Assignment 3

Overview

The purpose of the assignment is to `extend` the `GameCharacter()` class we had in assignment 2 through the use of three subclasses: `DogCharacter()`, `CatCharacter()`, and `AlienCharacter()`. All these characters are to have a default constructor, and unique constructors. In addition to the set, get, and modify method we created in assignment 2, we are to have a `makeSadNoise()`, which prints ":(" when called. Also, we have to have a `makeHappyNoise()` that calls "meow" for cats, "woof" for dogs, and "bzzrt" for aliens.

Pseudocodes

GameCharacter

For GameCharacter(), I made four changes:

- Changed class to abstract.
- Added `makeSadNoise()` and defined it.
- Added an abstract signature to `makeHappyNoise()`.
- Added default and special constructors.

```
Abstract Class GameCharacter

---Initialization variables for game character---

        ---Private---
        health = default 100
        happiness = default 100
        weight = default 100

        ---Public---
        none

---Constructors---

        1. Default values
        2. construct with health, happiness, weight set to defined values.

---Method overview---

        ---Private---
        isValid()
                This will check to see if changeAmount is valid.

        ---Public---
        modifyCharacteristic()
                Accepts an int changeAmount and String characteristic.
        getCharacteristic() (3 methods)
                Returns value of characteristic
        setCharacteristic() (3 methods)
                Sets the characteristic
        makeSadNoise()
                Prints a sad face

        ---Abstract---
        makeHappyNoise()
                Will print out the happy noise defined by subclass

---Method pseudocode---

        modifyCharacteristic():
                Set the characteristic string to all lowercase
                Check to see if the string is valid
                        If not, throw an exception
                Calls setCharacteristic() to modify characteristic

        getCharacteristic():
                Returns characteristic
        isValid():
                Add the characteristic to check validity
                        If impossible, it will throw an exception

        setCharacteristic():
                Sets the characteristic if the value is valid

        makeSadNoise():
                Will print out a sad face because the character is sad

        makeHappyNoise():
                Will print out animal's happy noise.
```

Cat, Dog, and Alien

```
---Classes---

CatCharacter extends GameCharacter
DogCharacter extends GameCharacter
AlienCharacter extends GameCharacter

---Methods---

All the above classes, in addition to extending the methods of GameCharacter, will include:

-Default constructor (will set Health, Happiness, and Weight to 100 by default)
-Special constructor that accepts custom values for Health, Happiness, Weight.
        This constructor will call the default constructor if the user inputs an invalid amount
for any one of the values.

-A makeHappyNoise that will print a string depending on class:
        Cats will print "Meow"
        Dogs will print "Woof"
        Aliens will print "Bzzrp"
```

Game Tester

```
**PRINT ALL ACTIONS AND COMPARE WITH EXPECTED VALUES**

Step 1) Test constructors

        make new dog, cat, and alien using default constructors
        make new dog using constructor with health 60 happiness 75 weight 90
        make new cat using constructor with health 110, happiness 50, weight 50 (Defaulting)
        make new alien using constructor with health -10, happiness 50, weight 50 (Defaulting)

Step 2) Checking these values.

        print the value of all the constructed instanced classes

Step 3) Changing values

        modify hapiness of dog by -50
        print happiness
        modify sadness by 50 (should throw)
        modify modify health by 100 (should throw)
        modify health by -100 (should throw)

Step 4) test noise

        call sad and happy noise for all types of subclasses
```

Code

GameCharacter

```java
abstract public class GameCharacter {

        //Variable declarations
        private int health, weight, happiness;

        //Constructors
        public GameCharacter() {
                this.health = 100;
                this.happiness = 100;
                this.weight = 100;
        }
        public GameCharacter(int health, int happiness, int weight) {
                this.health = health;
                this.happiness = happiness;
                this.weight = weight;
                //This will check and set defaults if invalid
                if ((0 > health || health > 100) ||
                        (0 > happiness || happiness > 100) ||
                        (0 > weight || weight > 100)) {
                                System.out.println("Invalid parameter");
                                this.health = 100;
                                this.happiness = 100;
                                this.weight = 100;
                }
        } //End constructors

        //This will check and set the characteristic
        public void modifyCharacteristic(int changeAmount, String characterist
ic) {
                //This will check and modify the characteristics, if valid.
                switch(characteristic.toLowerCase()) {
                case "health":
                        setHealth(changeAmount + health);
                        break;
                case "weight":
                        setWeight(changeAmount + weight);
                        break;
                case "happiness":
                        setHappiness(changeAmount + happiness);
                        break;
                default:
                        throw new IllegalArgumentException("Invalid char.");
                }
        } //End modifyCharacteristic()

        //These are the get functions
        public int getHealth() {
                return health;
        } //End getHealth()
        public int getWeight() {
                return weight;
        } //End getWeight()
        public int getHappiness() {
                return happiness;
        } //End getHappiness()

        //This are the set functions
        public void setHealth(int amount) {
                isValid(amount);
                this.health = amount;
        } //End setHealth()
        public void setWeight(int amount) {
                isValid(amount);
                this.weight = amount;
        } //End setWeight()
        public void setHappiness(int amount) {
                isValid(amount);
                this.happiness = amount;
```

```
	} //End setHappiness()

	//This will throw an exception if illegal change happens
	private boolean isValid(int amount) {
		if (0 <= (amount) && (amount) <= 100) {
			return true;
		} else {
			throw new IllegalArgumentException("Exceeds bounds.");
		}
	} //End isValid()

	//This will make a sad noise.
	public void makeSadNoise() {
		System.out.println(":(");
	} //End makeSadNoise()

	//A signature for makeHappyNoise()
	abstract public void makeHappyNoise();

} //End class GameCharacter
```

Subclasses

Dog

```
public class DogCharacter extends GameCharacter {

	//The constructors
	public DogCharacter() {
		super();
	}
	public DogCharacter(int health, int happiness, int weight) {
		super(health, happiness, weight);
	} //End constructors

	public void makeHappyNoise() {
		System.out.println("woof");
	} //End makeHappyNoise()

} //End DogCharacter()
```

Cat

```
public class CatCharacter extends GameCharacter {

	//The constructors.
	public CatCharacter() {
		super();
	}
	public CatCharacter(int health, int happiness, int weight) {
		super(health, happiness, weight);
	} //End constructors.

	public void makeHappyNoise() {
		System.out.println("meow");
	} //End makeHappyNoise()

} //End CatCharacter()
```

Alien

```java
public class AlienCharacter extends GameCharacter {

	//The constructors
	public AlienCharacter() {
		super();
	}
	public AlienCharacter(int health, int happiness, int weight) {
		super(health, happiness, weight);
	} //End constructors

	public void makeHappyNoise() {
		System.out.println("bzzrp");
	} //End makeHappyNoise()
}
```

GameCharacterTester2

```java
public class GameCharacterTester2 {

	public static void main(String[] args) {
		//====Step 1: Create instances====
		System.out.println("Creating new characters");
		DogCharacter dixie = new DogCharacter();
		CatCharacter nyan = new CatCharacter();
		AlienCharacter et = new AlienCharacter();
		DogCharacter dixie2 = new DogCharacter(50, 50, 50);
		System.out.println("Create an alien having NO FUN :(");
		AlienCharacter et2 = new AlienCharacter(50, -100, 50);
		System.out.println("Creating an overweight cat");
		CatCharacter garfield = new CatCharacter(50, 50, 1000);
		System.out.println("=====");

		//====Step 2: Check values====
		System.out.println("Printing 5 default characters:");
		System.out.println("Health: " + dixie.getHealth() +
				"| Happiness: " + dixie.getHappiness() +
				" | Weight: " + dixie.getWeight() );
		System.out.println("Health: " + nyan.getHealth() +
				" | Happiness: " + nyan.getHappiness() +
				" | Weight: " + nyan.getWeight());
		System.out.println("Health: " + et.getHealth() +
				" | Happiness: " + et.getHappiness() +
				" | Weight: " + et.getWeight());
		System.out.println("Health: " + et2.getHealth() +
				" | Happiness: " + et2.getHappiness() +
				" | Weight: " + et2.getWeight());
		System.out.println("Health: " + garfield.getHealth() +
				" | Happiness: " + garfield.getHappiness() +
				" | Weight: " + garfield.getWeight());
		System.out.println("Printing a custom character:");
		System.out.println("Health: " + dixie2.getHealth() +
				" | Happiness: " + dixie2.getHappiness() +
				" | Weight: " + dixie2.getWeight());
		System.out.println("=====");

		//====Step 3: Changing values====
		System.out.println("Changing character's values to all 0");
		dixie.setHealth(0);
		dixie.setHappiness(0);
		dixie.setWeight(0);
		System.out.println("Health: " + dixie.getHealth() +
				"| Happiness: " + dixie.getHappiness() +
				" | Weight: " + dixie.getWeight() );
		System.out.println("Modifying 100 and try to modify sadness");
		dixie.modifyCharacteristic(100, "HealTh");
		System.out.println("Health: " + dixie.getHealth());
```

```java
                try {
                        dixie.modifyCharacteristic(10, "sadNess");
                } catch (Exception e) {
                        System.out.println(e.getMessage());
                }
                System.out.println("Trying to set health to 1000 and -1");
                try {
                        dixie.setHealth(1000);
                } catch (Exception e) {
                        System.out.println(e.getMessage());
                } try {
                        dixie.setHealth(-1);
                } catch (Exception e) {
                        System.out.println(e.getMessage());
                }
                System.out.println("=====");

                //====Step 4: Testing noises====
                System.out.println("Dogs:");
                dixie.makeSadNoise();
                dixie.makeHappyNoise();
                System.out.println("Cats: ");
                nyan.makeSadNoise();
                nyan.makeHappyNoise();
                System.out.println("Aliens: ");
                et.makeSadNoise();
                et.makeHappyNoise();

        } //End main()
} //End class GameCharacterTester
```

Output of GameCharacterTester2

```
Creating new characters
Create an alien having NO FUN :(
Invalid parameter, using defaults.
Creating an overweight cat
Invalid parameter, using defaults.
=====
Printing 5 default characters:
Health: 100| Happiness: 100 | Weight: 100
Health: 100 | Happiness: 100 | Weight: 100
Health: 100 | Happiness: 100 | Weight: 100
Health: 100 | Happiness: 100 | Weight: 100
Health: 100 | Happiness: 100 | Weight: 100
Printing a custom constructed character:
Health: 50 | Happiness: 50 | Weight: 50
=====
Changing character's values to all 0
Health: 0| Happiness: 0 | Weight: 0
Modifying to 100 and try to modify sadness
Health: 100
Invalid char.
Trying to set health to 1000 and -1
Exceeds bounds.
Exceeds bounds.
=====
Dogs:
:(
woof
Cats:
:(
meow
Aliens:
:(
bzzrp
```