

Практическое занятие №16

Тема: составление программ с использованием ООП в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с ООП в IDE PyCharm Community.

Постановка задачи: разработать программу, обрабатывающая код с ООП

Текст программы:

1)

```
# Создайте класс "Калькулятор" с методами "сложение", "вычитание", "умножение" и "деление".  
Каждый метод должен  
# принимать два аргумента и возвращать результат операции.  
  
class Calculator:  
    def add(self, a, b):  
        return a + b  
  
    def subtract(self, a, b):  
        return a - b  
  
    def multiply(self, a, b):  
        return a * b  
  
    def division(self, a, b):  
        if b == 0:  
            raise ValueError("Деление на ноль невозможно")  
        return a / b  
  
# Пример использования класса с вводом значений с клавиатуры  
calc = Calculator()  
  
# Ввод значений с клавиатуры  
a = float(input("Введите первое число: "))  
b = float(input("Введите второе число: "))  
  
# Примеры операций  
print(f"Сложение: {a} + {b} = {calc.add(a, b)}")  
print(f"Вычитание: {a} - {b} = {calc.subtract(a, b)}")  
print(f"Умножение: {a} * {b} = {calc.multiply(a, b)}")  
try:  
    print(f"Деление: {a} / {b} = {calc.division(a, b)}")  
except ValueError as e:  
    print(e)
```

Протокол работы программы:

```
/home/student/Документы/PycharmProjects/IS-22/Proj_1sem_Eliseev/venv/bin/python
/home/student/Документы/PycharmProjects/IS-22/Proj_1sem_Eliseev/PZ_16/PZ_16.py
Введите первое число: 52
Введите второе число: 61
Сложение: 52.0 + 61.0 = 113.0
Вычитание: 52.0 - 61.0 = -9.0
Умножение: 52.0 * 61.0 = 3172.0
Деление: 52.0 / 61.0 = 0.8524590163934426
```

Process finished with exit code 0

2)

```
# Создание базового класса "Работник" и его наследование для создания классов "Менеджер" и
# "Инженер". В классе
# "Работник" будут общие методы, такие как "работать" и "получать зарплату", а классы-
# наследники будут иметь свои
# уникальные методы и свойства, такие как "управлять командой" и "проектировать системы".

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

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

    def get_salary(self):
        return f"{self.name}а зарплата {self.salary}."

class Manager(Worker):
    def __init__(self, name, salary, team):
        super().__init__(name, salary)
        self.team = team

    def manage_team(self):
        return f"{self.name} управляет командой."

class Engineer(Worker):
    def __init__(self, name, salary, projects):
        super().__init__(name, salary)
        self.projects = projects
```

```
def design_systems(self):  
    return f"{self.name} занимается проектированием систем."  
  
manager1 = Manager("Джон", 50000, ["Элис", "Боб", "Чарли"])  
engineer1 = Engineer("Элис", 60000, ["Проект А", "Проект Б"])  
  
print(manager1.work())  
print(manager1.get_salary())  
print(manager1.manage_team())  
  
print(engineer1.work())  
print(engineer1.get_salary())  
print(engineer1.design_systems())
```

Протокол работы программы:

```
/home/student/Документы/PycharmProjects/IS-22/Proj_1sem_Eliseev/venv/bin/python  
/home/student/Документы/PycharmProjects/IS-22/Proj_1sem_Eliseev/PZ_16/PZ_16_2.py  
Джон работает.  
Джона зарплата 50000.  
Джон управляет командой.  
Элис работает.  
Элиса зарплата 60000.  
Элис занимается проектированием систем.
```

Process finished with exit code 0

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