

Esercitazione 2

Creare una mazzo di carte da ramino $\{A, 2...10, J, Q, K\}_i, i \in \{C, Q, F, P\}$, due carte per tipo.

Mescolare il mazzo in modo casuale e mostrare le prime N carte (default N=10).

Controllare se tra le carte c'è una doppia ~~e mostrare una finestra~~, che dirà “hai vinto” se la si è trovata, “hai perso” altrimenti.

Esercitazione 2

Soluzione

```
package it.unitn.lingprogmod1. esercitazione2;
```

```
public class Carta {  
    String valore;  
    char seme;
```

```
    public Carta(String valore, char seme) {  
        this.valore = valore;  
        this.seme = seme;  
    }
```

```
    @Override
```

```
    public String toString() {  
        return "Carta{" +  
            "valore=" + valore +  
            ", seme=" + seme +  
            '}';  
    }
```

```
    boolean compare(Carta c) {  
        return (this.seme==c.seme) && (this.valore == c.valore);  
    }  
}
```

Carta.java

```
package it.unitn.lingprogmod1.esercitazione2;
import java.util.Collections;
import java.util.LinkedList;
```

```
public class Mazzo extends LinkedList<Carta> {
    String[] valore={"A","2","3","4","5","6","7","8","9","10","J","Q","K"};
    char[] seme={'C','Q','F','P'};
```

```
public Mazzo() {
    // for (int i=1;i<=2;i++) // opzione A
    for (String v : valore) {
        for (char s: seme) {
            Carta c = new Carta(v, s);
            this.add(c);
            this.add(c); // opzione B
        }
    }
}
```

```
public Mazzo(int n, Mazzo m) {
    for (int i=0;i<n;i++) {
        this.add(m.get(i));
    }
}
```

Mazzo.java

Due possibilità:

- Commentiamo Opzione A
- oppure
- Commentiamo Opzione B

Mazzo.java

```
void mescola() {  
    Collections.shuffle(this);  
}
```

```
@Override  
public String toString() {  
    String s="Mazzo {\n";  
    for (Carta c: this) {  
        s = s + c.toString()+"\n";  
    }  
    s=s+'}';  
    return s;  
}
```

```
public Carta trovaPrimaDoppia(){  
    int n=this.size();  
    for (int i=0; i<n-1;i++) {  
        Carta c1=this.get(i);  
        for (int j=i+1; j<n; j++) {  
            Carta c2=this.get(j);  
            if (c1.compare(c2)) {  
                System.out.println("found at "+i+" "+j+" : "+c1.toString());  
                return c1;  
            }  
        }  
    }  
}
```

```
public static void main(String a[]){  
    Mazzo m=new Mazzo();  
    m.mescola();  
    m.trovaPrimaDoppia();  
}
```

```
package it.unitn.lingprogmod1.esercitazione2;
```

```
public class Gioco {
```

```
    public static void main(String a[]) {
```

```
        Mazzo m = new Mazzo();
```

```
        m.mescola();
```

```
        Mazzo mazzetto = new Mazzo(10, m);
```

```
        System.out.println(mazzetto);
```

```
        Carta c = m.trovaPrimaDoppia();
```

```
        if (c == null) {
```

```
            System.out.println("Spiacente, hai perso");
```

```
        } else {
```

```
            System.out.println("hai vinto con " + c);
```

```
        }
```

```
    }
```

```
}
```

Gioco.java

Mazzo.java

Implementazione alternativa

```
public Carta trovaPrimaDoppia(true){
    for (Carta c: this) {
        if (indexOf(c)!=lastIndexOf(c)) {
            System.out.println("found at "+indexOf(c)+" "+lastIndexOf(c)+" : "+c.toString());
            return c;
        }
    }
    return null;
}
```

```
public static void main(String a[]){
    Mazzo m=new Mazzo();
    m.mescola();
    m.trovaPrimaDoppia();
    m.trovaPrimaDoppia(true);
}
```


Varianti

- Rendere N scegliibile dall'utente, e validarne la risposta.
- Dare, nella finestra finale, una rappresentazione grafica semplificata della carta vincente.
- Dopo la eventuale vittoria, scegliere a caso una carta dal mazzo. Se questa ha lo stesso valore della vincente (indipendentemente da seme), la vittoria vale doppio. Dare comunicazione all'utente dell'esito di tale operazione.

Esercitazione 2

Soluzione con I/O grafico

- Rendere N scegliibile dall'utente, e validarne la risposta.
- Controllare se tra le carte c'è una doppia e mostrare una finestra, che dirà "hai vinto" se la si è trovata, "hai perso" altrimenti.

Main.java

```
package it.unitn.lingprogmod1.esercitaz2grafica;
import javafx.application.Application;
import javafx.scene.control.Alert;
import javafx.scene.control.TextInputDialog;
import javafx.stage.Stage;
public class Main extends Application {
```

```
@Override
public void start(Stage primaryStage) throws Exception {
    int n = 10;
    boolean error = true;
    do {
        TextInputDialog dialog = new TextInputDialog("" + n);
        dialog.setTitle("Gioco a carte");
        dialog.setHeaderText("Quante carte?");
        dialog.setContentText(":");
        String s = dialog.showAndWait().get();
        try {
            n = Integer.parseInt(s);
            error = false;
        } catch (NumberFormatException e) {
            error = true;
        }
    } while (error);
}
```

Main.java

```
Mazzo m = new Mazzo();
m.mescola();
Mazzo mazzetto = new Mazzo(n, m);
System.out.println(mazzetto);
Carta c = mazzetto.trovaPrimaDoppia();
String msg;
if (c == null) {
    msg = "Spiacente, hai perso";
} else {
    msg = "hai vinto con " + c;
}
```

```
Alert alert = new Alert(Alert.AlertType.INFORMATION);
alert.setTitle("Gioco a carte");
alert.setHeaderText("Esito della partita:");
alert.setContentText(msg);
alert.showAndWait();
/*StackPane root = new StackPane();
primaryStage.setTitle("Hello World");
primaryStage.setScene(new Scene(root, 300, 275));
primaryStage.show();
*/
}

public static void main(String[] args) {
    launch(args);
}
}
```

Esercitazione 2

Soluzione «vittoria doppia»

- Dopo la eventuale vittoria, scegliere a caso una carta dal mazzo. Se questa ha lo stesso valore della vincente (indipendentemente da seme), la vittoria vale doppio. Dare comunicazione all'utente dell'esito di tale operazione.

Vittoria doppia:

In Mazzo.java

```
public Carta getRandom(int excluding) {  
    Random r=new Random();  
    int i=excluding+r.nextInt(this.size()-excluding);  
    return get(i);  
}
```

In Main.java:

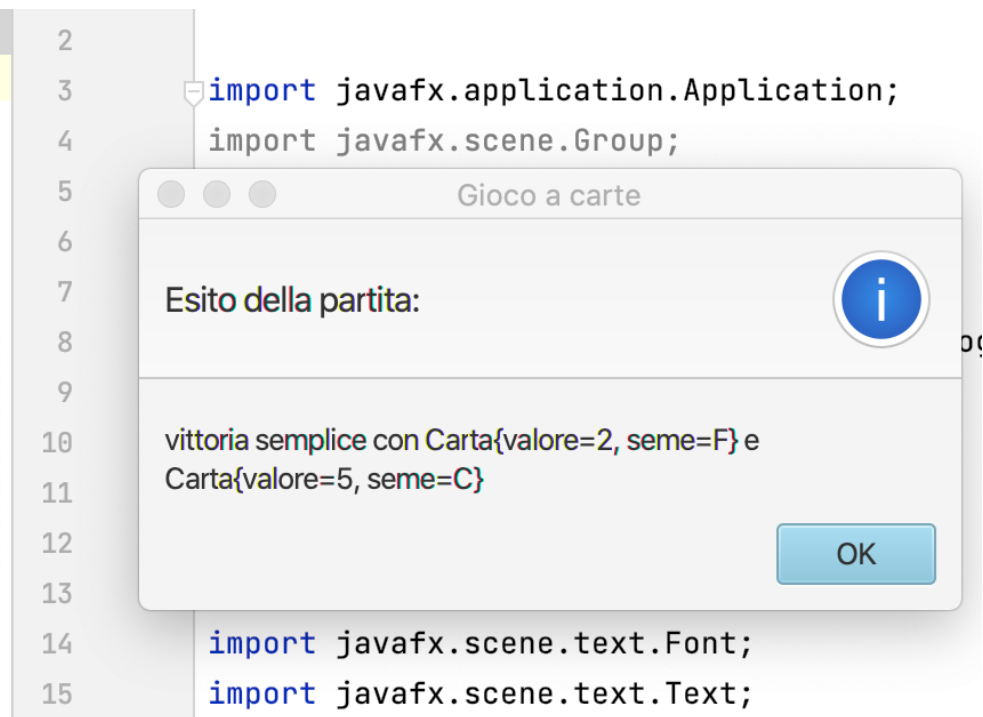
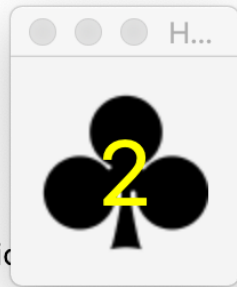
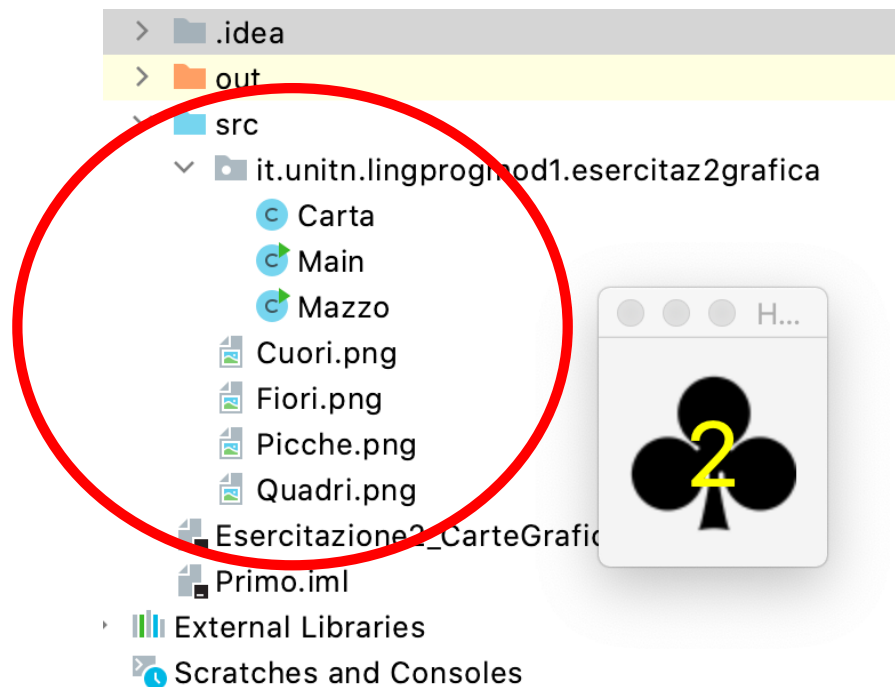
```
String msg;  
if (c == null) {  
    msg = "Spiacente, hai perso";  
} else {  
    Carta estratta=m.getRandom(n);  
    if (estratta.seme==c.seme) {  
        msg = "doppia vittoria con " + c + " e " + estratta;  
    } else {  
        msg = "vittoria semplice con " + c + " e " + estratta;  
    }  
}
```

Esercitazione 2

Soluzione con grafica

- Dare, nella finestra finale, una rappresentazione grafica semplificata della carta vincente.

Vittoria con grafica:



In Main.java:

Aggiungiamo
questo metodo

```
String getImageName(char c){
    String filename=null;
    switch (c) {
        case 'C':
            filename="Cuori.png";
            break;
        case 'Q':
            filename="Quadri.png";
            break;
        case 'F':
            filename="Fiori.png";
            break;
        case 'P':
            filename="Picche.png";
            break;
    }
    if (filename==null) return null;
    return filename;
}
```

Modifichiamo
start:

```
...
boolean vittoria=true;
if (c == null) {
    msg = "Spiacente, hai perso";
    vittoria=false;
} else {
    Carta estratta=m.getRandom(n);
    if (estratta.seme==c.seme) {
        msg = "doppia vittoria con "
            + c + " e " + estratta;
    } else {
        msg = "vittoria semplice con »
            + c + " e " + estratta;
    }
}
Alert alert = new Alert(
    Alert.AlertType.INFORMATION);
alert.setTitle("Gioco a carte");
alert.setHeaderText("Esito della partita:");
alert.setContentText(msg);
alert.show();
```

Vittoria con grafica

```
if (vittoria) {
    String filename = getImageName(c.seme);
    System.out.println(filename);
    Image image = new Image("file:src/" +
        filename, 80, 80, true, true);
    ImageView iw = new ImageView(image);
    Text t = new Text(c.valore);
    t.setFill(Paint.valueOf("YELLOW"));
    t.setFont(new Font(40));
    StackPane root = new StackPane(iw, t);
    primaryStage.setTitle("Hello World");
    primaryStage.setScene(new Scene(
        root, 100, 100));
    primaryStage.setX(400);
    primaryStage.setY(300);
    primaryStage.show();
}
```

Esercitazione 2

Razionalizzazione del codice: Modifica di Carta

In Carta.java:

```
private String getImageName(){
    String filename=null;
    switch (seme) {
        case 'C':
            filename="Cuori.png";
            break;
        case 'Q':
            filename="Quadri.png";
            break;
        case 'F':
            filename="Fiori.png";
            break;
        case 'P':
            filename="Picche.png";
            break;
    }
    if (filename==null) return null;
    return filename;
}
```

Aggiungiamo questi
metodi

```
public Node getGraphicCarta(){
    String filename = getImageName();
    Image image = new Image("file:src/" +
        filename, 80, 80, true, true);
    ImageView iw = new ImageView(image);
    Text t = new Text(valore);
    t.setFill(Paint.valueOf("YELLOW"));
    t.setFont(new Font(40));
    StackPane node = new StackPane(iw, t);
    return node;
}
```

Modifichiamo
start :

Modifica di Carta

```
if (vittoria) {
    Parent root =(Parent)c.getGraphicCarta();
    primaryStage.setTitle("Hello World");
    primaryStage.setScene(
        new Scene(root, 100, 100));
    primaryStage.setX(400);
    primaryStage.setY(300);
    primaryStage.show();
}
```

Esercitazione 2

Razionalizzazione del codice: Modifica della classe principale

Ristrutturazione main

Aggiungiamo questi
metodi

```
private int askUserForNumber(int default_n){
    int n=default_n;
    boolean error = true;
    do {
        TextInputDialog dialog =
            new TextInputDialog(" " + default_n);
        dialog.setTitle("Gioco a carte");
        dialog.setHeaderText("Quante carte?");
        dialog.setContentText(":");
        String s = dialog.showAndWait().get();
        try {
            n = Integer.parseInt(s);
            error = false;
        } catch (
            NumberFormatException e) {
            error = true;
        }
    } while (error);
    return n;
}
```

Modifichiamo start

```
private void showAlert(String msg) {
    Alert alert = new Alert(
        Alert.AlertType.INFORMATION);
    alert.setTitle("Gioco a carte");
    alert.setHeaderText(
        "Esito della partita:");
    alert.setContentText(msg);
    alert.show();
}
```

```
public void start(Stage primaryStage) throws Exception {
    int n = askUserForNumber(10);
    Mazzo m = new Mazzo();
    m.mescola();
    Mazzo mazzetto = new Mazzo(n, m);
    System.out.println(mazzetto);
    Carta c = mazzetto.trovaPrimaDoppia();
    String msg;
    boolean vittoria=true;
    if (c == null) {
        msg = "Spiacente, hai perso";
        vittoria=false;
    } else {
        Carta estratta=m.getRandom(n);
        if (estratta.seme==c.seme) {
            msg = "doppia vittoria con " + c + " e " + e
        } else {
            msg = "vittoria semplice con " + c + " e " + e
        }
    }
    showAlert(msg);
    if (vittoria) {...}
}
```

Set vs. List

Verificare che il mazzo creato aggiungendo le carte ad un Set contiene 104 elementi *se Carta non implementa la equals e la hashCode*, mentre ne contiene 52 *se Carta implementa correttamente equals ed hashCode*.

Spiegarne le ragioni.