

```

//@author maximilian raspe
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import javafx.scene.layout.VBox;
import javafx.scene.control.Label;
import javafx.scene.canvas.Canvas;

public class GUI extends Application {

    public static Label lb_text;
    Button btn_click1;
    Button btn_click2;
    public static TextField field;
    ButtonClickHandler handler1 = new ButtonClickHandler();
    ValueButtonHandler handler2 = new ValueButtonHandler();
    public static Canvas canvas = new Canvas(400, 200);

    public static void main(String[] args) {
        launch(args);
    }
    @Override
    public void start(Stage primaryStage) throws Exception {
        lb_text = new Label("0 Klicks");
        field = new TextField("5");
        btn_click1 = new Button("Klick");
        btn_click2 = new Button("Setze Klicks");
        canvas.getGraphicsContext2D().strokeRect(100,100,100,100);
        btn_click1.setOnAction(handler1);
        btn_click2.setOnAction(handler2);

        primaryStage.setTitle("GUI");
        VBox box = new VBox();

        box.getChildren().addAll(lb_text, btn_click1, btn_click2, field, canvas);

        Scene scene = new Scene(box, 400, 400);
        primaryStage.setScene(scene);

        primaryStage.show();
    }
}

//@author maximilian raspe
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.paint.Color;

```

```

public class ButtonClickHandler implements EventHandler<ActionEvent> {
    public static int counter = 0;
    @Override
    public void handle(ActionEvent event) { //setzt die klickzahl höher und malt die vierecke
        counter++;
        GUI.lb_text.setText(counter + " Klicks");
        GUI.canvas.getGraphicsContext2D().clearRect(0,0, 400, 200);
        for(int i = 1; i <= counter; i++) {
            GUI.canvas.getGraphicsContext2D().strokeRect(400 / counter, 200/ counter, 100 / i, 100 /
i);
            GUI.canvas.getGraphicsContext2D().fillRect(400 / ButtonClickHandler.counter, 200/
ButtonClickHandler.counter, 100 / i, 100 / i);
            GUI.canvas.getGraphicsContext2D().setFill(Color.DARKTURQUOISE);
        }
    }
}

```

```

//@author maximilian raspe
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.paint.Color;

```

```

import static java.lang.Math.random;

```

```

public class ValueButtonHandler implements EventHandler<ActionEvent> {

    @Override
    public void handle(ActionEvent event) { //malt die anzahl der vierecke
        String t = GUI.field.getText();
        try {
            int num = Integer.parseInt(t);
            ButtonClickHandler.counter = num;
            GUI.lb_text.setText(t + " Klicks");

            GUI.canvas.getGraphicsContext2D().clearRect(0,0, 400, 200);
            for(int i = 1; i <= ButtonClickHandler.counter; i++) {
                GUI.canvas.getGraphicsContext2D().strokeRect(400 / ButtonClickHandler.counter, 200/
ButtonClickHandler.counter, 100 / i, 100 / i);
                GUI.canvas.getGraphicsContext2D().fillRect(400 / ButtonClickHandler.counter, 200/
ButtonClickHandler.counter, 100 / i, 100 / i);
                GUI.canvas.getGraphicsContext2D().setFill(Color.DARKTURQUOISE);
            }
        } catch(NumberFormatException e) {
            System.out.println("Keine Zahl");
        }
    }
}

```

```

//@author maximilian raspe

```

```

public class Eintrag {

    private String produktname;
    private int menge;
    private double preis;
    private static double gesamtpreis;

    //gibt gesamtpreis zurück
    public double getGesamtpreis() {
        return gesamtpreis;
    }

    //Konstruktor für den artikel
    public Eintrag(String produktname, int menge, double preis) {
        this.produktname = produktname;
        this.menge = menge;
        this.preis = preis;
        gesamtpreis += preis;
    }

    public String getProduktname() {
        return produktname;
    }

    public int getMenge() {
        return menge;
    }

    public double getPreis() {
        return preis;
    }

    public String toString() {
        return this.produktname + " " + this.menge + " " + this.preis;
    }
}

//@author maximilian raspe
import java.util.ArrayList;
import java.util.List;

public class Kassenzettel {

    private String name;
    private List<Eintrag> artikel = new ArrayList<Eintrag>();

    //konstruktor für kassenzettel namen
    public Kassenzettel(String name) {
        this.name = name;
    }

    //fügt einträge hinzu
    public void addEintrag(Eintrag gegenstand) {
        artikel.add(gegenstand);
    }
}

```

```

    }
    public String toString() {
        String output = name + "\n\n";
        for(int i = 0; i <= artikel.size() - 1; i++) {
            output += artikel.get(i).getProduktname();
            output += "\n" + artikel.get(i).getMenge() + "x";
            output += "\t" + artikel.get(i).getPreis() + "€";
            output += "\n\n";
        }
        output += "-----";
        output += "\n" + "Gesamtpreis \t" + artikel.get(artikel.size() - 1).getGesamtpreis() + "€";
        return output;
    }
}

```

//@author maximilian raspe

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

String produkt;

int menge;

double preis;

boolean fertig = false;

Kassenzettel zettel = new Kassenzettel("Kassenzettel");

Scanner scan = new Scanner(System.in);

while(fertig == false) {

System.out.println("Produktnamen eingeben");

produkt = scan.nextLine();

if(produkt.equals("x")) {

fertig = true;

System.out.println(zettel);

break;

}

System.out.println("Menge eingeben");

menge = scan.nextInt();

System.out.println("Preis eingeben");

preis = scan.nextDouble();

zettel.addEintrag(new Eintrag(produkt, menge, preis));

scan.nextLine();

}

scan.close();

}

}

//@author maximilian raspe

import java.util.ArrayList;

import java.util.List;

```

public class Kiosk {

    private String name;
    private List<KioskInhalt> kioskinhalte = new ArrayList<KioskInhalt>();

    public Kiosk (String name) {
        this.name = name;
    }
    //fügt die kioskinhalte zu der liste hinzu
    public void addKioskInhalt(KioskInhalt inhalt) {
        kioskinhalte.add(inhalt);
    }
    public String toString() {
        String strReturn = name + ":\n";
        //geht die liste durch um so den artikel auszugeben
        for (KioskInhalt kioskInhalt : kioskinhalte) {
            strReturn += "\n" + kioskInhalt.getName();

            if (kioskInhalt instanceof Elektronisch) {
                strReturn += ", Stromverbrauch: " + ((Elektronisch) kioskInhalt).getStromverbrauch();
            }

            if (kioskInhalt instanceof Kaufbar) {
                strReturn += ", Preis: " + ((Kaufbar) kioskInhalt).getPreis();
            }
        }
        strReturn += "\n";
        return strReturn;
    }
}

```

```

//@author maximilian raspe
public class KioskInhalt {

```

```

    private String name;
    //konstruktor für kioskinhalt
    public KioskInhalt(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }
}

```

```

//@author maximilian raspe
public class Main {
    public static void main(String[] args) {
        Kiosk kiosk = new Kiosk("Kiosk");
        kiosk.addKioskInhalt(new Kasse("Peter"));
    }
}

```

```

        kiosk.addKioskInhalt(new EBookReader("Sony"));
        kiosk.addKioskInhalt(new Zeitung("Wochenblatt"));
        System.out.println(kiosk);
    }
}
//@author maximilian raspe
public class Zeitung extends KioskInhalt implements Kaufbar {

    private double preis = 0.5;
    //konstruktor für die zeitung
    public Zeitung(String name) {
        super(name);
    }
    @Override
    public void setPreis(double preis) {
        this.preis = preis;
    }

    @Override
    public double getPreis() {
        return this.preis;
    }
}
//@author maximilian raspe
public class EBookReader extends KioskInhalt implements Elektronisch, Kaufbar {

    double preis = 0.25;
    double stromverbrauch = 0.25;

    //konstruktor für ebookreader
    public EBookReader(String name) {
        super(name);
    }

    @Override
    public double getStromverbrauch() {
        return stromverbrauch;
    }

    @Override
    public void setPreis(double preis) {
        this.preis = preis;
    }

    @Override
    public double getPreis() {
        return preis;
    }
}
//@author maximilian raspe
public interface Elektronisch {

```

```

    double getStromverbrauch();
}
//@author maximilian raspe
public class Kasse extends KioskInhalt implements Elektronisch {

    double stromverbrauch = 0.5;
    //konstruktor für klassennamen überschreibt
    public Kasse(String name) {
        super(name);
    }

    @Override
    public double getStromverbrauch() {
        return 0.5;
    }
}
//@author maximilian raspe
public interface Kaufbar {

    double getPreis();

    void setPreis(double preis);
}

```