

Question 3: (60 minutes, 11.30-12.30)

1. Instructions

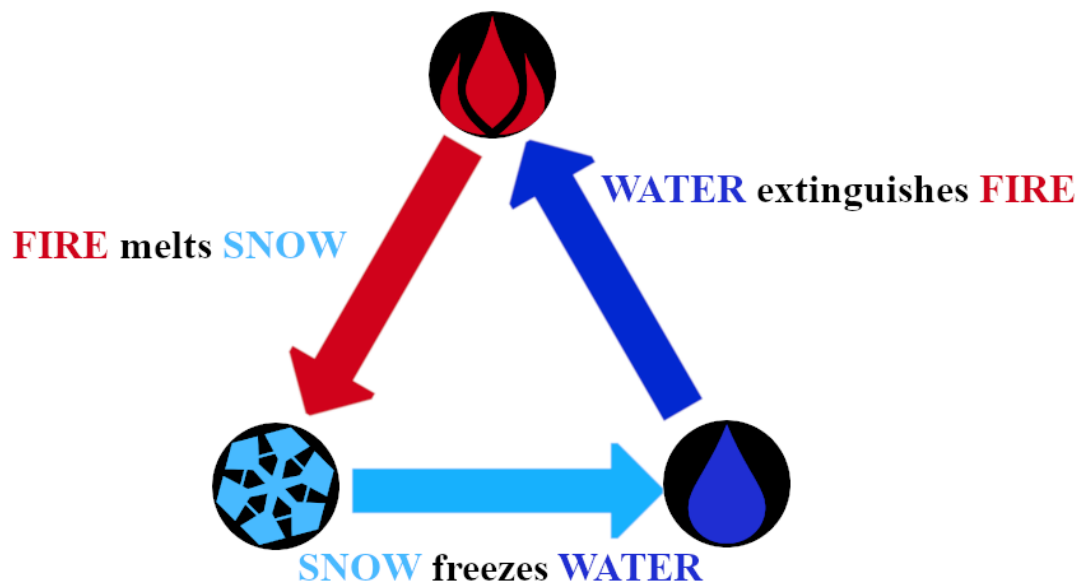
- 1) Create Java Project named “2110215_Midterm_Q3” in your workspace.
- 2) Copy all folders in “Q3.zip” to your project directory src folder.
- 3) You are to implement the following classes (details for each class are given in section 4.)
 - a) Penguin (package penguin)
 - b) FighterPenguin (package penguin)
 - c) ElementalPenguin (package penguin)
- 4) You also need to create a UML Diagram including the classes Penguin, FighterPenguin, and ElementalPenguin from the package penguin. (This is worth 2 points.) Save it as picture in the folder of package penguin.
- 5) JUnit for testing is in package test.
- 6) To submit:
 - a) go to src folder that you actually do the coding for this question.
 - b) Zip this question’s src folder. Name it YOUR-ID_Q3.zip (for example, 6332112121_Q3.zip)
 - c) Submit the zipped file as an assignment on MyCourseville.

2. Problem Statement: Penguin Impact



You are hired by MemeHoyo to finish the code for the game Penguin Impact after the previous programmer was fired for hacking Penguin Gems into his own account. Your task is to create new Penguin classes that supports PvP combat, including the elemental system.

The characters in this game are Penguins. There are NPC Penguins that has no ability to fight, Fighter Penguins who has a power level and can attack other Penguins, and Elemental Penguins who has elemental advantage and disadvantages over other Penguins with elements.



There are three elements in this game. Fire, Water, and Snow. Refer to the chart above for elemental advantage information. Fire beats Snow, Snow beats Water, and Water beats Fire. Elemental advantage allows a Penguin to deal double damage, while elemental disadvantage gives them a half damage penalty. Elemental Penguins do not gain advantage or disadvantage over non-elemental Penguins or Elemental Penguins of the same type.

3. Implementation Details

The class packages are summarized below. Note that only important methods that you either need to write and methods that you need to complete this question are listed below.

4.1 package penguin

4.1.1. `public class PenguinUtil` **/*ALREADY PROVIDED*/**

This class contains useful functions for creating your program.

a.) Variables

None.

b.) Methods

Method	Description
<u>+ String getTypeString(ElementalPenguin penguin)</u>	Static method. Returns a shorthand String of the input Penguin's element. Used for ElementalPenguin's toString.
<u>+ String makeToString(Penguin p)</u>	Static method. Returns a Penguin formatted as string. Used for Penguin's toString, FighterPenguin's toString, and ElementalPenguin's toString.
<u>+ String makeSpeech(Penguin p)</u>	Static method. Returns a formatted speech string. Used for Penguin's getSpeech.

4.1.2. **public enum PenguinType** **/*ALREADY PROVIDED*/**

This enum contains the values that can be used to initialize ElementalPenguin. There are three enum types here: PenguinType.FIRE, PenguinType.SNOW, PenguinType.WATER.

4.1.3. **public class Penguin** **/* You must implement this class from scratch*/**

This class represents a Penguin in the game world. This class will be used for non-combatant NPCs.

a.) Variables

Variable	Description
-String name	The name of this Penguin.
-int hp	The HP of this Penguin. If it reaches 0, it is defeated.
-int maxHp	The max HP of this Penguin.

-String catchphrase	This Penguin's catchphrase that it will say in combat.
---------------------	--------------------------------------------------------

b.) Methods

Method	Description
+ Penguin(String name, int hp, String catchphrase)	Constructor. Set variables to the given parameter values. maxHp is set to the value of hp. (The ordering of these parameter is important!)
Public getter/setter for everything	For setMaxHp and setHp, if the input is less than 0, set the value to 0. For setHp, if the input is greater than maxHp, set the value to maxHp.
+ String toString()	Returns this character's status. Use PenguinUtil's makeToString for this.
+ String getSpeech()	Returns this character's catchphrase. Use PenguinUtil's makeSpeech for this.

Important: There are 3 methods in PenguinTest (package test) that cannot be tested until other classes are finished. They are originally commented out. Please uncomment them to test them when you finish other classes.

4.1.4. public class FighterPenguin **/* You must implement this class from scratch*/**

This class represents an extension of the base Penguin class that can attack. This class will be used for player characters and enemies.

a.) Variables

Variable	Description
-int pow	The power level of this Penguin.

b.) Methods

Method	Description
+ FighterPenguin(String name, int hp, int pow, String catchphrase)	Constructor. Use the super constructor to set other values, and set the pow to pow.
Public getter/setter for pow.	For setPow, if the input is less than 0, set the value to 0.
+ int attack(Penguin target)	Deals damage to the target. Reduce the target Penguin's health by this Penguin's pow, then return the damage dealt.
+ String toString()	Returns this character's status. Use PenguinUtil's makeToString for this.

Important: There are 2 methods in FighterPenguinTest (package test) that cannot be tested until other classes are finished. They are originally commented out. Please uncomment them to test them when you finish other classes.

4.1.4. public class ElementalPenguin **/* You must implement this class from scratch*/**

This class represents an extension of the base FighterPenguin class whose attacks have an elemental attribute.

a.) Variables

Variable	Description
-PenguinType type	The elemental type of this Penguin. See the enum PenguinType for more information.

b.) Methods

Method	Description
+ ElementalPenguin(String name, int hp, int pow, PenguinType type, String catchphrase)	Constructor. Use the super constructor to set other values, and set the type to type.
Public getter/setter for type.	
+ int getElementalAdvantage(Penguin target)	Returns 0 if this Penguin has no advantage or disadvantage over target Penguin. Returns 1 if this Penguin has elemental advantage over target Penguin. Returns -1 if this Penguin has elemental disadvantage over target Penguin. (Elemental advantage is explained in Problem Statement, but to recap: Fire beats Snow, Snow beats Water, and Water beats Fire. If the target has no element or is the same element, there is no advantage or disadvantage.)

+ int attack(Penguin target)	<p>First, check for elemental advantage against the target.</p> <p>If there is elemental advantage, reduce the target Penguin's health by this Penguin's pow multiplied by 2.</p> <p>If there is elemental disadvantage, reduce the target Penguin's health by this Penguin's pow divided by 2 (rounded down).</p> <p>If there is no advantage or disadvantage, deal damage to target normally.</p> <p>Finally, return the damage dealt.</p>
+ String toString()	Returns this character's status. Use PenguinUtil's makeToString for this.

4.2 package game

4.2.1. public class Test **/*ALREADY PROVIDED*/**

This class is for testing the objects you wrote. You can use this class to do whatever you want; it will not be checked.

4.2.2. public class GameApp **/*ALREADY PROVIDED*/**

This class contains a simulation of the PvP system that you can use to test the objects you wrote.

4. Scoring Criteria

The scoring criteria is listed below. There are 3 JUnit files to test for the 3 files in the penguin package. The total score of this section is 38, which will be factored to a net score of 5. **Note that some tests start off commented, and will have to be uncommented to be properly tested.**

5.1 PenguinTest (Total: 12)

- a) testConstructor (2)
- b) testSetName (1)
- c) testSetHp (1)
- d) testSetMaxHp (1)
- e) testSetCatchphrase (1)
- f) testToString (uncomment after finishing ElementalPenguin) (1)
- g) testGetSpeech (1)
- h) testNotFighter (uncomment after finishing FighterPenguin) (2)
- i) testNotElemental (uncomment after finishing ElementalPenguin) (2)

5.2 FighterPenguinTest (Total: 10)

- a) testConstructor (2)
- b) testSetPow (1)
- c) testAttack (2)
- d) testToString (uncomment after finishing ElementalPenguin) (1)
- e) testNotElemental (uncomment after finishing ElementalPenguin) (2)
- f) testInheritance (2)

5.3 ElementalPenguinTest (Total: 14)

- a) testConstructor (2)
- b) testSetType (1)
- c) testGetElementalAdvantageFire (1)

- d) testGetElementalAdvantageSnow (1)
- e) testGetElementalAdvantageWater (1)
- f) testGetElementalAdvantageElementless (1)
- g) testGetElementalAdvantageSameElement (1)
- h) testAttackFire (1)
- i) testAttackSnow (1)
- j) testAttackWater (1)
- k) testToString (1)
- l) testInheritance (2)

5.4 UML Diagram of Penguin, FighterPenguin, and ElementalPenguin. Save the UML Diagram as png or jpg in the folder penguin. (2)