МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Лабораторная работа №7

**Работа с классами ч.3**

**по дисциплине**

**«Введение в информационные технологии»**

Выполнил: студент гр. БВТ2403

Косарев С.А.

Проверил:

Москва, 2024 г.

**Цель работы**: Разработать систему управления сотрудниками, демонстрирующую множественное наследование, инкапсуляцию и полиморфизм в Python. Система должна уметь обрабатывать различные типы сотрудников, включая менеджеров и технических специалистов, а также предоставлять возможность для расширения и добавления новых ролей.

**Оборудование:** компьютер с установленным программным обеспечением Python.

**Методы и технологии:**

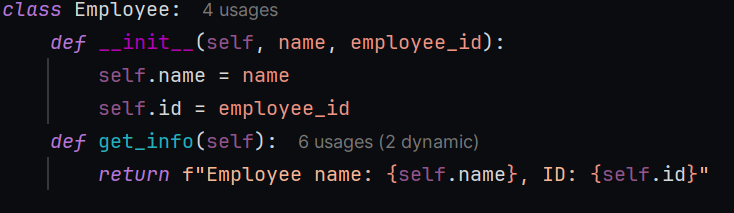
Основные функции, использованные в проекте, включают:

1. Для создания класса был использован метод class. Для инициализации переменных в классе была использована функция \_\_init\_\_.
2. Для забивания переменных в класс был использован self. Для создания функции get\_info() используется def get\_info(). Также была использована f-строка для вывода выражения с использованием переменной. Для вывода значений используется print. Для введения данных был использован input(). Для проверки выполнения/невыполнения условий используется if, else. Для обращения к родительскому классу был использован super.

Реализация:

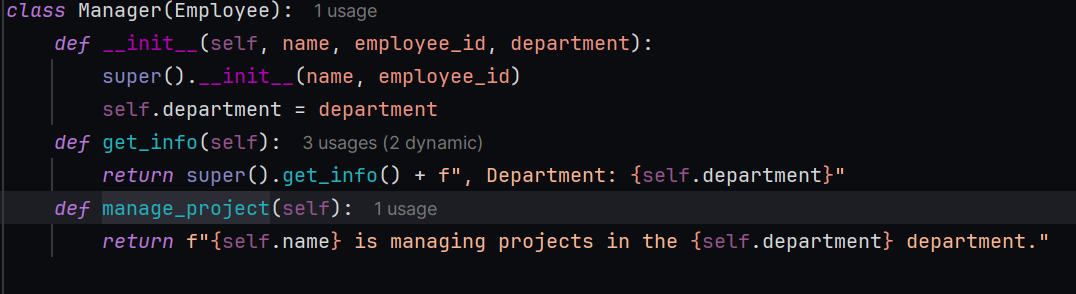
**Задание 1**

Создаем класс Employee с атрибутами name, id и методом get\_info, инициализируя атрибуты с помощью \_\_init\_\_.



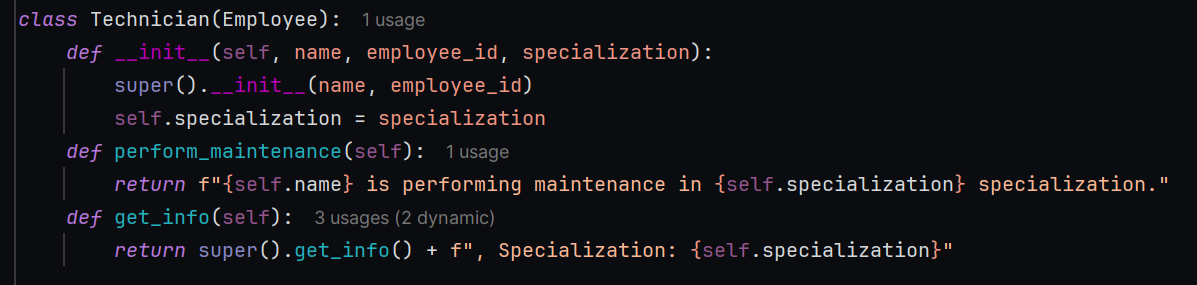
**Задание 2**

Создаем класс Manager, который наследует от класса employee такие атрибуты как имя и номер, с помощью super, и добавляем новые атрибуты. Также добавляем новый метод manage project.



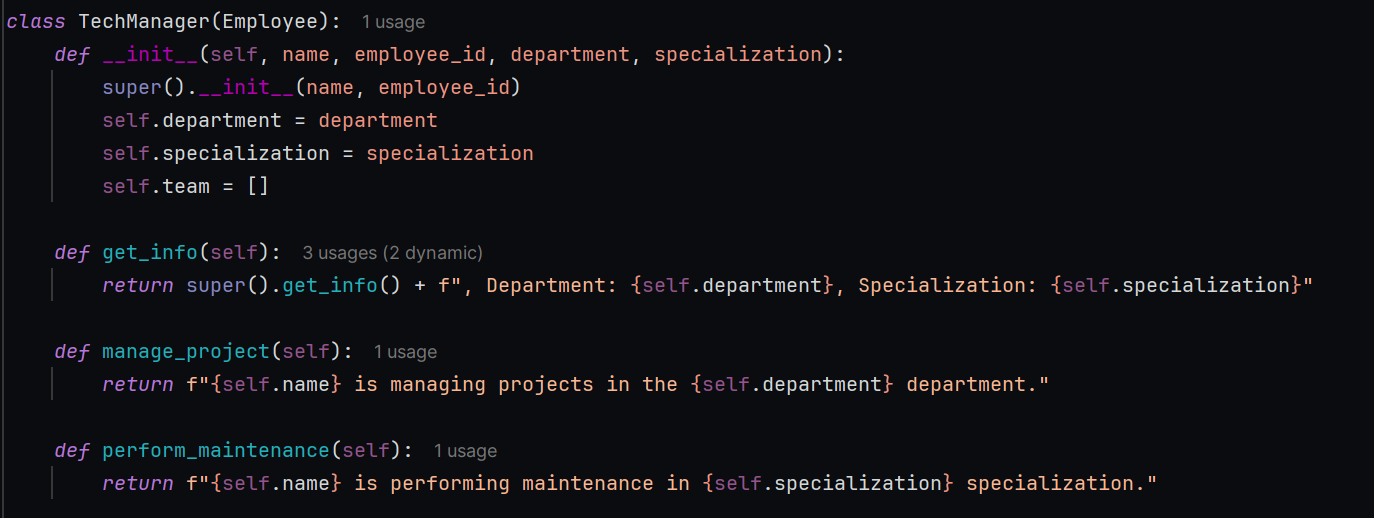
**Задание 3**

Создаем класс Technician, который наследует имя и номер. Также добавляем уникальные атрибуты, такие как специализация. Далле добавляем методы perform\_maintenance() и get\_info(), который наследуется от employee.



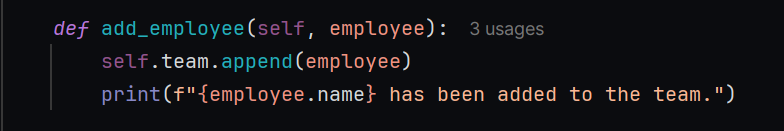
**Задание 4**

Создаем класс TechManager, который наследует атрибуты и методы от Manager и Technician, то есть добавляются методы perform\_maintenance(), manage\_project() и get\_info().



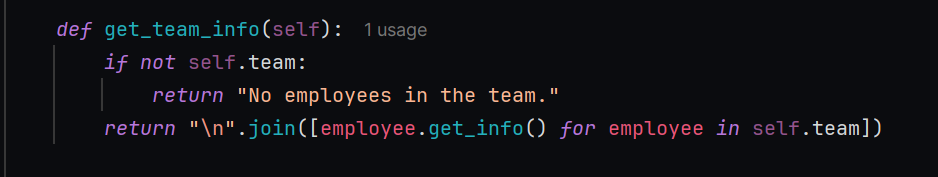
**Задание 5**

В класс TechManager добавляем метод добавления нового работника, предварительно заведя пустой список работников.



**Задание 6**

В класс TechManager добавляем метод, который позволяет просматривать весь список работников. Используем /n, для удобного выведения списка сотрудников.



**Задание 7**

Создание объектов выглядит так:



**Вывод:** Мы разработали систему управления сотрудниками, демонстрирующую множественное наследование, инкапсуляцию и полиморфизм в Python. Система включает в себя возможность расширения ролей.