

hr_registry

Дата создания: 01.11.2024, 10:05

Задание: Создание модулей HR-системы для управления сотрудниками

Цель: Разработать систему для управления данными сотрудников с использованием классов Employee, HR, и HRRegistry. Задача — построить структуру, которая позволяет хранить, обрабатывать и отображать информацию о сотрудниках.

Структура задания

1. Создайте структуру проекта:

- Создайте проект и папку src для исходного кода.
- Добавьте файл requirements.txt и создайте виртуальное окружение для установки зависимостей, если они появятся.

2. Класс Employee:

- **Цель:** Класс для представления сотрудника с основными полями и методами.
- **Основные поля:**
 - id (уникальный идентификатор сотрудника),
 - name (имя сотрудника),
 - position (должность),
 - salary (зарплата).
- **Основные методы:**
 - get_info(): возвращает информацию о сотруднике в виде строки.
 - promote(new_position: str): обновляет должность сотрудника.
 - increase_salary(amount: float): увеличивает зарплату сотрудника на указанную сумму.

• Пример использования:

```
emp = Employee(id=1, name="Alice Smith", position="Developer", salary=50000)
print(emp.get_info())
emp.promote("Senior Developer")
emp.increase_salary(5000)
```

3. Класс HR:

- **Цель:** Класс HR представляет отдел кадров, который может выполнять действия

над объектами Employee.

- **Основные методы:**

- hire(employee: Employee): принимает объект сотрудника и сохраняет его в списке сотрудников.
- terminate(employee_id: int): удаляет сотрудника из списка по его ID.
- get_employee(employee_id: int) -> Employee: возвращает объект сотрудника по ID.

- **Пример использования:**

```
hr = HR()
emp = Employee(id=2, name="Bob Brown", position="Manager", salary=70000)
hr.hire(emp)
hr.terminate(2)
```

4. Класс HRRegistry:

- **Цель:** Класс HRRegistry выступает как реестр сотрудников и предоставляет интерфейс для управления списком всех сотрудников.

- **Основные методы:**

- add_hr(hr: HR): добавляет HR в реестр.
- find_employee_by_name(name: str) -> Employee: ищет сотрудника по имени среди всех HR-отделов и возвращает объект Employee, если найден.
- list_all_employees(): выводит информацию о всех сотрудниках во всех отделах.

- **Пример использования:**

```
registry = HRRegistry()
hr1 = HR()
hr2 = HR()
registry.add_hr(hr1)
registry.add_hr(hr2)
```

5. **Дополнительное задание** (для продвинутых):

- Добавьте метод generate_employee_report() в HRRegistry, который собирает и возвращает отчёт обо всех сотрудниках, их должностях и зарплатах в формате текста или JSON.
- Пример реализации может включать экспорт данных в файл employees_report.json для внешнего анализа.