АБСТРАКТНЫЕ КЛАССЫ

ВВЕДЕНИЕ

Задачи урока

01 Познакомиться с абстрактными классами;

02 Понять, когда и для чего используются абстрактные классы

03 Познакомиться с тем, как в больших командах разработки используются абстрактные классы

Абстрактные классы

Абстрактные классы

это классы, которые предоставляют общий интерфейс для других классов, но сами по себе не могут быть созданы в экземплярах. Они являются обобщенной формой классов и предназначены для предоставления базовой функциональности и методов, которые должны быть реализованы в производных классах

В Python абстрактные классы реализуются через модуль abc (Abstract Base Classes). Для создания абстрактного класса необходимо импортировать модуль ABC и использовать декоратор @abstractmethod перед объявлением метода или свойства, которые должны быть реализованы в дочерних классах

Вот пример:

```
class Employee(ABC):

    def __init__(self, age, name, salary) -> None:
        self.age = age
        self.name = name
        self.salary = salary

    def set_salary(self, salary):
        self.salary = salary

    @abstractmethod
    def training(self):
        ...

class Accountant(Employee):

    def __init__(self, age, name, salary) -> None:
        super().__init__(age, name, salary)
        self.audit = False

    def set_audit(self, is_audit):
        self.audit = is_audit

    def conducts_audit(self):
        return self.audit

    def training(self):
        print("Обучение 1С")
        print("Обучение работе в УТ")

class Programmer(Employee):

    def __init__(self, age, name, salary) -> None:
        super().__init__(age, name, salary)

    def training(self):
        print("Обучение Jira")
        print("Обучение работе в Confluense")
        print("Обучение подключению по SSH к серверам")
```

В этом примере Employee является абстрактным классом, так как содержит абстрактный метод training(). Классы Accountant и Programmer являются производными от Employee и обязаны реализовать абстрактный метод training ()

При попытке создать экземпляр абстрактного класса, возникнет ошибка, так как абстрактные классы не могут быть непосредственно инициализированы. Однако их можно использовать для определения общего интерфейса и согласования поведения производных классов

Абстрактные классы

Полезны, когда требуется определить общие методы или свойства для группы связанных классов, но необходимость в их конкретной реализации зависит от контекста использования

Они помогают обеспечить структуру и согласованность в иерархии классов и облегчают разработку и поддержку программного кода

ИТОГИ

- Важно использовать абстрактные классы для описания основных методов, которые будут использоваться в классах наследниках. Если следующий программист решит использовать ваш абстрактный класс, ему будет необходимо реализовать все методы, которые были указаны в вашем базовом классе. Таким образом, мы приводим все к единому стандарту. И если в классе должен быть метод run(), то во всех наследуемых классах он будет присутствовать
- С абстрактными классами становится проще читать код других разработчиков, понимая основную логику и методы который должен реализовывать клас