

## Homework #4

RELEASE DATE: 04/26/2013

DUE DATE: 05/16/2013, noon

*As directed below, you need to submit your code to the designated place on the course website.*

*Any form of cheating, lying or plagiarism will not be tolerated. Students can get zero scores and/or get negative scores and/or fail the class and/or be kicked out of school and/or receive other punishments for those kinds of misconducts.*

*Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.*

*Since everyone needs to write the final solutions alone, there is absolutely no need to lend your homework solutions and/or source codes to your classmates at any time. In order to maximize the level of fairness in this class, lending and borrowing homework solutions are both regarded as dishonest behaviors and will be punished according to the honesty policy.*

*You need to write your homework report in English. For programming, we only allow Java.*

*This homework set comes with 200 points and 40 bonus points. In general, every homework set of ours would come with a full credit of 200 points.*

## 1 Description

The P00Pet is ready to breed, and fight! You have lots of flexibility to design a fight game between P00Pets this time. A fight game happens at a P00Arena, with two or more P00Pet instances inside. Each P00Pet comes with at least four properties, its health point (HP), its magic point (MP), its agility value (AGI), and its name, and with at two actions: moving, or performing a action on itself or other pets on the arena. The moving should depend on AGI, and the effect of a action should depend on MP. A action can be used to change the HP, MP, AGI, or position of the pet(s). The goal is to decrease the opponent's HP as soon as possible.

You are asked to implement and demonstrate a prototype of the fight game. As it is only a prototype and probably will be extended by the programming team later, both P00Pet and P00Arena need to be **abstract**. Nevertheless, you can resort to polymorphism and use subclasses of P00Pet and subclasses of P00Arena to make things work. You need to implement at least two kinds of (i.e. two different subclasses of) P00Pet, and one kind of P00Arena. You can see that there are some other abstract classes in the provided codes. Yes, you need to extend them appropriately.

To make things more realistic, your lazy team leader (a.k.a. CharlieL) did not give you many documents, but only all the source code. No, you cannot modify his source code because he is really stubborn, and you know that he'd only use the main method in P00Fight to check your code. So your only chance to please him is to use your creativity under the **contract** that he has given you. Good luck!

## 2 Requirements

- Implement your P00(PetKind) and P00(ArenaKind) classes, and any additional classes that you need.
- Write a short report with at **most** four A4 pages that contains the following items:
  - (1) your name and school ID
  - (2) the relations between the classes that you design
  - (3) the advantages of your design in terms of software engineering (say, co-development)
  - (4) the disadvantages of your design in terms of software engineering (say, co-development)
  - (5) the advantages of your design in terms of the interestingness of the game

- (6) the disadvantages of your design in terms of the interestingness of the game
- (7) any part that you implemented that is worth getting “bonus” points

You should submit your report in **PDF** format. See the course forum for some possible instructions for converting from Word to PDF.

### 3 Special Notes

- 40 bonus (a.k.a. extra) points will be allocated to award creativity—basically how interesting your game is.
- We will **REQUIRE** that you only include the files listed below. Some points may be deducted if you include additional files.
- The TAs will use the following lines on our LINUX machines to do the grading:

```
git clone your_git_repository
cp /home/faculty/htlin/htdocs/course/oop13spring/hw4/hw4.jar .
javac -cp hw4.jar -d . @source_files
source run.sh
```

### 4 Submission File

Similar to homework 3, we will ask you to use some GIT service to allow you to learn to manage your codes for the homework. Your GIT repository should contain the following items:

- `src/P00(PetKind).java` (at least two different kinds)
- `src/P00(ArenaKind).java` (at least one)
- `src/*.java`, which represent any other classes that you implemented, **you can use sub-directories for organizing the source code if you want**
- `source_files`: the list of the java files that you implemented (see above for javac)
- `run.sh`: your running script that should follow the sample `run.sh`
- `.gitignore` and `.gitkeep`: the former for ignoring some files from version control, the latter for keeping an empty directory
- **any other non-Java files (say, image) needed for your program**
- a PDF file `report.pdf`, which is your report file written in English
- an optional text file `MEDAL`, in case you want to use the gold medals, with the number of medals listed as a single number in the first line of the file. Use your medals wisely—usage cannot be retracted

**Please do not include any other files (e.g. class files) in the GIT repository.**