

OOP

Company Management System

In this assignment we are going to cover some of the major concepts of OOP such as Class, object, abstract classes and inheritance along with methods and variables.

Introduction

The idea behind this assignment is very simple, let's assume that we are preparing a system that will hold the record of different departments existing within the company and different employees working in those departments.

The idea is to create, update, read and delete records available within them. Obviously this can be done through procedural programming as well. As we know that using OOP concepts in these situations will help us to imitate the real time behavior of the actual functionality.

Program Organization

Let's explore the different Classes and methods we need to create and implement the above system.

1. *Class Department*: In this class, the constructor will accept the department name and create it, if it does not exist. Other methods in this class involved creating the department, updating the number of employees in the department, manager/lead of the department, retrieving the information about the department, deleting the existing record in the department or deleting the department itself.
2. *Class Employee*: Class that will consist of the information related to the employee, current department and project details, current manager, current role. The class will have different get and set methods to update and read the information for a particular employee. Delete operation should also be there to delete the record of an employee once he has moved out of the organization.

Tasks

1. *Class Department*:
 - a. *create_department*: This method will create the departments as per the requirement. The idea is that this method should work as a registration and automatically it should be invoked when the object is getting created.
 - b. *department_list()*: Print the list of all the departments created till now in the organization.
 - c. *delete_department()*: Delete the department from the organization and relieve the registered employee under that department.
 - d. *register_department()*: Register the department in the organization.

- e. *register_employee()*: Register the employee in the department.
- f. *remove_employee()*: Remove the employee from the department.
- g. *read_employee_role()*: Return detail of the work/project currently the employee is involved.
- h. *update_employee()*: Update the details of employees such as their role, manager in the department.

2. *Class Employee:*

- a. *create_employee()/register_employee()*: The method will register and assign the unique ID to the new employee. It will accept the details of employees such as Name, Skill, Role and Salary.
- b. *update_employee()*: Update the details of the employee based on the ID provided. The details of the employee can not be updated outside of the class. Implement getter and setter methods to do the same.
- c. *delete_employee()*: Delete the record of the employee once the person left the organization.
- d. *list_all_employee()*: List details of all the employees that are working in a particular department.
- e. *read_employee()*: Read all the details of the employee based on the given ID.
- f. *set_manager()*: Set the manager for the department
- g. *get_manager()*: Get the details of the current manager of the given organization.
- h. *list_all_manager()*: List the name of all the employees who are currently working as manager.

3. *Driver.py*: This python file will basically have the static commands to implement required features. This essentially will create different objects of the asked classes, and will perform CRUD operation on those classes. Please make sure that the method call that is supposed to happen with the Employee should happen with the help of the employee object only. Internally it the communication can happen between department and employee to implement some of the features.

Expected Tasks

1. This is just a minimal example and a practice assignment, feel free to add more methods to make this program more realistic in nature.
2. You should use appropriate data structures and data types to store the relevant data points so that the details can be updated or fetched in the most efficient way.
3. Please design the algorithm for each task before implementing it and do the analysis of it in terms of how efficient the implement method is and what are the alternative methods to it.
4. You can write three different python files to implement the above practice assignment, it will enable the understanding of importing python files/libraries in your code.