

Практическая работа № 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.

Задание 1

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

Текст программы:

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

class Bank:
    def __init__(self, amount, rate):
        self.amount = amount
        self.rate = rate

    def rate_calculation(self):
        interest_rate = self.amount * (self.rate / 100)
        print(f'Процентные начисления {interest_rate: .2f} рублей процентов.')

    def replenishment_money(self, amount):
        self.amount += amount
        print(f'Вам начислено {amount} рублей. На вашем счёте {self.amount}
рублей.')

    def withdraw_money(self, amount):
        if (amount <= self.amount):
            self.amount -= amount
            print(f'Вы сняли {amount} рублей. Остаток на счёте {self.amount}
рублей.')
        else:
            print('Недостаточно средств на счёте.')

bank1 = Bank(50000, 10)
print(f'На вашем счете {bank1.amount} рублей. Процентная ставка
{bank1.rate} %.\n')
```

```
bank1.withdraw_money(4000)
bank1.rate_calculation()
bank1.replenishment_money(17000)
bank1.rate_calculation()
```

На вашем счете 50000 рублей. Процентная ставка 10 %.

Вы сняли 4000 рублей. Остаток на счёте 46000 рублей.

Процентные начисления 4600.00 рублей процентов.

Вам начислено 17000 рублей. На вашем счёте 63000 рублей.

Процентные начисления 6300.00 рублей процентов.

Process finished with exit code 0

Задание 2

Постановка задачи: 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.

Текст программы:

```
# 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.

class Figure:
    def __init__(self, weight, height):
        self.weight = weight
        self.height = height

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

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

class Rectangle(Figure):
    pass
```

```

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

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

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

rect = Rectangle(5, 10)
print(f'Площадь прямоугольника - {rect.area()}\nПериметр прямоугольника - {rect.perimeter()}\n')

square = Square(5)
print(f'Площадь квадрата - {square.area()}\nПериметр квадрата - {square.perimeter()}')

```

Площадь прямоугольника - 50

Периметр прямоугольника - 30

Площадь квадрата - 25

Периметр квадрата - 20

Process finished with exit code 0

Задание 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 Bank:
    def __init__(self, amount, rate):
        self.amount = amount
        self.rate = rate

    def rate_calculation(self):
        interest_rate = self.amount * (self.rate / 100)
        return f'Процентные начисления {interest_rate: .2f} рублей процентов.'

    def replenishment_money(self, amount):
        self.amount += amount
        return f'Вам начислено {amount} рублей. На вашем счёте {self.amount} рублей.'

    def withdraw_money(self, amount):
        if (amount <= self.amount):
            self.amount -= amount
            return f'Вы сняли {amount} рублей. Остаток на счёте {self.amount} рублей.'
        else:
            return 'Недостаточно средств на счёте.'

def save_def(banks, filename):
    with open(filename, 'wb') as f:
        pickle.dump(banks, f)

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

Bank1 = Bank(2500, 7)
Bank2 = Bank(50000, 10)
Bank3 = Bank(120000, 12)

banks = [Bank1, Bank2, Bank3]
save_def(banks, 'banks.pkl')

loaded_banks = load_def('banks.pkl')

```

```
for bank in loaded_banks:
    print(f'На вашем счете {bank.amount} рублей. Процентная ставка {bank.rate} %.')
    print(bank.withdraw_money(1000))
    print(bank.rate_calculation())
    print(bank.replenishment_money(17000))
    print(bank.rate_calculation())
    print()
```

На вашем счете 2500 рублей. Процентная ставка 7 %.

Вы сняли 1000 рублей. Остаток на счёте 1500 рублей.

Процентные начисления 105.00 рублей процентов.

Вам начислено 17000 рублей. На вашем счёте 18500 рублей.

Процентные начисления 1295.00 рублей процентов.

На вашем счете 50000 рублей. Процентная ставка 10 %.

Вы сняли 1000 рублей. Остаток на счёте 49000 рублей.

Процентные начисления 4900.00 рублей процентов.

Вам начислено 17000 рублей. На вашем счёте 66000 рублей.

Процентные начисления 6600.00 рублей процентов.

На вашем счете 120000 рублей. Процентная ставка 12 %.

Вы сняли 1000 рублей. Остаток на счёте 119000 рублей.

Процентные начисления 14280.00 рублей процентов.

Вам начислено 17000 рублей. На вашем счёте 136000 рублей.

Процентные начисления 16320.00 рублей процентов.

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием ООП IDE PyCharm Community.

Были использованы языковые конструкции `def`, `if`, `return`. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на [GitHub](#).