

Codice Java corso Generation 2021

```
package model;

public class Automobile {

    public String marca, modello, targa;
    public boolean privato;
    public double kw;

    public Automobile(String marca, String modello, String targa, boolean
privato, double kw) {
        this.marca = marca;
        this.modello = modello;
        this.targa = targa;
        this.privato = privato;
        this.kw = kw;
    }

    //OVERLOAD dei metodi: stesso nome, ma FIRMA del metodo differente
nome(listaParametri)
    public double calcolaBollo(double sconto) {
        System.out.println("hai chiamato il metodo con un argomento di tipo
double");
        double valoreBollo = this.kw * sconto;

        if (this.privato == false ) {
            valoreBollo = valoreBollo - ( valoreBollo * 0.25 );
        }

        return valoreBollo;
    }
    public int calcolaBollo(int sconto) {
        System.out.println("hai chiamato il metodo con un argomento di tipo
intero");
        int valoreBollo = (int) (this.kw * sconto);

        if (this.privato == false ) {
            valoreBollo = (int) (valoreBollo - ( valoreBollo * 0.25 ));
        }

        return valoreBollo;
    }

    public double calcolaBollo() {
        System.out.println("hai chiamato il metodo senza argomenti");

        double valoreBollo = this.kw * 3.70;

        if (this.privato == false ) {
```

```

        valoreBollo = valoreBollo - ( valoreBollo * 0.25 );
    }

    return valoreBollo;
}

public String libretto() {
    String s = "Marca: " + this.marca + " Modello: " + this.modello + "
€" + this.calcolaBollo();
    return s;
}

@Override
public String toString() {
    return "Automobile [marca=" + marca + ", modello=" + modello + ",
targa=" + targa + ", calcolaBollo()="
        + calcolaBollo(2.50) + " ]";
}
}

```

```

package interfaccia;

import java.util.Scanner;

import model.Automobile;

public class GestioneACI {

    public static void main(String[] args) {

        //contenitore di automobili
        Automobile[] auto = new Automobile[3];
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < auto.length; i++) {

            System.out.println("Dati auto nr." + (i + 1));
            System.out.println("marca");
            String marca = sc.nextLine();

            System.out.println("modello");
            String modello = sc.nextLine();

            System.out.println("targa");
            String targa = sc.nextLine();

            System.out.println("p: per uso privato, c: per uso
commerciale");

```

```

        String uso = sc.nextLine();

        System.out.println("quanti KW");
        double kw = sc.nextDouble();
        sc.nextLine();

        auto[i] = new Automobile(marca, modello, targa,
        uso.equals("p"), kw);

    }

//      auto[0] = new Automobile("FIAT", "500", "AA550BB", true, 78); //uso
privato
//      auto[1] = new Automobile("Audi", "A8", "BB550CC", true, 150); //uso
privato
//      auto[2] = new Automobile("VW", "GOLF", "CC550DD", false, 100); //uso
commerciale
//      auto[3] = null;

//      for ( int i = 0; i < auto.length ; i++      ) {
//          if (auto[i] != null)
//              System.out.println(      auto[i]      );
//      }

        for (int i = 0; i < auto.length; i++) {
            if (auto[i] != null)
                System.out.println(      auto[i]      );
        }

    }

}

```

```

package giochi;

import java.util.Scanner;

public class CartaForbiciSasso {

    static int vittorieU = 0;
    static int vittorieP = 0;
    static int pareggi = 0;
    static Scanner input = new Scanner(System.in);

    public static void main(String[] args) {

        char utente;
        char pc;
    }
}

```

```
String msg;

for (int i = 0; i < 5; i++) {
    //1 chiedere all'utente p,f,c
    utente = sceltaUtente();

    //2 far scegliere un casuale tra pfc alla macchina
    pc = sceltaPC();

    //3 valutare se sono uguali: messaggio = "pareggio"
    if (utente == pc) {
        msg = "pareggio";
        pareggi++;
    } else {
        //4 altrimenti valutare chi ha vinto:

        msg = valutaGioco(utente, pc);

        //4a u == p -> se pc == f -> messaggio = pc perde altrimenti
        vince
        //4b u == c -> se pc == p -> messaggio = pc perde altrimenti
        vince
        //4c u == f -> se pc == c -> messaggio = pc perde altrimenti
        vince
    }

    //5 stampa messaggio
    System.out.println(msg);
} //chiudo ciclo

System.out.println("Pareggi " + pareggi);
System.out.println("Vittorie utente " + vittorieU);
System.out.println("Vittorie PC " + vittorieP);

//6 saluta ed esce, oppure rigioca se in un loop
System.out.println("Grazie per aver giocato!");
input.close();
}

private static String valutaGioco(char utente, char pc) {
    String s = null;

    if (utente == 'p') {
        if (pc == 'f') {
            s = "Vince utente";
            vittorieU++;
        } else {
            s = "Vince PC";
            vittorieP++;
        }
    } else if (utente == 'f') {
        if (pc == 'c') {

```

```

        s = "Vince utente";
        vittorieU++;
    } else {
        s = "Vince PC";
        vittorieP++;
    }
} else {
    if(pc == 'p') {
        s = "Vince utente";
        vittorieU++;
    } else {
        s = "Vince PC";
        vittorieP++;
    }
}

return s;
}

private static char sceltaPC() {
    char c = 0;
    double d = Math.random();
    if (d < 0.33) {
        c = 'p';
    } else if ( d < 0.66 ) {
        c = 'f';
    } else {
        c = 'c';
    }
    return c;
}

//metodo ausiliario per chiedere la scelta all'utente
private static char sceltaUtente() {

    System.out.println("Scegli tra pietra, forbici, carta");
    System.out.println("p: pietra");
    System.out.println("f: forbici");
    System.out.println("c: carta");
    String s = input.nextLine();
    char c = s.charAt(0);
    //input.close();
    return c;
}

}

```

```

package model;

public class Nota {

```

```

        //incapsulamento
        private boolean completa;
        private String descrizione;

        //costruttore metodo senza tipo ritorno, normalmente public, scompare
        quello di default
        public Nota(String descrizione) {
            this.descrizione = descrizione;
            this.completa = false;
            //System.out.println("ho creato la nota: " + this.descrizione);
        }

        public boolean isCompleta() {
            return completa; //      true/false
        }

        public void setCompleta(boolean completa) {
            this.completa = completa;
        }

        public String getDescrizione() {
            return descrizione;
        }

        @Override
        public String toString() {
            String s = "[ ]"; //non completa

            if (this.completa) {
                s = "[*]";
            }

            return s + " " + descrizione + "";
            // [ ] latte
            // [*] pane
        }

    }

```

```

package view;

import java.util.Scanner;

import controller.BloccoNote;
import model.Nota;

public class GestioneNote {

```

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

    BloccoNote notesBianco = new BloccoNote();
    String s = "";
    boolean esci = false;

    Scanner sc = new Scanner(System.in);

    while (!esci) {
        System.out.println("Cosa vuoi fare?");
        System.out.println("a) aggiungi nota");
        System.out.println("b) completa nota");
        System.out.println("c) elenco note");
        System.out.println("q) esci");

        s = sc.nextLine();

        if (s.equals("a")) {
            riempiBlocco(notesBianco);
        }

        if (s.equals("b")) {
            notesBianco.completaNota(5);
            eliminaNota(notesBianco);
        }

        if (s.equals("c")) {
            stampaBloccoNote(notesBianco);
        }

        if (s.equals("q")) {
            esci = true;
            System.out.println("arrivederci! Buon appetito :)");
        }
    }

}

private static void eliminaNota(BloccoNote notesBianco) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Quale nota vuoi completare? inserisci il numero");
    int s = sc.nextInt();
    notesBianco.completaNota(s);
}

private static void riempiBlocco(BloccoNote notesBianco) {
    //automatizzare
    Scanner sc = new Scanner(System.in);
    System.out.println("Inserisci voce");
}
```

```
        String s = sc.nextLine();
        notesBianco.addNota(s);
//        notesBianco.addNota("pane");
//        notesBianco.addNota("uova");
//        notesBianco.addNota("burro");
//        notesBianco.addNota("birra");
//        notesBianco.addNota("biscotti");
//        notesBianco.addNota("biscotti");
//        notesBianco.addNota("carote");
//        notesBianco.addNota("patate");
//        notesBianco.addNota("broccoli");
//        notesBianco.addNota("olio");
    }

    private static void stampaBloccoNote(BloccoNote notesBianco) {
        //stampare elenco
        Nota[] note = notesBianco.getLista();
        for (int i = 0; i < note.length; i++) {
            System.out.println("Nota. " + i + " - " + note[i]);
        }
    }
}
```

```
package controller;

import model.Nota;

public class BloccoNote {

    private Nota[] lista;//getLista()

    public BloccoNote() {
        this.lista = new Nota[10];
    }

//    private boolean isFull() {
//        boolean full = false;
//        int elementi = 0;
//        for (int i = 0; i < lista.length; i++) {
//            if(lista[i] != null)
//                elementi++;
//        }
//        if (elementi == lista.length)
//            full = true;
//        return full;
//    }

    private int primaPosizioneDisponibile() {
        int posizione = -1;
    }
}
```



```
        for (int i = 0; i < lista.length; i++) {
            if (lista[i] == null) {
                posizione = i;
                break;
            }
        }

        return posizione;
    }

    public void addNota(String testo) {

        //verificare se array è pieno
        if ( primaPosizioneDisponibile() == -1 ) {
            System.out.println("Non puoi inserire " + testo+ " perché il
blocco è già pieno!");
        } else {
            //se non è pieno, qual è la prima pos disponibile
            int pos = primaPosizioneDisponibile();
            //costruisco la nota
            Nota n = new Nota(testo);
            //metto la nota nella posizione ritornata
            lista[pos] = n;
            //System.out.println("ho inserito " + n.getDescrizione() + "
nella lista alla posizione " + pos );
        }

    } //fine metodo

    public Nota[] getLista() {
        return lista;
    }

    public void completaNota(int pos) {
        if (pos >= 0 && pos < lista.length)
            lista[pos].setCompleta(true);
        System.out.println("Completata la nota n. " + pos);
    }

}
```

```
package modelli;

public class ProvaBicicletta {

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

```
Bicicletta bici1 = new Bicicletta("Atala");
bici1.accelera(20);
bici1.cambia(2);

System.out.println(bici1.stampaReport());
bici1.cambia(4);
System.out.println(bici1.stampaReport());

Bicicletta bici2 = new Bicicletta("Specialized");
bici2.accelera(20);
bici2.cambia(5);
System.out.println(bici2.stampaReport());

}

}
```

```
package modelli;

public class Bicicletta {

    String marca;
    int marcia = 1;
    int frequenza = 0;
    int velocita = 0;
    int circonferenza = 1;

    public Bicicletta(String marcaBici) {
        marca = marcaBici;
    }

    public void accelera(int pedalata) {
        frequenza += pedalata;
        calcolaVelocita();
    }

    public void rallenta(int pedalata) {
        frequenza -= pedalata;
        calcolaVelocita();
    }

    public int calcolaVelocita() {
        velocita = marcia * frequenza * circonferenza;

        return velocita;
    }

    public void cambia(int nuovaMarcia) {
```

```
        marcia = nuovaMarcia;
        calcolaVelocita();
    }

    public String stampaReport() {
//        String report = "Marca " + marca + " velocita " + velocita;
        return "La Marca della bici è " + marca + ", la sua velocità " +
velocita;
    }

}
```

```
package model;

public class Studente {

    //contatore static: appartiene alla classe e serve per contare gli
    studenti costruiti
    private static int contatore = 1;
    private String nome, cognome;
    private int nMatricola;

    public Studente(String nome, String cognome) {
        this.nome = nome;
        this.cognome = cognome;
        //ogni volta che istanzio uno studente:
        //col valore di contatore attribuisco il valore alla matricola dello
        studente corrente (this)
        //poi, il valore di contatore si incrementa di uno
        this.nMatricola = Studente.contatore++;
    }

    public String getCognome() {
        return cognome;
    }

    @Override
    public String toString() {

        return nMatricola + ") " + nome + ", " + cognome;

    }

}
```

```
package view;

import controller.Registro;

public class GestioneAula {

    public static void main(String[] args) {
        Registro r = new Registro();
        r.faiAppello();
    }

}
```

```
package database;

import model.Studente;

public class DBStudenti {

    private Studente[] studenti;

    public DBStudenti() {
        this.studenti = new Studente[29];
        this.fakeDB();
    }

    private void fakeDB() {
        this.studenti[0] = new Studente("Alessandro", "Storto");
        this.studenti[1] = new Studente("Andrea", "Spinnato");
        this.studenti[2] = new Studente("Chiara", "Cavagnino");
        this.studenti[3] = new Studente("Chiara", "Favaro");
        this.studenti[4] = new Studente("Clarissa", "Ballocco");
        this.studenti[5] = new Studente("teodora elena", "tomescu");
        this.studenti[6] = new Studente("Fabio Denis", "Davico");
        this.studenti[7] = new Studente("Federica", "Liuzzi");
        this.studenti[8] = new Studente("Federico", "Delpiano");
        this.studenti[9] = new Studente("Flavio", "Locorotondo");
        this.studenti[10] = new Studente("Francesca", "Provero");
        this.studenti[11] = new Studente("Francesca", "Mollica");
        this.studenti[12] = new Studente("Gabriel", "Zirondi Ferreira da
Silva");
        this.studenti[13] = new Studente("george", "bejenaru");
        this.studenti[14] = new Studente("Giovenale", "Sergio");
        this.studenti[15] = new Studente("Giulia", "Bosco");
        this.studenti[16] = new Studente("Giuliana Elisa", "Ducano");
        this.studenti[17] = new Studente("Giulio", "Ricci");
        this.studenti[18] = new Studente("Irene", "Tempo");
        this.studenti[19] = new Studente("Kleandro", "Kulli");
        this.studenti[20] = new Studente("Laila", "Zoundi");
        this.studenti[21] = new Studente("Lorenzo", "Baialardo Marchese");
    }

}
```

```
        this.studenti[22] = new Studente("Matteo", "Billia");
        this.studenti[23] = new Studente("Matteo", "Inchingolo");
        this.studenti[24] = new Studente("Paolo", "Cai");
        this.studenti[25] = new Studente("Riccardo", "Placenti");
        this.studenti[26] = new Studente("Sibora", "Beja");
        this.studenti[27] = new Studente("Tommaso", "Amateis");
        this.studenti[28] = new Studente("Vincenzo", "Di Domenico");

    }

    public Studente[] getStudenti() {
        return studenti;
    }

}
```

```
package controller;

import java.util.Date;
import java.util.Scanner;

import database.DBStudenti;
import model.Studente;

public class Registro {

    //contenitore in cui riversati gli studenti del db
    private Studente[] studenti;

    //è l'oggetto db
    private DBStudenti db;

    public Registro() {
        db = new DBStudenti();
        this.studenti = db.getStudenti();
    }

    public void faiAppello() {
        Scanner sc = new Scanner(System.in);
        int presenti = 0;
        int assenti = 0;
        StringBuilder sbAssenti = new StringBuilder();
        StringBuilder sbPresenti = new StringBuilder();
        //per ogni posizione io trovo un oggetto Studente
        for (Studente studente : studenti) {
            System.out.println("E' presente lo studente");
        }
    }
}
```

```

        System.out.println(studente);
        String risposta = sc.nextLine();
        if (risposta.equals("s")) {
            presenti++;
            sbPresenti.append(studente.getCognome()).append(" - ");
        } else {
            assenti++;
            if (studente != null)
                sbAssenti.append(studente.getCognome()).append(" - ");
        }
    } //fine for
    Date oggi = new Date();
    System.out.println("Oggi " + oggi);
    System.out.println("sono presenti " + presenti);
    System.out.println(sbPresenti.toString());

    System.out.println("sono assenti " + assenti);
    System.out.println(sbAssenti.toString());
    sc.close();
}

}

```

```

package esercizi;

import java.util.Scanner;

/*
 * Scrivere una funzione (metodo) per calcolare l'importo di una tassa
secondo la seguente tabella:
* Fino a 10.000 €, l'importo della tassa è del 10%
* Fino a 20.000 €, l'importo della tassa è del 10% per i primi 10.000 €,
del 7 % sul restante.
* Fino a 30.000 €, l'importo è ancora del 10% per i primi 10.000 €, poi del
7% fino a 20.000 ed infine il 5% sul restante.
* Oltre i 30.000 € ci si comporta come prima, aggiungendo un ulteriore 3%
sulla porzione oltre i 30.000 €.
* Per ogni importo non valido si ritorni -1 (qualunque importo che non sia
un numero reale e positivo).

* Esempi: Un importo di 10000, dovrebbe tornare 1000 (1 è il 10% di 10)
* Un importo di 21000, dovrebbe tornare 1750 (10% di 10 + 7% di 10 + 5% di
1)
*/

public class CalcolaTassa {

    static final double TAX10 = 0.1; //Fino a 10.000 €, l'importo della
tassa è del 10%
    static final double TAX20 = 0.07; //da 10 a 20K del 7 % sul restante.
    static final double TAX30 = 0.05; //da 20 a 30K del 5%

```

```

static final double TAX_OVER = 0.03; //oltre 30K del 3%
static final double SCAGLIONE = 10000;

public static void main(String[] args) {

    double importo = getInput();
    double tasso = 0;

    tasso = calcolaTasse(importo, tasso);

    System.out.println(calcolaTasse(5));

    System.out.println("Dato l'importo imponibile di € " + importo);
    if (tasso >= 0)
        System.out.println("Le tasse da pagare sono di € " + tasso);
    if (tasso < 0) {
        System.out.println("Sei andato sotto! Non ci sono tasse da
pagare. Buona fortuna per l'anno prossimo!");
    }

}

private static double calcolaTasse(double importo) {

    if (importo <= 1) {
        return 1;
    } else {
        return (importo * calcolaTasse(importo - 1));
    }

    //double tasso=0;

    //21000 ciclo (x 21 - (10 * t1)), (11 - (10*t2) ), (1 - (1*t3)) ,
(x 10)esci

    //return tasso;

}

private static double calcolaTasse(double importo, double tasso) {
    if (importo <= 0) {
        //return -1
        //da pagare 0 - zero
        tasso = -1;
    } else if (importo <= SCAGLIONE ) {
        tasso = importo * TAX10;
    } else if (importo <= 2 * SCAGLIONE ) {
        tasso = SCAGLIONE * TAX10 + (importo - SCAGLIONE) * TAX20;
    } else if (importo <= 3 * SCAGLIONE ) {
        tasso = (SCAGLIONE * TAX10) + (SCAGLIONE * TAX20) + (importo -

```

```
2 * SCAGLIONE ) * TAX30;
    } else {
        tassa = (SCAGLIONE * TAX10) + (SCAGLIONE * TAX20) + (SCAGLIONE
* TAX30) + (importo - 3 * SCAGLIONE ) * TAX_OVER;
    }
    return tassa;
}

private static double getInput() {
    Scanner sc = new Scanner(System.in);
    System.out.println("Inserisci importo imponibile (maggiore di 0)");
    return sc.nextDouble();
}

}
```

```
package teol2.stringhe;

public class ProvaString01 {

    public static void main(String[] args) {

        String s1 = "ciao";
        String s2 = " mondo";
        String s4 = s1 + s2;

        String s3 = "ciao Mondo";
        if ( s4.equalsIgnoreCase(s3) )
            System.out.println("sono uguali");
        else
            System.out.println("sono diverse");
        System.out.println(s4);

        String s5 = new String("prova");
        String s6 = new String("prova");

        s5 += " pippo";

        if (s5 == s6)
            System.out.println("sono uguali");
        else
            System.out.println("sono diverse");

    }

}
```



```
package teo12.stringhe;

/**
 * Questa classe serve per testare il ns stringbuilder
 * @author mauro
 *
 */
public class ProvaStringBuilder {

    /**
     *
     * @param args - passare 0, 1 o più argomenti per far partire questo
    programma
     */
    public static void main(String[] args) {

        StringBuilder sb = new StringBuilder();

        String[] re = {
            "romolo",
            "numa pompilio",
            "tullo ostilio",
            "anco marzio",
            "servio tullio",
            "tarquinio prisco",
            "tarquinio il superbo"
        };

        for (String string : re) {
            sb.append(string).append(" - ");
        }

        //System.out.println(sb.toString());

        String stringona = sb.toString();

        String modificata = stringona.replaceAll(" - ", "|");
        System.out.println(modificata);

        //stringona.length()//lunghezza
        //String[] split = stringona.split(" - ");
        //System.out.println(split[0]);

    }

}
```

```
package teo11.array;

public class Interi {
```

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

    //dichiarazione di un array di interi
    int[] voti;

    //inizializzato un oggetto di tipo array (contenitore indicizzato)
    voti = new int[10];

    //nell'array length è una proprietà dell'oggetto
    int lunghezza = voti.length;
    int somma = 0;
    //
    System.out.println(lunghezza);

    //
    voti[0] = 5;
    //
    voti[1] = 1;
    //
    //...
    //ciclo per riempire l'array voti con 10 valori casuali tra 1 e 10
    for (int i = 0; i < voti.length; i++) {

        voti[i] = (int) (Math.random() * lunghezza) + 1;
    }

    //ciclo per leggere i valori contenuti nell'array voti
    for (int i = 0; i < voti.length; i++) {
        System.out.println("Il voto alla posizione " + i + " è: " +
voti[i]);
        somma += voti[i];
    }
    System.out.println("La somma dei voti casuali è: " + somma);
    System.out.println("La media dei voti è: " + ((double)
somma/lunghezza));
}

}
```

```
package teo11.array;

import java.util.Arrays;

public class Stringhe {

    public static void main(String[] args) {

        //inizializzo
        //
        nani = new String[7];

        //inserisco i valori(riferimenti agli oggetti)
        //
        nani[0] = "pisolo";//string inizializzato con apici
    }
}
```

```
//      nani[1] = new String("mammolo");//string inizializzata con
operatore new

//inizializzo shorthand scorciatoia
//dichiaro array di stringhe
String[] nani = {"pisolo", "mammolo", "eolo",
                "brontolo", "cucciolo", "dotto", "gongolo"};

for (int i = 0; i < nani.length; i++) {
    System.out.println(nani[i]);
}
//ordino l'array col metodo statico .sort() della classe
java.util.Arrays
Arrays.sort(nani);
//dichiaro una string inizializzata a null (è un oggetto nullo)
String nanoCorrente = null;

System.out.println("nani ordinati");
for (int i = 0; i < nani.length; i++) {
    //assegno alla var il valore della string corrente
    nanoCorrente = nani[i];

    System.out.println("-----");
    System.out.println(nanoCorrente);
    System.out.println("-----");
    System.out.println(nanoCorrente.charAt(0));
    System.out.println(nanoCorrente.equals("pisolo"));
    System.out.println(nanoCorrente.toUpperCase());
    System.out.println(nanoCorrente.substring(2));
    if(nanoCorrente.length() <= 3)
        System.out.println(nanoCorrente.substring(2, 5));
    System.out.println(nanoCorrente.startsWith("b"));
    System.out.println(nanoCorrente.endsWith("lo"));
    System.out.println(nanoCorrente.contains("ol"));
    System.out.println(nanoCorrente.repeat(3));
    System.out.println(nanoCorrente.replaceAll("o", "@"));
    System.out.println("-----");

}

}

}
```

```
package giochi;

import java.util.Scanner;

public class CartaForbiciSasso {
```

```
static int vittorieU = 0;
static int vittorieP = 0;
static int pareggi = 0;
static Scanner input = new Scanner(System.in);

public static void main(String[] args) {

    char utente;
    char pc;

    String msg;

    for (int i = 0; i < 5; i++) {
        //1 chiedere all'utente p,f,c
        utente = sceltaUtente();

        //2 far scegliere un casuale tra pfc alla macchina
        pc = sceltaPC();

        //3 valutare se sono uguali: messaggio = "pareggio"
        if (utente == pc) {
            msg = "pareggio";
            pareggi++;
        } else {
            //4 altrimenti valutare chi ha vinto:

            msg = valutaGioco(utente, pc);

            //4a u == p -> se pc == f -> messaggio = pc perde altrimenti
vince
            //4b u == c -> se pc == p -> messaggio = pc perde altrimenti
vince
            //4c u == f -> se pc == c -> messaggio = pc perde altrimenti
vince
        }

        //5 stampa messaggio
        System.out.println(msg);
    } //chiudo ciclo

    System.out.println("Pareggi " + pareggi);
    System.out.println("Vittorie utente " + vittorieU);
    System.out.println("Vittorie PC " + vittorieP);

    //6 saluta ed esce, oppure rigioca se in un loop
    System.out.println("Grazie per aver giocato!");
    input.close();
}

private static String valutaGioco(char utente, char pc) {
    String s = null;
```

```
        if (utente == 'p') {
            if(pc == 'f') {
                s = "Vince utente";
                vittorieU++;
            } else {
                s = "Vince PC";
                vittorieP++;
            }
        } else if (utente == 'f') {
            if(pc == 'c') {
                s = "Vince utente";
                vittorieU++;
            } else {
                s = "Vince PC";
                vittorieP++;
            }
        } else {
            if(pc == 'p') {
                s = "Vince utente";
                vittorieU++;
            } else {
                s = "Vince PC";
                vittorieP++;
            }
        }
    }

    return s;
}

private static char sceltaPC() {
    char c = 0;
    double d = Math.random();
    if (d < 0.33) {
        c = 'p';
    } else if ( d < 0.66 ) {
        c = 'f';
    } else {
        c = 'c';
    }
    return c;
}

//metodo ausiliario per chiedere la scelta all'utente
private static char sceltaUtente() {

    System.out.println("Scegli tra pietra, forbici, carta");
    System.out.println("p: pietra");
    System.out.println("f: forbici");
    System.out.println("c: carta");
    String s = input.nextLine();
    char c = s.charAt(0);
    //input.close();
    return c;
}
```

```
}  
  
}
```

```
package esercizi;  
  
public class ProvaFor {  
  
    public static void main(String[] args) {  
  
        //struttura di iterazione, quando conosco a priori il numero di  
        ripetizioni  
        for ( int i = 10; i >= 0 ; i-- ) {  
            //questa istruzione si ripeterà 10 volte  
            System.out.println("ciao " + i);  
        }  
  
    }  
  
}
```

```
package esercizi;  
  
public class ProvaIf {  
  
    public static void main(String[] args) {  
  
        double casuale = Math.random();  
  
        System.out.println(casuale);  
  
        if (casuale < 0.5) {  
            System.out.println("La mia risposta è SI!");  
        } else {  
            System.out.println("La mia risposta è NO!");  
        }  
  
    }  
  
}
```

```
package esercizi;  
  
public class TipiInteri {  
  
    public static void main(String[] args) {
```

```
//byte : 8bit -128 +127
//short : 16bit
//int : 32bit
//long : 64bit

int base = 10, altezza = 5;
int area = base * altezza / 2;

System.out.println(area);

base = 23;
altezza = 12;
area = base * altezza / 2;
System.out.println(area);

}

}
```

```
package esercizi;

public class TipoCarattere {

    public static void main(String[] args) {

        //char 16bit

        char a = 980;//65 ASCII
        char b = 'B';//66

        char speciale = '\u0024';

        System.out.println(a);

    }

}
```

```
package esercizi;

public class GiustoSbagliato {

    public static void main(String[] args) {

        int usciteSI = 0;
        int usciteNO = 0;
        double casuale = 0.0;
        final int RIPETIZIONI = 10;
```

```
//creo un ciclo

for ( int i = 0; i < RIPETIZIONI ; i++ ) { //inizio
    casuale = Math.random();

    if (casuale < 0.5) {
        usciteSI++;
    } else {
        usciteNO++;
    }
} //fine ciclo

System.out.println("SI: " +usciteSI);
System.out.println("NO: " +usciteNO);

}

}
```

```
package esercizi;

public class TipiReali {

    public static void main(String[] args) {

        //float 32bit
        //double 64bit

        double prezzo = 4.70;
        final double IVA = 0.22;

        double imposta = prezzo * IVA;
        double totale = prezzo + imposta;

        System.out.println("Prezzo imponibile: " + prezzo );
        System.out.println("Aliquota IVA 22%: " + IVA );
        System.out.println("Totale: " + (prezzo + imposta) );

    }

}
```

```
package esercizi;

public class HelloWorld {

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



```
//variabile di tipo string
String saluto = "Ciao";
System.out.println(saluto);

}

}
```

```
package esercizi;

public class TipoBoolean {

    public static void main(String[] args) {

        //boolean true, false

        boolean acceso = false;

        boolean caldo = true;

        caldo = false;

        acceso = true;

        if ( 4 != 5 ) {
            System.out.println("Il motore è acceso");
        } else {
            System.out.println("Il motore è spento");
        }

    }

}
```

```
package esercizi;

import java.util.Scanner;

public class Saluto {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Come ti chiami?");
        String risposta = sc.nextLine();

        System.out.println("Benvenuto " + risposta);

    }

}
```

```
}  
  
}
```

```
package model;  
  
import java.util.Date;  
  
public class Noleggio {  
  
    private Date dataNoleggio;  
    private Autista autista;  
    private Automobile auto;  
    private int giorni;  
  
    public Noleggio(Autista autista, Automobile auto, int giorni) {  
        this.autista = autista;  
        this.auto = auto;  
        this.giorni = giorni;  
        this.dataNoleggio = new Date();  
    }  
  
    @Override  
    public String toString() {  
        return "Noleggio [dataNoleggio=" + dataNoleggio + ", autista=" +  
autista + ", auto=" + auto + ", giorni=" +  
        + giorni + " ]";  
    }  
  
}
```

```
package model;  
  
public class Autista {  
  
    private String nome, cognome, patente;  
  
    public Autista(String nome, String cognome, String patente) {  
        this.nome = nome;  
        this.cognome = cognome;  
        this.patente = patente;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public String getCognome() {  
        return cognome;  
    }  
  
}
```

```
    }

    public String getPatente() {
        return patente;
    }

    @Override
    public String toString() {
        return "Autista [nome=" + nome + ", cognome=" + cognome + ",\npatente=" + patente + "]\n";
    }

}
```

```
package model;

public class Automobile {

    private String marca, modello;
    private int cilindrata, ruoteMotrici, consumo;

    public Automobile(String marca, String modello, int cilindrata, int
ruoteMotrici, int consumo) {
        this.marca = marca;
        this.modello = modello;
        this.cilindrata = cilindrata;
        this.ruoteMotrici = ruoteMotrici;
        this.consumo = consumo;
    }

    public Automobile() {
        this.marca = "Generica";
        this.modello = "Generale";
    }

    public String getMarca() {
        return marca;
    }

    public void setMarca(String marca) {
        this.marca = marca;
    }

    public String getModello() {
        return modello;
    }

    public void setModello(String modello) {
        this.modello = modello;
    }
}
```

```
}

public int getCilindrata() {
    return cilindrata;
}

public void setCilindrata(int cilindrata) {
    this.cilindrata = cilindrata;
}

public int getRuoteMotrici() {
    return ruoteMotrici;
}

public void setRuoteMotrici(int ruoteMotrici) {
    this.ruoteMotrici = ruoteMotrici;
}

public int getConsumo() {
    return consumo;
}

public void setConsumo(int consumo) {
    this.consumo = consumo;
}

@Override
public String toString() {
    return "Automobile [marca=" + marca + ", modello=" + modello + ",\ncilindrata=" + cilindrata + "]\n";
}

}
```

```
package view;

import java.util.Scanner;

import controller.AutistaController;
import controller.AutomobileController;
import controller.NoleggioController;
import model.Autista;
import model.Automobile;
import model.Noleggio;

public class GestioneAutoNoleggio {

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

```
Scanner sc = new Scanner(System.in);

boolean ripeti = true;

String risposta = null;
String cliente = null;
String auto = null;
int nGiorni = 0;

AutomobileController ac = new AutomobileController();
AutistaController auc = new AutistaController();
NoleggioController nc = new NoleggioController();

while(ripeti) {
    System.out.println("Scegli opzione");
    System.out.println("a) elenco mezzi");
    System.out.println("b) elenco autisti");
    System.out.println("c) crea noleggio");
    System.out.println("e) elenco noleggi");
    System.out.println("q) per uscire");

    risposta = sc.nextLine();

    if(risposta.equals("a")) {
        for (Automobile automob : ac.getAuto()) {
            System.out.println(automob);
        }
    }

    if(risposta.equals("b")) {
        for (Autista a : auc.getAutisti()) {
            System.out.println(a);
        }
    }

    if(risposta.equals("c")) {
        System.out.println("Codice cliente");
        cliente = sc.nextLine();
        System.out.println("Codice auto");
        auto = sc.nextLine();
        System.out.println("numero giorni");
        nGiorni = sc.nextInt();

        Autista autista = auc.getAutista(cliente);
        Automobile automobile = ac.getAutoByTarga(auto);

        nc.noleggia(autista, automobile, nGiorni);
    }

    if (risposta.equals("e")) {
        for (Noleggio noleggio : nc.getNoleggi()) {
            System.out.println(noleggio);
        }
    }
}
```

```

        }

        }

        if (risposta.equals("q")) {
            System.out.println("arrivederci");
            ripeti = false;
        }

    }

}

```

```

package controller;

import model.Autista;

public class AutistaController {

    private Autista[] autisti = {
        new Autista("paolino", "paperino", "PAP12"),
        new Autista("pippo", "pippi", "PIP36"),
        new Autista("pluto", "pluti", "PLU45"),
    };

    public Autista[] getAutisti() {
        return autisti;
    }

    public Autista getAutista(String nome) {
        for (Autista autista : autisti) {
            if (autista.getNome().equals(nome))
                return autista;
        }
        return null;
    }

}

```

```

package controller;

import model.Autista;
import model.Automobile;
import model.Noleggio;

```

```
public class NoleggioController {

    private Noleggio[] noleggi = new Noleggio[100];
    private static int nNoleggi = 0;

    public Noleggio[] getNoleggi() {
        return noleggi;
    }

    public void noleggia(Autista a, Automobile auto, int giorni) {

        noleggi[nNoleggi] = new Noleggio(a, auto, giorni);
        nNoleggi++;
    }

}
```

```
package controller;

import model.Automobile;

public class AutomobileController {

    private Automobile[] auto = {
        new Automobile(),
        new Automobile("fiat", "panda", 1100, 2, 12)
    };

    public Automobile[] getAuto() {
        return auto;
    }

    public Automobile getOne(int i) {
        return auto[i];
    }

    public Automobile getAutoByTarga(String targa) {
        for (Automobile automobile : auto) {
            if (automobile.getModello().equals(targa))
                return automobile;
        }
        return null;
    }

}
```

```
package model;

public class Libro {

    private String titolo;
    private String codice;
    private double prezzo;

    public Libro(String titolo, String codice, double prezzo) {
        this.titolo = titolo;
        this.codice = codice;
        this.prezzo = prezzo;
    }

    public String getTitolo() {
        return titolo;
    }

    public void setTitolo(String titolo) {
        this.titolo = titolo;
    }

    public String getCodice() {
        return codice;
    }

    public void setCodice(String codice) {
        this.codice = codice;
    }

    public double getPrezzo() {
        return prezzo;
    }

    public void setPrezzo(double prezzo) {
        this.prezzo = prezzo;
    }

    @Override
    public String toString() {
        return "Libro [titolo=" + titolo + ", codice=" + codice + ",  
prezzo=" + prezzo + "]\n";
    }

}
```

```
package view;

import java.util.ArrayList;
```



```
import controller.LibroCtrl;
import model.Libro;

public class MultiStore {

    public static void main(String[] args) {

        Libro l1 = new Libro("E babbo morto. Una storia di Natale", "B01",
10.45);
        Libro l2 = new Libro("A Cambiare l'acqua ai fiori", "B02", 17.10);

        LibroCtrl lc = new LibroCtrl();

        ArrayList<Libro> libri = lc.getLibri();

        System.out.println(libri.size());
        lc.addLibro(l1);
        libri.add(l2);
        libri.add(l2);
        libri.add(l2);
        lc.addLibro("Fiori per i Bastardi di Pizzofalcone", "B03", 17.57);
        //System.out.println(libri.get(0));
        libri.remove(0);

        for (Libro libro : libri) {
            System.out.println(libro);
        }

    }

}
```

```
package controller;

import java.util.ArrayList;

import model.Libro;

public class LibroCtrl {

    //Libro[] non è ridimensionabile: length è una proprietà, nota già
    all'inizio
    //ArrayList: è una specie di array, ridimensionabile, solo OGGETTI
    private ArrayList<Libro> libri;

    public LibroCtrl() {
        this.libri = new ArrayList<Libro>();
    }

}
```

```
public ArrayList<Libro> getLibri() {
    return libri;
}

public void addLibro(Libro l) {
    if(l.getTitolo().startsWith("A"))
        return;
    this.libri.add(l);
}

//overload dei metodi
public void addLibro(String titolo, String codice, double prezzo) {
    if(titolo.startsWith("A"))
        return;
    Libro l = new Libro(titolo, codice, prezzo);
    this.libri.add(l);
}

}
```