Студент группы ИС-23 Халанский Я.Р.

**Практическое занятие №16**

**Тема:** составление программ с использованием ООП.

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление 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 print(f"Withdrawn amount: {bank.withdraw(500)}")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 составление программ с ООП в IDE PyCharm Community.