

Студент группы ИС-25 Несповитый А.Е.

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.

Постановка задачи

Блок заданий 1

Создайте класс "Товар" с атрибутами "название", "цена" и "количество". Напишите метод, который выводит информацию о товаре в формате "Название: название, Цена: цена, Количество: кол-во"

Блок заданий 2

Создайте базовый класс "Фигура" со свойствами "ширина" и "высота". От этого класса унаследуйте классы "Прямоугольник" и "Квадрат". Для класса "Квадрат" переопределите методы, связанные с вычислением площади и периметра.

Блок заданий 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:

```
import pickle

# Блок заданий 1
class Product:
    def __init__(self, name, price, quantity):
        self.name = name
        self.price = price
        self.quantity = quantity

    def info(self):
        print(f'Название: {self.name}, Цена: {self.price},  
Количество: {self.quantity}')

# Блок заданий 2
class Figure:
    def __init__(self, width, height):
        self.width = width
        self.height = height

class Rectangle(Figure):
    def area(self):
        return self.width * self.height
```

```

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

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

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

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

# Блок заданий 3
def save_def(products):
    with open('products.pickle', 'wb') as f:
        pickle.dump(products, f)

def load_def():
    with open('products.pickle', 'rb') as f:
        products = pickle.load(f)
    return products

# Создаем экземпляры класса Product
product1 = Product('Товар1', 100, 5)
product2 = Product('Товар2', 200, 3)
product3 = Product('Товар3', 150, 7)

# Выводим информацию о товарах
product1.info()
product2.info()
product3.info()

# Сохраняем информацию о товарах
save_def([product1, product2, product3])

# Загружаем информацию из файла и выводим
loaded_products = load_def()
for product in loaded_products:
    product.info()

# Создаем экземпляры классов Rectangle и Square
rectangle = Rectangle(10, 5)

```

```
square = Square(5)

# Вычисляем площадь и периметр прямоугольника
print(f'Площадь прямоугольника: {rectangle.area()}')
print(f'Периметр прямоугольника: {rectangle.perimeter()}')

# Вычисляем площадь и периметр квадрата
print(f'Площадь квадрата: {square.area()}')
print(f'Периметр квадрата: {square.perimeter()}')
```

Протокол программы:

Название: Товар1, Цена: 100, Количество: 5

Название: Товар2, Цена: 200, Количество: 3

Название: Товар3, Цена: 150, Количество: 7

Название: Товар1, Цена: 100, Количество: 5

Название: Товар2, Цена: 200, Количество: 3

Название: Товар3, Цена: 150, Количество: 7

Площадь прямоугольника: 50

Периметр прямоугольника: 30

Площадь квадрата: 25

Периметр квадрата: 20

Process finished with exit code 0

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