Lab 2: Inheritance

1. Instruction

1. Click the provided link on MyCourseVille to create your own repository.
2. Create a "java project" and set the project name in this format **Lab2_2024_1_{ID}_{FIRSTNAME}**, e.g., **Lab2_2024_1_6531999921_Somsak**.
3. Initialize Git in your project directory
   - 3.1. Add .gitignore and set up your git.
   - 3.2. Create your remote repository from a given link.
   - 3.3. Commit and push initial codes to your GitHub repository. Use force push if necessary.
4. Implement all the classes and methods following the details given in the problem statement file which you can download from CourseVille.
   - 4.1. You should create a commit with meaningful messages when you finish each part of your program.
   - 4.2. Don't wait until you finish all features to create a commit.
5. Test your codes with the provided JUnit test cases, they are inside package **test.grader**
   - 5.1. If you want to create your own test cases, please put them inside package **test.student**
   - 5.2. Aside from passing all test cases, your program must be able to run properly without any runtime errors.
6. After finishing the program, create a UML diagram for classes in package **monkey** and put the result image (UML.png) at the root of your project folder.
7. Export your project into a jar file called **Lab2_2024_1_{ID}** and place it at the root directory of your project. Make sure you export all your source (.java) files. You can open the jar file with any zip software to check it. Jar file example -> **Lab2_2024_1_6531234521.jar.**
8. Push all other commits to your GitHub repository.

2. Problem Statement: Lopburi Monkey War

In the beautiful land called “Lopburi” lies the realm of the Monkeys. One day, The Apes, who have evolved from the stray monkeys of long ago, arrived to invade and attempt to destroy Lopburi. As the Monkey King, you need to assemble an army and lead them to fight back against the invasion of the malevolent Apes, thus restoring Lopburi to its former greatness.

In this lab, we will be coding a game regarding the situation above. In this game, you will command your hand-picked army of Monkeys to fight against the invading Apes. If any monkey (or ape) has a health point (HP) equal to 0, that monkey will die. There are 5 types of monkeys:

- BaseMonkey
- MuscleMonkey
- MommyMonkey
- UgabugagaMonkey
- Ape (which is technically a descendant of monkey)

Every monkey has a *hp*, an *atk*, and a *def* (a health point, an attack power, and a defense power respectively) Each type can attack their enemy.

A BaseMonkey is a monkey who doesn’t have any special skill.

- When a BaseMonkey attacks, it will deal damage equal to its *atk* subtracted by the enemy *def* to the targeted ape.

A MuscleMonkey is a monkey who possesses a skill named “*buff*”

- When a MuscleMonkey attacks, it will deal damage = 2 * (its *atk* subtract with the enemy *def)* to the targeted ape.
- When a MuscleMonkey uses *buff*, it will increase its own *atk* and *def*.

A MommyMonkey is a monkey who possesses a skill named “*birth*”

- MommyMonkey cannot attack.
- When a MommyMonkey uses *birth*, it will create a BaseMonkey with default attributes.

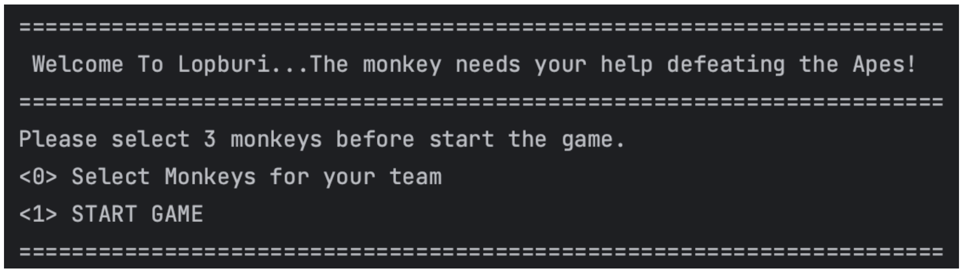
A UgabugagaMonkey is a monkey who possesses a skill named “*heal*”

- When a UgabugagaMonkey attacks, it will deal damage = its *atk* subtract with the enemy *def* to the targeted ape.
- When a UgabugagaMonkey uses *heal*, it will heal the targeted monkey according to its HEAL value.

An Ape is a monkey who possesses a skill named “*Attack AOE*”

- When an Ape attacks, it will deal damage = its *atk* subtract with the enemy *def* to the targeted ape.
- When an Ape uses skill attackAOE, it will attack all monkeys in the fields (see details later).

The program example is shown below. The program should be run from the Main class in the package main.

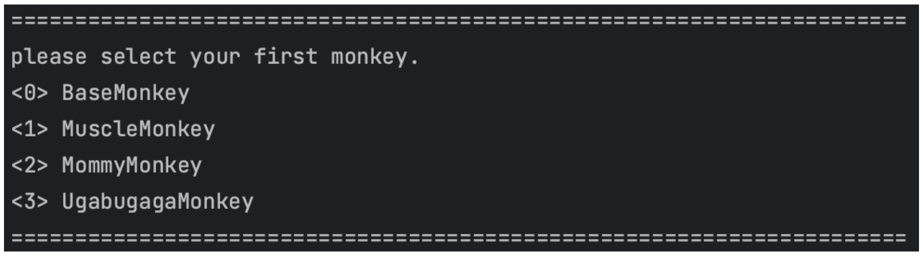


```
======================================================================
 Welcome To Lopburi...The monkey needs your help defeating the Apes!
======================================================================
Please select 3 monkeys before start the game.
<0> Select Monkeys for your team
<1> START GAME
======================================================================
```

(start game menu)

There are 2 options:

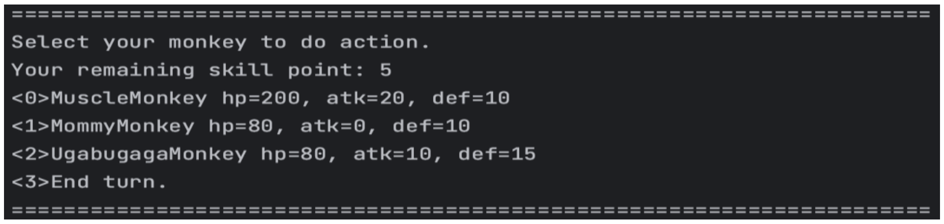
1. <0> Select monkey for your team
    - This option will present four types of monkeys that you can choose from to help you fight the ape. You will need to select three monkeys.



```
======================================================================
please select your first monkey.
<0> BaseMonkey
<1> MuscleMonkey
<2> MommyMonkey
<3> UgabugagaMonkey
======================================================================
```

(Select monkey menu)

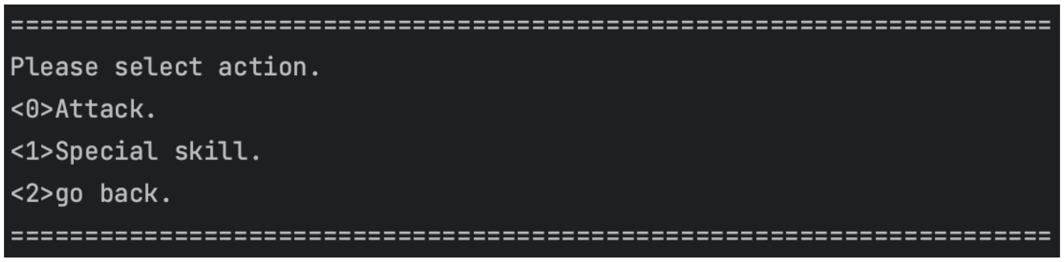
2. <1> START GAME
    - This choice will begin the game. Once the game starts, a menu will show up, letting you pick a monkey to take actions. Your turn will finish either after using 5 skill points or if you press “<i>End turn” to end your turn.



```
======================================================================
Select your monkey to do action.
Your remaining skill point: 5
<0>MuscleMonkey hp=200, atk=20, def=10
<1>MommyMonkey hp=80, atk=0, def=10
<2>UgabugagaMonkey hp=80, atk=10, def=15
<3>End turn.
======================================================================
```

(select monkey menu)

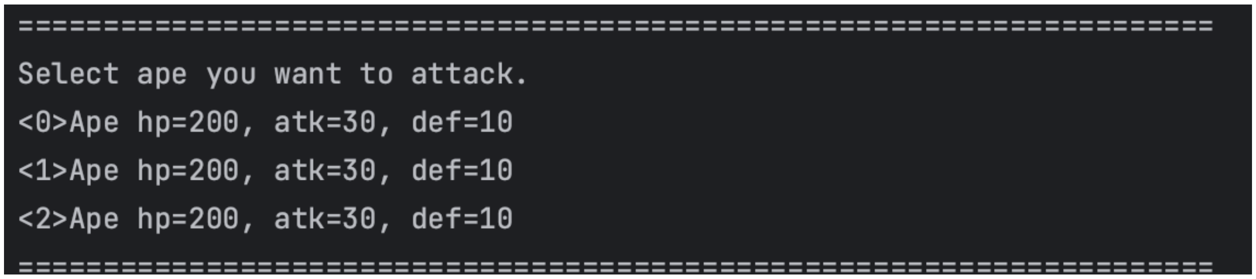
Once you've selected a monkey, the action menu will appear, allowing you to choose from three actions. The first is a normal attack, the second is a special skill that depends on the type of monkey, and the last option is to go back to reselect the monkey for performing an action.



```
===============================================================
Please select action.
<0>Attack.
<1>Special skill.
<2>go back.
===============================================================
```

(action menu)

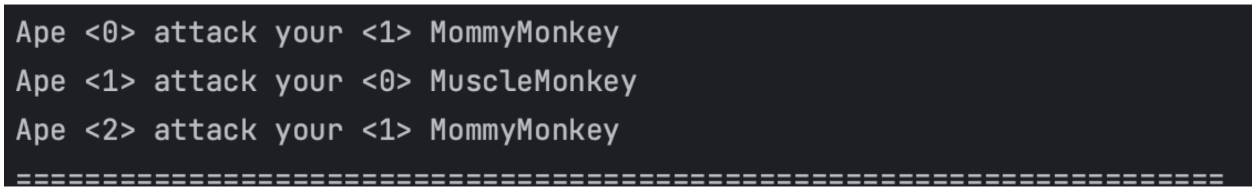
If you decide to attack the enemy, the menu will display a list of apes. You will need to choose the specific ape you want to attack.



```
======================================================================
Select ape you want to attack.
<0>Ape hp=200, atk=30, def=10
<1>Ape hp=200, atk=30, def=10
<2>Ape hp=200, atk=30, def=10
======================================================================
```

(list of apes)

When your turn is finished, the game will switch to the Ape's turn. You'll then see what actions the apes take during their turn.



```
Ape <0> attack your <1> MommyMonkey
Ape <1> attack your <0> MuscleMonkey
Ape <2> attack your <1> MommyMonkey
=======================================================================
```

(this is what apes do in their turn)

The game will end when your entire team is defeated or when all the apes are defeated.

3. Implementation Details

3.1 Package monkey /* You must implement this package from scratch */

3.1.1 Class BaseMonkey /* You must implement this class from scratch */

Variable

Name	Description
- int maxHp	This is BaseMonkey’s maxHp. maxHp cannot be less than 0.
- int hp	This is BaseMonkey’s hp. Hp cannot be less than 0 and cannot be more than maxHp.
- int atk	This is BaseMonkey’s atk. Atk cannot be less than 0.
- int def	This is BaseMonkey’s def. Def cannot be less than 0.

Method

Name	Description
+ BaseMonkey()	Initialize this BaseMonkey with the following attribute set -maxHp to 30 -Hp to maxHp -atk to 20 -def to 5
+ BaseMonkey(int maxHp, int atk, int def)	Initialize this BaseMonkey with the following attribute set -maxHp to maxHp -Hp to maxHp -atk to atk -def to def
+ void attack(BaseMonkey m)	This monkey attacks the given monkey, then the enemy would be dealt damage equal to this monkey’s atk power subtracted by enemy’s def . The damage dealt is the HP change. The damage value cannot be below 0.
+ String getType()	Return a monkey’s type as String. You can use getSimpleName() to simplify the return value from getClass().
+ String toString()	Return string in this following format. <Monkey’s Type> hp=<hp>, atk=<atk>, def=<def> (example is in the last picture in the previous page)

+ getter,setter of all fields	getter, setter of all fields. The value of hp cannot go below 0 or beyond maxHp. The value of atk and def cannot go below 0.

3.1.2 Class MommyMonkey /* You must implement this class from scratch */

This is a type of Monkey.

Method

Name	Description
+ MommyMonkey(int maxHp, int atk, int def)	Initialize this MommyMonkey with the following attribute set -maxHp to maxHp -Hp to maxHp -atk to atk -def to def Hint: Try to use “super”
+ void attack(BaseMonkey m)	Do nothing
+ void birth()	Initialize BaseMonkey by calling default constructor then add it to the monkey container in the game system. Hint:you can see a usage example of the game system in class Main(In method skillFlow(BaseMonkey m)).

3.1.3 Class MuscleMonkey /* You must implement this class from scratch */

This is a type of Monkey

Variable

Name	Description
- final int POWER_UP	Set power up to 4

Method

Name	Description
+ MuscleMonkey(int maxHp,int atk , int def)	Initialize this MuscleMonkey with the following attribute set -maxHp to maxHp -Hp to maxHp -atk to atk -def to def

	Hint: Try to use “super”
+ int getPowerUp()	Return the value of POWER_UP.
+ void attack(BaseMonkey m)	Attack the given monkey 2 times (try calling method of superclass).
+ void buff()	Increase this monkey atk and def by power up

3.1.4 Class UgabugagaMonkey /* You must implement this class from scratch */

This is a type of Monkey.

Variable

Name	Description
- final int DEBUFF	Set DEBUFF to 1
- final int HEAL	Set HEAL to 10

Method

Name	Description
+ UgabugagaMonkey(int maxHp, int atk, int def)	Initialize this UgabugagaMonkey with the following attribute set -maxHp to maxHp -Hp to maxHp -atk to atk -def to def Hint: Try to use “super”
+ void attack(BaseMonkey m)	Attack the given monkey just like BaseMonkey. Then decrease the attacked monkey’s atk and def by DEBUFF.
+ void heal(BaseMonkey m)	Heal the given monkey by HEAL value without exceeding the maximumHP.
+ int getDebuff()	Get the value of DEBUFF.
+ int getHeal()	Get the value of HEAL.

3.1.5 Class Ape /* You must implement this class from scratch */

This is a type of Monkey.

Method

Name	Description

+ Ape(int maxHp, int atk, int def)	Initialize this Ape with the following attribute set -maxHp to maxHp -Hp to maxHp -atk to atk -def to def Hint: Try to use “super”
+ void attack(BaseMonkey m)	Attack the given monkey.
+ void void attackAOE()	Attack all of monkeys in the game Hint: -How to get all monkeys in the game. Check GameSystem class. -GameSystem class is where the ArrayList of monkeyContainer is.Which method return an array of monkeyContainer?

3.2 Package logic.game /* You must implement something in this package */

3.2.1 Class GameSystem /* You must implement 3 methods in this class */

Variable

Name	Description
- final ArrayList<BaseMonkey> monkeyContainer	Initialize ArrayList monkey container.
- final ArrayList<BaseMonkey> apeContainer	Initialize ArrayList ape container.
- int gameState	Set to 0. This is the state of the game.
- int sp	Set sp to 5.
- boolean gameEnd	True when monkey container or ape container is empty.
- GameSystem instance	A static instance of a game system. This will make sure that there is only one game system when our program is running.

Method

Name	Description
+ GameSystem getInstance()	Get a game system.
- GameSystem()	This is the Constructor. Add all the ape to its ArrayList. Set gameEnd to false.
+ void addApe()	/*You must implement this method */ Add 3 new apes to its container Each of the apes has

	-200 maxHp. -30 atk. -10 def.
+ boolean isGameEnd()	Check that the game is over.
+ void printMonkeyStatus()	Print monkey status with format.
+ void removeDeadEntity(ArrayList<BaseMonkey> entityContainer)	/*You must implement this method */ Loop over an entityContainer, then check whether each entity's hp is equal to 0. If the entity's hp is equal to 0, then (1) remove that entity from entityContainer and (2) call method showDeadMessage.
+ void showDeadMessage(BaseMonkey monkey, int index)	Show a dead monkey message. The index refers to position of that monkey in its container.
+ BaseMonkey fight(BaseMonkey a, BaseMonkey b)	/*You must implement this method */ This is a static method. Execute the code that has a attack b. Return b.
+ getter setter of the field	Getter/setter of the field.

3.3 Package application

3.3.1 Class Main

This class is the main program. You don't have to implement anything in this class. You can test the program by running this class.

4.Score Criteria

All Class (Total 68)

ApeTest (Total 10)

- testConstructor (2)
- testBadConstructor (2)
- testAttack (3)
- testAttackAOE (3)

BaseMonkeyTest (Total 13)

- testDefaultConstructor (1)
- testConstructor (2)
- testBadConstructor (2)
- testAttack (3)
- testGetType (1)
- testSetHp (1)
- testSetAtk (1)
- testSetDef (1)
- testToString (1)

GameSystemTest (Total 15)

- testAddApe (5)
- testRemoveDeadEntity (5)
- testFight (5)

MommyMonkeyTest (Total 10)

- testConstructor (2)
- testBadConstructor (2)
- testAttack (3)
- testBirth (3)

MuscleMonkeyTest (Total 10)

- testConstructor (2)
- testBadConstructor (2)
- testAttack (3)
- testBuff (3)

UgabugagaMonkeyTest (Total 10)

- testConstructor (2)
- testBadConstructor (2)
- testAttack (3)
- testHeal (3)

UML for all the monkeys (Total 8)

- identifier/methods/naming convention ผิด => หัก -1 คะแนน/Class
- UML structure ผิด => หัก -2 คะแนน/Class

TOTAL AllClass + UML = 68 + 8 = 76 (Will be scaled to 2.5%)