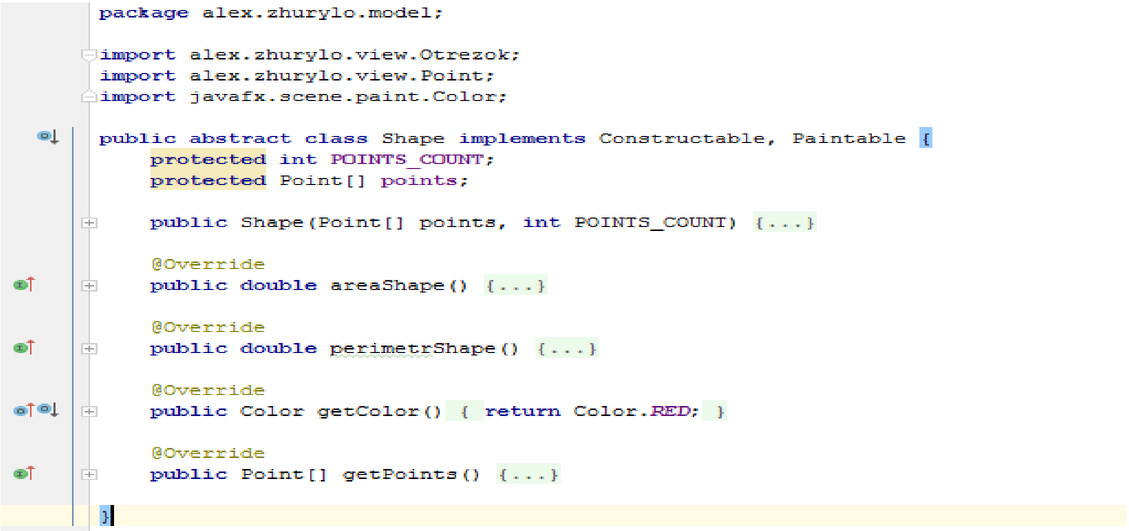
This file provides examples of the use of the principles of OOP, and also contains links to the corresponding code in the repository.

Principle 1 Inheritance

Example 1. Project Shape

Code Description 1

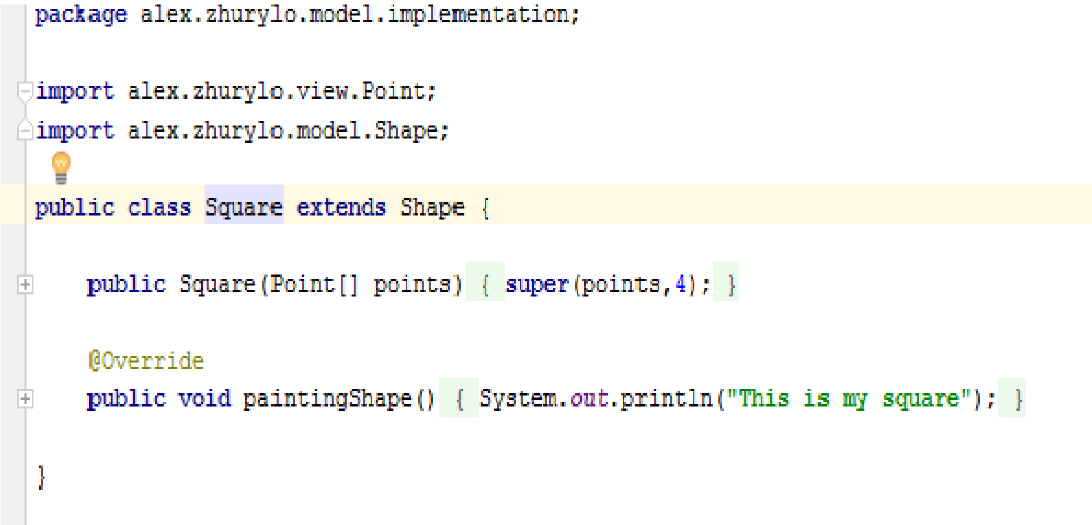
https://github.com/sasha96/InterLink/blob/master/Shape/src/alex/zhurylo/model/Shape.java



```
package alex.zhurylo.model;

import alex.zhurylo.view.Otrezok;
import alex.zhurylo.view.Point;
import javafx.scene.paint.Color;

public abstract class Shape implements Constructable, Paintable {
    protected int POINTS_COUNT;
    protected Point[] points;

    public Shape(Point[] points, int POINTS_COUNT) {...}

    @Override
    public double areaShape() {...}

    @Override
    public double perimetrShape() {...}

    @Override
    public Color getColor() { return Color.RED; }

    @Override
    public Point[] getPoints() {...}

}
```

.

https://github.com/sasha96/InterLink/blob/master/Shape/src/alex/zhurylo/model/implementation/Square.java

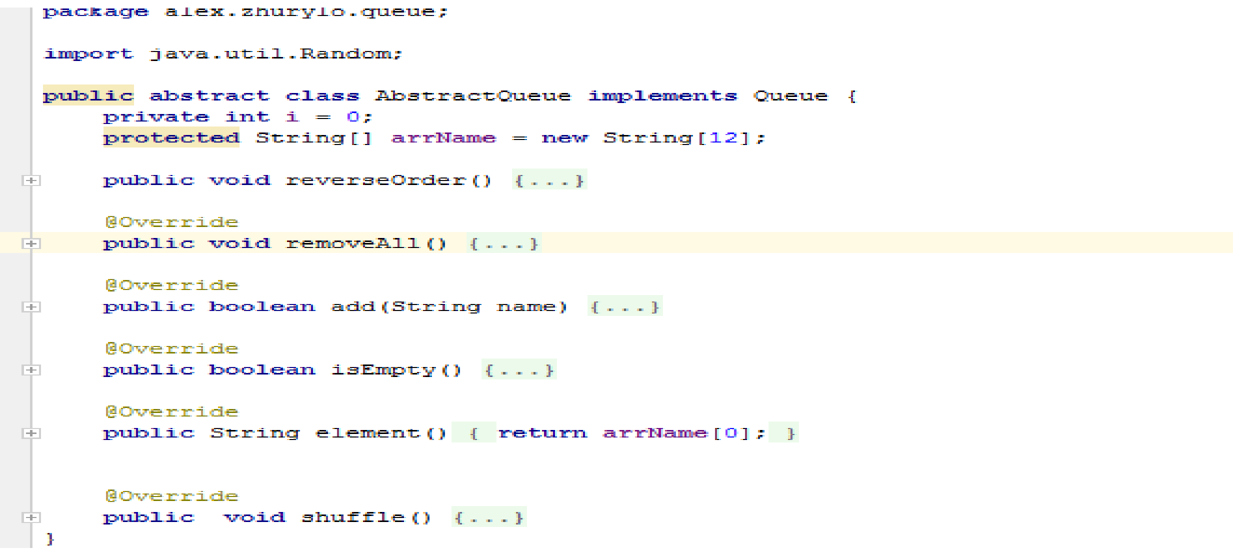


```
package alex.zhurylo.model.implementation;

import alex.zhurylo.view.Point;
import alex.zhurylo.model.Shape;

public class Square extends Shape {

    public Square(Point[] points) { super(points,4); }

    @Override
    public void paintingShape() { System.out.println("This is my square"); }

}
```

Example 2. Project Queue

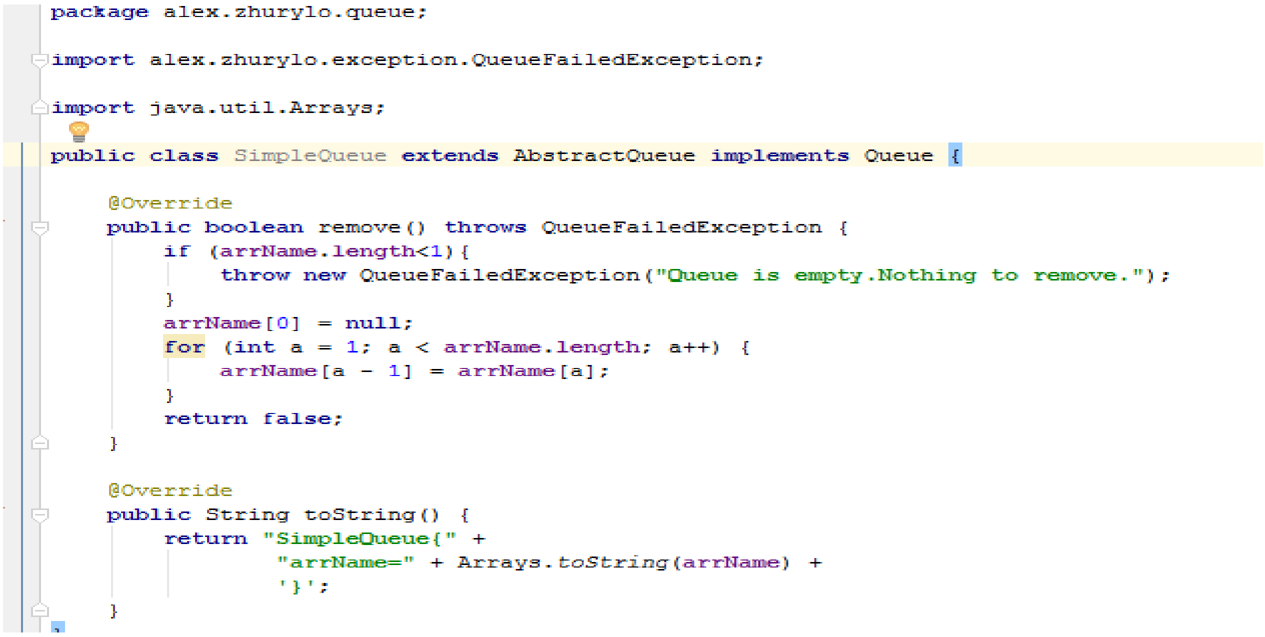
Code Description 2

https://github.com/sasha96/InterLink/blob/master/Queue/src/alex/zhurylo/queue/AbstractQueue.java



```
package alex.zhurylo.queue;

import java.util.Random;

public abstract class AbstractQueue implements Queue {
    private int i = 0;
    protected String[] arrName = new String[12];

    public void reverseOrder() {...}

    @Override
    public void removeAll() {...}

    @Override
    public boolean add(String name) {...}

    @Override
    public boolean isEmpty() {...}

    @Override
    public String element() { return arrName[0]; }


    @Override
    public  void shuffle() {...}
}
```

https://github.com/sasha96/InterLink/blob/master/Queue/src/alex/zhurylo/queue/SimpleQueue.java



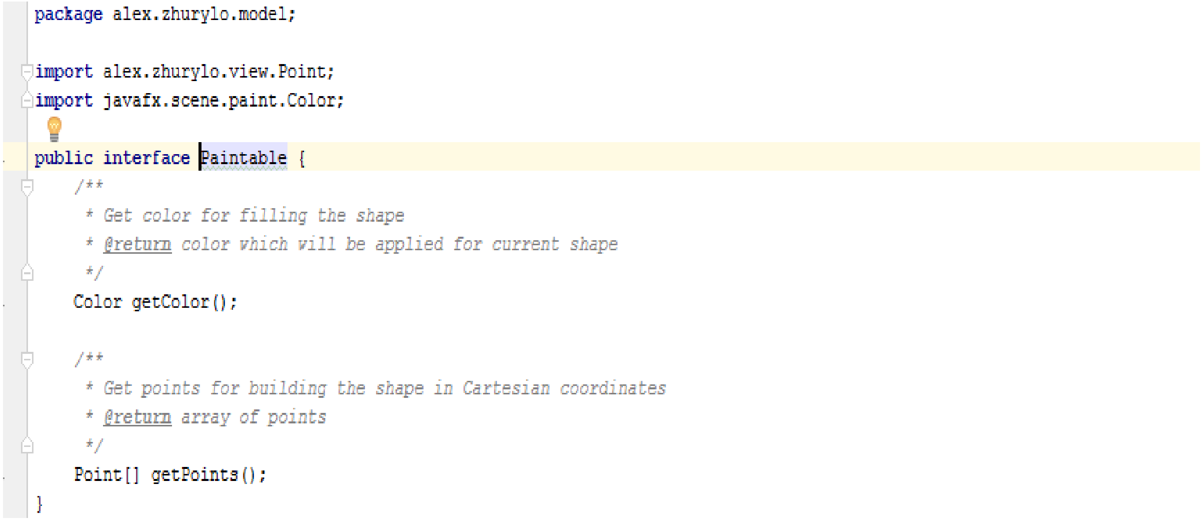
```
package alex.zhurylo.queue;

import alex.zhurylo.exception.QueueFailedException;

import java.util.Arrays;

public class SimpleQueue extends AbstractQueue implements Queue {

    @Override
    public boolean remove() throws QueueFailedException {
        if (arrName.length<1){
            throw new QueueFailedException("Queue is empty.Nothing to remove.");
        }
        arrName[0] = null;
        for (int a = 1; a < arrName.length; a++) {
            arrName[a - 1] = arrName[a];
        }
        return false;
    }

    @Override
    public String toString() {
        return "SimpleQueue{" +
                "arrName=" + Arrays.toString(arrName) +
                '}';
    }
}
```

Principle 2 Polymorphism

Example 1. Project Shape

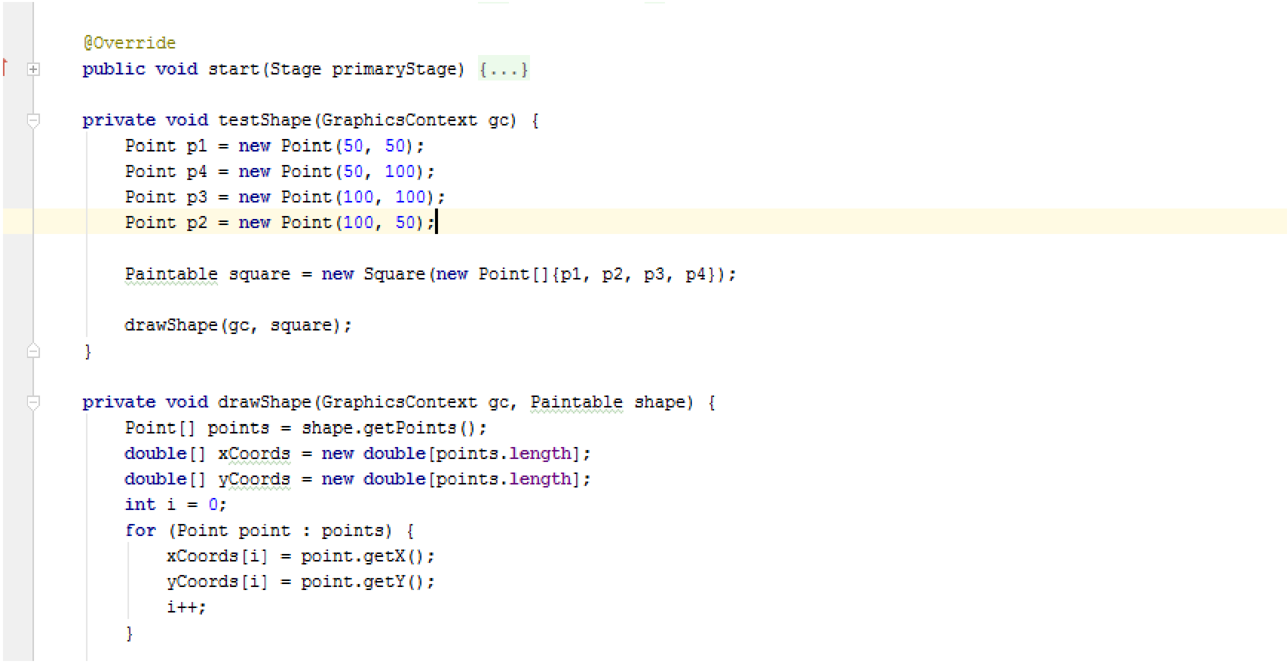
Code Description 1

https://github.com/sasha96/InterLink/blob/master/Shape/src/alex/zhurylo/model/Paintable.java



```java
package alex.zhurylo.model;

import alex.zhurylo.view.Point;
import javafx.scene.paint.Color;

public interface Paintable {
    /**
     * Get color for filling the shape
     * @return color which will be applied for current shape
     */
    Color getColor();

    /**
     * Get points for building the shape in Cartesian coordinates
     * @return array of points
     */
    Point[] getPoints();
}
```

https://github.com/sasha96/InterLink/blob/master/Shape/src/alex/zhurylo/view/Visual.java

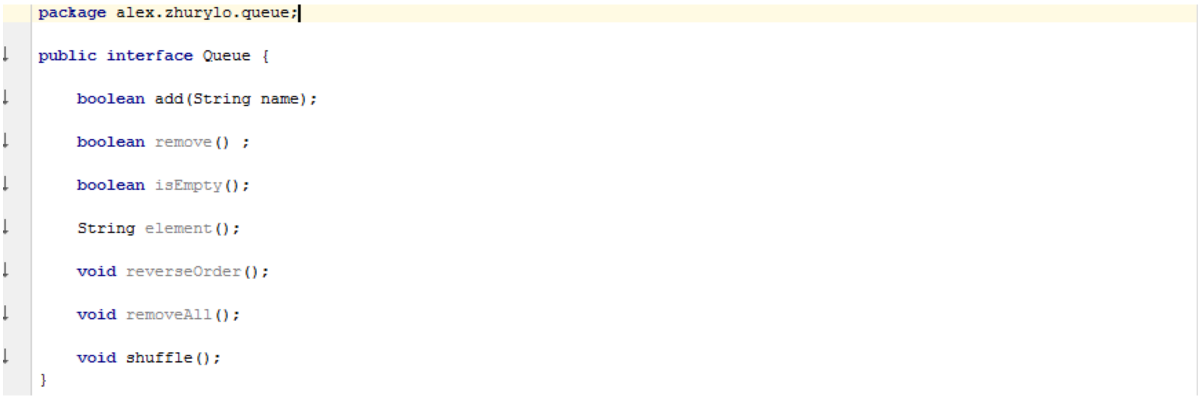


```java
    @Override
    public void start(Stage primaryStage) {...}

    private void testShape(GraphicsContext gc) {
        Point p1 = new Point(50, 50);
        Point p4 = new Point(50, 100);
        Point p3 = new Point(100, 100);
        Point p2 = new Point(100, 50);

        Paintable square = new Square(new Point[]{p1, p2, p3, p4});

        drawShape(gc, square);
    }

    private void drawShape(GraphicsContext gc, Paintable shape) {
        Point[] points = shape.getPoints();
        double[] xCoords = new double[points.length];
        double[] yCoords = new double[points.length];
        int i = 0;
        for (Point point : points) {
            xCoords[i] = point.getX();
            yCoords[i] = point.getY();
            i++;
        }
```

Example 2. Project Queue

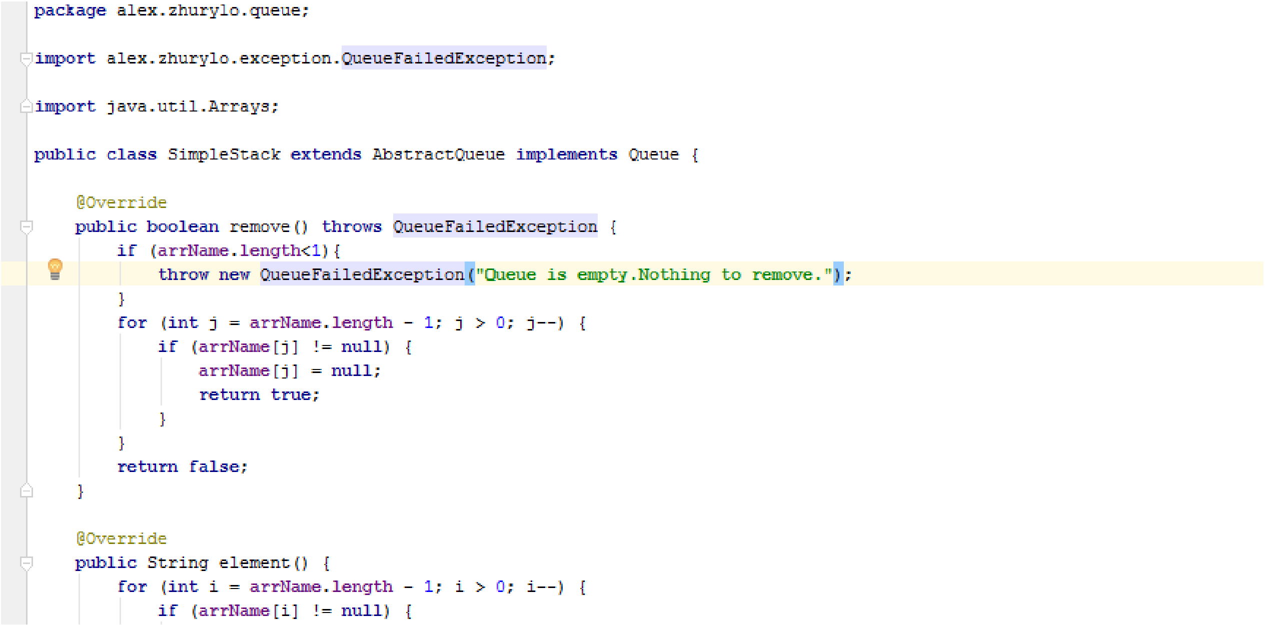
Code Description 2

https://github.com/sasha96/InterLink/blob/master/Queue/src/alex/zhurylo/queue/Queue.java



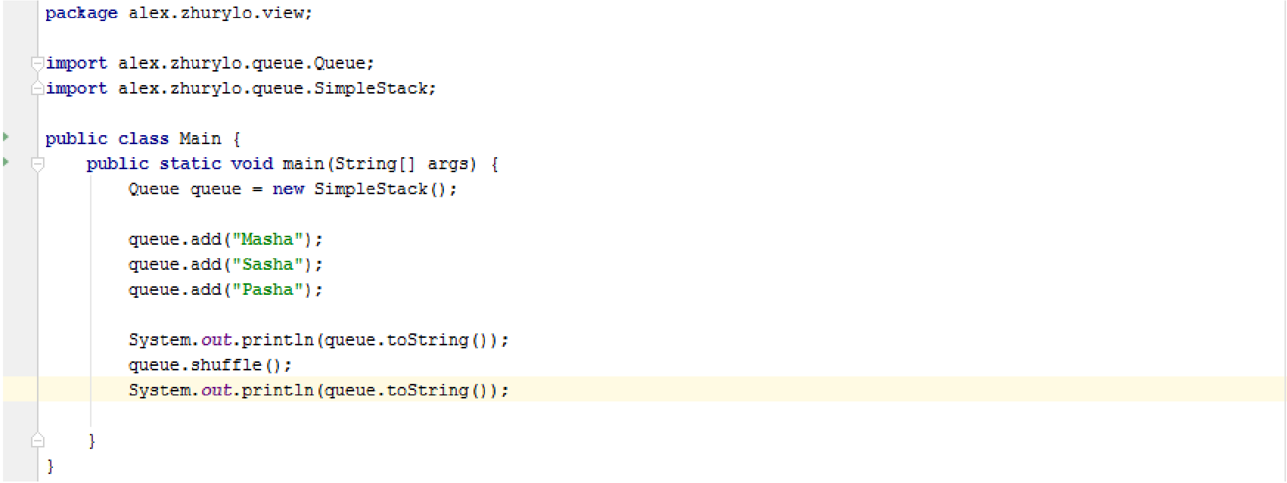
```java
package alex.zhurylo.queue;

public interface Queue {

    boolean add(String name);

    boolean remove() ;

    boolean isEmpty();

    String element();

    void reverseOrder();

    void removeAll();

    void shuffle();
}
```

https://github.com/sasha96/InterLink/blob/master/Queue/src/alex/zhurylo/queue/SimpleStack.java



```java
package alex.zhurylo.queue;

import alex.zhurylo.exception.QueueFailedException;

import java.util.Arrays;

public class SimpleStack extends AbstractQueue implements Queue {

    @Override
    public boolean remove() throws QueueFailedException {
        if (arrName.length<1){
            throw new QueueFailedException("Queue is empty.Nothing to remove.");
        }
        for (int j = arrName.length - 1; j > 0; j--) {
            if (arrName[j] != null) {
                arrName[j] = null;
                return true;
            }
        }
        return false;
    }

    @Override
    public String element() {
        for (int i = arrName.length - 1; i > 0; i--) {
            if (arrName[i] != null) {
```

https://github.com/sasha96/InterLink/blob/master/Queue/src/alex/zhurylo/view/Main.java



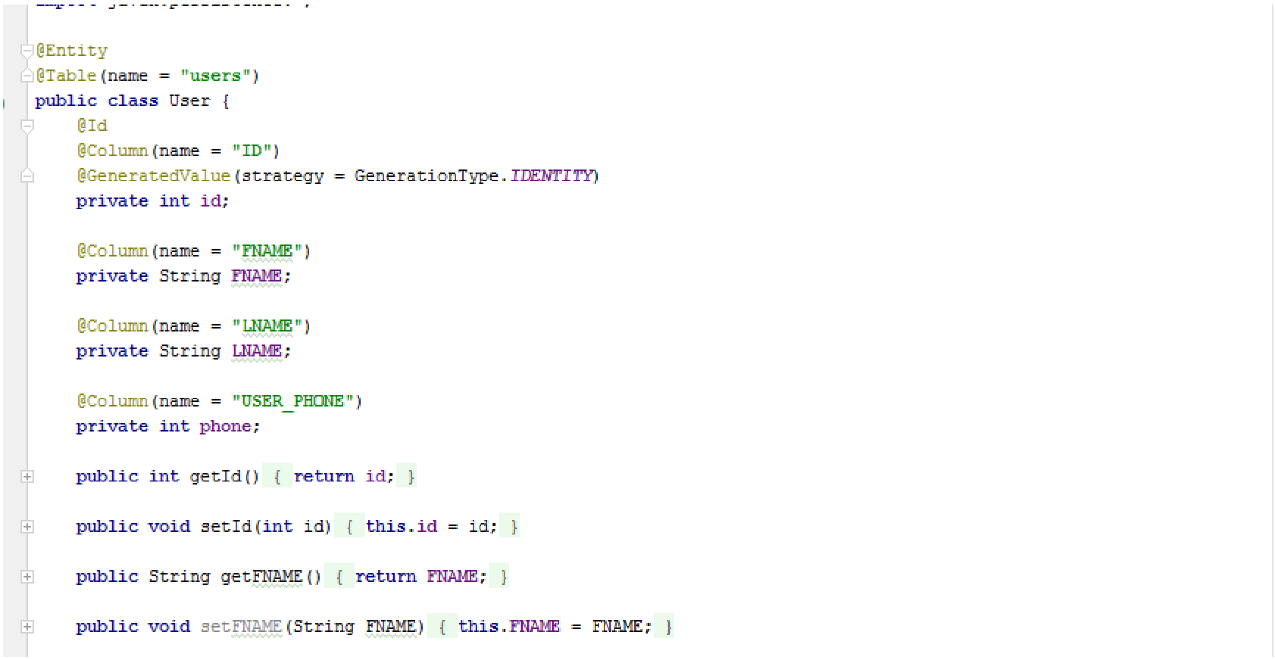
```java
package alex.zhurylo.view;

import alex.zhurylo.queue.Queue;
import alex.zhurylo.queue.SimpleStack;

public class Main {
    public static void main(String[] args) {
        Queue queue = new SimpleStack();

        queue.add("Masha");
        queue.add("Sasha");
        queue.add("Pasha");

        System.out.println(queue.toString());
        queue.shuffle();
        System.out.println(queue.toString());

    }
}
```

Principle 1 Encapsulation

Example 1. Project UserManager

Code Description 1

https://github.com/sasha96/InterLink/blob/master/UserManager/src/main/java/com/zhurylo/sasha/model/User.java

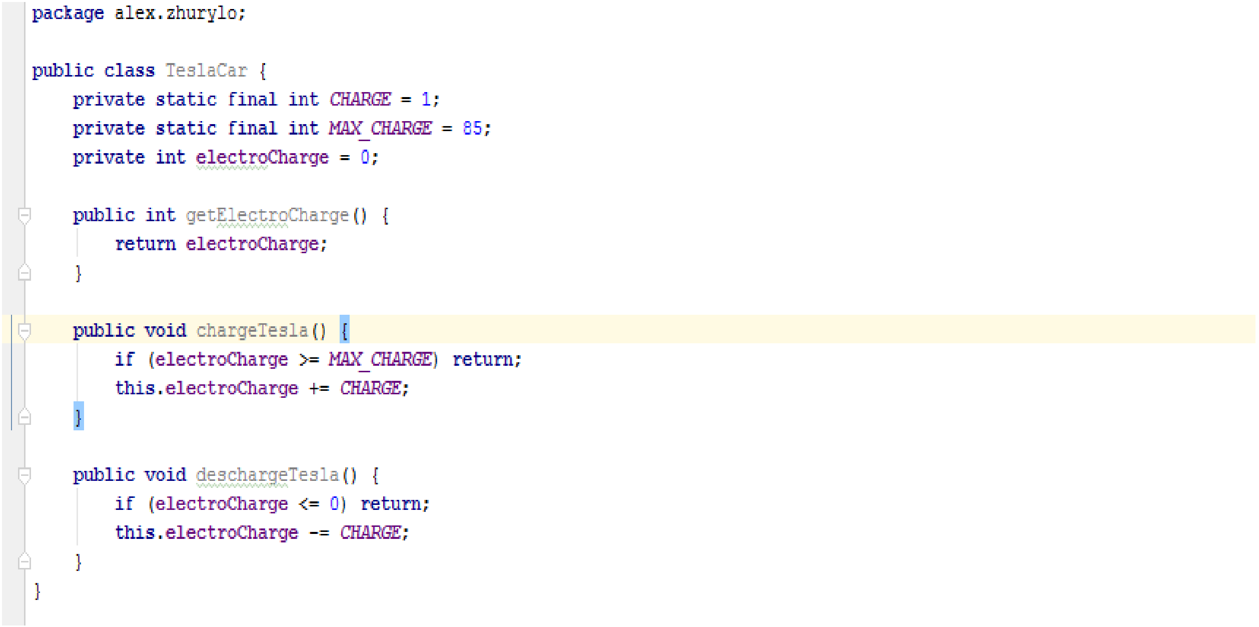


```
@Entity
@Table(name = "users")
public class User {
    @Id
    @Column(name = "ID")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    @Column(name = "FNAME")
    private String FNAME;

    @Column(name = "LNAME")
    private String LNAME;

    @Column(name = "USER_PHONE")
    private int phone;

    public int getId() { return id; }

    public void setId(int id) { this.id = id; }

    public String getFNAME() { return FNAME; }

    public void setFNAME(String FNAME) { this.FNAME = FNAME; }
```

Example 2. Project TeslaCar

Code Description 2

https://github.com/sasha96/InterLink/blob/master/Car/src/alex/zhurylo/TeslaCar.java



```
package alex.zhurylo;

public class TeslaCar {
    private static final int CHARGE = 1;
    private static final int MAX_CHARGE = 85;
    private int electroCharge = 0;

    public int getElectroCharge() {
        return electroCharge;
    }

    public void chargeTesla() {
        if (electroCharge >= MAX_CHARGE) return;
        this.electroCharge += CHARGE;
    }

    public void deschargeTesla() {
        if (electroCharge <= 0) return;
        this.electroCharge -= CHARGE;
    }
}
```