

Практическая №17
Кобелев Евгений ИС-24

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.

Подзадача №1

Постановка задачи: написать класс Calculator у которого есть статические методы суммы, вычитания, умножения и деления.

Текст программы:

```
class Calculator:
    @staticmethod
    def sum(first, second):
        return first + second

    @staticmethod
    def subtract(first, second):
        return first - second

    @staticmethod
    def multiply(first, second):
        return first * second

    @staticmethod
    def division(first, second):
        if second != 0:
            return first / second
        else:
            return ValueError('Нельзя делить на ноль!')

print(Calculator.sum(2, 2))
print(Calculator.subtract(2, 2))
print(Calculator.multiply(2, 2))
print(Calculator.division(2, 2))
print(Calculator.division(2, 0))
```

Протокол работы программы:

4
0
4
1.0
Нельзя делить на ноль!

Process finished with exit code 0

Подзадача №2

Постановка задачи: написать общий класс Worker от которого будут наследоваться подклассы Manager и Engineer. У Worker есть общие для всех методы work и getSalary. У Manager есть

метод manage, а у Engineer есть метод design.

Код программы:

```
class Worker:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def work(self):
        print(f"{self.name} работает...")

    def getSalary(self, money):
        print(f"{self.name} получает зарплату в размере {money} рублей!")

class Manager(Worker):
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def manage(self):
        print(f"{self.name} управляет командой!")

class Engineer(Worker):
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def design(self):
        print(f"{self.name} проектирует системы!")

worker = Worker('Anton', 17)
manager = Manager('Eugene', 18)
engineer = Engineer('Max', 17)

worker.work()
worker.getSalary(40000)

manager.work()
manager.getSalary(40000)
manager.manage()

engineer.work()
engineer.getSalary(40000)
engineer.design()
```

Протокол работы программы:

Anton работает...
Anton получает зарплату в размере 40000 рублей!
Eugene работает...
Eugene получает зарплату в размере 40000 рублей!
Eugene управляет командой!
Max работает...
Max получает зарплату в размере 40000 рублей!
Max проектирует системы!

Process finished with exit code 0

Вывод:

В процессе выполнения практического занятия выработал навыки составления программ с использованием ООП и наследования в IDE PyCharm Community. Были использованы языковые конструкции class, self и декоратор @staticmethod. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на Github.