

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.

Задача 1.

Постановка задачи.

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

Текст программы:

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

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

    def read_book_info(self):
        return f"Название: {self.title} \nАвтор: {self.author} \nКоличество страниц: {self.num_pages}"

my_book = Book('Мастер и Маргарита', 'Михаил Булгаков', 384) print(my_book.read_book_info())
```

Протокол работы программы:

```
C:\Users\artur\PycharmProject
s\IS-
25_Galstyan_Artur\venv\Script
s\python.exe
"C:\Users\artur\OneDrive\Рабо
чий стол\manakova
2semestr\pz 16\pz 16.1.py"
Название: Мастер и
Маргарита
Автор: Михаил Булгаков
Количество страниц: 384
```

```
Process finished with exit code
0
```

Задача 2.

Постановка задачи.

Создание базового класса "Фигура" и его наследование для создания классов "Квадрат", "Прямоугольник" и "Круг". Класс "Фигура" будет иметь общие методы, такие как вычисление площади и периметра, а классы-наследники будут иметь специфичные методы и свойства.

Текст программы:

```
# Создание базового класса "Фигура" и его наследование для создания классов "Квадрат",  
"Прямоугольник" и "Круг".  
# Класс "Фигура" будет иметь общие методы, такие как вычисление площади и периметра, # а  
классы-наследники будут иметь специфичные методы и свойства.
```

```
class Figure:  
    def __init__(self):  
        pass  
  
    def area(self):  
        raise NotImplementedError  
  
    def perimeter(self):  
        raise NotImplementedError
```

```

class Square(Figure): def
    init_(self, side):
        self.side = side

Текст программы: def area(self):
    return self.side ** 2

def perimeter(self):
    return 4 * self.side

class Rectangle(Figure):
    def _init_(self, width, height):
        self.width = width self.height
        = height

    def area(self):
        return self.width * self.height

    def perimeter(self):
        return 2 * (self.width + self.height)

class Circle(Figure):
    def _init_(self, radius): self.radius =
        radius

    def area(self):
        return 3.14 * (self.radius ** 2)

    def perimeter(self):
        return 2 * 3.14 * self.radius

square = Square(6)
print(f"Площадь квадрата: {square.area()}") print(f"Периметр
квадрата: {square.perimeter()} \n")

rectangle = Rectangle(8, 6)
print(f"Площадь треугольника: {rectangle.area()}")
print(f"Периметр треугольника: {rectangle.perimeter()} \n")

circle = Circle(3)
print(f"Площадь круга: {circle.area()}") print(f"Периметр
круга: {circle.perimeter()}")

```

Протокол работы программы:

C:\Users\artur\PycharmProjects\IS-25_Galstyan_Artur\venv\Scripts\python.exe "C:\Users\artur\OneDrive\Рабочий стол\manakova 2semestr\pz 16\pz 16.2.py"

Площадь квадрата: 36

Периметр квадрата: 24

Площадь треугольника: 48

Периметр треугольника: 28

Площадь круга: 28.26

Периметр круга: 18.84

Process finished with exit code 0

Задача 3.

Постановка задачи.

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:

```
# 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.
import pickle

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

    def read_book_info(self):
        return f"Название: {self.title} \nАвтор: {self.author} \nКоличество страниц: {self.num_pages}"

def save_def(books):
    with open('books_data.bin', 'wb') as file:
        pickle.dump(books, file)

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

book1 = Book('Мастер и Маргарита', 'Михаил Булгаков', 384)
book2 = Book('Евгений Онегин', 'Александр Пушкин', 224)
book3 = Book('Муму', 'Иван Тургенев', 220)

books_list = [book1, book2, book3]
save_def(books_list)
loaded_books = load_def()

for book in loaded_books:
```

Протокол работы программы:

C:\Users\artur\PycharmProjects\IS-25_Galstyan_Artur\venv\Scripts\python.exe "C:\Users\artur\OneDrive\Рабочий стол\manakova 2semestr\pz 16\pz 16.3.py"

Название: Мастер и Маргарита

Автор: Михаил Булгаков

Количество страниц: 384

Название: Евгений Онегин

Автор: Александр Пушкин

Количество страниц: 224

Название: Муму

Автор: Иван Тургенев

Количество страниц: 220

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с ООП в IDE PyCharm Community.