотнесите функции Django с их описаниями:
- 31 Уровень изоляции, который используется по умолчанию в SQLite, – ...
- 32 Уровень изоляции, который блокирует базу данных для всех объектов чтения и записи до завершения транзакции, – ...
- 33 SQL-запрос, который используется для создания таблицы в SQLite, – ...
- 34 Ключевой аспект, который отличает SQLite от MySQL, – ...





- 35) SQL-оператор для вставки данных в таблицу ...
- 36) Метод ... в sqlite3 используется для извлечения всех строк из результата запроса
- 37) Метод, который используется для подключения к базе данных SQLite в Python, – это ...
- 38) Атрибут `row_factory`, который используется для доступа к данным по именам столбцов, – это ...
- 39) Упорядочьте этапы работы с транзакциями в SQLite:
- 40) Соотнесите описание функционала с правильным классом или функцией:
- 41) Необходим скрипт на Python, который принимает от пользователя имя и выводит приветственное сообщение. Напишите данный скрипт.
- 42) Необходима программа на Python, которая правильно считывает возраст пользователя и выводит его год рождения. Приведите данную программу.
- 43) Необходима программа на Python, которая правильно считывает строку текста и выводит количество слов в этой строке. Приведите данную программу.
- 44) Необходим скрипт на Python, который верно заменяет символ «o» в строке Python на «\$». Напишите данный скрипт.
- 45) Необходим скрипт на Python, который удаляет повторяющиеся элементы из списка [2, 4, 5, 1, 2, 7, 8, 4, 1]. Напишите данный скрипт.
- 46) Необходима программа на Python, которая принимает от пользователя два числа и проверяет следующие условия (выводит результаты проверок): - Оба числа положительные. - Хотя бы одно из чисел отрицательное. - Оба числа делятся на 5 без остатка. Приведите данную программу.
- 47) Ниже приведен скрипт на Python: 

```
string = "Python упрощает программирование с помощью простого синтаксиса"
for index in range(len(string)):
    if index % 2 == 0:
        print(string[index], end="")
```

 Что является результатом вывода этого скрипта?





- 48) Ниже приведен код на Python:
- ```
month = int(input("Введите номер месяца (от 1 до 12): "))
if 1 <= month <= 12:
    if month in [12, 1, 2]:
        season = "Зима"
    elif month in [3, 4, 5]:
        season = "Весна"
    elif month in [6, 7, 8]:
        season = "Лето"
    else:
        season = "Осень"
    print("Время года:", season)
else:
    print("Ошибка: введите номер месяца от 1 до 12.")
```
- Что выполняет этот код?
- 49) Требуется программа на Python, которая запрашивает у пользователя число и пытается вывести его квадрат. Если пользователь вводит нечисловое значение, программа обрабатывает ошибку и выводит сообщение об ошибке. И в любом случае программа должна вывести сообщение о завершении работы. Приведите такую программу.
- 50) В Python существует функция, которая проверяет, содержится ли элемент в списке. Приведите эту функцию.
- 51) Ниже приведен следующий код в Python:
- ```
def infinite_generator():
    num = 1
    while True:
        yield num
        num += 1
def generator_2(num):
    for i in infinite_generator():
        yield num ** i
gen = generator_2(3)
for _ in range(4):
    result = next(gen)
    print(result)
```
- Какой результат будет выведен при выполнении цикла в последней итерации?
- 52) Ниже приведен декоратор в Python.
- ```
def log_function_call(func):
    def wrapper(*args, **kwargs):
        print(f"Функция {func.__name__} была вызвана с аргументами: {args} и {kwargs}")
        return func(*args, **kwargs)
    return wrapper
```
- Что делает данный декоратор?
- 53) Ниже приведен код, написанный на Python.
- ```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        return f"{self.name} издает звук"
class Dog(Animal):
    def __init__(self, name, breed):
        super().__init__(name)
        self.breed = breed
    def speak(self):
        return f"{self.name} лает"
class Cat(Animal):
    def speak(self):
        return f"{self.name} мяукает"
def make_animal_speak(animal):
    print(animal.speak())
# Создаем объекты
dog = Dog("Барбос", "Овчарка")
cat = Cat("Мурка")
make_animal_speak(dog)
make_animal_speak(cat)
```
- Какой результат будет напечатан после выполнения данного кода?





- 54) Ниже приведен код на Python. `from abc import ABC, abstractmethod class Animal(ABC): def __init__(self, name): self.name = name @abstractmethod def sound(self): pass class Dog(Animal): def sound(self): return "Гав" class Cat(Animal): def sound(self): return "Мяу" # Создание экземпляра класса Animal animal = Animal("Барбос")` Что произойдет при попытке создания экземпляра класса Animal в следующем коде?
- 55) Ниже приведен код на Python. `class Vector: def __init__(self, x, y): self.x = x self.y = y def __add__(self, other): if isinstance(other, Vector): return Vector(self.x + other.x, self.y + other.y) return NotImplemented def __repr__(self): return f"Vector({self.x}, {self.y})" v1 = Vector(1, 2) v2 = Vector(3, 4) result = v1 + v2 print(result)` Что произойдет при перегрузке оператора + для класса Vector в следующем коде?
- 56) Ниже приведен код с использованием Tkinter для создания интерфейса. `import tkinter as tk def say_hello(): print("Привет!") root = tk.Tk() root.title("Пример окна") btn = tk.Button(root, text="Нажми меня", command=say_hello) btn.pack() root.mainloop()` Что произойдет, если выполнить данный код?
- 57) Ниже приведен код с использованием Tkinter. `import tkinter as tk class MyApp: def __init__(self, master): self.master = master master.title("MyApp") self.label = tk.Label(master, text="Hello, Tkinter!") self.label.pack() self.button = tk.Button(master, text="Click Me", command=self.change_text) self.button.pack() def change_text(self): self.label.config(text="Text Changed!") root = tk.Tk() app = MyApp(root) root.mainloop()` Что произойдет при запуске приложения?
- 58) Ниже приведен код с использованием Tkinter. `import tkinter as tk class MyApp: def __init__(self, master): self.master = master master.title("MyApp") self.frame = tk.Frame(master) self.frame.pack() self.canvas = tk.Canvas(self.frame, width=100, height=100, bg='blue') self.canvas.pack() self.button = tk.Button(self.frame, text="Draw", command=self.draw_rectangle) self.button.pack() def draw_rectangle(self): self.canvas.create_rectangle(10, 10, 90, 90, fill='red') root = tk.Tk() app = MyApp(root) root.mainloop()` Что произойдет при запуске приложения?





- 59) Ниже представлен код на Python. `from django.db import models`  
`class Person(models.Model):`    `name =`  
`models.CharField(max_length=100)`    `age = models.IntegerField()`  
`def __str__(self):`    `return f"{self.name}, Age: {self.age}"`    Что  
делает метод `__str__` в модели Django Person в этом коде?
- 60) Ниже представлен код на Python. `from django.http import`  
`HttpResponse` `from django.shortcuts import render` `def`  
`greeting(request):`    `name = request.GET.get('name', 'Гость')`    `return`  
`render(request, 'greeting.html', {'name': name})`    Что произойдет,  
если этот код выполнится при обращении к URL  
`http://example.com/greeting?name=Иван?`
- 61) Ниже представлен код на Python. `from django.http import`  
`HttpResponse` `from .models import Bb` `def index(request):`    `s =`  
`'Объявления\r\n\r\n'`    `for bb in Bb.objects.order_by('-published'):`  
`s += bb.title + '\r\n' + bb.content + '\r\n'`    `return HttpResponse(s,`  
`content_type='text/plain; charset=utf-8')`    Какой результат будет  
возвращен пользователю при запросе по URL, который вызывает  
функцию `index`?
- 62) Ниже представлен код. `import sqlite3` # Подключение к базе  
данных SQLite `conn = sqlite3.connect('example.db')` `cursor =`  
`conn.cursor()` # Выполнение запросов SQLite `cursor.execute('CREATE`  
`TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY, name TEXT)')`  
`cursor.execute('INSERT INTO users (name) VALUES (?)', ('Alice',))`  
`conn.commit()` # Проверка содержимого таблицы  
`cursor.execute('SELECT * FROM users')` `rows = cursor.fetchall()` #  
Закрытие соединения `conn.close()` `print(rows)`    Что выведет этот  
код?





- 63) Ниже представлен код, который использует именованные параметры в запросе к базе данных SQLite.
- ```
import sqlite3
def fetch_tasks_by_status_and_project_named(db_name, status, project_name):
    connection = sqlite3.connect(db_name)
    # Установка row_factory для получения данных в виде словарей
    connection.row_factory = sqlite3.Row
    cursor = connection.cursor()
    # Определение запроса с именованными параметрами и явным указанием колонок
    query = """ SELECT id, priority, description, status, deadline, completed, project FROM tasks WHERE status = :status AND project = :project """
    # Выполнение запроса с использованием именованных параметров
    cursor.execute(query, {"status": status, "project": project_name})
    rows = cursor.fetchall()
    for row in rows:
        print(f"ID: {row['id']}, Priority: {row['priority']}, Description: {row['description']}, Status: {row['status']}, Deadline: {row['deadline']}, Completed: {row['completed']}, Project: {row['project']}")
    connection.close()
if __name__ == "__main__":
    fetch_tasks_by_status_and_project_named("projects_and_tasks.db", "pending", "Learn Python")
```
- Что произойдет, если в запросе используются неверные имена именованных параметров (например, :stat вместо :status)?
- 64) Ниже представлен код, в котором используется управление транзакциями для добавления нового проекта и задач в базу данных.
- ```
import sqlite3
def demo_transactions(db_name):
    connection = sqlite3.connect(db_name)
    cursor = connection.cursor()
    try:
        cursor.execute("INSERT INTO projects (name, description, deadline) VALUES ('Project Y', 'Описание проекта Y', '2024-11-01')")
        cursor.execute("INSERT INTO tasks (priority, description, status, deadline, completed, project) VALUES (1, 'Задача 1 для Project Y', 'pending', '2024-10-15', NULL, 'Project Y'), (2, 'Задача 2 для Project Y', 'in progress', '2024-10-30', NULL, 'Project Y')")
        raise sqlite3.Error("Искусственная ошибка для демонстрации rollback.")
    except sqlite3.Error as e:
        # В случае ошибки откат всех изменений
        connection.rollback()
    finally:
        connection.close()
demo_transactions('projects_and_tasks.db')
```
- Что произойдет, если возникает ошибка при выполнении транзакции?
- 65) Механизм включения объектов одного класса в качестве атрибутов другого класса – это ...
- 66) Декоратор для создания статического метода – ...







- 67) Метод, определенный внутри класса и связанный с самим классом, а не с его экземпляром, – это ...
- 68) Специальные функции, принимающие класс в качестве аргумента и возвращающие его измененную версию, – это ...
- 69) Процесс создания конкретных объектов на основе класса – это ...
- 70) Метод в Python, возвращающий строковое представление объекта для отладки, – это метод `__str__`
- 71) Метод, позволяющий в Python дочернему классу вызвать метод родительского класса, – это метод `super()`
- 72) Конструктор класса в Python автоматически вызывается при создании нового экземпляра и называется `__init__`
- 73) Декоратор `@...` в Python позволяет методам класса работать с самим классом, а не с его экземплярами
- 74) Для того чтобы в Python пользовательский класс поддерживал операцию индексации (например, `obj[0]`), необходимо реализовать метод `__getitem__`
- 75) Соотнесите методы перегрузки операторов в Python с их функциями:
- 76) Расположите фрагменты кода, определяющие иерархию классов, в том порядке, в котором Python будет искать метод в классе X (порядок разрешения методов (MRO) для класса X):
- 77) Основная роль цикла обработки событий в событийно-ориентированном программировании – ...
- 78) При запуске основного цикла обработки событий в Tkinter происходит ...
- 79) Базовый класс для всех виджетов в Tkinter – ...
- 80) Виджет в Tkinter, предназначенный для отображения какой-либо надписи без возможности редактирования пользователем, – это ...
- 81) Класс, предоставляющий методы для рисования графических элементов в Tkinter, – это ...
- 82) Метод, который размещает виджеты в виде блоков в Tkinter, – это ...
- 83) Виджет, используемый для создания кнопок в Tkinter, – это ...







- 84 Библиотека Python, которая позволяет создавать пользовательские интерфейсы с поддержкой мультитач и написана на сочетании Python и Cython, – это ...
- 85 Класс в Tkinter, который предоставляет возможность работы с таблицами и списками, – это ...
- 86 Параметр, используемый для привязки функции к событию нажатия кнопки при создании кнопки в Tkinter, – это ...
- 87 Установите правильный порядок шагов создания простого окна в Tkinter:
- 88 Соотнесите виджеты Tkinter с их функциональным назначением:
- 89 Файл URL маршрутов проекта, определяющий, какие контроллеры (views) будут обрабатывать запросы для каждого URL адреса, – это ...
- 90 Команда для создания новой миграции базы данных в Django – python manage.py: ...
- 91 Основной паттерн архитектуры, используемый в Django: ...
- 92 Способ подключения HTML-шаблонов в Django: ...
- 93 HTML-тег, отвечающий за создание гиперссылок, – это тег
- 94 Безопасный протокол передачи данных, который поддерживает шифрование посредством криптографических протоколов SSL и TLS, – это протокол ...
- 95 Раздел HTML-документа, содержащий основной контент страницы, – это раздел
- 96 Модуль или файл, в котором содержатся настройки приложения в проекте Django: ...
- 97 Файл \_\_init\_\_.py указывает на то, что папка является пакетом Python
- 98 Соотнесите команды Django с их назначением:
- 99 Соотнесите функции Django с их описаниями:
- 100 Расположите в правильной последовательности части URL-адреса, слева направо:





- 101) Функция, которая используется для выполнения SQL-запросов в Python с помощью sqlite3, – ...
- 102) Название функции в sqlite3, которая устанавливает объект Row в качестве типа для результирующего набора запроса, – ...
- 103) При вызове метода rollback в контексте SQLite ...
- 104) Метод, который необходимо вызвать для сохранения изменений в базе данных SQLite после выполнения запросов, – это метод ...
- 105) Команда ... служит для получения списка таблиц в базе данных MySQL.
- 106) Тип данных, используемый для хранения строк в SQLite, – ...
- 107) Для обозначения позиционного параметра в SQL-запросе используется символ ...
- 108) Метод для выполнения запросов в виде списка строк – это метод ...
- 109) Функция для установления соединения с базой данных SQLite – это функция ...
- 110) Упорядочьте этапы работы с транзакциями в SQLite:
- 111) Соотнесите описание функционала с правильным классом или функцией:

