

Codice Java corso TSS 2021

./Prj_05_BloccoNote/src/model/Nota.java

```
package model;

public class Nota {

    //proprietà, attributi, fields
    public String testo;
    public boolean completa;

    //costruttore dell'oggetto nota, prende in argomento un testo da
    //inserire nella nota e
    //di default assegna il valore false alla proprietà della nota
    public Nota(String testo) {
        //System.out.println("Creazione nota con testo " + testo);
        this.testo = testo;
        this.completa = false;
    }

    //il metodo setCompleta serve per dichiarare completata la nota, prende
    //in argomento un valore
    //boolean e lo assegna come valore della proprietà "completa"
    public void setCompleta(boolean b) {
        this.completa = b;
    }

    @Override
    public String toString() {

        String done = "[ ] ";
        if (completa) {
            done = "[*] ";
        }

        return "" + done + testo;
    }
}
```

./Prj_05_BloccoNote/src/view/BloccoNote.java

```
package view;

import java.util.Scanner;
```

```
import controller.NoteController;

public class BloccoNote {

    //stato interno
    static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) {

        NoteController bloccoBlu = new NoteController();

        BloccoNote.menu();

        String nota = aggiungiNota();
        int pos = posizione();

        bloccoBlu.aggiungi(nota, pos);
        bloccoBlu.aggiungi(aggiungiNota(), posizione());
        bloccoBlu.aggiungi(aggiungiNota(), posizione());
        bloccoBlu.aggiungi(aggiungiNota(), posizione());
        bloccoBlu.aggiungi(aggiungiNota(), posizione());

        bloccoBlu.mostraTutte();
        sc.close();
    }

    private static void menu() {
        System.out.println("Blocco Note");
        System.out.println("1) per inserire una nuova nota");
        System.out.println("2) per completare una nota");
        System.out.println("3) mostra elenco completo");
        System.out.println("0) per uscire");
    }

    private static String aggiungiNota() {

        String s = sc.nextLine();
        return s;
    }

    private static int posizione() {
        int i = sc.nextInt();
        return i;
    }

}
```

./Prj_05_BloccoNote/src/controller/NoteController.java

```
package controller;

import model.Nota;

public class NoteController {

    public Nota[] lista;
    private final int DIM = 10;

    public NoteController() {
        this.lista = new Nota[DIM];
        //System.out.println("Ho costruito un oggetto di tipo controller
delle note: NoteController");
    }

    public void aggiungi(String s, int pos) {
        //creo una nuova nota
        Nota n = new Nota(s);
        //inserisco la nota nell'array di note
        lista[pos] = n;
        //System.out.println("Nota creata e aggiunta");
    }

    //questo metodo modifica la proprieta completa della nota con posizione
    passata in argomento
    public void completa(int i) {
        this.lista[i].setCompleta(true);
    }

    public void mostraTutte() {
        System.out.println("-----");
        System.out.println("-----BLOCCO NOTE-----");
        System.out.println("-----");
        for (int i = 0; i < lista.length; i++) {
            if(lista[i] == null ) {
                System.out.println("*****");
            } else

                System.out.println(lista[i]);
        }
        System.out.println("-----");
    }

}
```

./PietraForbiciCarta/src/giochi/CartaForbiciSasso.java

```
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;
    }

}
```

./Prj_09_DistributoreAutomatico2/src/model/Alimento.java

```
package model;

public class Alimento {

    //proprietà
    private String nome;
    private double prezzo;
    private final String TEMP = "4°";

    public Alimento(String nome) {
        this.nome = nome;
        this.prezzo = 0.30;
    }

    public Alimento(String nome, double prezzo) {
        this.nome = nome;
        this.prezzo = prezzo;
    }

    //getters setters
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public double getPrezzo() {
        return prezzo;
    }
    public void setPrezzo(double prezzo) {
        this.prezzo = prezzo;
    }

    @Override
    public String toString() {
        return "" + nome + " - € " + prezzo + " " + TEMP;
    }
}
```

```
}
```

./Prj_09_DistributoreAutomatico2/src/model/Caffe.java

```
package model;

public class Caffe extends Bevanda{

    public Caffe(double prezzo) {
        super("Caff ");
        this.prezzo = prezzo;
        this.acqua = 80;
        System.out.println("Macina la q di caff  desiderata per il tipo di caff ");
        System.out.println("Aggiungi l'acqua calda");
        System.out.println("versa il caff  nel bicchiere");
        System.out.println("bevanda pronta");
        System.out.println("emetti un segnale sonoro: beeeep");
        System.out.println("scrivi sul display bevanda pronta");
    }

    @Override
    public String toString() {
        return "[nome=" + this.nome + ", prezzo=" + prezzo + ", TEMP=" + TEMP + ", acqua=" + acqua + "]";
    }

}
```

./Prj_09_DistributoreAutomatico2/src/model/Bevanda.java

```
package model;

public class Bevanda {

    //propriet 
    protected String nome;
    protected double prezzo;
    protected String TEMP = "40 ";
    int acqua;

}
```

```

    public Bevanda(String nome) {
        this.nome = nome;
        this.prezzo = 0.30;
        System.out.println("Fai cadere il bicchiere");
        System.out.println("Se l'utente vuole lo zucchero, metti la q
desiderata");
        System.out.println("fai cadere la palettina, se vuole lo
zucchero");
    }

    public Bevanda(String nome, double prezzo) {
        this.nome = nome;
        this.prezzo = prezzo;
    }

    //getters setters
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public double getPrezzo() {
        return prezzo;
    }
    public void setPrezzo(double prezzo) {
        this.prezzo = prezzo;
    }

    @Override
    public String toString() {
        return "" + nome + " - € " + prezzo + " " + TEMP;
    }

}

```

./Prj_09_DistributoreAutomatico2/src/model/The.java

```

package model;

public class The extends Bevanda {

    public The(double prezzo) {
        super("The");
        this.prezzo = prezzo;
        this.acqua = 160;
        this.TEMP = "30°";
    }
}

```



```
        System.out.println("Aggiungi la q di polvere di the");
        System.out.println("Aggiungi l'acqua calda");
        System.out.println("versa il the nel bicchiere");
        System.out.println("bevanda pronta");
        System.out.println("emetti un segnale sonoro: beeeeeep");
        System.out.println("scrivi sul display bevanda pronta");
    }

    @Override
    public String toString() {
        return "[nome=" + nome + ", prezzo=" + prezzo + ", TEMP=" + TEMP +
", acqua=" + acqua + "]";
    }

}
```

./Prj_09_DistributoreAutomatico2/src/model/Ginseng.java

```
package model;

public class Ginseng extends Bevanda {

    public Ginseng(String nome) {
        super(nome);
        System.out.println("ho preparato un ginseng solo per matteo");
    }

}
```

./Prj_09_DistributoreAutomatico2/src/view/Distributore.java

```
package view;

import java.util.Scanner;

import controller.DistributoreCtrl;
import controller.Messaggi;
import model.Alimento;
import model.Bevanda;
import model.Caffe;
import model.Ginseng;
import model.The;
```

```
public class Distributore {

    static DistributoreCtrl controllerBevande = new DistributoreCtrl();
    static DistributoreCtrl controllerAlimenti = new DistributoreCtrl();

    public static void main(String[] args) {

        Bevanda b = new Bevanda("caffè");
        Caffè c = new Caffè(0.3);
        c.setNome("caffè lavazza oro");
        The t = new The(0.3);
        Ginseng g = new Ginseng("Per Matteo");
        g.setNome("Matteo non lo vuole più");
        System.out.println("-----");
        System.out.println(b);
        System.out.println("-----");
        System.out.println(c);
        System.out.println("-----");
        System.out.println(t);
        System.out.println("-----");
        System.out.println(g);
        System.out.println("-----");

        System.out.println(Messaggi.NOME);

        System.out.println(Messaggi.SCEGLI);

        String nomeBevanda = scegliBevanda();

        System.out.println("Hai scelto " + nomeBevanda);

        double prezzo = 0.30;

        boolean pagato = faiPagare(prezzo);

        if (pagato) {

            System.out.println(Messaggi.PREPARAZIONE);
            boolean pronta = preparazione(nomeBevanda);

            if (pronta) {
                System.out.println("E' pronta");
                System.out.println("beeeep");
            }

            } else {

            }

        }

        private static boolean faiPagare(double prezzo) {
```

```
        boolean tuttoOk = false;
        double monete = 0.0;

        System.out.println(prezzo);
        System.out.println(Messaggi.INSERISCI);

        Scanner sc = new Scanner(System.in);
        monete = sc.nextDouble();

        if (monete == prezzo) {
            tuttoOk = true;
        } else if (monete < prezzo) {
            System.out.println("Denaro insufficiente");
        } else {
            System.out.println("resto = " + (monete - prezzo));
            tuttoOk = true;
        }
        sc.close();
        return tuttoOk;
    }

    private static boolean preparazione(String nomeBevanda) {
        boolean preparazione = false;
        switch (nomeBevanda) {
            case "caffè":
                Caffè pronta = controllerBevande.getCaffè();
                System.out.println("La bevanda " + pronta);
                preparazione = true;
                break;
            case "the":
                The t = controllerBevande.getThe();
                System.out.println("La bevanda " + t);
                preparazione = true;
                break;
            default:
                System.out.println("Bevanda non disponibile");
                break;
        }
        return preparazione;
    }

    private static String scegliBevanda() {
        Scanner sc = new Scanner(System.in);
        String scelta = sc.nextLine();

        return scelta;
    }
}
```

./Prj_09_DistributoreAutomatico2/src/controller/Messaggi.java

```
package controller;

public class Messaggi {
    //static: appartiene alla classe Distributore e NON agli oggetti di
    tipo distributore
    public static final String NOME = "Distributore automatico";

    public static final String SCEGLI = "Scegli alimento o bevanda";

    public static final String INSERISCI = "Inserisci importo";

    public static final String PREPARAZIONE = "Bevanda in preparazione";
}
```

./Prj_09_DistributoreAutomatico2/src/controller/DistributoreCtrl.java

```
package controller;

import model.Alimento;
import model.Bevanda;
import model.Caffe;
import model.The;

public class DistributoreCtrl {

    public Bevanda getBevanda(String nome) {

        Bevanda b = new Bevanda(nome);
        System.out.println("bevanda preparata");
        return b;
    }

    public Caffe getCaffe() {
        Caffe c = new Caffe(0.3);
        return c;
    }

    public The getThe() {
        The t = new The(0.3);
        return t;
    }

    public Alimento getAlimento(String nome) {
        Alimento a = new Alimento(nome);
        System.out.println("Alimento prelevato");
    }
}
```

```
        return a;
    }
}
```

./Prj_03_Giochi/src/programmini/GiustaDecisione.java

```
package programmini;

public class GiustaDecisione {

    public void gioca() {

        final int RIPETIZIONI = 10;

        String decisione1 = "SI";
        String decisione2 = "NO";
        double casuale;
        int numeroSi=0, numeroNo=0;

        //inizializzo un ciclo for
        for (int i = 0; i < RIPETIZIONI; i++) {

            casuale = Math.random();

            if ( casuale < 0.5 ) {

                //System.out.println(decisione1);
                numeroSi++;

            }
            else {
                //System.out.println(decisione2);
                numeroNo++;
            }
        }
        System.out.println("-----");
        System.out.println("La risposta è: ");
        if (numeroSi == numeroNo)
            System.out.println("NON SO");
        else if (numeroSi > numeroNo)
            System.out.println(decisione1);
        else
            System.out.println(decisione2);
        System.out.println("-----");

    }
}
```

./Prj_03_Giochi/src/programmini/GuessNumber.java

```
package programmini;

import java.util.Scanner;

public class GuessNumber {

    public void gioca() {

        boolean trovato = false;
        int contatore = 0;

        int x = (int) ( Math.random() * 100 ) + 1;

        Scanner input = new Scanner(System.in);

        System.out.println("come ti chiami");
        String nome = input.nextLine();
        System.out.println("Benvenuto " + nome + " gioca con noi");

        while(!trovato) {
            contatore++;
            System.out.println("Inserisci un numero intero tra 1 e 100");

            int a = input.nextInt();

            if (x == a) {
                System.out.println("hai indovinato in " + contatore + " tentativi!");
                trovato = true;
            }
            else if (x < a) {
                System.out.println("troppo grande");
            }
            else {
                System.out.println("troppo piccolo");
            }

        } //fine ciclo while
    }

}
```

./Prj_03_Giochi/src/programmini/Giochi.java

```
package programmini;

import java.util.Scanner;
```

```
public class Giochi {

    static Scanner scan = new Scanner(System.in);

    public static void main(String[] args) {

        boolean esci = false;

        while(!esci) {
            menu();
            int i = getInput("Scegli un numero per continuare");

            switch (i) {
                case 1:
                    System.out.println("Vuoi giocare a Giusta decisione");
                    //GiustaDecisione.gioca();
                    GiustaDecisione gd = new GiustaDecisione();
                    gd.gioca();

                    break;
                case 2:
                    System.out.println("Vuoi giocare a Indovina il numero");
                    GuessNumber gn = new GuessNumber();
                    gn.gioca();

                    break;
                case 3:
                    System.out.println("Vuoi giocare a PFC");
                    PietraForbiciCarta pfc = new PietraForbiciCarta();
                    pfc.gioca();
                    break;
                case 0:
                    esci = true;
                    System.out.println("arrivederci...");
                    break;
                default:
                    System.out.println("Opzione non disponibile");
                    break;
            }
        }

        //metodi ausiliari
        private static int getInput(String messaggio) {
            System.out.println(messaggio);
            int s = Giochi.scan.nextInt();
            System.out.println("Hai scritto " + s);
            return s;
        }

        private static void menu() {
```

```
        System.out.println("-----");
        System.out.println("1) Giusta decisione");
        System.out.println("2) Indovina il numero");
        System.out.println("3) Pietra, forbici, carta");
        System.out.println("0) Esci");
        System.out.println("-----");
    }

}
```

./Prj_03_Giochi/src/programmini/PietraForbiciCarta.java

```
package programmini;

import java.util.Scanner;

public class PietraForbiciCarta {

    String[] mosse = {"pietra", "forbici", "carta"};

    Scanner sc = new Scanner(System.in);

    final String MSG = "Fai la mossa giusta!";
    final String VPC = "Vince computer";
    final String VUM = "Vince giocatore umano";
    final String VPA = "Pareggio";

    int sceltaUmano=0, sceltaPC=0, vittorieU, vittorieP, pareggi, partite;

    public void scegli() {
        System.out.println(MSG);
        for (int i = 0; i < mosse.length; i++) {
            System.out.println("Scegli " + i + " per " + mosse[i]);
        }

        sceltaUmano = sc.nextInt();
    }

    public void sceltaCasuale() {
        sceltaPC = (int) (Math.random() * mosse.length);
    }

    public void valutazione() {

        partite++;
        System.out.println("Umano ha scelto " + mosse[sceitaUmano]);
        System.out.println("PC ha scelto " + mosse[sceitaPC]);
    }
}
```



```
        if (sceltaUmano == sceltaPC) {
            pareggi++;
            System.out.println(VPA);
        }
        else {

            switch (sceltaUmano) {
                case 0:
                    if (sceltaPC == 1) {
                        vittorieU++;
                    } else {
                        vittorieP++;
                    }
                    break;
                case 1:
                    if (sceltaPC == 2) {
                        vittorieU++;
                    } else {
                        vittorieP++;
                    }
                    break;
                case 2:
                    if (sceltaPC == 0) {
                        vittorieU++;
                    } else {
                        vittorieP++;
                    }
                    break;

                default:
                    System.out.println("Errore di sistema...");
                    break;
            }
        }
    }

    public void mostraPunteggi() {
        System.out.println("Partite giocate " + partite);
        System.out.println("Pareggi " + pareggi);
        System.out.println("Vittorie umano " + vittorieU);
        System.out.println("Vittorie pc " + vittorieP);
    }

    public void gioca() {
        //PietraForbiciCarta pfc = new PietraForbiciCarta();
        boolean gioca = true;

        while(gioca) {
            scegli();
            sceltaCasuale();
            valutazione();
            if (vittorieU == 3 || vittorieP == 3)
                gioca = false;
        }
    }
}
```

```

    }

    mostraPunteggi();
}

}

```

./metadata/.plugins/org.eclipse.wst.server.core/tmp0/work/Catalina/localhost/Prj_13_VideoGames/org/apache/jsp/index_jsp.java

```

/*
 * Generated by the Jasper component of Apache Tomcat
 * Version: Apache Tomcat/9.0.41
 * Generated at: 2020-12-18 14:53:45 UTC
 * Note: The last modified time of this file was set to
 *       the last modified time of the source file after
 *       generation to assist with modification tracking.
 */
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class index_jsp extends org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent,
               org.apache.jasper.runtime.JspSourceImports {

    private static final javax.servlet.jsp.JspFactory _jspxFactory =
        javax.servlet.jsp.JspFactory.getDefaultFactory();

    private static java.util.Map<java.lang.String,java.lang.Long>
    _jspx_dependants;

    static {
        _jspx_dependants = new
java.util.HashMap<java.lang.String,java.lang.Long>(2);
        _jspx_dependants.put("/login.html", Long.valueOf(1608302476277L));
        _jspx_dependants.put("/riservata.jsp", Long.valueOf(1608302974695L));
    }

    private static final java.util.Set<java.lang.String>
    _jspx_imports_packages;

    private static final java.util.Set<java.lang.String>
    _jspx_imports_classes;

    static {
        _jspx_imports_packages = new java.util.HashSet<>();
    }

```

```

        _jspx_imports_packages.add("javax.servlet");
        _jspx_imports_packages.add("javax.servlet.http");
        _jspx_imports_packages.add("javax.servlet.jsp");
        _jspx_imports_classes = null;
    }

    private volatile javax.el.ExpressionFactory _el_expressionfactory;
    private volatile org.apache.tomcat.InstanceManager _jsp_instancemanager;

    public java.util.Map<java.lang.String,java.lang.Long> getDependants() {
        return _jspx_dependants;
    }

    public java.util.Set<java.lang.String> getPackageImports() {
        return _jspx_imports_packages;
    }

    public java.util.Set<java.lang.String> getClassImports() {
        return _jspx_imports_classes;
    }

    public javax.el.ExpressionFactory _jsp_getExpressionFactory() {
        if (_el_expressionfactory == null) {
            synchronized (this) {
                if (_el_expressionfactory == null) {
                    _el_expressionfactory =
                    _jspxFactory.getJspApplicationContext(getServletConfig().getServletContext(
                    )).getExpressionFactory();
                }
            }
        }
        return _el_expressionfactory;
    }

    public org.apache.tomcat.InstanceManager _jsp_getInstanceManager() {
        if (_jsp_instancemanager == null) {
            synchronized (this) {
                if (_jsp_instancemanager == null) {
                    _jsp_instancemanager =
                    org.apache.jasper.runtime.InstanceManagerFactory.getInstanceManager(getServ
                    letConfig());
                }
            }
        }
        return _jsp_instancemanager;
    }

    public void _jspInit() {
    }

    public void _jspDestroy() {
    }

    public void _jspService(final javax.servlet.http.HttpServletRequest

```

```

request, final javax.servlet.http.HttpServletResponse response)
    throws java.io.IOException, javax.servlet.ServletException {

    if
(!javax.servlet.DispatcherType.ERROR.equals(request.getDispatcherType())) {
        final java.lang.String _jspx_method = request.getMethod();
        if ("OPTIONS".equals(_jspx_method)) {
            response.setHeader("Allow", "GET, HEAD, POST, OPTIONS");
            return;
        }
        if (!"GET".equals(_jspx_method) && !"POST".equals(_jspx_method) &&
!"HEAD".equals(_jspx_method)) {
            response.setHeader("Allow", "GET, HEAD, POST, OPTIONS");
            response.sendError(HttpServletResponse.SC_METHOD_NOT_ALLOWED, "JSPs
only permit GET, POST or HEAD. Jasper also permits OPTIONS");
            return;
        }
    }

    final javax.servlet.jsp.PageContext pageContext;
    javax.servlet.http.HttpSession session = null;
    final javax.servlet.ServletContext application;
    final javax.servlet.ServletConfig config;
    javax.servlet.jsp.JspWriter out = null;
    final java.lang.Object page = this;
    javax.servlet.jsp.JspWriter _jspx_out = null;
    javax.servlet.jsp.PageContext _jspx_page_context = null;

    try {
        response.setContentType("text/html; charset=UTF-8");
        pageContext = _jspxFactory.getPageContext(this, request, response,
            null, true, 8192, true);
        _jspx_page_context = pageContext;
        application = pageContext.getServletContext();
        config = pageContext.getServletConfig();
        session = pageContext.getSession();
        out = pageContext.getOut();
        _jspx_out = out;

        out.write("\n");
        out.write("<!DOCTYPE html>\n");
        out.write("<html>\n");
        out.write("<head>\n");
        out.write("<meta charset=\"UTF-8\">\n");
        out.write("<title>VideoGiochi</title>\n");
        out.write("\n");
        out.write("<link rel=\"stylesheet\" href=\"css/style.css\">\n");
        out.write("</head>\n");
        out.write("<body>\n");
        out.write("<h1>VideoGiochi</h1>\n");
        out.write("<p>Il mio sito web di videogiochi</p>\n");
        out.write("\n");
        out.write("\n");
    }
}

```

```
        out.write("\n");
        out.write("\n");

        if (request.getParameter("nome") != null &&
            request.getParameter("password") != null){
            String nome = request.getParameter("nome");
            String passw = request.getParameter("password");

            if (nome.equals("mauro") && passw.equals("12345")){
                out.print("Benvenuto utente " + nome);

                out.write("\n");
                out.write("\t\t\t\n");
                out.write("\t\t\t");
                out.write("<h1>Area riservata</h1>\n");
                out.write("\n");
                out.write("<h2>Benvenuto ");
                out.print(nome);
                out.write("</h2>");
                out.write("\t\n");
                out.write("\t\t\t\n");
                out.write("\t\t\t");

            } else {
                out.print("Non puoi entrare!");
            }
        } else {

            out.write("\n");
            out.write("\t\t\t\n");
            out.write("\t\t\t");
            out.write("<form action=\"\" method=\"post\">\n");
            out.write("\t<input type=\"text\" name=\"nome\"");
            out.write("placeholder=\"inserire username\">\n");
            out.write("\t<input type=\"password\" name=\"password\"");
            out.write("placeholder=\"inserire password\">\n");
            out.write("\t<input type=\"submit\" value=\"Login\">\n");
            out.write("</form>\t");
            out.write("\t\n");
            out.write("\t\t\t\n");
            out.write("\t\t\t");

            out.print("inserisci user e pw per entrare");
        }

        out.write("\n");
        out.write("<script src=\"js/script.js\"></script>\n");
        out.write("</body>\n");
        out.write("</html>");
```

```

    } catch (java.lang.Throwable t) {
        if (!(t instanceof javax.servlet.jsp.SkipPageException)){
            out = _jspx_out;
            if (out != null && out.getBufferSize() != 0)
                try {
                    if (response.isCommitted()) {
                        out.flush();
                    } else {
                        out.clearBuffer();
                    }
                } catch (java.io.IOException e) {}
            if (_jspx_page_context != null)
                _jspx_page_context.handlePageException(t);
            else throw new ServletException(t);
        }
    } finally {
        _jspxFactory.releasePageContext(_jspx_page_context);
    }
}
}
}

```

./metadata/.plugins/org.eclipse.wst.server.core/tmp0/work/Catalina/localhost/Prj_13_VideoGames/org/apache/jsp/riservata_jsp.java

```

/*
 * Generated by the Jasper component of Apache Tomcat
 * Version: Apache Tomcat/9.0.41
 * Generated at: 2020-12-18 14:47:16 UTC
 * Note: The last modified time of this file was set to
 *       the last modified time of the source file after
 *       generation to assist with modification tracking.
 */
package org.apache.jsp;

import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.jsp.*;

public final class riservata_jsp extends
org.apache.jasper.runtime.HttpJspBase
    implements org.apache.jasper.runtime.JspSourceDependent,
               org.apache.jasper.runtime.JspSourceImports {

    private static final javax.servlet.jsp.JspFactory _jspxFactory =
        javax.servlet.jsp.JspFactory.getDefaultFactory();

    private static java.util.Map<java.lang.String,java.lang.Long>
    _jspx_dependants;

    private static final java.util.Set<java.lang.String>

```

```
_jspx_imports_packages;

private static final java.util.Set<java.lang.String>
_jsp_imports_classes;

static {
    _jspx_imports_packages = new java.util.HashSet<>();
    _jspx_imports_packages.add("javax.servlet");
    _jspx_imports_packages.add("javax.servlet.http");
    _jspx_imports_packages.add("javax.servlet.jsp");
    _jspx_imports_classes = null;
}

private volatile javax.el.ExpressionFactory _el_expressionfactory;
private volatile org.apache.tomcat.InstanceManager _jsp_instancemanager;

public java.util.Map<java.lang.String,java.lang.Long> getDependants() {
    return _jspx_dependants;
}

public java.util.Set<java.lang.String> getPackageImports() {
    return _jspx_imports_packages;
}

public java.util.Set<java.lang.String> getClassImports() {
    return _jspx_imports_classes;
}

public javax.el.ExpressionFactory _jsp_getExpressionFactory() {
    if (_el_expressionfactory == null) {
        synchronized (this) {
            if (_el_expressionfactory == null) {
                _el_expressionfactory =
_jspFactory.getJspApplicationContext(getServletConfig().getServletContext(
)).getExpressionFactory();
            }
        }
    }
    return _el_expressionfactory;
}

public org.apache.tomcat.InstanceManager _jsp_getInstanceManager() {
    if (_jsp_instancemanager == null) {
        synchronized (this) {
            if (_jsp_instancemanager == null) {
                _jsp_instancemanager =
org.apache.jasper.runtime.InstanceManagerFactory.getInstanceManager(getServ
letConfig());
            }
        }
    }
    return _jsp_instancemanager;
}
```

```

    public void _jspInit() {
    }

    public void _jspDestroy() {
    }

    public void _jspService(final javax.servlet.http.HttpServletRequest
request, final javax.servlet.http.HttpServletResponse response)
        throws java.io.IOException, javax.servlet.ServletException {

        if
(!javax.servlet.DispatcherType.ERROR.equals(request.getDispatcherType())) {
            final java.lang.String _jspx_method = request.getMethod();
            if ("OPTIONS".equals(_jspx_method)) {
                response.setHeader("Allow", "GET, HEAD, POST, OPTIONS");
                return;
            }
            if (!"GET".equals(_jspx_method) && !"POST".equals(_jspx_method) &&
!"HEAD".equals(_jspx_method)) {
                response.setHeader("Allow", "GET, HEAD, POST, OPTIONS");
                response.sendError(HttpServletResponse.SC_METHOD_NOT_ALLOWED, "JSPs
only permit GET, POST or HEAD. Jasper also permits OPTIONS");
                return;
            }
        }

        final javax.servlet.jsp.PageContext pageContext;
        javax.servlet.http.HttpSession session = null;
        final javax.servlet.ServletContext application;
        final javax.servlet.ServletConfig config;
        javax.servlet.jsp.JspWriter out = null;
        final java.lang.Object page = this;
        javax.servlet.jsp.JspWriter _jspx_out = null;
        javax.servlet.jsp.PageContext _jspx_page_context = null;

        try {
            response.setContentType("text/html");
            pageContext = _jspxFactory.getPageContext(this, request, response,
                null, true, 8192, true);
            _jspx_page_context = pageContext;
            application = pageContext.getServletContext();
            config = pageContext.getServletConfig();
            session = pageContext.getSession();
            out = pageContext.getOut();
            _jspx_out = out;

            out.write("<h1>Area riservata</h1>\n");
            out.write("\n");
            out.print(nome);
        } catch (java.lang.Throwable t) {
            if (!(t instanceof javax.servlet.jsp.SkipPageException)){
                out = _jspx_out;
                if (out != null && out.getBufferSize() != 0)

```



```
        try {
            if (response.isCommitted()) {
                out.flush();
            } else {
                out.clearBuffer();
            }
        } catch (java.io.IOException e) {}
        if (_jspx_page_context != null)
            _jspx_page_context.handlePageException(t);
        else throw new ServletException(t);
    }
    finally {
        _jspxFactory.releasePageContext(_jspx_page_context);
    }
}
}
```

./Prj_09_DistributoreAutomatico/src/model/Alimento.java

```
package model;

public class Alimento {

    //proprietà
    private String nome;
    private double prezzo;
    private final String TEMP = "4°";

    public Alimento(String nome) {
        this.nome = nome;
        this.prezzo = 0.30;
    }

    public Alimento(String nome, double prezzo) {
        this.nome = nome;
        this.prezzo = prezzo;
    }

    //getters setters
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public double getPrezzo() {
        return prezzo;
    }
    public void setPrezzo(double prezzo) {
        this.prezzo = prezzo;
    }
}
```

```
    }

    @Override
    public String toString() {
        return "" + nome + " - € " + prezzo + " " + TEMP;
    }
}
```

./Prj_09_DistributoreAutomatico/src/model/Bevanda.java

```
package model;

public class Bevanda {

    //proprietà
    private String nome;
    private double prezzo;
    private final String TEMP = "40°";

    public Bevanda(String nome) {
        this.nome = nome;
        this.prezzo = 0.30;
    }

    public Bevanda(String nome, double prezzo) {
        this.nome = nome;
        this.prezzo = prezzo;
    }

    //getters setters
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public double getPrezzo() {
        return prezzo;
    }
    public void setPrezzo(double prezzo) {
        this.prezzo = prezzo;
    }

    @Override
    public String toString() {
        return "" + nome + " - € " + prezzo + " " + TEMP;
    }
}
```

```
}
```

./Prj_09_DistributoreAutomatico/src/view/Distributore.java

```
package view;

import controller.DistributoreCtrl;
import model.Alimento;
import model.Bevanda;

public class Distributore {

    //static: appartiene alla classe Distributore e NON agli oggetti di
    //tipo distributore
    static final String NOME = "Distributore automatico";

    static final String SCEGLI = "Scegli alimento o bevanda";

    static final String INSERISCI = "Inserisci importo";

    static final String PREPARAZIONE = "Bevanda in preparazione";

    static DistributoreCtrl controllerBevande = new DistributoreCtrl();
    static DistributoreCtrl controllerAlimenti = new DistributoreCtrl();

    public static void main(String[] args) {

        System.out.println(Distributore.NOME);

        System.out.println(Distributore.SCEGLI);

        String nomeBevanda = "Caff  ";

        System.out.println(nomeBevanda);

        double prezzo = 0.30;

        System.out.println(prezzo);

        System.out.println(INSERISCI);

        double monete = 0.30;

        //calcolo resto
        //calcolo del denaro mancante

        if (prezzo == monete) {
            System.out.println(PREPARAZIONE);
            Bevanda pronta = controllerBevande.getBevanda(nomeBevanda);
        }
    }
}
```

```
        System.out.println("La bevanda " + pronta);
        System.out.println("E' pronta");
        System.out.println("beeeep");
    }

    Alimento a = controllerAlimenti.getAlimento("Panino");
    System.out.println(a);
}

}
```

./Prj_09_DistributoreAutomatico/src/controller/DistributoreCtrl.java

```
package controller;

import model.Alimento;
import model.Bevanda;

public class DistributoreCtrl {

    public Bevanda getBevanda(String nome) {

        Bevanda b = new Bevanda(nome);
        System.out.println("bevanda preparata");
        return b;
    }

    public Alimento getAlimento(String nome) {
        Alimento a = new Alimento(nome);
        System.out.println("Alimento prelevato");
        return a;
    }
}
```

./Prj_11_Calcolatrice/src/it/bogliaccino/calcolatrice/Calcolatrice.java

```
package it.bogliaccino.calcolatrice;

public class Calcolatrice {

    public static void main(String[] args) {

        String input1 = MathHelper.getInput(Etichette.OPE1);
        String input2 = MathHelper.getInput(Etichette.OPE2);
        String operatore = MathHelper.getInput(Etichette.OPERATORE);
```

```
//MathHelper.chiudiScanner();

    int inp1 = MathHelper.trasformaIntero(input1);
    int inp2 = MathHelper.trasformaIntero(input2);
    double risultato = MathHelper.calcola(inp1, inp2, operatore);

    System.out.println(input1 + " - " + inp1);
    System.out.println(input2 + " - " + inp2);
    System.out.println(operatore);
    System.out.printf(Etichette.display + "%5.2f", risultato);
}

}
```

./Prj_11_Calcolatrice/src/it/bogliaccino/calcolatrice/Etichette.java

```
package it.bogliaccino.calcolatrice;

public class Etichette {
    public static final String OPE1 = "Inserire il primo operando";
    public static final String OPE2 = "Inserire il secondo operando";
    public static final String OPERATORE = "Inserire l'operatore aritmetico";
    public static final String MSG_FAIL = "Operatore non riconosciuto!";
    public static final String MSG_INPUT = "L'input non è corretto";

    public static String display = "Il risultato è: ";
}
```

./Prj_11_Calcolatrice/src/it/bogliaccino/calcolatrice/MathHelper.java

```
package it.bogliaccino.calcolatrice;

import java.util.Scanner;

public class MathHelper {

    private static Scanner sc = new Scanner(System.in);

    public static double calcola(int inp1, int inp2, String operatore) {
        double risultato = 0;
        switch (operatore) {
            case "+":
                risultato = inp1 + inp2;
        }
    }
}
```

```
        break;
    case "-":
        risultato = inp1 - inp2;
        break;
    case "*":
        risultato = inp1 * inp2;
        break;
    case "/":
        risultato = (double) inp1 / inp2;
        break;

    default:
        System.out.println(Etichette.MSG_FAIL + operatore);
        break;
    }
    return risultato;
}

public static int trasformaIntero(String input1) {

    int i = 0;

    try {
        i = Integer.parseInt(input1);
    } catch (NumberFormatException e) {
        System.err.println(Etichette.MSG_INPUT);
    }

    return i;
}

public static String getInput(String messaggio) {

    System.out.println(messaggio);
    String s = sc.nextLine();

    return s;
}

public static void chiudiScanner() {
    sc.close();
}

}
```

./Prj_10_Studenti/src/model/Studente.java

```
package model;

public class Studente {
```

```
//variabile di proprietà della classe: static
private static int contatore = 1;

//stato interno
private String nome;
private String cognome;
private int eta;
private int matricola;

//costruttore dell'oggetto studente

public Studente(String n, String c, int e) {
    nome = n;
    cognome = c;
    eta = 2020 - e;
    this.setMatricola();
}

//getters accessori, setters mutatori

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getCognome() {
    return cognome;
}

public void setCognome(String cognome) {
    this.cognome = cognome;
}

public int getEta() {
    return eta;
}

public void setEta(int eta) {
    this.eta = eta;
}

public int getMatricola() {
    return matricola;
}

public void setMatricola() {
    this.matricola = Studente.contatore++;
}
```

```
@Override
public String toString() {
    return matricola + ") " + nome + ", " + cognome + ", " + eta + ""
        + "";
}

}
```

./Prj_10_Studenti/src/view/GestioneAula.java

```
package view;

import java.util.ArrayList;
import java.util.Scanner;

import controller.Registro;
import model.Studente;

public class GestioneAula {

    private static ArrayList<Studente> presenti = new ArrayList<>();
    private static ArrayList<Studente> assenti = new ArrayList<>();

    public static void main(String[] args) {

        Registro aula1f = new Registro();
        //riempio di studenti il registro di aula1f
        aula1f.inizializza();
        //faiAppello(aula1f);

        ////        aula1f.getStudenti();
        //        for (Studente s : aula1f.getStudenti()) {
        //            System.out.println(s);
        //        }
        //
        faiAppello(aula1f);

        System.out.println("PRESENTI " + presenti.size());
        for (Studente s : presenti) {
            System.out.println(s);
        }

        System.out.println("ASSENTI " + assenti.size());
        for (Studente s : assenti) {
            System.out.println(s);
        }
    }
}
```



```
        //calcolaMedia(aula1f);

    }

    private static void calcolaMedia(Registro aula1f) {
        int somma=0;
    }

    private static void faiAppello(Registro aula1f) {
        Scanner sc = new Scanner(System.in);
        for (Studiante s : aula1f.getStudentiAL()) {
            System.out.println("E' presente lo studente");
            System.out.println(s);
            if (sc.nextLine().equals("s"))
                presenti.add(s);
            else
                assenti.add(s);
        }
    }
}
```

./Prj_10_Studenti/src/database/DB.java

```
package database;

import java.util.ArrayList;

import model.Studiante;

public class DB {

    //persistenza
    //1) scarico i dati da un database, per ogni record creo un oggetto di
    tipo studente
    //2) leggo un file di testo, con valori separati da virgola e per ogni
    linea creo un oggetto di tipo studente

    private Studiante[] studentiAR;
```

```
private ArrayList<Studente> studentiAL;

public DB() {
    studentiAR = new Studente[18];
    studentiAL = new ArrayList<>();
    this.init();
}

private void init(){
    //inizializzo l'array di studenti
    addStudente(new Studente("Amal", "El Arabi", 2000), 0);
    addStudente(new Studente("Ayoub", "Bettaieb", 2001), 1);
    addStudente(new Studente("Dario", "Iapicca", 1995), 2);
    addStudente(new Studente("Davide", "Beltramo", 1997), 3);
    addStudente(new Studente("Diego", "Pellegrino", 1998), 4);
    addStudente(new Studente("Emanuele", "Sandrone", 1996), 5);
    addStudente(new Studente("Federica", "Corbezzolo", 1995), 6);
    addStudente(new Studente("giacomo", "mosso", 1998), 7);
    addStudente(new Studente("Giada", "Squillaci", 1993), 8);
    addStudente(new Studente("Luca", "Basello", 1996), 9);
    addStudente(new Studente("Maryam", "Alimohammadi", 1987), 10);
    addStudente(new Studente("Matteo", "Tosin", 1994), 11);
    addStudente(new Studente("Mattia", "Vazzoler", 1989), 12);
    addStudente(new Studente("Mirko", "Franco", 1990), 13);
    addStudente(new Studente("Nabil", "Bougarech", 1993), 14);
    addStudente(new Studente("Rebecca", "Ghergo", 1996), 15);
    addStudente(new Studente("Salvatore", "Zollino", 1994), 16);
    addStudente(new Studente("Tommaso", "Angeli", 1999), 17);

    //inizializzo i valori dell'AL
    for (Studente studente : studentiAR) {
        this.studentiAL.add(studente);
    }
}

private void addStudente(Studente studente, int i) {
    studentiAR[i] = studente;
}

public Studente[] getStudentiAR() {
    return studentiAR;
}

public ArrayList<Studente> getStudentiAL() {
    return studentiAL;
}

}
```

./Prj_10_Studenti/src/controller/Registro.java

```
package controller;

import java.util.ArrayList;

import database.DB;
import model.Studente;

public class Registro {

    //registro vuoto con 18 posti liberi
    private Studente[] studenti = new Studente[18];
    private ArrayList<Studente> studentiAL = new ArrayList<>();

    public void inizializza() {
        DB db = new DB();
        studenti = db.getStudentiAR();
        studentiAL = db.getStudentiAL();
    }

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

    public ArrayList<Studente> getStudentiAL() {
        return studentiAL;
    }

}
```

./Prj_06_LanciaDadi/src/model/Dado.java

```
package model;

import java.util.Random;

public class Dado {

    byte facce;
    Random random;

    public Dado(byte facce) {
        this.facce = facce;
        this.random = new Random();
    }

}
```

```
    }

    public byte lancia() {
        return (byte) (this.random.nextInt(this.facce) + 1);
    }

}
```

./Prj_06_LanciaDadi/src/view/LanciaDadi.java

```
package view;

import controller.GiocoController;
import model.Dado;

public class LanciaDadi {

    public static void main(String[] args) {

        GiocoController gc = new GiocoController();
        GiocoController gc2 = new GiocoController();

        gc.gioco1();
        gc2.gioco2();

    }

}
```

./Prj_06_LanciaDadi/src/controller/GiocoController.java

```
package controller;

import model.Dado;

public class GiocoController {
    public void gioco1() {
        final int LIMITE = 100000;
        int successi = 0;

        Dado d1 = new Dado((byte) 6);
        Dado d2 = new Dado((byte) 6);
```

```
        long start = System.currentTimeMillis();

        for (int i = 0; i < LIMITE; i++) {

            byte res1 = d1.lancia();
            byte res2 = d2.lancia();

            //      System.out.print("Dado 1: " + res1);
            //      System.out.println(" Dado 2: " + res2);
            if (res1+res2 == 7) {
                //System.out.println("VITTORIA");
                successi++;
            }
        } //fine ciclo
        long stop = System.currentTimeMillis();
        System.out.println("Percentuale successi "+
((double)successi/LIMITE) * 100 + "%");
        System.out.println("Tempo di elaborazione " + (stop-start));
    }
    public void gioco2() {
        final int LIMITE = 100000;
        int successi = 0;

        Dado d1 = new Dado((byte) 6);
        Dado d2 = new Dado((byte) 6);

        long start = System.currentTimeMillis();

        for (int i = 0; i < LIMITE; i++) {

            byte res1 = d1.lancia();
            byte res2 = d2.lancia();

            //      System.out.print("Dado 1: " + res1);
            //      System.out.println(" Dado 2: " + res2);
            if (res1 == res2) {
                //System.out.println("VITTORIA");
                successi++;
            }
        } //fine ciclo
        long stop = System.currentTimeMillis();
        System.out.println("Percentuale successi "+
((double)successi/LIMITE) * 100 + "%");
        System.out.println("Tempo di elaborazione " + (stop-start));
    }
}
```

./Laboratorio/src/model/Voto.java

```
package model;

public class Voto {

    private String materia;
    private int valutazione;

    public Voto(String materia, int valutazione) {
        setMateria(materia);
        setValutazione(valutazione);
    }

    public Voto(int valutazione) {
        setMateria("Test laboratorio");
        setValutazione(valutazione);
    }

    public int getValutazione() {
        return valutazione;
    }
    private void setValutazione(int valutazione) {

        if (valutazione >= 0 && valutazione <= 30)
            this.valutazione = valutazione;
    }
    public String getMateria() {
        return materia;
    }
    private void setMateria(String materia) {
        this.materia = materia;
    }

    @Override
    public String toString() {
        return "Voto [materia=" + materia + ", valutazione=" + valutazione
+ " ]";
    }

}
```

./Laboratorio/src/scarabocchi/OrdinalInteri.java

```
package scarabocchi;

import java.util.ArrayList;
import java.util.Arrays;
```

```
import java.util.Collections;

import model.Voto;

public class OrdinaInteri {

    public static void main(String[] args) {

        int[] voti = {30,26,28,24,22};

        Arrays.sort(voti);

        for (int i : voti) {
            System.out.println(i);
        }

        ArrayList<Integer> voti2 = new ArrayList<>();
        voti2.add(30);
        voti2.add(26);
        voti2.add(28);
        voti2.add(24);
        voti2.add(22);

        Collections.sort(voti2);

        for (Integer integer : voti2) {
            System.out.println(integer);
        }

        ArrayList<Voto> voti3 = new ArrayList<Voto>();
        voti3.add(new Voto(30));
        voti3.add(new Voto(26));
        voti3.add(new Voto(28));
        voti3.add(new Voto(24));
        voti3.add(new Voto(22));

        //Collections.sort(voti3);

        for (Voto voto : voti3) {
            System.out.println(voto);
        }

    }

}
```

./Laboratorio/src/scarabocchi/Eccezioni01.java

```
package scarabocchi;
```

```
public class Eccezioni01 {

    public static void main(String[] args) {

        int[] voti = {5,4,6,7,8,10};

        //      for (int i = 0; i < 7; i++) {
        //          System.out.println(voti[i]);
        //      }
        //

        //voti[-1] = 12;

        int a = 7;
        int b = 0;

        System.out.println(a/b);

        System.out.println("la cosa che volevo dirti davvero è...");

    }

}
```

./Laboratorio/src/view/GestioneVoti.java

```
package view;

import java.util.Scanner;

import controller.VotiController;
import model.Voto;

public class GestioneVoti {
    public static void main(String[] args) {
        VotiController vc = new VotiController();
        String richiesta = null;
        Scanner sc = new Scanner(System.in);
        boolean continua = true;

        while(continua) {

            menu();

            richiesta = sc.nextLine();

            if (richiesta.equals("a")) {
```



```
        assegnaVoto(vc);

    }

    if (richiesta.equals("b")) {
        //stampo il report
        report(vc);

    }

    if (richiesta.equals("q")) {
        continua = false;
        System.out.println("arrivederci!");
    }
}

}

private static void assegnaVoto(VotiController vc) {
    Scanner sc = new Scanner(System.in);

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

    System.out.println("Inserisci voto");
    int voto = sc.nextInt();

    if (materia != null) {
        vc.addVoto(materia, voto);
    } else {
        vc.addVoto(voto);
    }
}

private static void menu() {
    System.out.println();
    System.out.println("-----");
    System.out.println("Benvenuto in Gestione Voti");
    System.out.println("a) per inserire voti");
    System.out.println("b) per mostrare elenco e media");
    System.out.println("q) per uscire");
    System.out.println("-----");
    System.out.println();
}

private static void report(VotiController vc) {
    for (Voto v : vc.getVoti()) {
        System.out.println(v);
    }

    System.out.println("la media è " + vc.calcolaMedia());
}
```

```
        System.out.println("Il voto di laurea è: " +  
vc.calcolaVotoLaurea());  
    }  
  
}
```

./Laboratorio/src/controller/VotiController.java

```
package controller;  
  
import java.util.ArrayList;  
  
import model.Voto;  
  
public class VotiController {  
  
    private ArrayList<Voto> voti = new ArrayList<>();  
  
    public ArrayList<Voto> getVoti() {  
        return voti;  
    }  
  
    public void addVoto(int v) {  
        //conversione automatica -> boxing  
        voti.add(new Voto(v));  
    }  
  
    public void addVoto(String materia, int voto) {  
        voti.add(new Voto(materia, voto));  
    }  
  
    public double calcolaMedia() {  
        double somma=0;  
        //somma dei voti  
        for (Voto v : voti) {  
            somma += v.getValutazione();  
        }  
  
        //numero di voti  
  
        //somma/numVoti  
  
        return somma / voti.size();  
    }  
  
    public double calcolaVotoLaurea() {  
        double media = this.calcolaMedia();  
    }  
}
```

```
        return media / 3 * 11;
    }

}
```

./Prj_02_Biblioteca/src/model/Libro.java

```
package model;

public class Libro {

    public String titolo;
    public String autori;
    public int pagine;

    public Libro(String titolo, String autori, int pagine) {
        this.titolo = titolo;
        this.autori = autori;
        this.pagine = pagine;
    }

    public String mostra() {
        return titolo + ", " + autori + ", " + pagine;
    }

}
```

./Prj_02_Biblioteca/src/view/Gestione.java

```
package view;

import controller.Schedario;
import model.Libro;

public class Gestione {

    public static void main(String[] args) {

        Schedario narrativaStr = new Schedario(4);

        Libro l1 = new Libro("Harry Potter", "Rowling", 100);
```

```
        Libro l2 = new Libro("Il signore degli anelli", "Tolkien", 1000);
        Libro l3 = new Libro("Io, robot", "Asimov", 500);
        Libro l4 = new Libro("Io, Diego", "D.A. Maradona", 1500);

        narrativaStr.aggiungiLibro(l1, 0);
        narrativaStr.aggiungiLibro(l2, 1);
        narrativaStr.aggiungiLibro(l3, 2);
        narrativaStr.aggiungiLibro(l4, 3);

        narrativaStr.elenco();

    }

}
```

./Prj_02_Biblioteca/src/controller/Schedario.java

```
package controller;

import model.Libro;

public class Schedario {

    public Libro[] schedario;

    public Schedario(int i) {
        this.schedario = new Libro[i];
    }

    public void aggiungiLibro(Libro l, int i) {
        this.schedario[i] = l;
    }

    public void elenco() {
        for (int i = 0; i < schedario.length; i++) {
            System.out.println( "Libro " + i + ": " +
            schedario[i].mostra());
        }
    }

}
```

./Prj_12_Geometria/src/model/Punto.java

```
package model;

public class Punto {

    private int x, y;

    //overload
    public Punto() {
        this.x = 0;
        this.y = 0;
    }

    //overload
    public Punto(int x) {
        this.x = x;
        this.y = 0;
    }

    //overload
    public Punto(int y, boolean t) {
        if (t == true)
            this.x = 1;
        else
            this.x = 0;
        this.y = y;
    }

    //overload
    public Punto(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public int getX() {
        return x;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getY() {
        return y;
    }

    public void setY(int y) {
        this.y = y;
    }

    @Override
    public String toString() {
        return "(" + x + ", " + y + ")";
    }
}
```

```
}
```

./Prj_12_Geometria/src/model/Rettangolo.java

```
package model;

public class Rettangolo {

    private Segmento base, altezza;

    public Rettangolo(Segmento b, Segmento h) {
        this.base = b;
        this.altezza = h;
    }

    public double perimetro() {
        System.out.println("Uso la formula del rettangolo");
        return 2 * (this.base.lunghezza() + this.altezza.lunghezza());
    }

    public double superficie() {
        return this.base.lunghezza() * this.altezza.lunghezza();
    }

}
```

./Prj_12_Geometria/src/model/Quadrato.java

```
package model;

public class Quadrato extends Rettangolo {

    private Segmento lato;

    public Quadrato(Segmento l) {
        super(l, l);
        this.lato=l;
    }

    @Override
    public double perimetro() {
        System.out.println("Uso la formula del quadrato");
        return this.lato.lunghezza()*4;
    }

    @Override
```

```
    public double superficie() {  
        // TODO Auto-generated method stub  
        return this.lato.lunghezza() * this.lato.lunghezza();  
    }  
  
}
```

./Prj_12_Geometria/src/model/Segmento.java

```
package model;  
  
public class Segmento {  
  
    private Punto a, b;  
  
    public Segmento(Punto a, Punto b) {  
        this.a = a;  
        this.b = b;  
    }  
  
    public double lunghezza() {  
        double d = 0;  
  
        double q1 = Math.pow((a.getX() - b.getX()), 2);  
        double q2 = Math.pow((a.getY() - b.getY()), 2);  
  
        d = Math.sqrt( q1 + q2 );  
  
        return d;  
    }  
  
    @Override  
    public String toString() {  
        return a + ", " + b;  
    }  
  
}
```

./Prj_12_Geometria/src/model/Triangolo.java

```
package model;  
  
public class Triangolo {  
  
    private Segmento ab, ac, cb;  
    private Punto a, b, c;  

```

```
public Triangolo(Punto a, Punto b, Punto c) {
    this.a = a;
    this.b = b;
    this.c = c;

    this.ab = new Segmento(a, b);
    this.ac = new Segmento(a, c);
    this.cb = new Segmento(c, b);
}

public double perimetro() {
    return this.ab.lunghezza() +
this.ac.lunghezza()+this.cb.lunghezza();
}

public double superficieErone() {
    double sp = this.perimetro()/2;
    double s = 0;

    s = Math.sqrt(sp * ( sp - this.ab.lunghezza() ) * ( sp -
this.ac.lunghezza() )
        * ( sp - this.cb.lunghezza() ) );
    return s;
}
}
```

./Prj_12_Geometria/src/view/ProvaGeometrica.java

```
package view;

import model.Punto;
import model.Quadrato;
import model.Rettangolo;
import model.Segmento;
import model.Triangolo;

public class ProvaGeometrica {

    public static void main(String[] args) {

        Punto p1 = new Punto(1,1);
        Punto p2 = new Punto(5,1);
        Punto p3 = new Punto(1,4);

        Segmento ab = new Segmento(p1, p2);
        Segmento ac = new Segmento(p1, p3);
    }
}
```



```
        Segmento cb = new Segmento(p3, p2);

//        System.out.println("Segmento ab: " + ab);
//        System.out.println("Lunghezza ab: " + ab.lunghezza());
//
//        System.out.println("Segmento ac: " + ac);
//        System.out.println("Lunghezza ac: " + ac.lunghezza());
//
//        System.out.println("Segmento cb: " + cb);
//        System.out.println("Lunghezza cb: " + cb.lunghezza());

        Triangolo t = new Triangolo(p1, p2, p3);
//        System.out.println(t.perimetro());
//        System.out.println(t.superficieErone());

        Rettangolo r = new Rettangolo(ab, ac);
        System.out.println(r.perimetro());
        System.out.println(r.superficie());

        Rettangolo q = new Quadrato(ab);
        System.out.println(q.perimetro());
        System.out.println(q.superficie());

        Rettangolo[] quadrilateri = {r, q};

        for (Rettangolo r1 : quadrilateri) {
            if (r1 instanceof Quadrato)
                System.out.println("E' un QUADRATO!!!!");
            System.out.println(r1.perimetro());
        }

    }

}
```

./Prj_01_Studenti/src/model/Studente.java

```
package model;

public class Studente {

    //stato interno
    public String nome;
    public String cognome;
    public int eta;
    public int matricola;

    //costruttore dell'oggetto studente
```

```
public Studente(String n, String c, int e) {
    nome = n;
    cognome = c;
    eta = 2020 - e;
}

//metodi dell'oggetto studente

public String scheda() {
    return nome + " " + cognome + " " + eta;
}

}
```

./Prj_01_Studenti/src/view/GestioneAula.java

```
package view;

import java.util.Scanner;

import controller.Registro;
import model.Studente;

public class GestioneAula {

    public static void main(String[] args) {

        Registro aula1f = new Registro();
        Registro aula1d = new Registro();
        //riempio di studenti il registro di aula1f
        aula1f.inizializza();
        faiAppello(aula1f);
        //calcolaMedia(aula1f);
        System.out.println("finito...");

    }

    private static void calcolaMedia(Registro aula1f) {
        int somma=0;
        for (int i = 0; i < aula1f.studenti.length; i++) {
            System.out.println(aula1f.studenti[i].eta);
            somma += aula1f.studenti[i].eta;
        }
        System.out.println("L'età media dell'aula è: " +
(somma/aula1f.studenti.length));
    }
}
```

```
private static void faiAppello(Registro aula1f) {
    int presenti = 0;

    Scanner sc = new Scanner(System.in);

    //scorri il registro (array di studente)
    for (int i = 0; i < aula1f.studenti.length; i++) {

        System.out.println( "E' presente lo studente " +
        aula1f.studenti[i].scheda() + "?" );
        String s = sc.nextLine();
        if (s.startsWith("s")) {
            presenti++;
        }
    }

    System.out.println("sono presenti " + presenti + " studenti.");
}

}
```

./Prj_01_Studenti/src/controller/Registro.java

```
package controller;

import model.Studente;

public class Registro {

    //registro vuoto con 18 posti liberi
    public Studente[] studenti = new Studente[18];

    public Registro() {
        System.out.println("Ho costruito un registro");
    }

    public void inizializza() {
        System.out.println("inizializzo il registro");
        addStudente(new Studente("Amal", "El Arabi", 2000), 0);
        addStudente(new Studente("Ayoub", "Bettaieb", 2001), 1);
        addStudente(new Studente("Dario", "Iapicca", 1995), 2);
    }
}
```

```

        addStudente(new Studente("Davide", "Beltramo", 1997), 3);
        addStudente(new Studente("Diego", "Pellegrino", 1998), 4);
        addStudente(new Studente("Emanuele", "Sandrone", 1996), 5);
        addStudente(new Studente("Federica", "Corbezzolo", 1995), 6);
        addStudente(new Studente("giacomo", "mosso", 1998), 7);
        addStudente(new Studente("Giada", "Squillaci", 1993), 8);
        addStudente(new Studente("Luca", "Basello", 1996), 9);
        addStudente(new Studente("Maryam", "Alimohammadi", 1987), 10);
        addStudente(new Studente("Matteo", "Tosin", 1994), 11);
        addStudente(new Studente("Mattia", "Vazzoler", 1989), 12);
        addStudente(new Studente("Mirko", "Franco", 1990), 13);
        addStudente(new Studente("Nabil", "Bougarech", 1993), 14);
        addStudente(new Studente("Rebecca", "Ghergo", 1996), 15);
        addStudente(new Studente("Salvatore", "Zollino", 1994), 16);
        addStudente(new Studente("Tommaso", "Angeli", 1999), 17);
        System.out.println("registro inizializzato");

    }

    //ogni volta che aggiungo uno studente occupo una delle 18 posizioni
    public void addStudente(Studente s, int i) {
        studenti[i] = s;
    }

}

```

./Prj_08_TCommerce/src/model/Prodotto.java

```

package model;
/**
 * Classe che rappresenta un geneico prodotto ...
 * @author mauro
 *
 */
public class Prodotto {

    private int codProdotto;
    private String descrizione;
    private double prezzo;

    /**
     * Costruttore di un generico Prodotto
     * @param codProdotto inserire un codice prodotto di tipo intero
     * @param descrizione inserire una stringa descrittiva del prodotto
     * @param prezzo inserire il prezzo del prodotto
     */
    public Prodotto(int codProdotto, String descrizione, double prezzo) {
        this.codProdotto = codProdotto;
    }
}

```

```

        this.descrizione = descrizione;
        this.prezzo = prezzo;
    }

    //getters (accessori) and setters(mutatori)
    public int getCodProdotto() {
        return codProdotto;
    }
    public void setCodProdotto(int codProdotto) {
        this.codProdotto = codProdotto;
    }
    public String getDescrizione() {
        return descrizione.toUpperCase();
    }
    public void setDescrizione(String descrizione) {
        this.descrizione = descrizione;
    }
    public double getPrezzo() {
        return prezzo;
    }
    public void setPrezzo(double prezzo) {
        this.prezzo = prezzo;
    }
}

@Override
public String toString() {
    return "\"" + this.codProdotto + ", " + this.getDescrizione() + ", "
+ this.prezzo + "\"";
}

}

```

./Prj_08_TCommerce/src/view/Negozio.java

```

package view;

import controller.Carrello;
import controller.Prodotti;
import model.Prodotto;

public class Negozio {

    public static void main(String[] args) {

        Prodotti negozio1 = new Prodotti();

        //stampo elenco prodotti
        for (Prodotto p : negozio1.getMagazzino()) {

```

```

        System.out.println(p);
    }
    //chiedo all'utente di scegliere dei prodotti

    Carrello utente1 = new Carrello();
    Prodotto p1 = negozio1.getMagazzino()[0];
    Prodotto p2 = negozio1.getMagazzino()[1];
    Prodotto p3 = negozio1.getMagazzino()[2];

    //aggiungo i prodotti al carrello
    utente1.aggiungiAlCarrello(p1);
    utente1.aggiungiAlCarrello(p1);
    utente1.aggiungiAlCarrello(p1);
    utente1.aggiungiAlCarrello(p2);
    utente1.aggiungiAlCarrello(p1);
    utente1.aggiungiAlCarrello(p3);
    utente1.aggiungiAlCarrello(p1);

    //stampo elenco carrello
    System.out.println("Il carrello contiene");
    for (Prodotto p : utente1.getContenitore()) {
        System.out.println(p);
    }
    System.out.println("Prezzo totale: " + utente1.calcolaPrezzo());
}
}

```

./Prj_08_TCommerce/src/controller/Magazzino.java

```

package controller;

import model.Prodotto;

public class Magazzino {

    public static Prodotto[] getAll() {

        Prodotto[] array = {
            new Prodotto(1, "Pasta", 0.85),
            new Prodotto(2, "Riso", 1.85),
            new Prodotto(3, "Biscotti", 1.15),
            new Prodotto(4, "Latte", 1.55),
            new Prodotto(5, "Pane", 1.99),
            new Prodotto(6, "Vino", 2.85),
            new Prodotto(7, "Sale", 0.85),
            new Prodotto(8, "Pepe", 0.95),
            new Prodotto(9, "Grissini", 1.35),
            new Prodotto(10, "Mele", 1.05),
        };
    }
}

```

```
        };

        return array;
    }

}
```

./Prj_08_TCommerce/src/controller/Prodotti.java

```
package controller;

import model.Prodotto;

public class Prodotti {

    private Prodotto[] magazzino;

    public Prodotti() {
        this.magazzino = Magazzino.getAll();
    }

    public Prodotto[] getMagazzino() {
        return magazzino;
    }

}
```

./Prj_08_TCommerce/src/controller/Carrello.java

```
package controller;

import java.util.ArrayList;
import model.Prodotto;

public class Carrello {

    private ArrayList<Prodotto> contenitore;

    public Carrello() {
        this.contenitore = new ArrayList<>();
    }

    public void aggiungiAlCarrello(Prodotto p) {
        this.contenitore.add(p);
    }

}
```

```
    public ArrayList<Prodotto> getContenitore() {
        return contenitore;
    }

    public double calcolaPrezzo() {
        double prezzo = 0;

        for (Prodotto p : contenitore) {
            prezzo += p.getPrezzo();
        }

        return prezzo;
    }
}
```

./Prj_04_Banca/src/model/Conto.java

```
package model;

public class Conto {

    public int numeroConto;
    private double saldo;
    private static int contatore = 0;
    public Cliente[] titolare;

    public Conto(Cliente[] titolare) {

        this.numeroConto = ++contatore;
        this.titolare = titolare;
        this.saldo = 0.0;
        System.out.println("Nuovo conto numero "+this.numeroConto+" creato  
con successo!");
    }

    public void versamento(double ammontare) {
        this.saldo += ammontare;
    }

    public void prelievo(double ammontare) {
        if (this.saldo >= ammontare)
            this.saldo -= ammontare;
        else
            System.out.println("Non puoi prelevare " + ammontare + "  
disponibilità: " +this.saldo);
    }
}
```



```
@Override
public String toString() {

    String titolari = "";
    for (int i = 0; i < titolare.length; i++) {
        titolari += titolare[i];
        titolari += ", ";
    }

    return this.numeroConto+") "+ titolari + " saldo: "+this.saldo ;
}

}
```

./Prj_04_Banca/src/model/Cliente.java

```
package model;

public class Cliente {

    private String nome, cognome;

    public Cliente(String nome, String cognome) {
        this.nome = nome;
        this.cognome = cognome;
    }

    @Override
    public String toString() {
        return "" + nome + ", " + cognome + " - ";
    }

}
```

./Prj_04_Banca/src/view/GestioneBanca.java

```
package view;

import model.Cliente;
import model.Conto;

public class GestioneBanca {

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

```
        Cliente c1 = new Cliente("Pippo", "De pippis");
        Cliente c2 = new Cliente("Tex", "Willer");

        Cliente[] titolari = {c1, c2};
        Cliente[] titolari2 = {c2};

        Conto conto1 = new Conto(titolari);
        Conto conto2 = new Conto(titolari2);

        System.out.println(conto1);
        System.out.println(conto2);

        conto1.versamento(100);
        System.out.println(conto1);
        conto1.prelievo(200);
        System.out.println(conto1);

    }

}
```

./Prj_04_Banca/src/controller/Filiale.java

```
package controller;

public class Filiale {

}
```