

Практическое занятие №16

Тема: Составление программ с использованием ООП.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community.

Вариант 16

Постановка задачи №1:

Создайте класс «Книга», который имеет атрибуты название, автор и количество страниц. Добавьте методы для чтения и записи книги

Текст программы:

```
# Блок 1

class Book:
    def __init__(self, title, author, num_pages):
        self.title = title
        self.author = author
        self.num_pages = num_pages
        self.content = ""

    def read(self):
        return self.content

    def write(self, text):
        self.content += text

# Пример использования
book1 = Book("Программирование по Python", "Guido van Rossum", 400)
book1.write("Chapter 1: Ознакомление с Python\n")
book1.write("Chapter 2: Переменные и типы данных\n")

print("Название книги:", book1.title)
print("Автор книги:", book1.author)
print("Количество страниц:", book1.num_pages, end="\n\n")
print("Содержание:")
print(book1.read())

print("\n")
```

Протокол работы программы:

Название книги: Программирование по Python

Автор книги: Guido van Rossum

Количество страниц: 400

Содержание:

Chapter 1: Ознакомление с Python

Chapter 2: Переменные и типы данных

Постановка задачи №2:

Создайте класс "Фрукт", который содержит информацию о наименовании и весе фрукта. Создайте классы "Яблоко" и "Апельсин", которые наследуются от класса

"Фрукт" и содержат информацию о цвете.

Текст программы:

```
# Блок 2

class Fruit:
    def __init__(self, name, weight):
        self.name = name
        self.weight = weight

class Apple(Fruit):
    def __init__(self, name, weight, color):
        super().__init__(name, weight)
        self.color = color

class Orange(Fruit):
    def __init__(self, name, weight, color):
        super().__init__(name, weight)
        self.color = color

# Пример использования
apple = Apple("Яблоко", 150, "Красный")
orange = Orange("Апельсин", 200, "Оранжевый")

print("Название яблока:", apple.name)
print(f"Вес яблока: {apple.weight} грамм")
print("Цвет яблока:", apple.color)

print("Название апельсина:", orange.name)
print(f"Вес апельсина: {orange.weight} грамм")
print("Цвет апельсина:", orange.color)

print("\n")
```

Протокол работы программы:

Название яблока: Яблоко

Вес яблока: 150 грамм

Цвет яблока: Красный

Название апельсина: Апельсин

Вес апельсина: 200 грамм

Цвет апельсина: Оранжевый

Постановка задачи №3:

Для задачи из блока 1 создать две функции, `save_def` и `load_def`, которые позволяют сохранять информацию из экземпляров класса (3 шт.) в файл и загружать ее обратно. Использовать модуль `pickle` для сериализации и десериализации объектов Python в бинарном формате.

Текст программы:

```
# Блок 3

def save_def(obj, filename):
    with open(filename, 'wb') as file:
        pickle.dump(obj, file)

def load_def(filename):
    with open(filename, 'rb') as file:
        return pickle.load(file)

# Пример использования
book2 = Book("Машинное обучение", "Andrew Ng", 300)
save_def(book2, "book2.pkl")

loaded_book = load_def("book2.pkl")

print("Название загруженной книги:", loaded_book.title)
print("Автор загруженной книги:", loaded_book.author)
print("Количество страниц загруженной книги:", loaded_book.num_pages)
```

Протокол работы программы:

Название загруженной книги: Машинное обучение

Автор загруженной книги: Andrew Ng

Количество страниц загруженной книги: 300

Вывод: В процессе выполнения практического занятия закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ, работы с ООП в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.
Готовые программные коды выложены на **GitHub**