Greta Sorritelli

EMPORIO

ARTICOLO

**import** java.util.ArrayList;

**public** **abstract** **class** Articolo {

**private** Lettera reparto;

**private** Misura unita;

**protected** Number quantita;

**private** **double** prezzoEuro;

**private** String codice;

**private** ArrayList<String> descrizione;

**private** **double** consumoWatt;

**public** Articolo(Lettera reparto, Misura unita, **double** quantita, **double** prezzo, String codice, ArrayList<String> descrizione) {

**this**.reparto = reparto;

**this**.unita = unita;

**this**.quantita = quantita;

**this**.prezzoEuro = prezzo;

**this**.codice = codice;

**this**.descrizione = descrizione;

}

**public** Articolo(Lettera reparto, Misura unita, **int** quantita, **double** prezzo, String codice, **double** consumoWatt, ArrayList<String> descrizione) {

**this**.reparto = reparto;

**this**.unita = unita;

**this**.quantita = quantita;

**this**.prezzoEuro = prezzo;

**this**.codice = codice;

**this**.descrizione = descrizione;

**this**.consumoWatt = consumoWatt;

}

**public** String toString() {

**return** reparto + " " + unita + " " + quantita + " " + prezzoEuro + " " + codice + " " +consumoWatt + " " + descrizione;

}

**public** Lettera getReparto() {

**return** reparto;

}

**public** Misura getUnita() {

**return** unita;

}

**public** Number getQuantita() {

**return** quantita;

}

**public** **double** getPrezzoEuro() {

**return** prezzoEuro;

}

**public** String getCodice() {

**return** codice;

}

**public** ArrayList<String> getDescrizione() {

**return** descrizione;

}

**public** **double** getConsumoWatt() {

**return** consumoWatt;

}

}

ARTICOLO NON NUMERABILE

**import** java.util.ArrayList;

**public** **class** ArticoloNonNumerabile **extends** Articolo {

**public** ArticoloNonNumerabile(Lettera reparto, Misura unita, **double** quantita, **double** prezzoEuro, String codice, ArrayList<String> descrizione) {

**super**(reparto, unita, quantita, prezzoEuro, codice, descrizione);

}

**public** **boolean** Vendita(String codice,**double** quantita) {

**if**(quantita < **super**.quantita.doubleValue()) {

**super**.quantita= **super**.quantita.doubleValue() - quantita ;

**return** **true**;

}

**else** **if**(quantita > **super**.quantita.doubleValue() ) {

**return** **false**;

}

**else** {

Utilita.*listaArticoli*.remove(codice);

**return** **true**;

}

}

**public** String toString() {

**return** **super**.toString();

}

}

ARTICOLO NUMERABILE

**import** java.util.ArrayList;

**public** **class** ArticoloNumerabile **extends** Articolo {

**public** ArticoloNumerabile(Lettera reparto, Misura unita, **int** quantita, **double** prezzoEuro, String codice, **double** consumoWatt, ArrayList<String> descrizione) {

**super**(reparto, unita, quantita, prezzoEuro, codice, consumoWatt,descrizione );

}

**public** **boolean** Vendita(String codice,**int** quantita) {

**if**(quantita < **super**.quantita.intValue()) {

**super**.quantita= **super**.quantita.intValue() - quantita ;

**return** **true**;

}

**else** **if**(quantita > **super**.quantita.intValue() ) {

**return** **false**;

}

**else** {

Utilita.*listaArticoli*.remove(codice);

**return** **true**;

}

}

**public** String toString() {

**return** **super**.toString();

}

}

LETTERA

**public** **enum** Lettera {

***E***('E'), //Elettricità

***F***('F'), //Ferramenta

***L***('L'), //Legname

***C***('C'), //Casa

***S***('S'); //Sport e tempo libero

**private** **char** x;

Lettera(**char** x) {

**this**.x = x ;

}

**public** **char** getLet() {

**return** x ;

}

}

MISURA

**public** **enum** Misura {

***U***('U'), //unità

***K***('K'), //kilo

***L***('L'), //litro

***M***('M'); //metro

**private** **char** y;

Misura(**char** y) {

**this**.y = y;

}

**public** **char** getMis() {

**return** y;

}

}

ECCEZIONE CODICE

**public** **class** EccezioneCodice **extends** Exception {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** EccezioneCodice(String codice) {

**super**("Codice errato");

}

}

ECCEZIONE CODICE ESISTENTE

**public** **class** EccezioneCodiceEsistente **extends** Exception {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** EccezioneCodiceEsistente() {

System.***out***.println("Nessun codice inserito.");

}

**public** EccezioneCodiceEsistente (String codice) {

**super**("Codice Esistente.");

}

**public** EccezioneCodiceEsistente(String codice, Throwable error) {

**super**("Errore Codice Esistente.");

}

}

UTILITA

**import** java.util.HashMap;

**import** java.util.Map;

**public** **class** Utilita {

**public** **static** **int** *elementiNonRiconosciuti* = 0;

**public** **static** Map<String, Articolo> *listaArticoli* = **new** HashMap<>();

**private** **static** **boolean** controlloLunghezza(String codice) {

**if** (codice.length() == 8) {

**return** **true**;

} **else** {

**return** **false**;

}

}

**private** **static** **boolean** isNumeric(String codice) { // controllo numeri da 1 a 9

**return** codice != **null** && codice.matches("\\d+");

}

**public** **static** **boolean** isNumber(String s) { // controllo numeri negativi

**return** s != **null** && s.matches("\\d.\*");

}

**public** **static** **boolean** controlloCodiceEsistente(String codice, Map<String, Articolo> map)

**throws** EccezioneCodiceEsistente {

**if** (map.containsKey(codice)) {

**return** **false**;

} **else** {

**return** **true**;

}

}

**public** **static** **boolean** controlloCodice(String codice) **throws** EccezioneCodice {

**if** (*controlloLunghezza*(codice) && *isNumeric*(codice)) {

**return** **true**;

} **else** {

**return** **false**;

}

}

**public** **static** **boolean** controlloCodice(String codice, Map<String, Articolo> map)

**throws** EccezioneCodice, EccezioneCodiceEsistente {

**if** (*controlloLunghezza*(codice) && *isNumeric*(codice) && *controlloCodiceEsistente*(codice, map)) {

**return** **true**;

} **else** {

**return** **false**;

}

}

}

OPERAZIONI MAGAZZINO

**import** java.io.BufferedReader;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.Map;

**import** java.util.NoSuchElementException;

**import** java.io.InputStreamReader;

**public** **class** OperazioniMagazzino {

**public** **void** CalcoloValoreMagazzino() {

**double** valoreTotale = 0;

**for** (Map.Entry<String, Articolo> entry : Utilita.*listaArticoli*.entrySet()) {

Articolo value = entry.getValue();

Number quantita = value.getQuantita();

**double** prezzoEuro = value.getPrezzoEuro();

**double** prodotto = quantita.doubleValue() \* prezzoEuro;

valoreTotale += prodotto;

}

System.***out***.print("Il valore totale del magazzino è: " + valoreTotale + "\n");

}

**public** **void** VenditaArticolo() **throws** IOException, EccezioneCodice {

BufferedReader brInput = **new** BufferedReader(**new** InputStreamReader(System.***in***)); // scrittura da console

System.***out***.print("Inserisci il codice: ");

**try** {

String codice = brInput.readLine();

**if** (Utilita.*controlloCodice*(codice) && Utilita.*listaArticoli*.containsKey(codice)) {

System.***out***.println("Inserisci la quantita da vendere: ");

String quantita = brInput.readLine();

**if** (Utilita.*listaArticoli*.get(codice).getUnita() == Misura.***U***) {

ArticoloNumerabile articoloNumerabile = (ArticoloNumerabile) Utilita.*listaArticoli*.get(codice);

**try** {

**int** quantitaInt = Integer.*parseInt*(quantita);

**if** (articoloNumerabile.Vendita(codice, quantitaInt)) {

System.***out***.println("Articolo venduto correttamente.");

} **else** {

System.***out***.println(

"Non è possibile vendere il prodotto, perchè la quantità in magazzino non è sufficiente.");

}

} **catch** (NumberFormatException e) {

System.***out***.println("Quantità non riconosciuta.");

}

} **else** {

ArticoloNonNumerabile articoloNonNumerabile = (ArticoloNonNumerabile) Utilita.*listaArticoli*

.get(codice);

**try** {

**double** quantitaDouble = Double.*parseDouble*(quantita);

**if** (articoloNonNumerabile.Vendita(codice, quantitaDouble)) {

System.***out***.println("Articolo venduto correttamente.");

} **else** {

System.***out***.println(

"Non è possibile vendere il prodotto, perchè la quantita in magazzino non è sufficiente.");

}

} **catch** (NumberFormatException e) {

System.***out***.println("Quantità non riconosciuta.");

}

}

} **else** {

System.***out***.println("Codice prodotto non riconosciuto.");

}

} **finally** {

}

}

**public** **void** InserimentoArticolo() **throws** IOException, NoSuchElementException, IllegalStateException,

EccezioneCodice, EccezioneCodiceEsistente {

System.***out***.println(

"Inserisci i parametri dell'articolo (Reparto;Unità;Quantità;Prezzo in euro;Codice univoco;Consumo in watt;Descrizione;)");

BufferedReader nInput = **new** BufferedReader(**new** InputStreamReader(System.***in***)); // scrittura da console

String nuovo = **null**;

**if** ((nuovo = nInput.readLine()) != **null**) {

List<String> campi = **new** ArrayList<String>();

campi = Arrays.*asList*(nuovo.split(";"));

**boolean** primo;

**boolean** secondo;

**switch** (campi.get(0)) {

**case** "E":

**case** "F":

**case** "L":

**case** "C":

**case** "S":

primo = **true**;

**break**;

**default**:

primo = **false**;

**break**;

}

**switch** (campi.get(1)) {

**case** "U":

**case** "K":

**case** "L":

**case** "M":

secondo = **true**;

**break**;

**default**:

secondo = **false**;

**break**;

}

**if** (primo && secondo) {

**if** (Utilita.*controlloCodice*(campi.get(4), Utilita.*listaArticoli*)) {

**if** (Utilita.*isNumber*(campi.get(2)) && Utilita.*isNumber*(campi.get(3))) {

**if** (Utilita.*isNumber*(campi.get(5))) {

Lettera reparto = **null**;

Misura unita = **null**;

**int** quantitaInt = 0;

**double** quantitaDouble = 0;

**double** prezzoEuro = 0;

String rep = (String) campi.get(0);

String uni = (String) campi.get(1);

String qDouble = (String) campi.get(2);

String pre = (String) campi.get(3);

String cod = (String) campi.get(4);

String cons = (String) campi.get(5);

**double** cons1 = Double.*parseDouble*(cons);

**if** (rep.equals("E")) {

reparto = Lettera.***E***;

} **else** **if** (rep.equals("F")) {

reparto = Lettera.***F***;

} **else** **if** (rep.equals("L")) {

reparto = Lettera.***L***;

} **else** **if** (rep.equals("C")) {

reparto = Lettera.***C***;

} **else** **if** (rep.equals("S")) {

reparto = Lettera.***S***;

}

**if** (uni.equals("U")) {

unita = Misura.***U***;

**int** q1 = Integer.*parseInt*(qDouble);

quantitaInt = q1;

} **else** **if** (uni.equals("K")) {

unita = Misura.***K***;

**double** q2 = Double.*parseDouble*(qDouble);

quantitaDouble = q2;

} **else** **if** (uni.equals("M")) {

unita = Misura.***M***;

**double** q2 = Double.*parseDouble*(qDouble);

quantitaDouble = q2;

} **else** **if** (uni.equals("L")) {

unita = Misura.***L***;

**double** q2 = Double.*parseDouble*(qDouble);

quantitaDouble = q2;

}

**else** {

Utilita.*elementiNonRiconosciuti* += 1;

}

**double** p = Double.*parseDouble*(pre);

prezzoEuro = p;

**if** (Utilita.*controlloCodice*(cod, Utilita.*listaArticoli*) == **false**) {

Utilita.*elementiNonRiconosciuti* += 1;

}

ArrayList<String> descrizione = **new** ArrayList<String>();

**for** (**int** i = 6; i < campi.size(); i++) {

descrizione.add(campi.get(i));

}

**if** (unita == Misura.***U***) {

ArticoloNumerabile articolo = **new** ArticoloNumerabile(reparto, unita, quantitaInt,

prezzoEuro, cod, cons1, descrizione);

Utilita.*listaArticoli*.put(cod, articolo);

} **else** **if** (unita == Misura.***K*** || unita == Misura.***M*** || unita == Misura.***L***) {

ArticoloNonNumerabile articoloN = **new** ArticoloNonNumerabile(reparto, unita,

quantitaDouble, prezzoEuro, cod, descrizione);

Utilita.*listaArticoli*.put(cod, articoloN);

}

}

} **else** {

System.***out***.println("Valori numerici non validi.");

}

} **else** {

System.***out***.println("Codice errato,articolo presente.");

}

} **else** {

System.***out***.println("Reparto o unità di misura non validi.");

}

}

}

**public** **void** RicercaArticolo() **throws** IOException, EccezioneCodice {

System.***out***.println("Inserisci il codice dell'articolo da cercare:");

BufferedReader rInput = **new** BufferedReader(**new** InputStreamReader(System.***in***)); // scrittura da console

String s = rInput.readLine();

**if** (Utilita.*controlloCodice*(s)) {

**try** {

String valore = Utilita.*listaArticoli*.get(s).toString();

System.***out***.println("L'articolo corrispondente è: " + valore);

} **catch** (Exception e) {

System.***out***.println("A nessun articolo è associato questo codice.");

}

} **else** {

System.***out***.println("Errore nella ricerca dell'articolo.");

}

}

}

LETTURA FILE

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** LetturaFile {

**public** **void** LetturaTXT() **throws** EccezioneCodice, EccezioneCodiceEsistente {

**try** {

File txt = **new** File("EMPORIO.TXT");

BufferedReader br = **new** BufferedReader(**new** FileReader(txt));

**while** (**true**) {

String line = **null**;

line = br.readLine();

**if** (line == **null**)

**break**;

List<String> parametri = **new** ArrayList<String>();

parametri = Arrays.*asList*(line.split(";"));

Lettera reparto = **null**;

Misura unita = **null**;

**int** quantitaInt = 0;

**double** quantitaDouble = 0;

**double** prezzoEuro = 0;

String codice = **null**;

**double** consumoWatt = 0;

String rep = (String) parametri.get(0);

String uni = (String) parametri.get(1);

String qDouble = (String) parametri.get(2);

String pre = (String) parametri.get(3);

String cod = (String) parametri.get(4);

String cons = (String) parametri.get(5);

consumoWatt = Double.*parseDouble*(cons);

codice = cod;

**if** (rep.length() == 1) {

**if** (rep.equals("E")) {

reparto = Lettera.***E***;

} **else** **if** (rep.equals("F")) {

reparto = Lettera.***F***;

} **else** **if** (rep.equals("L")) {

reparto = Lettera.***L***;

} **else** **if** (rep.equals("C")) {

reparto = Lettera.***C***;

} **else** **if** (rep.equals("S")) {

reparto = Lettera.***S***;

}

}

**if** (uni.length() > 0) {

**if** (uni.equals("U")) {

unita = Misura.***U***;

**int** q1 = Integer.*parseInt*(qDouble);

quantitaInt = q1;

} **else** **if** (uni.equals("K")) {

unita = Misura.***K***;

**double** q2 = Double.*parseDouble*(qDouble);

quantitaDouble = q2;

} **else** **if** (uni.equals("M")) {

unita = Misura.***M***;

**double** q2 = Double.*parseDouble*(qDouble);

quantitaDouble = q2;

} **else** **if** (uni.equals("L")) {

unita = Misura.***L***;

**double** q2 = Double.*parseDouble*(qDouble);

quantitaDouble = q2;

}

**else** {

Utilita.*elementiNonRiconosciuti* += 1;

}

} **else** {

Utilita.*elementiNonRiconosciuti* += 1;

}

**double** p = Double.*parseDouble*(pre);

prezzoEuro = p;

**if** (Utilita.*controlloCodice*(codice) == **false**) {

Utilita.*elementiNonRiconosciuti* += 1;

}

ArrayList<String> descrizione = **new** ArrayList<String>();

**for** (**int** i = 6; i < parametri.size(); i++) {

descrizione.add(parametri.get(i));

}

**if** (unita == Misura.***U***) {

ArticoloNumerabile articolo = **new** ArticoloNumerabile(reparto, unita, quantitaInt, prezzoEuro,

codice, consumoWatt, descrizione);

Utilita.*listaArticoli*.put(codice, articolo);

} **else** {

ArticoloNonNumerabile articoloN = **new** ArticoloNonNumerabile(reparto, unita, quantitaDouble,

prezzoEuro, codice, descrizione);

Utilita.*listaArticoli*.put(codice, articoloN);

}

}

br.close();

} **catch** (IllegalArgumentException e) {

Utilita.*elementiNonRiconosciuti* += 1;

System.***out***.println("Errore durante il caricamento del file: uno o più parametri hanno un formato errato.");

} **catch** (ArrayIndexOutOfBoundsException e) {

Utilita.*elementiNonRiconosciuti* += 1;

System.***out***.println("Errore durante il caricamento del file: numero di parametri errato.");

} **catch** (IOException e) {

System.***out***.println("Errore");

}

}

}

SCRITTURA FILE

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.io.IOException;

**import** java.util.\*;

**public** **class** ScritturaFile {

**public** **void** ScritturaNelFile() **throws** FileNotFoundException, ArrayIndexOutOfBoundsException {

File txt = **new** File("EMPORIO.TXT");

BufferedReader br;

**try** {

**int** presenza = 0;

**for** (Map.Entry<String, Articolo> entry1 : Utilita.*listaArticoli*.entrySet()) {

String code = entry1.getValue().getCodice();

br = **new** BufferedReader(**new** FileReader(txt));

String linea1 = br.readLine();

**while** (linea1 != **null**) {

List<String> linea = **new** ArrayList<String>();

linea = Arrays.*asList*(linea1.split(";"));

String cod = linea.get(4);

**if** (cod.equals(code)) {

presenza++;

}

linea1 = br.readLine();

}

**if** (presenza < 1) {

ScritturaTXT(code);

}

br.close();

presenza = 0;

}

} **catch** (IOException e) {

System.***out***.println("Errore lettura file di testo.");

}

}

**private** **void** ScritturaTXT(String code) {

BufferedWriter bw;

FileWriter fw;

**try** {

String lines = " ";

String descrizione = " ";

**for** (Map.Entry<String, Articolo> entry : Utilita.*listaArticoli*.entrySet()) {

**if** (entry.getKey().equals(code)) {

ArrayList<String> desc = entry.getValue().getDescrizione();

**for** (**int** i = 0; i < desc.size(); i++) {

descrizione = desc.get(i);

}

lines += "\n" + entry.getValue().getReparto() + ";" + entry.getValue().getUnita() + ";"

+ entry.getValue().getQuantita() + ";" + entry.getValue().getPrezzoEuro() + ";"

+ entry.getValue().getCodice() + ";" + entry.getValue().getConsumoWatt();

**if** (descrizione != **null** && descrizione.length() > 0) {

lines += ";" + descrizione + ";";

}

fw = **new** FileWriter("EMPORIO.TXT", **true**);

bw = **new** BufferedWriter(fw);

bw.write(lines);

bw.close();

}

}

} **catch** (IOException e) {

System.***out***.println("Impossibile scrivere nel file!");

}

System.***out***.println("Azione completata!");

}

}

MAIN

**import** java.io.BufferedReader;

**import** java.io.InputStreamReader;

**public** **class** Main {

**public** **static** **void** main(String[] args) **throws** Exception, NullPointerException {

LetturaFile rf = **new** LetturaFile();

rf.LetturaTXT();

System.***out***.print("\nSeleziona un'azione digitando il numero corrispondente.\n"

+ "-------------------------------------------------------\n"

+ "| 1 - calcolo del valore totale del magazzino |\n"

+ "| 2 - vendita di un prodotto |\n"

+ "| 3 - inserimento di un prodotto |\n"

+ "| 4 - ricerca di un prodotto mediante codice univoco |\n"

+ "| 5 - stampa delle righe non riconosciute |\n"

+ "| 6 - stampa di tutti gli articoli |\n"

+ "| 7 - terminare il programma |\n"

+ "-------------------------------------------------------\n");

BufferedReader brInput = **new** BufferedReader(**new** InputStreamReader(System.***in***)); // scrittura da console

**int** inputNumb = 0;

**while** (inputNumb < 7) {

System.***out***.print("Inserisci il numero dell'azione da eseguire: ");

**try** {

inputNumb = Integer.*parseInt*(brInput.readLine());

} **catch** (NumberFormatException e) {

System.***out***.println("Formato non valido, inserisci un numero tra quelli presenti.");

} **catch** (IllegalArgumentException e) {

System.***out***.println("Formato non valido, inserisci un numero tra quelli presenti.");

}

OperazioniMagazzino m = **new** OperazioniMagazzino();

**switch** (inputNumb) {

**case** 0:

**break**;

**case** 1:

m.CalcoloValoreMagazzino();

**break**;

**case** 2:

m.VenditaArticolo();

**break**;

**case** 3:

m.InserimentoArticolo();

**break**;

**case** 4:

m.RicercaArticolo();

**break**;

**case** 5:

System.***out***.println("Elementi non riconosciuti: " + Utilita.*elementiNonRiconosciuti*);

**break**;

**case** 6:

**for** (String name : Utilita.*listaArticoli*.keySet()) {

String key = name;

String value = Utilita.*listaArticoli*.get(name).toString();

System.***out***.println(key + " - " + value);

}

**break**;

**case** 7:

ScritturaFile wf = **new** ScritturaFile();

wf.ScritturaNelFile();

**break**;

}

}

System.***out***.println("\n\nGrazie per aver usato il nostro programma. ");

}

}

EMPORIO.TXT

S;U;20;10.0;73341252;0.0;palla da calcio;

F;U;10;5.5;21253378;0.0;cacciavite;

L;U;5;13.0;34957621;0.0;pannello;

E;U;20;23.0;25896314;25.0;lampada;

F;L;21.3;25.0;25874569;0.0;vernice;

C;U;2;50.0;61293504;0.0;tappeto;

S;U;20;12.0;73253252;0.0;palla da bowling;

L;K;30.5;12.0;33445566;0.0;legna;

S;U;20;12.0;73251252;0.0;palla da basket;

Titolo: **Emporio**

Si vuole realizzare un programma che gestisce gli articoli di un emporio. Gli articoli sono memorizzati in un file dal nome EMPORIO.TXT. I dati relativi a ciascun articolo sono presenti in una riga del file. Ad esempio

E;U;10;1,5;87678350;Lampada;150

Indica che l'articolo è del reparto [E] elettricità, viene venduto a pezzo [U], ve ne sono 10, costa 1.5 Euro, ha un codice univoco, si tratta di una lampada e consuma 150 Watt.

La prima lettera indica il reparto: [E] elettricità, [F] ferramenta, [L] legno, etc.

La seconda lettera l'unità di misura: [U] per unità, [K] per chilo, [L] per litro, ecc. Di seguito il numero di pezzi presenti in magazzino. Quindi il prezzo espresso in Euro seguito da un codice di 8 cifre che rappresenta il codice a barre univoco per ogni articolo.

A seguire altre caratteristiche specifiche per il tipo di articolo. Il separatore dei valori è ";".

Il programma legge ciascuna riga e crea un oggetto per ogni articolo. Se la riga rappresenta un oggetto non riconosciuto dal programma, viene ignorato. Un contatore indica le righe non riconosciute. Se vi sono due articolo con lo stesso codice univoco viene segnalato un errore. La quantità non può essere un numero negativo.

Una volta caricato l'inventario si possono eseguire le seguenti operazioni:

\* calcolo del valore totale del magazzino

\* vendita di un prodotto

\* inserimento di un prodotto

\* ricerca di un prodotto mediante codice univoco.

Vi possono essere operazioni che possono essere applicate ad articoli differenti per tipologia. Ad esempio il calcolo della resa. ovvero il rapporto costo diviso unità di misura è applicabile solo agli articoli divisibili come vernici, lampada, etc.

Il progetto deve essere corredato dalle seguenti statistiche, ovvero la numerosità di ciascun categoria di costrutti presenti nel progetto.

Classi \_\_ Interfacce \_\_ Classi astratte \_\_ Sovraccarico di metodi/costruttori \_\_ Sovrascrittura di metodi \_\_ Eccezioni \_\_ Enumerazioni \_\_

public protected private

Attributi \_\_ \_\_ \_\_

Metodi \_\_ \_\_ \_\_

classi attributi metodi

Uso di final \_\_ \_\_ \_\_

Uso di static \_\_ \_\_ \_\_