CS136 Lab Section 1

Joshua Pollock

21 March 2017

**Lab 6 - RPG**

**Problem Statement**

In this lab, we will be required to create a roleplaying game. This lab will be using inheritance and interfaces to complete the lab requirements.

**Constraints:**

1. The arena should contain two of each character type
2. The three character types are: Mage, Warrior, and Beserker
3. Dead contestants should be described as dead
4. When taking damage, the amount of damage and who took damage should be displayed
5. The winner should be picked if they are the last man standing
6. The beserker: deals 20 smashing damage, may hit themselves, have 35 health, take ½ damage from fire and double from slashing
7. The warrior: targets the person with the largest amount of health, never attack themselves, deal 10-16 slashing damage, have 20-40 health, and have a 25% chance to dodge incoming slashes and smashes
8. The mage: deals 0 to 6 fire damage to everyone in the arena (including themselves), 10-60 health, when killed they deal 5 fire damage to the attacker

**Assumptions:**

1. The player class may be an interface
2. The players will attack in the turn: beserker=>mage=>warrior

**Features:**

* new Beserker(“”)
* new Mage(“”)
* new Warrior(“”)
* .isAlive()
* .getHP()
* .getType()
* .getName()

**Planning:**

We first started out this lab by creating multiple UML diagrams for our classes. We also decided that the Player class would be an interface instead of being extended from. The UMLs that we concluded on were:

|  |
| --- |
| Arena |
| Main(args: String[]) void |

|  |
| --- |
| Beserker |
| Beserker(name:String)  doDMG() Double  isAlive() boolean  takeDamage(type:String,amount:double) void  getName() String  getHP() Double  getType() String |

|  |
| --- |
| Mage |
| Mage(name:String)  doDMG() Double  isAlive() boolean  takeDamage(type:String,amount:double) void  getName() String  getHP() Double  getType() String |

|  |
| --- |
| Interface Player |
| isAlive(); boolean  takeDamage(String type,double amount); void  doDMG(); Double  getName(); String  getHP(); Double  getType(); String |

|  |
| --- |
| Warrior |
| Warrior(name:String)  doDMG() Double  isAlive() boolean  takeDamage(type:String,amount:double) void  getName() String  getHP() Double  getType() String |

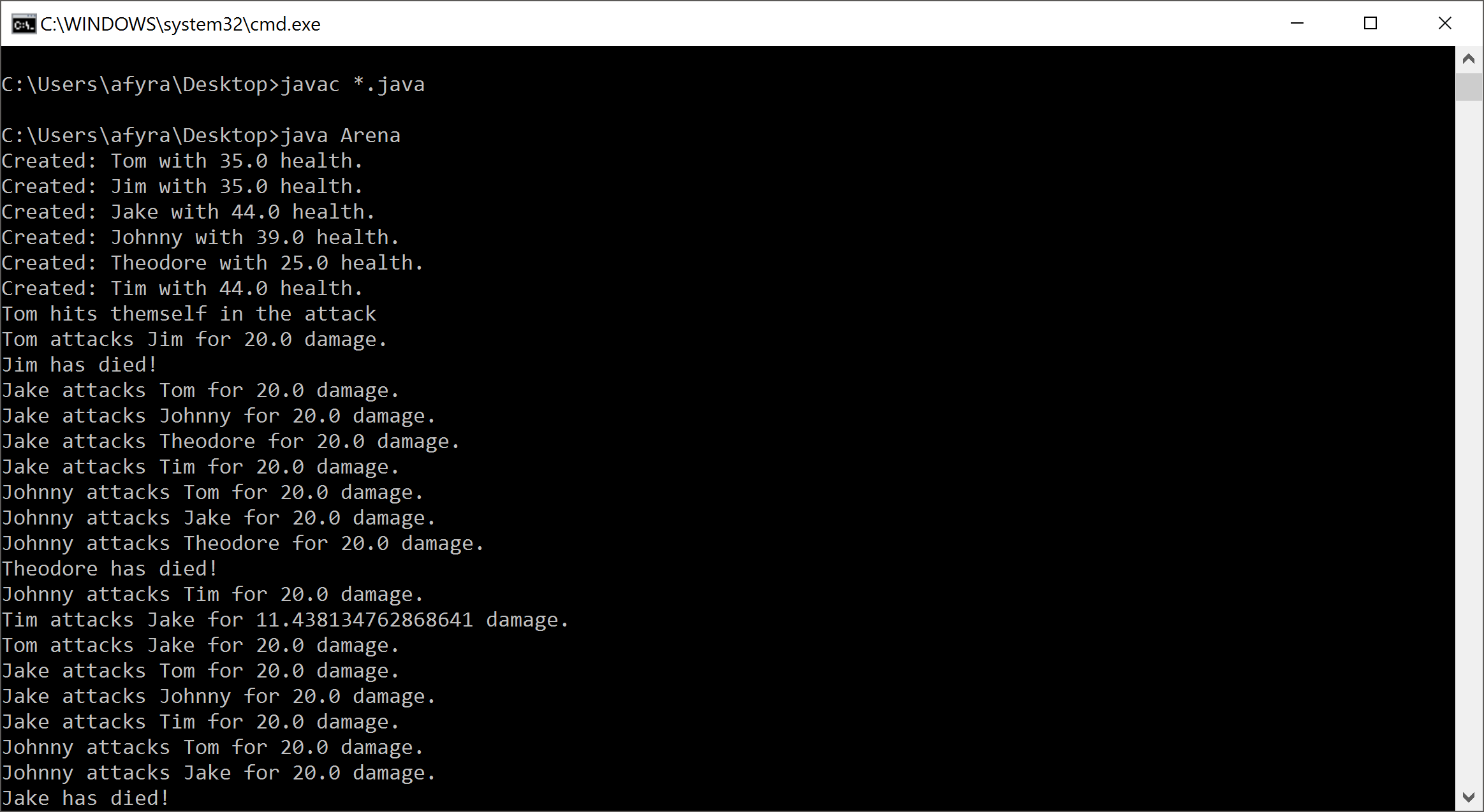
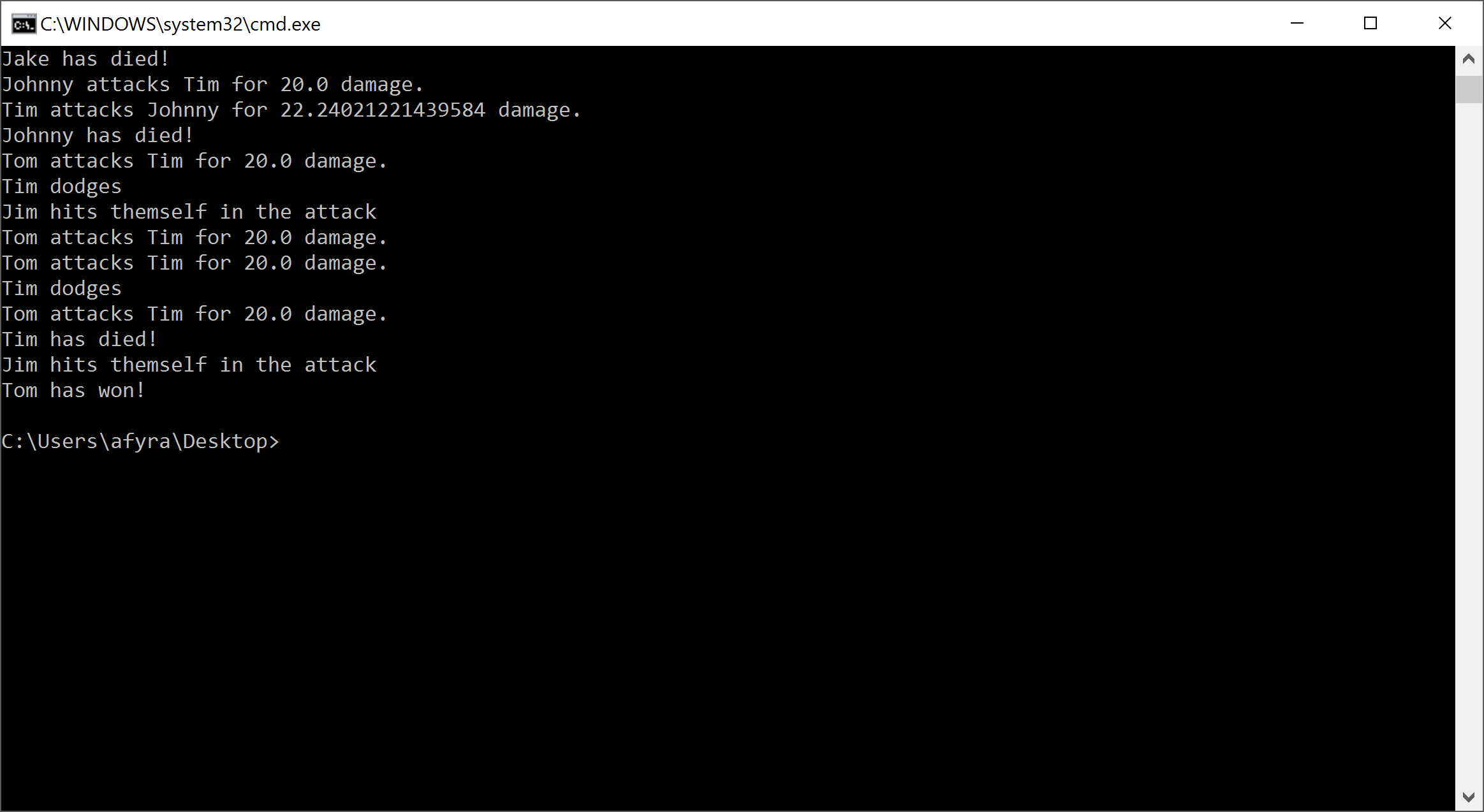
We decided against using an arraylist to store characters in. We believed it would take far too much time and effort to implement them in an array.

**Implementation:**

­­­­

Implementing this code was quite easy. We used the Python RPG lab as a skeleton for this, and easily decided on what variables and methods we would need to create. Using the Player interface, we knew what methods we needed to implement into each class. We changed the takeDamage and doDMG methods based on the lab constraints for each player.

**Running Application:**

Our solution compiles and runs as it is supposed to. It seems that the warriors have an advantage and win at a high percentage rate. The code is quite long and will not be screenshotted here, as it would take at least 10 pages.

**Reflection & Refactor:**

Overall, we are quite satisfied with our code. It satisfies all requirements and constraints given by the lab. The Arena class could easily be condensed down into separate methods. The current solution is quite long and repetitive. We had a bit of an issue trying to get the attacks to work properly. This was quickly sorted out by checking to see if the attacker was alive before attacking. More changes included adding in the checker to see if a mage dies once attacked. This turned out to be a simple if statement. The rest of the code was simply copy and pasting the attacks and changing a few words here and there. We are quite happy with the code and see very little to improve.