

Абстракция

№ урока: 4 **Курс:** Python Базовый

Средства обучения: Персональный компьютер/ноутбук стандартной производительности

Обзор, цель и назначение урока

Познакомиться с тем, что такое абстракция и как она реализована в Python. В данном уроке будет рассмотрено само понятие абстракции и после, на практических примерах будет показано как она реализована в Python.

Изучив материал данного занятия, учащийся сможет:

- Понимать, что такое абстракция и зачем она нужна.
- Знать, как она реализована в Python и уметь применить ее на практике.

Содержание урока

1. Что такое абстракция
2. Как она применима на практике
3. Решение задач

Резюме

- Завершающее понятие в ООП это абстракция. Данное понятие не имеет реализацию через ключевые слова или специальные символы. Абстракция является скомбинированным принципом из ранее изученных наследования, инкапсуляции и полиморфизма.
- Абстракция в ООП - это сокрытие сложности реализации программного продукта от конечного пользователя путем создания простых и понятных интерфейсов (речь о так называемых классах).
- Основную идею абстракции можно по началу легко перепутать с инкапсуляцией. Ведь сокрытие данных от пользователя появляется в обоих определениях. В случае инкапсуляции - это простое сокрытие атрибутов и методов, а абстракция - это сокрытие целых классов или групп классов путем построения архитектуры программного продукта (как раз за счет ООП).
- Умение мыслить абстрактно является ключом к пониманию идеи ООП.
- По сути, все приемы ООП являются простыми (вы их быстро поймете), но самый сложный момент - это проектирование архитектуры (как организовать классы? какие им дать атрибуты и методы? как все связать в единое целое?). Данный навык обязательно придет к вам с практикой и опытом.

Закрепление материала

- Что такое абстракция?
- Зачем нужна абстракция?
- Что такое архитектура программного продукта?

- Существует ли специальный синтаксис для абстракции в Python?

Дополнительное задание

Изучить теорию абстракции в ООП. Каким образом абстракция ООП помогает в разработке программных продуктов? Каково ее назначение?

Самостоятельная деятельность учащегося

1. Организуйте архитектуру приложения “База данных” (псевдо). В роли базы данных у вас будет класс Database, который будет хранить данные в виде переменной списка.
2. Класс Database должен иметь методы read_data(criteria), write_data(element).
3. Для элемента данных напишите класс Data. В данном случае мы будем хранить данные о пользователях. Data будет иметь атрибуты: country, name, age, gender, height, weight.
4. В классе Database метод read_data будет принимать на вход аргумент criteria, который является словарем вида {"age": 25}, после чего метод вернет отдельный список всех элементов у которых данное условие истинно.

Подсказка: чтобы получить у объекта класса значение его атрибута как у словаря, используйте следующий синтаксис: your_class_instance.__dict__.get('name').

PS: организуйте правильную инкапсуляцию. Вы должны добавлять элементы в класс Database только через метод write, но никак не напрямую через атрибут elements.

Рекомендуемые ресурсы

- <https://www.faceprep.in/python/abstraction-in-python/#:~:text=Abstraction%20in%20Python%3F-.Abstraction%20in%20Python%20is%20the%20process%20of%20hiding%20the%20real,using%20abstract%20classes%20and%20methods.>
- [Абстракция в ООП](#)
- <https://www.rupython.com/27563-27563.html>
-