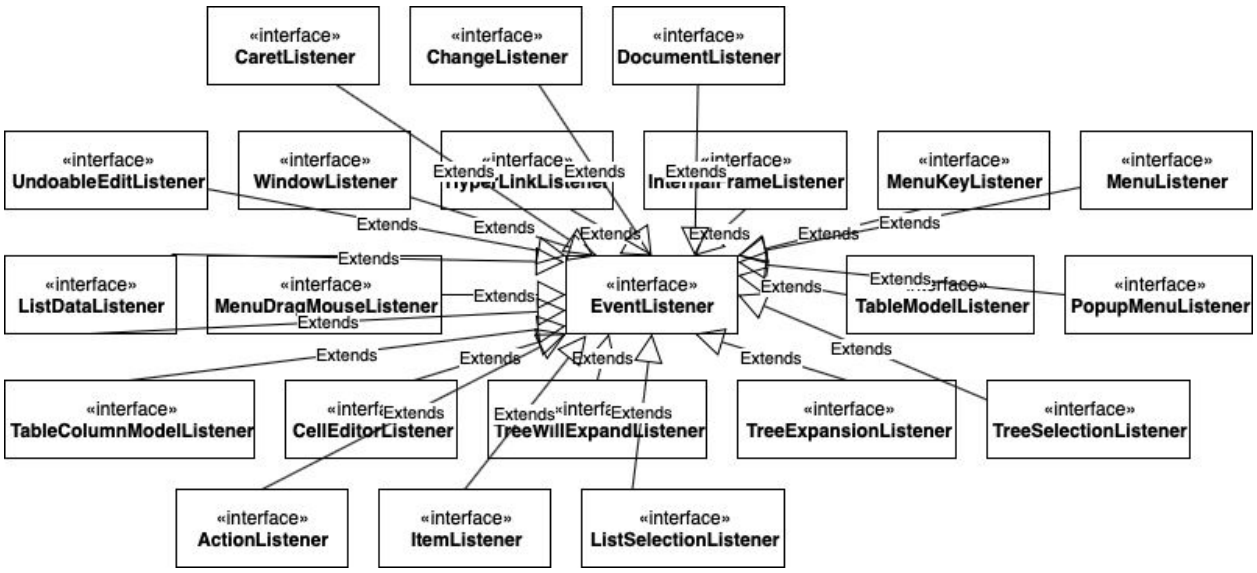
Tabassum Fabiha
tf2478

1.1 - Proxy Pattern

- A Social Security Number is a proxy for a person, as it represents the identity of a person and is often used as a stand in for online or paper forms as a substitute of the individual themselves.
- Java constructs
    - Rectangle rect = new Rectangle(0, 0, 5, 5);
        - // rect is a proxy
    - ArrayList<Integer> nums = new ArrayList<Integer>();
        - // nums is a proxy
    - String str = “hello”;
        - // str is a proxy
    - i = 0;
      j = 5;
      if (i >= 0 || j < 10)
      i += 1;
        - // if (i >= 0) is a proxy

1.2 - Observer Pattern



- The diagram is wide, which means that all the listeners must be inherently unique, and there must not be overlap between them.

1.3 - Visitor Pattern

- When you visit a nail salon to get a manicure you give control of hands to a manicurist.
    - You’re hands are protected from being harmed as the manicurist will want to provide a good enough service so that you will be a returning customer.

```
public class NailSalon {
        public static void main( String[] args ) {
                Person me = new Person();
                me.pay( new Manicurist() );
        }
}

public interface Manicure
{
   public void doNails(Person p);
}

public class Person {
        public void pay(Manicurist m) {
                m.doNails(this);
        }

        String nailStyle = “oval”;
}

public class Manicurists implements Manicure {
        public void doNails(Person p) {
                p.nailStyle = “almond”;
        }
}
```

1.4 - Shape Interface

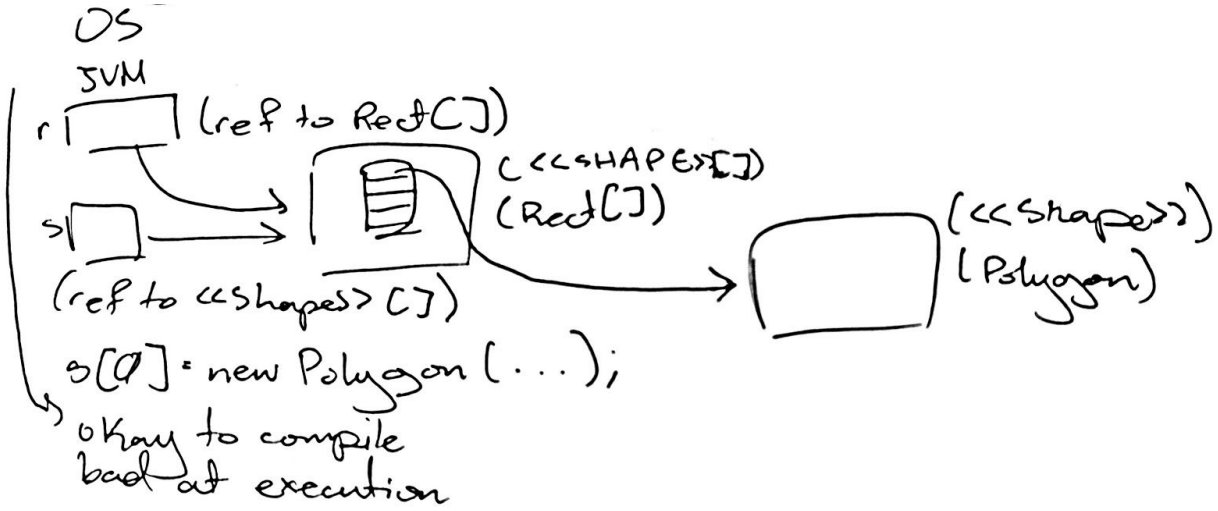
- RectangleShape (all subclasses)
    - Arc2D
        - Arc2D.Double
        - Arc2D.Float
    - Ellipse2D
        - Ellipse2D.Double
        - Ellipse2D.Float
    - Rectangle2D
        - Rectangle
            - DefaultCaret
                - BasicTextUI.BasicCaret
        - Rectangle2D.Double
        - Rectangle2D.Float
    - RoundRectangle2D

        - RoundRectangle2D.Double
        - RoundRectangle2D.Float
- The Rectangle class is unusual in that its instance variables: x, y, height, and width, are all public and can be changed directly without needing to invoke a method.

1.5 - Tagging Interfaces

- Tagging Interfaces
    - Cloneable (java.lang)
        - This interface is used as a flag to indicate that the method clone() has been overwritten for the class that implements it.
    - Remote (java.rmi)
        - This interface is used as a flag to indicate that any interface that extends this has methods that can be remotely accessed.
    - EventListener (java.util)
        - This interface is used as a flag to indicate that any interface that extends it will be an event listener.

1.6 - Reflection



- At compilation, the compiler checks line by line to make sure the syntax is appropriate, and for all the lines, the types and the values of those types match for each line. However at execution the JVM finds that it is trying to place a Polygon inside of an array of Rectangles, which is not possible because A Polygon does not have all the abilities to be a Rectangle.
- Error:
    - Exception in thread "main" java.lang.ArrayStoreException: java.awt.Polygon
    - at Tester.main(Tester.java:8)

1.7 - Hashcode

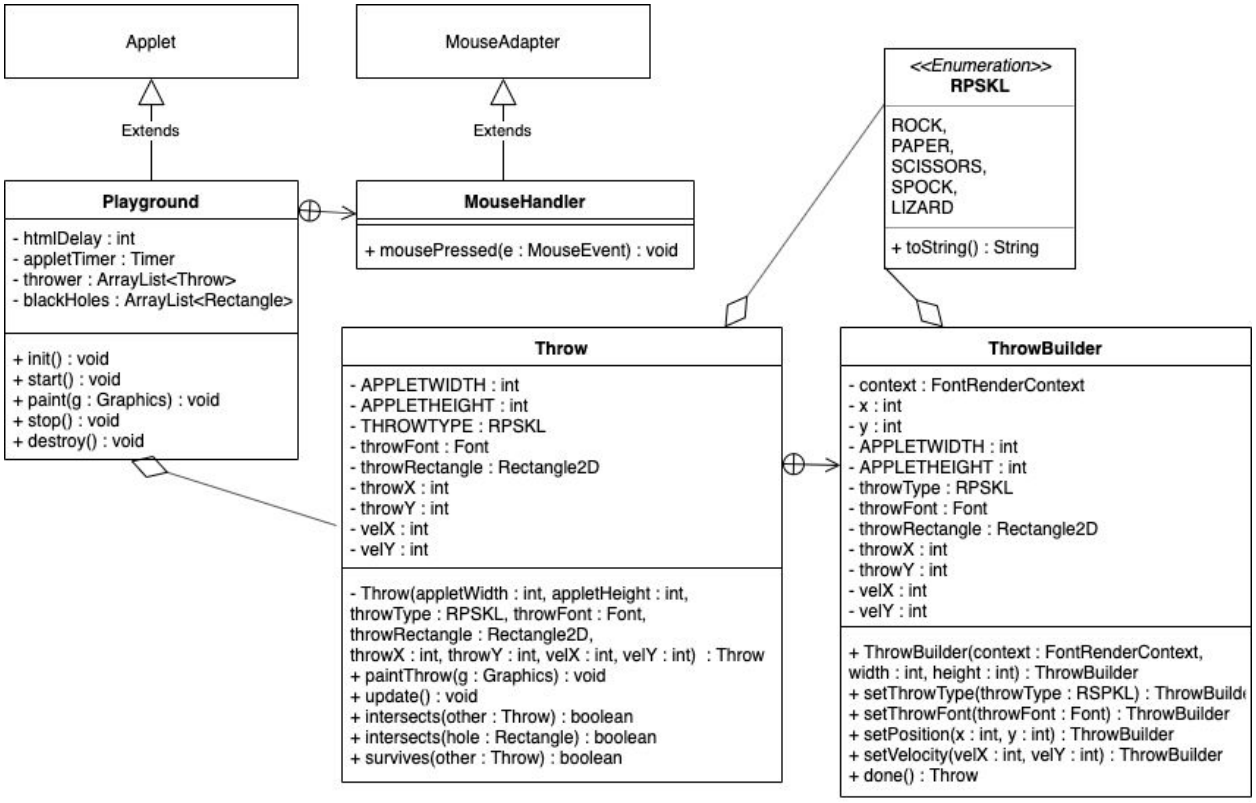
```
private int[] nums = new int[6];
```

```
// nums is filled with values satisfying the invariant
public int hashcode() {
        int hash = nums[0] * nums[1] * nums[2] * nums[3] * nums[4] * nums[5] / 32
        // need to divide by 32 because 2^31 is slightly too small to be able to map 16
        // products per int
}
myPowerBall.nums // {19, 2, 4, 8, 1, 1}
yourPowerBall.nums // {7, 11, 2, 4, 1, 2}

myPowerBall.hashcode() // 38
yourPowerBall.hashcode() // 38
Therefore myPowerBall.hashcode() == yourPowerBall.hashcode()

However myPowerball.equals(yourPowerBall) is false because the equals method will
compare each individual element in myPowerball.nums to the element in the same
placement in yourPowerball.nums
myPowerBall.nums[0] // 19
yourPowerBall.nums[0] // 7
Therefore myPowerball.equals(yourPowerBall) is false even though the hashcodes of
both are the same
```

1.8 - UML


Applet
MouseAdapter
Extends
Extends
<<Enumeration>>
RPSKL
ROCK,
PAPER,
SCISSORS,
SPOCK,
LIZARD
+ toString() : String
Playground
- htmlDelay : int
- appletTimer : Timer
- thrower : ArrayList<Throw>
- blackHoles : ArrayList<Rectangle>
+ init() : void
+ start() : void
+ paint(g : Graphics) : void
+ stop() : void
+ destroy() : void
MouseHandler
+ mousePressed(e : MouseEvent) : void
Throw
- APPLETWIDTH : int
- APPLETHEIGHT : int
- THROWTYPE : RPSKL
- throwFont : Font
- throwRectangle : Rectangle2D
- throwX : int
- throwY : int
- velX : int
- velY : int
- Throw(appletWidth : int, appletHeight : int, throwType : RPSKL, throwFont : Font, throwRectangle : Rectangle2D, throwX : int, throwY : int, velX : int, velY : int) : Throw
+ paintThrow(g : Graphics) : void
+ update() : void
+ intersects(other : Throw) : boolean
+ intersects(hole : Rectangle) : boolean
+ survives(other : Throw) : boolean
ThrowBuilder
- context : FontRenderContext
- x : int
- y : int
- APPLETWIDTH : int
- APPLETHEIGHT : int
- throwType : RPSKL
- throwFont : Font
- throwRectangle : Rectangle2D
- throwX : int
- throwY : int
- velX : int
- velY : int
+ ThrowBuilder(context : FontRenderContext, width : int, height : int) : ThrowBuilder
+ setThrowType(throwType : RSPKL) : ThrowBuild
+ setThrowFont(throwFont : Font) : ThrowBuilder
+ setPosition(x : int, y : int) : ThrowBuilder
+ setVelocity(velX : int, velY : int) : ThrowBuilder
+ done() : Throw