Esercizio 1: Gestione di una scuola

Si desidera creare un'applicazione per gestire una lista di studenti presso una scuola. Ogni studente è caratterizzato da un nome, un cognome e un numero di matricola univoco. L'applicazione deve permettere di:

1. Aggiungere uno studente alla lista.
2. Rimuovere uno studente dalla lista utilizzando il numero di matricola.
3. Visualizzare l'elenco completo degli studenti.
4. Ricerca di uno studente utilizzando il numero di matricola e visualizzazione delle sue informazioni.
5. Calcolare la media dell'età degli studenti presenti nella lista.

Implementare un'applicazione che soddisfi i requisiti sopra elencati utilizzando la classe **ArrayList** per gestire la lista degli studenti.

import java.util.ArrayList;

class Studente {

private String nome;

private String cognome;

private int numeroMatricola;

private int eta;

public Studente(String nome, String cognome, int numeroMatricola, int eta) {

this.nome = nome;

this.cognome = cognome;

this.numeroMatricola = numeroMatricola;

this.eta = eta;

}

// Metodi getter e setter

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 getNumeroMatricola() {

return numeroMatricola;

}

public void setNumeroMatricola(int numeroMatricola) {

this.numeroMatricola = numeroMatricola;

}

public int getEta() {

return eta;

}

public void setEta(int eta) {

this.eta = eta;

}

@Override

public String toString() {

return "Studente{" +

"nome='" + nome + '\'' +

", cognome='" + cognome + '\'' +

", numeroMatricola=" + numeroMatricola +

", eta=" + eta +

'}';

}

}

class Scuola {

private ArrayList<Studente> listaStudenti;

public Scuola() {

this.listaStudenti = new ArrayList<>();

}

// Metodo per aggiungere uno studente alla lista

public void aggiungiStudente(Studente studente) {

listaStudenti.add(studente);

System.out.println("Studente aggiunto con successo.");

}

// Metodo per rimuovere uno studente dalla lista utilizzando il numero di matricola

public void rimuoviStudente(int numeroMatricola) {

for (Studente studente : listaStudenti) {

if (studente.getNumeroMatricola() == numeroMatricola) {

listaStudenti.remove(studente);

System.out.println("Studente rimosso con successo.");

return;

}

}

System.out.println("Studente non trovato con il numero di matricola specificato.");

}

// Metodo per visualizzare l'elenco completo degli studenti

public void visualizzaElencoStudenti() {

if (listaStudenti.isEmpty()) {

System.out.println("Nessuno studente presente nella lista.");

} else {

System.out.println("Elenco degli studenti:");

for (Studente studente : listaStudenti) {

System.out.println(studente);

}

}

}

// Metodo per trovare uno studente utilizzando il numero di matricola e visualizzare le sue informazioni

public void ricercaStudente(int numeroMatricola) {

for (Studente studente : listaStudenti) {

if (studente.getNumeroMatricola() == numeroMatricola) {

System.out.println("Studente trovato:");

System.out.println(studente);

return;

}

}

System.out.println("Studente non trovato con il numero di matricola specificato.");

}

// Metodo per calcolare la media dell'età degli studenti presenti nella lista

public double calcolaMediaEta() {

if (listaStudenti.isEmpty()) {

return 0;

}

int sommaEta = 0;

for (Studente studente : listaStudenti) {

sommaEta += studente.getEta();

}

return (double) sommaEta / listaStudenti.size();

}

}

public class Main {

public static void main(String[] args) {

Scuola scuola = new Scuola();

// Aggiunta degli studenti

scuola.aggiungiStudente(new Studente("Mario", "Rossi", 1001, 20));

scuola.aggiungiStudente(new Studente("Luigi", "Verdi", 1002, 21));

scuola.aggiungiStudente(new Studente("Giulia", "Bianchi", 1003, 22));

// Visualizzazione dell'elenco degli studenti

scuola.visualizzaElencoStudenti();

// Ricerca di uno studente utilizzando il numero di matricola

scuola.ricercaStudente(1002);

// Rimozione di uno studente utilizzando il numero di matricola

scuola.rimuoviStudente(1002);

// Calcolo della media dell'età degli studenti

System.out.println("Media dell'età degli studenti: " + scuola.calcolaMediaEta());

}

}

Esercizio 2 - Concessionaria

Si desidera creare un'applicazione per gestire una flotta di auto presso una concessionaria. Ogni auto è caratterizzata da un modello, una marca, un anno di produzione, un prezzo di vendita e uno stato (disponibile o venduta). L'applicazione deve permettere di:

1. Aggiungere un'auto alla flotta.
2. Visualizzare l'elenco completo delle auto disponibili.
3. Visualizzare l'elenco completo delle auto vendute.
4. Vendere un'auto dalla flotta.
5. Calcolare il totale delle vendite effettuate.
6. Visualizzare le informazioni dettagliate di un'auto utilizzando il modello.

import java.util.ArrayList;

// Definizione della classe Auto

class Auto {

private String modello;

private String marca;

private int annoProduzione;

private double prezzo;

private boolean disponibile;

public Auto(String modello, String marca, int annoProduzione, double prezzo) {

this.modello = modello;

this.marca = marca;

this.annoProduzione = annoProduzione;

this.prezzo = prezzo;

this.disponibile = true;

}

// Metodi getter e setter

public String getModello() {

return modello;

}

public void setModello(String modello) {

this.modello = modello;

}

public String getMarca() {

return marca;

}

public void setMarca(String marca) {

this.marca = marca;

}

public int getAnnoProduzione() {

return annoProduzione;

}

public void setAnnoProduzione(int annoProduzione) {

this.annoProduzione = annoProduzione;

}

public double getPrezzo() {

return prezzo;

}

public void setPrezzo(double prezzo) {

this.prezzo = prezzo;

}

public boolean isDisponibile() {

return disponibile;

}

public void setDisponibile(boolean disponibile) {

this.disponibile = disponibile;

}

@Override

public String toString() {

return "Auto{" +

"modello='" + modello + '\'' +

", marca='" + marca + '\'' +

", annoProduzione=" + annoProduzione +

", prezzo=" + prezzo +

", disponibile=" + disponibile +

'}';

}

}

// Definizione della classe Concessionaria

class Concessionaria {

private ArrayList<Auto> flottaAuto;

private ArrayList<Auto> autoVendute;

public Concessionaria() {

this.flottaAuto = new ArrayList<>();

this.autoVendute = new ArrayList<>();

}

// Metodo per aggiungere un'auto alla flotta

public void aggiungiAuto(Auto auto) {

flottaAuto.add(auto);

System.out.println("Auto aggiunta con successo alla flotta.");

}

// Metodo per visualizzare l'elenco completo delle auto disponibili

public void visualizzaAutoDisponibili() {

if (flottaAuto.isEmpty()) {

System.out.println("Nessuna auto disponibile nella flotta.");

} else {

System.out.println("Auto disponibili presso la concessionaria:");

for (Auto auto : flottaAuto) {

if (auto.isDisponibile()) {

System.out.println(auto);

}

}

}

}

// Metodo per visualizzare l'elenco completo delle auto vendute

public void visualizzaAutoVendute() {

if (autoVendute.isEmpty()) {

System.out.println("Nessuna auto venduta.");

} else {

System.out.println("Auto vendute presso la concessionaria:");

for (Auto auto : autoVendute) {

System.out.println(auto);

}

}

}

// Metodo per vendere un'auto dalla flotta

public void vendiAuto(String modello) {

for (Auto auto : flottaAuto) {

if (auto.getModello().equals(modello) && auto.isDisponibile()) {

auto.setDisponibile(false);

autoVendute.add(auto);

System.out.println("Auto venduta con successo.");

return;

}

}

System.out.println("Auto non trovata o già venduta.");

}

// Metodo per calcolare il totale delle vendite effettuate

public double calcolaTotaleVendite() {

double totale = 0;

for (Auto auto : autoVendute) {

totale += auto.getPrezzo();

}

return totale;

}

// Metodo per visualizzare le informazioni dettagliate di un'auto utilizzando il modello

public void visualizzaDettagliAuto(String modello) {

for (Auto auto : flottaAuto) {

if (auto.getModello().equals(modello)) {

System.out.println("Dettagli dell'auto:");

System.out.println(auto);

return;

}

}

System.out.println("Auto non trovata.");

}

}

public class Main {

public static void main(String[] args) {

Concessionaria concessionaria = new Concessionaria();

// Aggiunta di alcune auto alla flotta

concessionaria.aggiungiAuto(new Auto("Giulia", "Alfa Romeo", 2022, 40000));

concessionaria.aggiungiAuto(new Auto("A4", "Audi", 2021, 45000));

concessionaria.aggiungiAuto(new Auto("Civic", "Honda", 2020, 35000));

// Visualizzazione delle auto disponibili

System.out.println("Auto disponibili:");

concessionaria.visualizzaAutoDisponibili();

// Vendita di un'auto

concessionaria.vendiAuto("Giulia");

// Visualizzazione delle auto disponibili e vendute

System.out.println("\nAuto disponibili dopo la vendita:");

concessionaria.visualizzaAutoDisponibili();

System.out.println("\nAuto vendute:");

concessionaria.visualizzaAutoVendute();

// Calcolo del totale delle vendite effettuate

System.out.println("\nTotale delle vendite effettuate: " + concessionaria.calcolaTotaleVendite());

// Visualizzazione dei dettagli di un'auto

System.out.println("\nDettagli dell'auto Giulia:");

concessionaria.visualizzaDettagliAuto("Giulia");

}

}