**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**Московский технический университет связи и информатики**

**(МТУСИ)**

**Отчёт**

**Лабораторная работа № 7**

«Работа с классами ч.3»

**по дисциплине: «**Введение в информационные технологии**»**

Выполнил:

студент группы БВТ2402

Зубенок Руслан Юрьевич

Москва

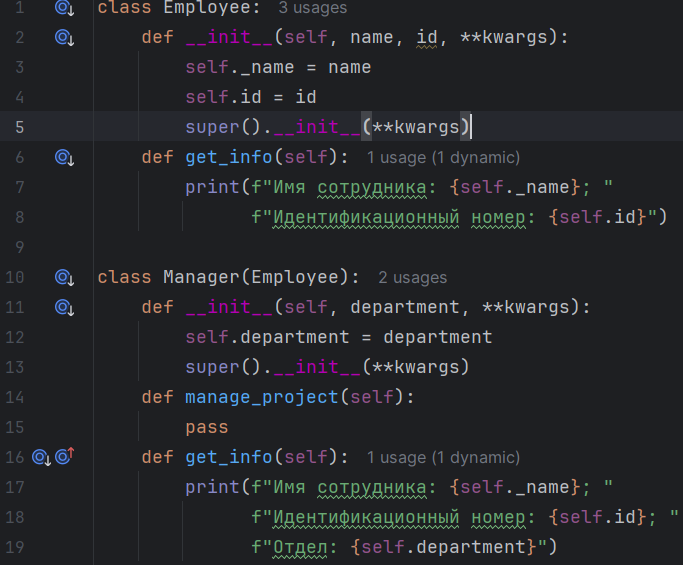
2024

**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.

**Задание**

1. Создали класс Employee с общими атрибутами, такими как **name** (имя), **id** (идентификационный номер) и методами, например, **get\_info**(), который возвращает базовую информацию о сотруднике.
2. Создали класс **Manager** с дополнительными атрибутами, такими как **department** (отдел) и методами, например, **manage\_project**(), символизирующим управление проектами.
3. Создали класс **Technician** с уникальными атрибутами, такими как **specialization** (специализация), и методами, например, **perform\_maintenance**(), означающим выполнение технического обслуживания.
4. Создали класс **TechManager**, который наследует как Manager, так и Technician. Этот класс должен комбинировать управленческие способности и технические навыки, например, иметь методы для управления проектами и выполнения технического обслуживания.
5. Добавили метод **add\_employee**(), который позволяет **TechManager** добавлять сотрудников в список подчинённых.
6. Реализовали метод **get\_team\_info**(), который выводит информацию о всех подчинённых сотрудниках.
7. Создали объекты каждого класса и демонстрируйте их функциональность.

Исполняемый файл:







Вывод:

