CSE 241 Homework Assignment 5

DUE

May 6, 2019, 23:55

Description

• This is an individual assignment. Please do not collaborate.

• If you think that this document does not clearly describe the assignment, ask questions before it's too late.

You won't be given a chance to correct any mistakes.

Scope: University Management System

You are expected to develop university management system.

Main Problem: Personal Management

Your Program is storyteller. Actors of the story are 10 employees who work in the University. It tells a story which composes 50 random actions. Actions are described below. Actions can be taken by only actor stated below. Actors can be Lecturer, ResearchAssistant, Secretary or Officer, described in UML diagram. They are inherited from Employee. As you see, some actions can be taken by all employee types.

Actions Types

Number: Name by **Actor.function Effects**
1: document by Officer.makeDoc happiness:-2 contribution:3
2: slackness by Employee.drinkTea happiness:+5 contribution:-2
3: project by ResearchAssistant.research happiness:+3 contribution: +4
4: lesson by Lecturer.giveLesson happiness:+1 contribution: +5
5: seminar by AcademicPersonnel.seeSuccessfulStudent happiness:+10 contribution: +0
6: academicPaper by AcademicPersonnel.makePublish happiness:+2 contribution:+5
7: administration by AdministrativePersonnel.manageProcess happiness:-1 contribution:+2
8: HomeworkTime by Lecturer.giveHW happiness:-2 contribution:+1
9: homeworkTimeout by ResearchAssistant.readHW happiness:-3 contribution:+2
10: incident by Secretary.receivePetition happiness:-1 contribution:-1
11: solution by Employee.submitPetition happiness:+1 contribution -2

For example, only officer can do "makeDoc" for the action document. But the action slackness might be done by functions drinkTea of all employees.

Requirements:

1. Your program will not expect any input. When it begins, it will process and prints all story.
2. Program should read txt file, personnellist.txt. It indicates Employee. There are 20 personnel examples in list, 10 of them should be selected to work in the university.
3. There is a University data type in program. It employs Employee and it expects that personnel contributes.
4. Actions should be managed with enumeration.
5. You should separate compilation into header files and implementation files.

Test Scenarios

Firstly, program reads employeelist.txt. Then, University employs 10 of them randomly. In other mean, 10 employees are selected randomly from the list and University employs them. Then, 50 random actions are processed, and these actions will be taken by related employee, which stated in the list. These actions affect happiness of employee and contribution of university. Functions of classes tell the story(print which action happens) and do effects.

For example;

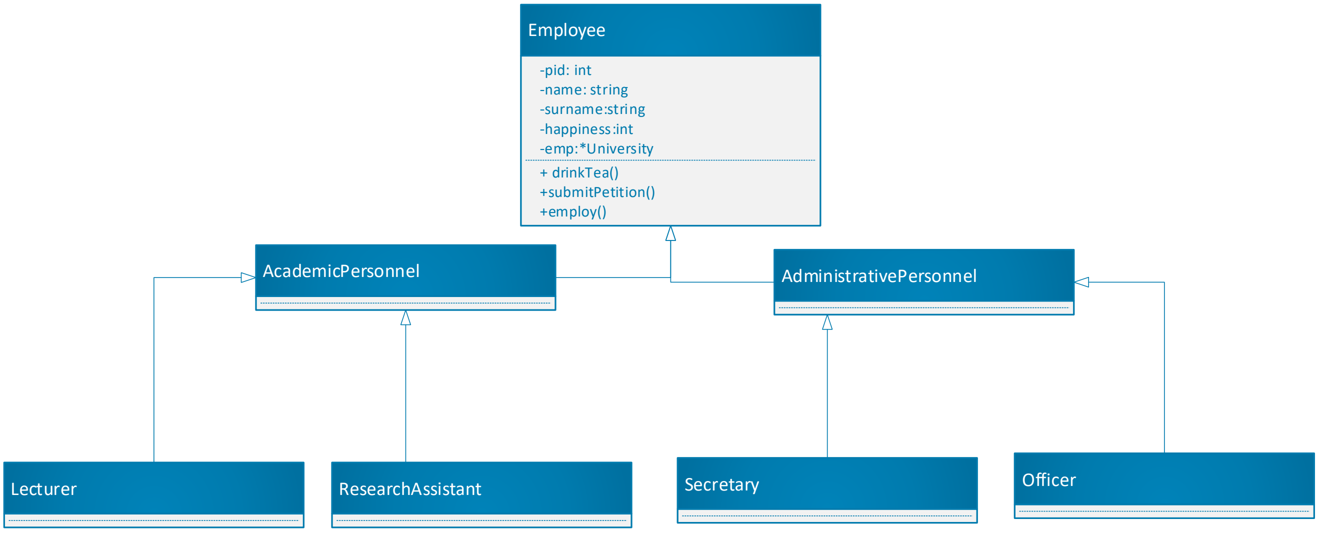
Suppose that University employs A(Lecturer), B(ResearchAssistant), C(Lecturer), D(Officer), E(Secretary) and first action is slackness. Then Program selects one employee, let us say that it is A, working in the university and it takes the action by doing function related. This function, which is drinkTea, prints the action and do effects, increasing the happiness of the actor and contribution of the university. So, happiness of A is 5 and contribution of uni is -2. Let Second action be administration. Then program selects one AdministrativePersonnel, let us say that it is D, working in the university and it takes the action by doing function related. This function, which is manageProcess, prints the action and do effects, increasing the happiness of the actor and contribution of the university. So, happiness of D is -1 and contribution of the uni is +2.

So, it prints like that;

```
(Output)> A have slackness. Therefore, A drinks tea. Happiness of A is 5, contribution of uni is -2.
(Output)> D have administration. Therefore, D manage process. Happiness of D is -1, contribution of uni is 0.
```

As you see, contribution of university is common, i.e. contribution of university is affected by all action. But happiness of actor is affected by only the action the actor takes. After finishing all actions, happiness of all actors and contribution of university will be shown.

UML Diagram



Employee is only class which all functions and variables are stated in UML diagram. Other classes have some discrete functions which stated in Action types.

Remarks

- Do not use any elements which is not covered in class.
- Do not submit your code without testing it with several different scenarios.
- Write comments in your code.
- Prepare a Makefile for your programme.
- You are free to define your functions or classes.

Turn in:

- Source code of a complete C++ program. Name of the file should be in this format: <full_name>_<id>.cpp.
- Example: gokhan_kaya_000000.cpp. Please do not use any Turkish special characters.
- You don't need to use an IDE for this assignment. Your code will be compiled and run in a command window.
- Your code will be compiled and tested on a Linux machine(Ubuntu). GCC will be used
- Make sure you don't get compile errors when you issue this command : g++ <full_name>_<id>.cpp.
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may loose points.)
- You may not get full credit if your implementation contradicts with the statements in this document.

Late Submission

- (0,24] hours: -20%
- (24,48] hours: -40%
- (48,72] hours: -60%
- (72,-) hours: -100%

Grading (Tentative)

- Max Grade : 100.
- Multiple tests(at least 5) will be performed.

All of the followings are possible deductions from Max Grade.

- #define HARD_CODED_VALUES -10.
- No submission: -100. (be consistent in doing this and your overall grade will converge to N/A) (To be specific: if you miss 3 assignments you'll get N/A)
- Compile errors: -100.
- Irrelevant code: -100.
- Major parts are missing: -100.
- Unnecessarily long code: -30.
- Using language elements and libraries which are not allowed: -100.
- Not caring about the structure and efficiency: -30. (avoid using hard-coded values, avoid hard-to-follow expressions, avoid code repetition, avoid unnecessary loops).
- Significant number of compiler warnings: -10.
- Not commented enough: -5. (Comments are in English).
- Source code encoding is not UTF-8 and characters are not properly displayed: -5. (You can use 'Visual Studio Code', 'Sublime Text', 'Atom' etc. . . Check the character encoding of your text editor and set it to UTF-8).
- Missing or wrong output values: Fails the test.
- Output format is wrong: -30.
- Infinite loop: Fails the test.
- Segmentation fault: Fails the test.
- Fails 5 or more random tests: -100.
- Fails the test: deduction up to 20.
- Prints anything extra: -30.
- Unwanted chars and spaces in output.txt: -30.
- Submission includes files other than the expected: -10.
- Submission does not follow the file naming convention: -10.
- Sharing or inheriting code: -200