

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 №1:

```
Создайте класс «Банк», который имеет атрибуты суммы денег и процентной ставки.  
# Добавьте методы для вычисления процентных начислений и снятия денег  
# 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.
```

Тип алгоритма: циклический

Текст программы:

```
import pickle  
  
class Bank:  
    def __init__(self, initial_balance, interest_rate):  
        self.balance = initial_balance  
        self.interest_rate = interest_rate  
  
    def calculate_interest(self):  
        interest = self.balance * (self.interest_rate / 100)  
        return interest  
  
    def withdraw(self, amount):  
        if amount <= self.balance:  
            self.balance -= amount  
            return amount  
        else:  
            return "Insufficient funds"  
  
def save_data(data, filename):  
    with open(filename, 'wb') as file:  
        pickle.dump(data, file)  
  
def load_data(filename):  
    with open(filename, 'rb') as file:  
        data = pickle.load(file)  
        return data  
  
bank1 = Bank(1000, 5)  
bank2 = Bank(2000, 3)  
bank3 = Bank(1500, 4)
```

```

save_data([bank1, bank2, bank3], "banks_data.txt")

loaded_data = load_data("banks_data.txt")

for bank in loaded_data:
    print(f"Initial Balance: {bank.balance}, Interest Rate: {bank.interest_rate}")
    print(f"Interest Earned: {bank.calculate_interest()}")
    print(f"Withdrawn amount: {bank.withdraw(500)}")

```

Протокол программы:

Initial amount in the account: 1000

Interest for the account: 20.0

Amount in the account after withdrawal: 900

Initial amount in the account: 5000

Interest for the account: 150.0

Amount in the account after withdrawal: 4900

Initial amount in the account: 200

Interest for the account: 3.0

Amount in the account after withdrawal: 100

Process finished with exit code 0

Постановка задачи №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.

Тип алгоритма: циклический

Текст программы:

```

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_length):
        super().__init__(side_length, side_length)

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

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

rectangle = Rectangle(5, 10)
print(f"Rectangle Area: {rectangle.area()}")
print(f"Rectangle Perimeter: {rectangle.perimeter()}")

square = Square(5)
print(f"Square Area: {square.area()}")
print(f"Square Perimeter: {square.perimeter()}")
```

Протокол программы:

Rectangle Area: 50

Rectangle Perimeter: 30

Square Area: 25

Square Perimeter: 20

Process finished with exit code 0

Вывод: я закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления программ с ООП в IDE PyCharm Community.

