МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

Протокол лабораторної роботи №6

з дисципліни об'єктно-орієнтоване програмування

на тему: «Абстрактні класи та інтерфейси. Механізм зворотного виклику»

Виконала студентка групи

ІПЗ-212

Ярош Є.В.

Прийняв

Рудніченко М.Д.

Одеса, 2022

ЗМІСТ

[ВСТУП](#_30j0zll) 3

[ХІД РОБОТИ](#_8fy8djbnwuy9) 4

[ВИСНОВОК](#_2et92p0) 9

# ВСТУП

Мета роботи - ознайомитись з механізмом абстрактних класів та інтерфейсів, розібратися з тим, коли і для чого необхідно використовувати абстрактні класи та інтерфейси, вивчити використання механізму зворотного виклику.

# ХІД РОБОТИ

1. Переробіть лабораторну роботу 5 за допомогою механізму абстрактних класів.

Код програми:

import java.util.Scanner;

abstract class Human {

private String surname;

private String name;

private int age;

public void setSurname(String surname) {

this.surname = surname;

};

public void setName(String name) {

this.name = name;

};

public void setAge(int age) {

this.age = age;

};

public String getSurname() {

return surname;

};

public String getName() {

return name;

};

public int getAge() {

return age;

};

public abstract void printInfo();

};

class Person extends Human {

@Override

public void printInfo() {

System.*out*.println("Person " + getSurname() + " " + getName() + ", age: " + getAge());

};

};

class Student extends Human {

private int group;

private int cardNumber;

public void setGroup(int group) {

this.group = group;

};

public void setCardNumber(int cardNumber) {

this.cardNumber = cardNumber;

};

public int getGroup() {

return group;

};

public int getCardNumber() {

return cardNumber;

};

@Override

public void printInfo() {

System.*out*.println("Student of " + getGroup() + " group " + getSurname() + " " + getName() +

", age: " + getAge() + ". \nStudent's card number: " + getCardNumber());

};

};

class Lecturer extends Human {

private String chair;

private int salary;

public void setChair(String chair) {

this.chair = chair;

};

public void setSalary(int salary) {

this.salary = salary;

};

public float getSalary() {

return salary;

};

public String getChair() {

return chair;

};

@Override

public void printInfo() {

System.*out*.println("Lecturer of " + getChair() + " chair " + getSurname() + " " + getName() +

", age: " + getAge() + ". \nSalary: " + getSalary());

};

};

public class Main {

public static void main(String [] args) {

System.*out*.println("Enter info about person:");

Scanner in = new Scanner(System.*in*);

Human[] people = new Human[9];

System.*out*.println("Students:");

for (int i = 0; i < 3; i++) {

Human student = new Student();

System.*out*.println("Enter surname: ");

student.setSurname(in.next());

System.*out*.println("Enter name: ");

student.setName(in.next());

System.*out*.println("Enter age: ");

student.setAge(in.nextInt());

System.*out*.println("Enter group: ");

Student student1 = (Student) student;

student1.setGroup(in.nextInt());

System.*out*.println("Enter Student's card number: ");

student1.setCardNumber(in.nextInt());

people[i] = student;

};

System.*out*.println("Lecturers:");

for (int i = 3; i < 6; i++) {

Human lecturer = new Lecturer();

System.*out*.println("Enter surname: ");

lecturer.setSurname(in.next());

System.*out*.println("Enter name: ");

lecturer.setName(in.next());

System.*out*.println("Enter age: ");

lecturer.setAge(in.nextInt());

System.*out*.println("Enter chair: ");

Lecturer lecturer1 = (Lecturer) lecturer;

lecturer1.setChair((in.next() + " " + in.next()));

System.*out*.println("Enter salary: ");

lecturer1.setSalary(in.nextInt());

people[i] = lecturer;

};

System.*out*.println("Persons:");

for (int i = 6; i < 9; i++) {

Human person = new Person();

System.*out*.println("Enter surname: ");

person.setSurname(in.next());

System.*out*.println("Enter name: ");

person.setName(in.next());

System.*out*.println("Enter age: ");

person.setAge(in.nextInt());

people[i] = person;

};

in.close();

for (int i = 0; i < 9;i++) {

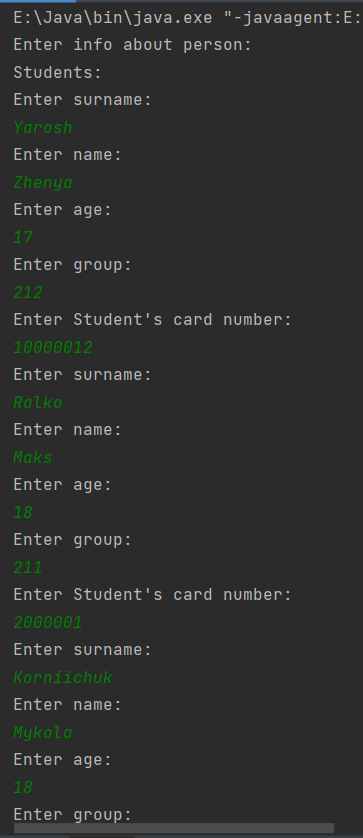
people[i].printInfo();

};

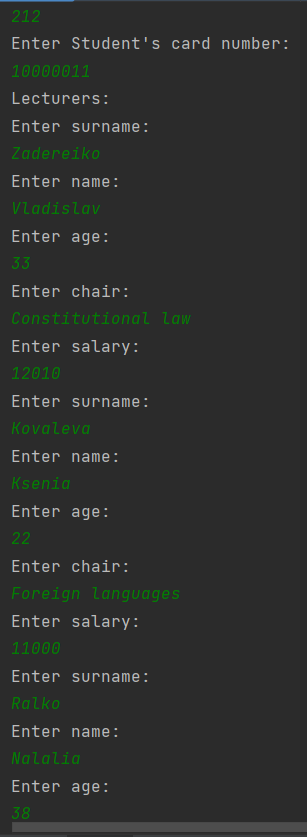
};

}

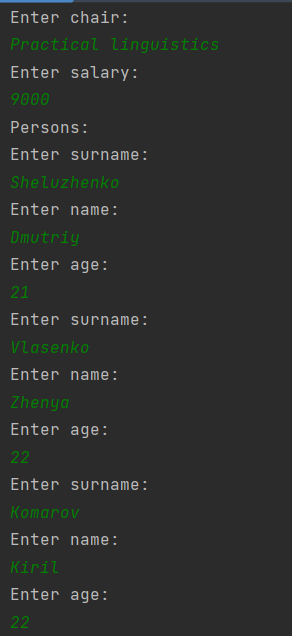
Результат роботи програми наведено на рис.1. - рис.4.



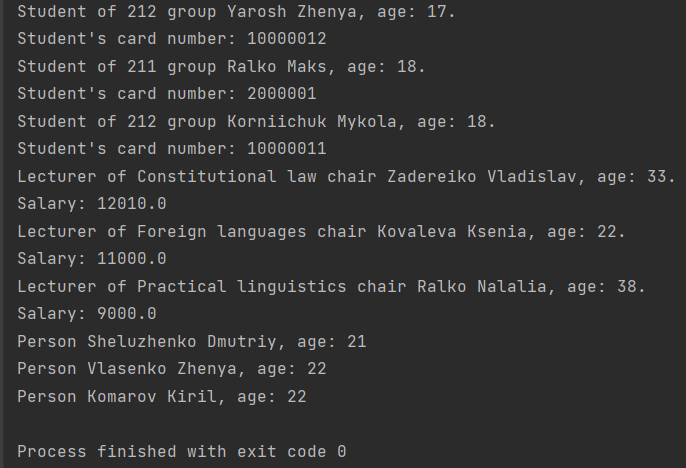
Риcунок 1 – Результат роботи програми.



Риcунок 2 – Результат роботи програми.



Риcунок 3 – Результат роботи програми.



Риcунок 4 – Результат роботи програми.

2. Переробіть лабораторну роботу 5 за допомогою механізму інтерфейсів

Код програми:

import java.util.Scanner;

interface Human {

public void setSurname(String surname);

public void setName(String name);

public void setAge(int age);

public String getSurname();

public String getName();

public int getAge();

public void printInfo();

};

class Person implements Human {

private String surname;

private String name;

private int age;

@Override

public void setSurname(String surname) {

this.surname = surname;

};

@Override

public void setName(String name) {

this.name = name;

};

@Override

public void setAge(int age) {

this.age = age;

};

@Override

public String getSurname() {

return surname;

};

@Override

public String getName() {

return name;

};

@Override

public int getAge() {

return age;

};

@Override

public void printInfo() {

System.*out*.println("Person " + getSurname() + " " + getName() + ", age: " + getAge());

};

};

class Student extends Person {

private int group;

private int cardNumber;

public void setGroup(int group) {

this.group = group;

};

public void setCardNumber(int cardNumber) {

this.cardNumber = cardNumber;

};

public int getGroup() {

return group;

};

public int getCardNumber() {

return cardNumber;

};

@Override

public void printInfo() {

System.*out*.println("Student of " + getGroup() + " group " + getSurname() + " " + getName() +

", age: " + getAge() + ". \nStudent's card number: " + getCardNumber());

};

};

class Lecturer extends Person {

private String chair;

private int salary;

public void setChair(String chair) {

this.chair = chair;

};

public void setSalary(int salary) {

this.salary = salary;

};

public float getSalary() {

return salary;

};

public String getChair() {

return chair;

};

@Override

public void printInfo() {

System.*out*.println("Lecturer of " + getChair() + " chair " + getSurname() + " " + getName() +

", age: " + getAge() + ". \nSalary: " + getSalary());

};

};

public class Main {

public static void main(String [] args) {

System.*out*.println("Enter info about person:");

Scanner in = new Scanner(System.*in*);

Person[] people = new Person[9];

System.*out*.println("Students:");

for (int i = 0; i < 3; i++) {

Person student = new Student();

System.*out*.println("Enter surname: ");

student.setSurname(in.next());

System.*out*.println("Enter name: ");

student.setName(in.next());

System.*out*.println("Enter age: ");

student.setAge(in.nextInt());

System.*out*.println("Enter group: ");

Student student1 = (Student) student;

student1.setGroup(in.nextInt());

System.*out*.println("Enter Student's card number: ");

student1.setCardNumber(in.nextInt());

people[i] = student;

};

System.*out*.println("Lecturers:");

for (int i = 3; i < 6; i++) {

Person lecturer = new Lecturer();

System.*out*.println("Enter surname: ");

lecturer.setSurname(in.next());

System.*out*.println("Enter name: ");

lecturer.setName(in.next());

System.*out*.println("Enter age: ");

lecturer.setAge(in.nextInt());

System.*out*.println("Enter chair: ");

Lecturer lecturer1 = (Lecturer) lecturer;

lecturer1.setChair((in.next() + " " + in.next()));

System.*out*.println("Enter salary: ");

lecturer1.setSalary(in.nextInt());

people[i] = lecturer;

};

System.*out*.println("Persons:");

for (int i = 6; i < 9; i++) {

Person person = new Person();

System.*out*.println("Enter surname: ");

person.setSurname(in.next());

System.*out*.println("Enter name: ");

person.setName(in.next());

System.*out*.println("Enter age: ");

person.setAge(in.nextInt());

people[i] = person;

};

in.close();

for (int i = 0; i < 9;i++) {

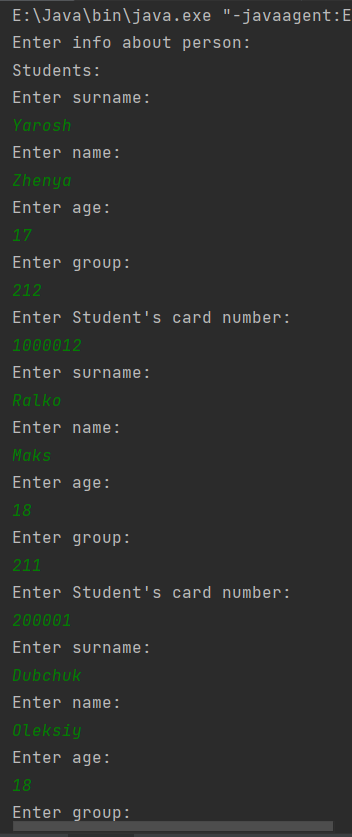
people[i].printInfo();

};

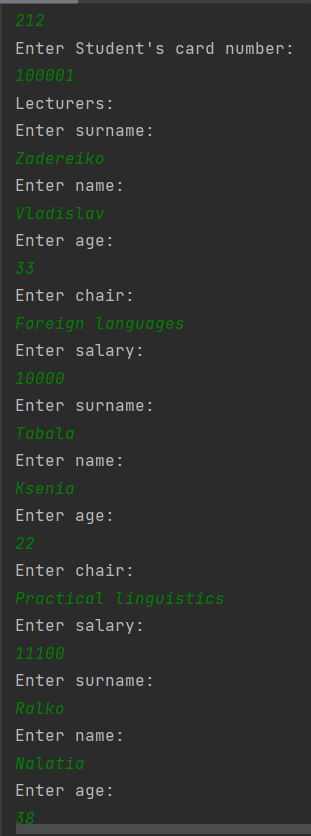
};

}

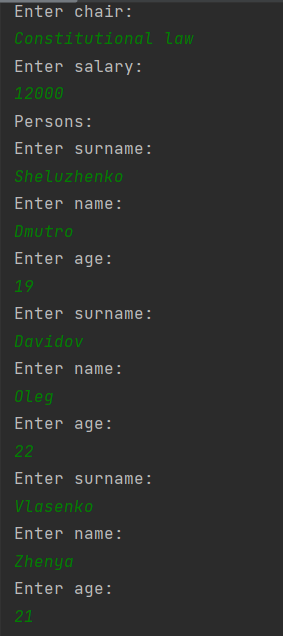
Результат роботи програми наведено на рис.5. - рис.8.



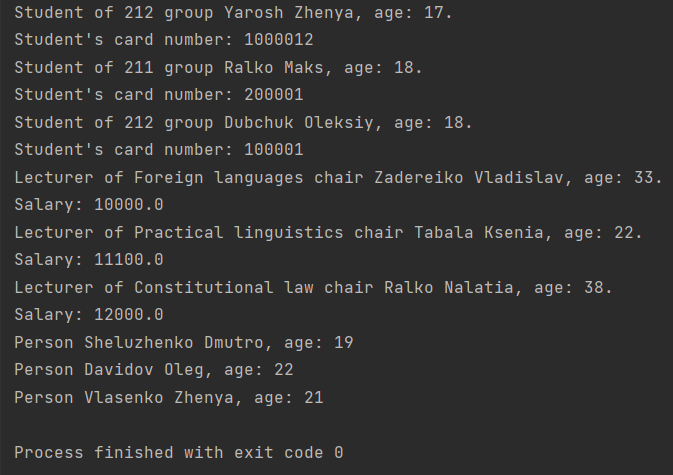
Риcунок 5 – Результат роботи програми.



Риcунок 6 – Результат роботи програми.



Риcунок 7 – Результат роботи програми.



Риcунок 8 – Результат роботи програми.

3. Незалежно від варіанта, необхідно наступне:

* У пакеті carshop.cars створити абстрактний клас Саг, що містить такі поля та методи:

int speed;

boolean isSellOut;

double regularPrice;

String color;

double getSalePrice();

* У пакеті carshop.cars створити клас Truck, який розширює клас Саг і містить такі поля та методи:

int weight;

// якщо вага більша за 2000, знижка 10%.

double getSalePrice();

* У пакеті carshop.cars створити клас Ford, який розширює клас Саг, що містить такі поля та методи:

int year;

int manufacturerDiscount;

// (Визначається як ціна мінус знижка виробника)

double getSalePrice();

* У пакеті carshop.cars створити клас Sedan, який розширює клас Саг і містить такі поля та методи:

int length;

// (Якщо довжина більше 20, знижка 5%)

double getSalePrice();

* У пакеті carshop.interface створити два інтерфейси Admin та Customer, які містять такі методи:

Admin:

getIncome() - сумарна ціна проданих автомобілів

Customer

getCarsPrice()

getCarColors()

getCarPrice(int id)

getCarColor(int id)

purchaseCar(int id)

* У пакеті carshop.impl створити клас MyOwnAutoShop, що реалізує інтерфейси Admin та Customer. Він містить об'єкт класу Sedan, 2 об'єкти класу Ford, 2 об'єкти класу Truck. Об'єкти зберігати як масиву. У цьому випадку id у методах буде індексом елемента у масиві.
* У пакеті carshop створити клас, що містить статичний метод main, який повинен демонструвати роботу програми.

У кожному варіанті необхідно створити проект, у ньому кілька пакетів, розмістити у цих пакетах необхідний інтерфейс (або абстрактний клас) та конкретні реалізації, а також основний клас:

Код класу Car:

package carshop.cars;

public abstract class Car {

int speed;

public void setSpeed(int speed) {

this.speed = speed;

}

public boolean isSellOut = false;

public double regularPrice;

public void setRegularPrice(int regularPrice) {

this.regularPrice = regularPrice;

}

public String color;

public void setColor(String color) {

this.color = color;

}

public double getSalePrice(double regularPrice) {

return regularPrice;

};

};

Код класу Ford:

package carshop.cars;

public class Ford extends Car {

int year;

public void setYear(int year) {

this.year = year;

}

int manufacturerDiscount;

public void setManufacturerDiscount(int manufacturerDiscount) {

this.manufacturerDiscount = manufacturerDiscount;

}

double salePrice;

@Override

public double getSalePrice(double regularPrice) {

double salePrice = regularPrice - manufacturerDiscount;

return salePrice;

};

};

Код класу Sedan:

package carshop.cars;

public class Sedan extends Car {

int length;

public void setLength(int length) {

this.length = length;

}

double salePrice;

@Override

public double getSalePrice(double regularPrice) {

if (length > 20) { salePrice = regularPrice - (regularPrice / 20 );}

return salePrice;

};

};

Код класу Truck:

package carshop.cars;

public class Truck extends Car {

int weight;

public void setWeight(int weight) {

this.weight = weight;

}

double salePrice;

@Override

public double getSalePrice(double regularPrice) {

if (weight > 2000) { salePrice = regularPrice - (regularPrice / 10);}

else salePrice = regularPrice;

return salePrice;

};

};

Код інтерфейсу Customer:

package carshop.intrface;

public interface Customer {

public void getCarsPrice();

public void getCarsColors();

public double getCarPrice(int id);

public String getCarColor(int id);

public void purchaseCar(int id);

}

Код інтерфейсу Admin:

package carshop.intrface;

public interface Admin {

public double getIncome();

}

Код, що імплементує класи та інтерфейси:

package carshop.impl;

import carshop.cars.\*;

import carshop.intrface.Admin;

import carshop.intrface.Customer;

import java.util.Scanner;

public class MyOwnAutoShop implements Admin, Customer {

@Override

public double getIncome() {

double income = 0;

for (Car car : shop) {

if (car.isSellOut == true) income += car.getSalePrice(car.regularPrice);

}

return income;

};

@Override

public void getCarsPrice() {

for (Car car : shop) {

System.*out*.println(car.regularPrice);

}

};

@Override

public void getCarsColors() {

for (Car car : shop) {

System.*out*.println(car.color);

};

};

@Override

public double getCarPrice(int id) {

return shop[id].getSalePrice(shop[id].regularPrice);

};

@Override

public String getCarColor(int id) {

return shop[id].color;

};

@Override

public void purchaseCar(int id) {

shop[id].isSellOut = true;

};

Car[] shop = new Car[5]; {

Sedan sedan = new Sedan();

shop[0] = sedan;

Ford ford1 = new Ford();

shop[1] = ford1;

Ford ford2 = new Ford();

shop[2] = ford2;

Truck truck1 = new Truck();

shop[3] = truck1;

Truck truck2 = new Truck();

shop[4] = truck2;

Scanner in = new Scanner(System.*in*);

System.*out*.println("Please enter the details about the cars");

System.*out*.println("Please enter the details about sedan");

System.*out*.println("Enter speed: ");

sedan.setSpeed(in.nextInt());

System.*out*.println("Enter regular price: ");

sedan.setRegularPrice(in.nextInt());

System.*out*.println("Enter color: ");

sedan.setColor(in.next());

System.*out*.println("Enter length: ");

sedan.setLength(in.nextInt());

System.*out*.println("Please enter the details about ford");

System.*out*.println("Enter speed: ");

ford1.setSpeed(in.nextInt());

System.*out*.println("Enter regular price: ");

ford1.setRegularPrice(in.nextInt());

System.*out*.println("Enter color: ");

ford1.setColor(in.next());

System.*out*.println("Enter year: ");

ford1.setYear(in.nextInt());

System.*out*.println("Enter manufacturer discount: ");

ford1.setManufacturerDiscount(in.nextInt());

System.*out*.println("Please enter the details about ford");

System.*out*.println("Enter speed: ");

ford2.setSpeed(in.nextInt());

System.*out*.println("Enter regular price: ");

ford2.setRegularPrice(in.nextInt());

System.*out*.println("Enter color: ");

ford2.setColor(in.next());

System.*out*.println("Enter year: ");

ford2.setYear(in.nextInt());

System.*out*.println("Enter manufacturer discount: ");

ford2.setManufacturerDiscount(in.nextInt());

System.*out*.println("Please enter the details about truck");

System.*out*.println("Enter speed: ");

truck1.setSpeed(in.nextInt());

System.*out*.println("Enter regular price: ");

truck1.setRegularPrice(in.nextInt());

System.*out*.println("Enter color: ");

truck1.setColor(in.next());

System.*out*.println("Enter weight: ");

truck1.setWeight(in.nextInt());

System.*out*.println("Please enter the details about truck");

System.*out*.println("Enter speed: ");

truck2.setSpeed(in.nextInt());

System.*out*.println("Enter regular price: ");

truck2.setRegularPrice(in.nextInt());

System.*out*.println("Enter color: ");

truck2.setColor(in.next());

System.*out*.println("Enter weight: ");

truck2.setWeight(in.nextInt());

System.*out*.println("Welcome to CarShop!");

System.*out*.println("Here's our assortement of cars:");

System.*out*.println("The colors we have: ");

getCarsColors();

System.*out*.println("The pricing we have: ");

getCarsPrice();

System.*out*.println("Enter id of car to see the details about (to end enter 5): ");

int help = in.nextInt();

while (help != 5) {

System.*out*.println("The price is: " + getCarPrice(help));

System.*out*.println("The color is: " + getCarColor(help));

help = in.nextInt();

}

System.*out*.println("Enter id of car you want to buy (to end purchase enter 5): ");

help = in.nextInt();

while (help != 5) {

purchaseCar(help);

help = in.nextInt();

}

in.close();

System.*out*.println("Income of carshop is: " + getIncome());

}

};

Код головного класу застосунка:

import carshop.impl.MyOwnAutoShop;

public class App {

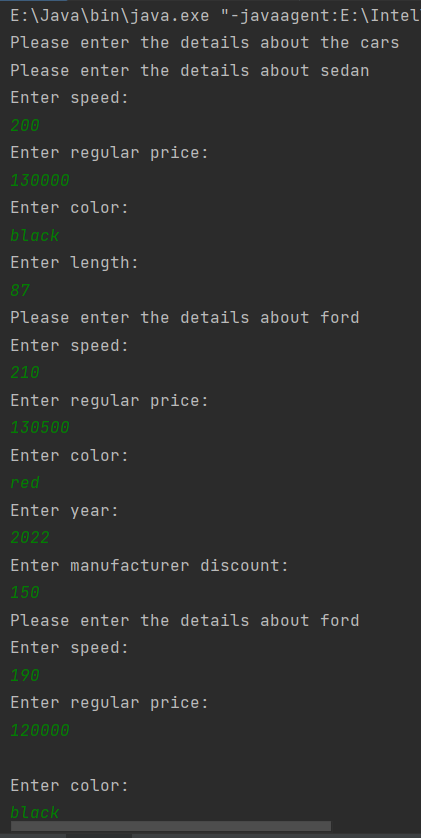
public static void main(String[] args) {

MyOwnAutoShop myShop = new MyOwnAutoShop();

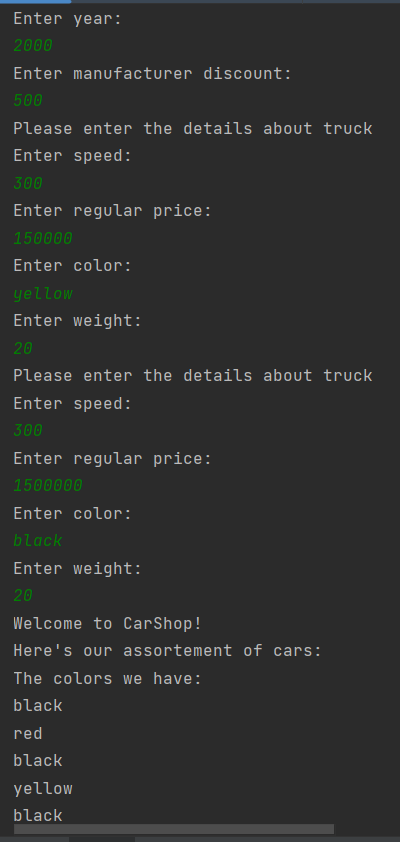
}

}

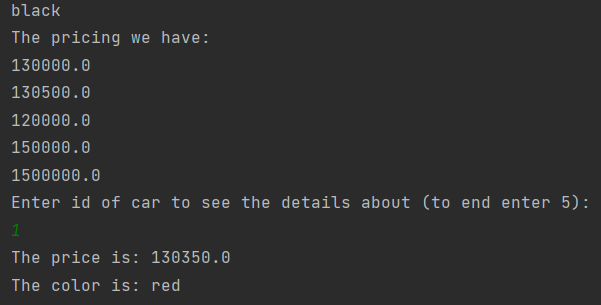
Результат роботи програми наведено на рис.9 - рис.11.



Риcунок 9 – Результат роботи програми.



Риcунок 10 – Результат роботи програми.



Риcунок 11 – Результат роботи програми.

3. Варіант: 12

Необхідно описати інтерфейс, що містить одну функцію: int count(int number);

Даний інтерфейс дозволяє враховувати характеристики чисел. Необхідно реалізувати два класи для цього інтерфейсу. Перший клас повинен обчислити кількість символів у десятковому записі для цього числа. Другий клас повинен обчислити кількість різних простих чисел, які є множниками цього числа. Необхідно, щоб програма запросила у користувача число та видала статистичні відомості для обох класів.

Код програми:

import java.util.\*;

interface Function {

int count (int number);

}

class SymbolsCount implements Function {

@Override

public int count (int number) {

int length = String.*valueOf*(number).length();

return length;

};

}

class NumberFactors implements Function {

@Override

public int count (int number) {

int factor = 0;

if (number < 2) factor = 1;

int c = 2;

HashMap<Integer, Boolean> visited = new HashMap<>();

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

visited.put(i, false);

}

while (number > 1) {

if (number % c == 0) {

if(visited.containsKey(c)){

if (!visited.get(c)) {

factor++;

visited.put(c, true);

}

}

number /= c;

}

else

c++;

}

return factor;

};

}

public class Numbers {

public static void main(String[] args) {

System.*out*.println("Input your number: ");

Scanner in = new Scanner(System.*in*);

SymbolsCount num = new SymbolsCount();

NumberFactors number = new NumberFactors();

int user;

user = in.nextInt();

in.close();

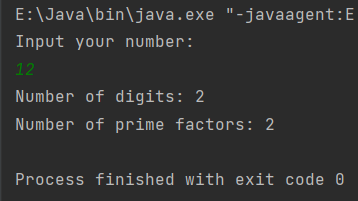
System.*out*.println("Number of digits: " + num.count(user));

System.*out*.println("Number of prime factors: " + number.count(user));

}

}

Результат роботи програми наведено на рис.12



Риcунок 12 – Результат роботи програми.

# ВИСНОВОК

Протягом виконання лабораторної роботи, я ознайомилася з механізмом абстрактних класів та інтерфейсів, розібралася з тим, коли і для чого необхідно використовувати абстрактні класи та інтерфейси, вивчила використання механізму зворотного виклику.