Assignment 2: DC Universe

Specifics

Today's Date:	**Monday, January 27, 2014**
Due Date:	Friday, February 7, 2014 @ 5pm
Type:	Individual Assignment
Value:	30 marks

This assignment is based on the material from Inheritance, Abstract Classes, and Interfaces. It is worth 30 marks. This assignment is to be completed individually. You are not allowed to work on it with anyone.

Design the characters for an MMORPG

MMORPG means "massively multiplayer online role-playing game": a video game played online which involves players from every country participating in a virtual world. Every player has an "avatar": a character which represents the player.

The DC Universe is the comic book universe which includes famous superheroes such as Batman, Superman, and Wonder Woman.

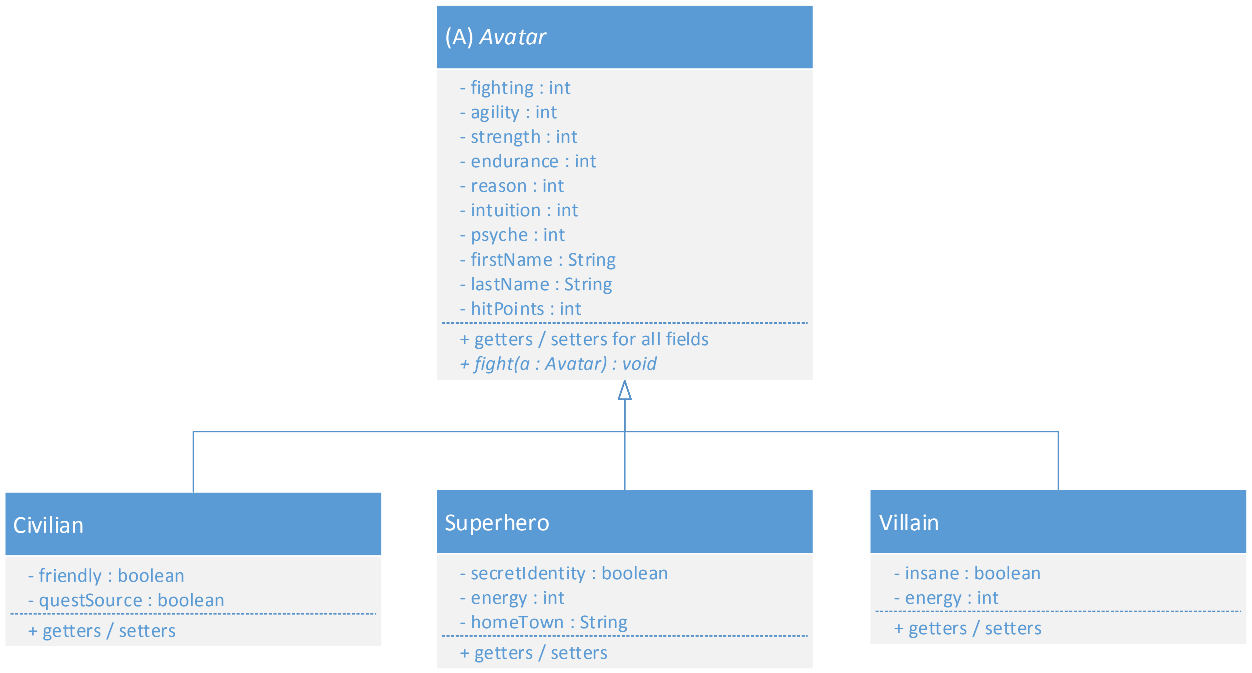
In this assignment, you will use classes, inheritance, abstract class(es), and interfaces to design the hero characters which will become a player's avatars. You are not designing gameplay: you are just modeling.

Part 1: Class design

Every character in the online universe will be a type of Avatar. Avatar is an abstract class (note the "(A)" in the class name). There will be three basic types of Avatars: Superheroes, Villains, and Civilians.

Create a class hierarchy which supports this design. Build all classes in the **model** package.

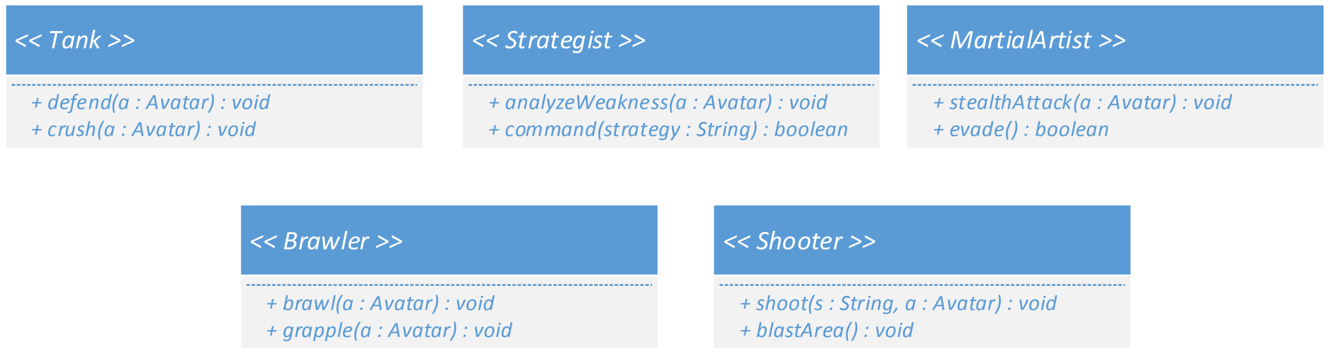
Remember: abstract classes and methods are always written in *italics*.



Part 2: Interface design

Every hero in the DC Comics universe is a particular type. There are five types of heroes: Tanks, Brawlers, Shooters, Strategists, and MartialArtists.

Design these five types as interfaces based upon the following UML models. Build all interfaces in the **model.interfaces** package.

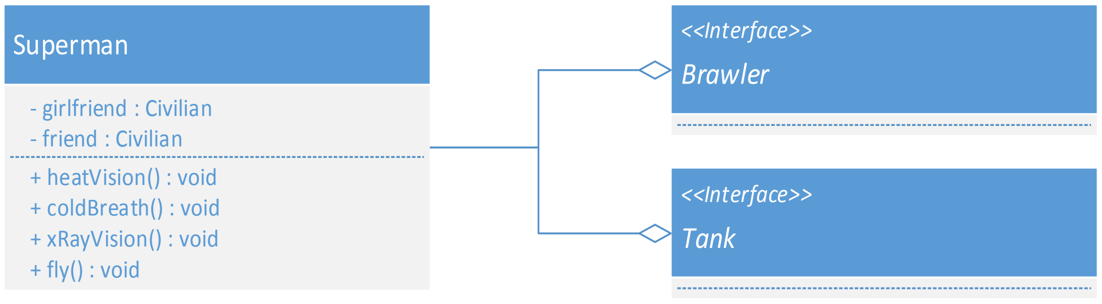

<< Tank >>
+ defend(a : Avatar) : void
+ crush(a : Avatar) : void
<< Strategist >>
+ analyzeWeakness(a : Avatar) : void
+ command(strategy : String) : boolean
<< MartialArtist >>
+ stealthAttack(a : Avatar) : void
+ evade() : boolean
<< Brawler >>
+ brawl(a : Avatar) : void
+ grapple(a : Avatar) : void
<< Shooter >>
+ shoot(s : String, a : Avatar) : void
+ blastArea() : void

Part 3: Design heroes

Extend the Superhero class to create five of the most famous heroes in the DC Universe. Build all heroes in the **model.hero** package:

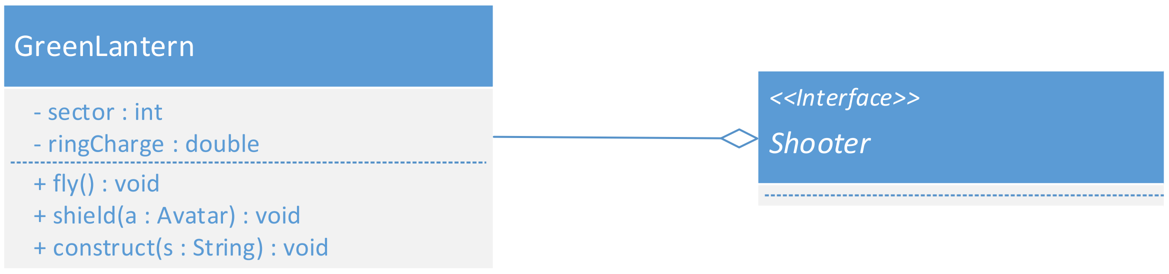
Superman

The Man of Steel has a girlfriend (Lois Lane) and a friend (Jimmy Olsen).



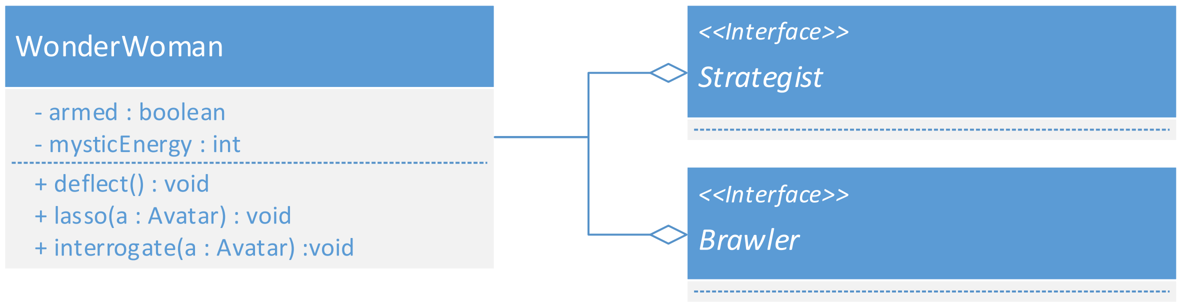
Green Lantern

Green Lantern is part of a corps of interstellar peace officers – "space cops" – known as The Green Lantern Corps. Each Green Lantern patrols a particular sector of space and can create energy constructs with a special power ring. The power ring contains a charge and must be recharged regularly.



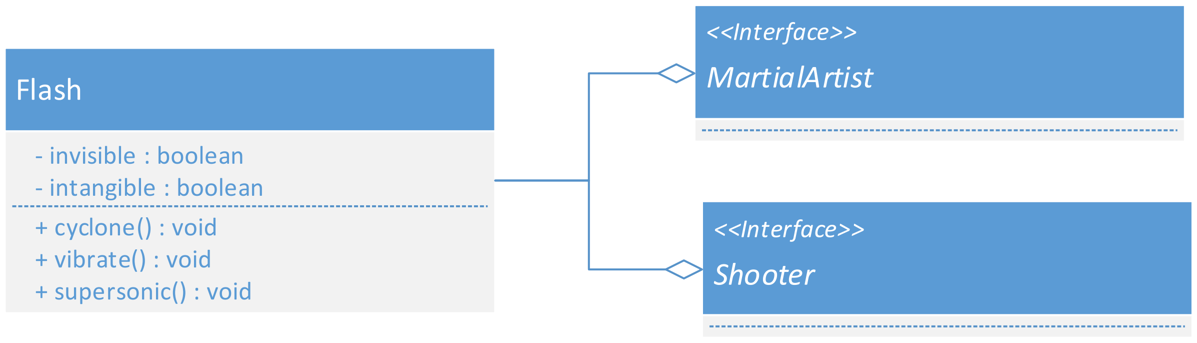
Wonder Woman

Diana is a warrior princess from the land of Themyscira. She is the leader, chief diplomat, and representative of the Amazons. She is sometimes armed with a sword or shield, and almost always carries her magic Lasso of Truth, which helps her interrogate subjects. She deflects weapons with her indestructible magic bracelets.



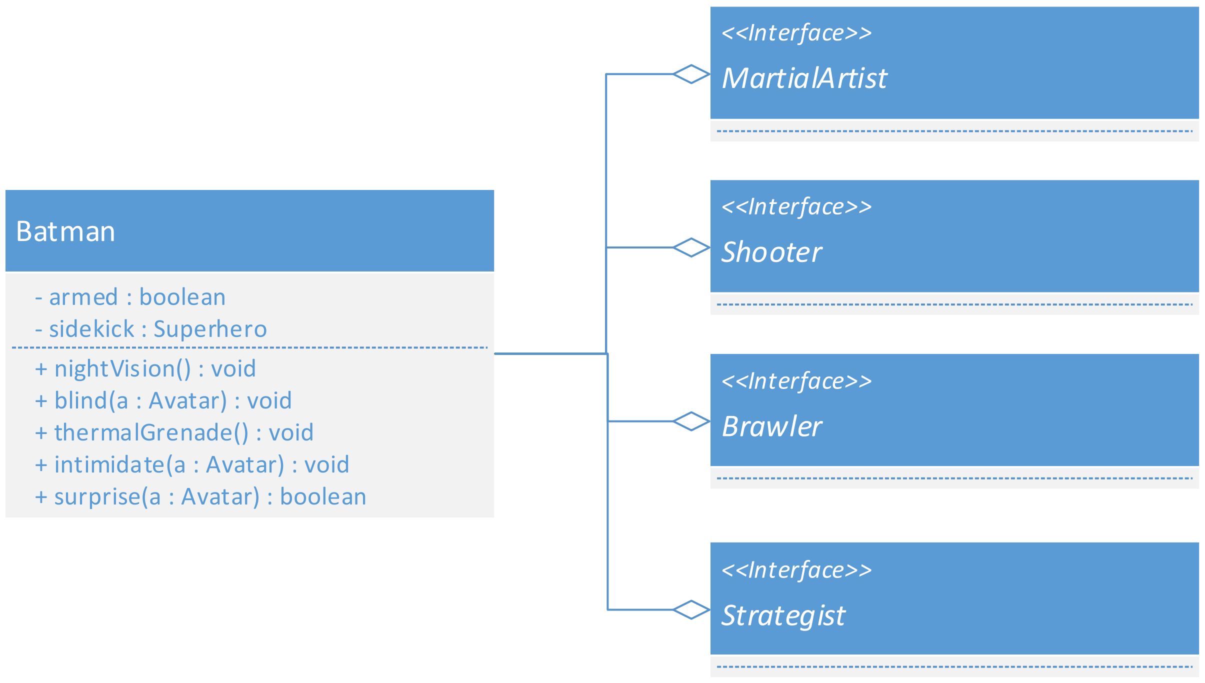
Flash

Flash moves at superhuman speed. Although he is not an accomplished martial artist, he moves so fast that his attacks are always unexpected and he is able to evade most attacks. He does not "shoot" any weapon, either, but because of his super speed, he is able to close the distance between him and his target at almost lightspeed – faster than any bullet can travel – and create cyclones which will affect an area of people. He travels so fast, he can be invisible or intangible.



Batman

The dark knight detective is one of the most fearsome – and famous – characters in the DC Universe. He is an accomplished martial artist and combatant with a huge arsenal of devices on his utility belt. He always has a strategy for winning. Sometimes, he is accompanied by his sidekicks, Robin or Batgirl.



Assumptions and Hints

- Abstract classes and methods are always modeled with *italics*.
- For each method, all that is required is to print an appropriate message to the console saying what each hero does. Each message should include the class and method. For example:

```
System.out.println("Flash evades! >>> So fast, can’t hit him");
System.out.println("Green Lantern blasts the area!");
```

- Some methods take an Avatar or type of Avatar as an argument. In this case, you may choose another hero to be the argument.

  ```
  System.out.println("Superman crushes " + avatarName);
  ```

- You must implement the fight(a : Avatar) method in each specific hero class. Each hero fights must a different way (ie, use a different print string).
- Include a constructor which sets the basic properties of an Avatar.

Part 4: Gameplay

Before the gameplay will be built, you must demonstrate to the design team that all five heroes work and can execute their "powers" (ie, methods), even those that are inherited, implemented, or overridden.

Create a Gameplay class in the **launch** package which includes a main() method. This is the only class in your project which should contain a main() method. In the method, create an instance of each hero (you may add them in to an array if you wish). Initialize each hero with the constructor, getters and setters. Execute all of the heroes' methods – even those that are inherited, implemented, or overridden.

- You must use Javadoc for your classes and methods. Use at least 2 @ tags per class, and at least 1 @ tag per method.
- Use comments as appropriate. Many methods do not need comments.

Part 5: Submission

Zip up everything from the project folder. Click **File > Export Project > To Zip**. Submit the project as a zip file named as your first and last name, e.g., **matsswan.zip**

You must properly comment your code and write Javadoc. You do not have to generate the Javadoc for this assignment. However, I must be able to generate the Javadoc without errors.

Your assignment's source code must be completed according to the department submission requirements and be submitted on or before the beginning of class on the due date by **placing it in the Slate2 dropbox for Assignment 2.**

Do not e-mail me your submissions. If your assignment is late, contact me for a dropbox extension. Do not e-mail me your submissions.

Evaluation

You'll be graded based on the following criteria:

- **Functionality**: program functions according to specifications
- **Efficiency**: uses variables where and only when necessary; program doesn't define variables that are never used, nor does it use too many variables for unnecessary tasks; program logic is written concisely and is not cluttered with unnecessary tasks.
- **Style**: proper indentation and spacing, use of comments/documentation; all identifiers are descriptive and valid; variables are defined with appropriate types and converted when required.
- **Discretionary**: all instructions regarding submissions and program specifications have been followed; submission was completed and submitted as requested in a timely fashion; techniques discussed in class have been used.

FAQ

If you have any questions, please post them to the Slate2 discussion board. If you send me questions, I will ask you to place them on the discussion board.