Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем

и программной инженерии

Лабораторная работа № 3

“ Интерфейсы. Наследование.”

по дисциплине

«Платформонезависимое программирование»

Выполнил:

ст. гр. ПРИ-118

В. В. Баранов

Принял:

А. Т. Казаев

Владимир 2020

**Цель работы**

Освоить базовые принципы ООП. Получить практические навыки по работе с классами и объектами на платформе Java. Научиться использовать наследование, абстрактные классы, интерфейсы.

**Выполнение работы**

**Задание 1.** Независимо от варианта, необходимо сделать следующее в среде Netbeans:

* В пакете carshop.cars создать абстрактный класс Car, содержащий следующие поля и методы:

int speed;

boolean isSellOut;

double regularPrice;

String color;

double getSalePrice();

* В пакете carshop.cars создать класс Truck, расширяющий класс Car и содержащий следующие поля и методы:

int weight;

// 2000, 10%.

double getSalePrice();

* В пакете carshop.cars создать класс Ford, расширяющий класс Car, содержащий следующие поля и методы:

int year;

int manufacturerDiscount;

// ( )

double getSalePrice();

* В пакете carshop.cars создать класс Sedan, расширяющий класс Car и содержащий следующие поля и методы:

int length;

// ( 20, 5%)

double getSalePrice();

* В пакете carshop.interface создать два интерфейса Admin и Customer, содержащие следующие методы:

Admin:

getIncome()

Customer:

getCarsPrice()

getCarColors()

getCarPrice(int id)

getCarColor(int id)

purchaseCar(int id)

* В пакете carshop.impl создать класс MyOwnAutoShop реализующий интерфейсы Admin и Customer. Он содержит объект класса Sedan, 2 объекта класса Ford, 2 объекта класса Truck. Объекты хранить в виде массива. В этом случае id в методах будет индексом элемента в массиве.
* В пакете carshop, создать класс, содержащий статический метод main, который должен демонстрировать работу приложения.

Исходный код:

carshop.cars.Car.java

1. package carshop.cars;
2. **public** **abstract** **class** **Car** {
3. **public** int speed;
4. **public** boolean isSellOut;
5. **public** double regularPrice;
6. **public** String color;
7. **public** **abstract** double getSalePrice();
9. **public** Car(int speed, double regularPrice, String color) {
10. **this**.speed = speed;
11. **this**.isSellOut = **false**;
12. **this**.regularPrice = regularPrice;
13. **this**.color = color;
14. }
15. }

carshop.cars.Truck.java

1. package carshop.cars;
2. **public** **class** **Truck** extends Car {
3. **public** int weight;
5. **public** Truck (int speed, double regularPrice, String color, int weight) {
6. super(speed,regularPrice,color);
7. **this**.weight = weight;
8. }
10. **public** double getSalePrice() {
11. **return** (weight > 2000) ? regularPrice \* 0.9 : regularPrice;
12. }
13. }

carshop.cars.Ford.java

1. package carshop.cars;
2. **public** **class** **Ford** extends Car {
3. **public** int year;
4. **public** int manufacturerDiscount;
6. **public** Ford (int speed, double regularPrice, String color, int year, int manufacturerDiscount) {
7. super(speed,regularPrice,color);
8. **this**.year = year;
9. **this**.manufacturerDiscount = manufacturerDiscount;
10. }
11. **public** double getSalePrice() {
12. **return** regularPrice - manufacturerDiscount;
13. }
14. }

carshop.cars.Sedan.java

1. package carshop.cars;
2. **public** **class** **Sedan** extends Car {
3. **public** int length;
4. **public** Sedan(int speed, double regularPrice, String color, int length) {
5. super(speed,regularPrice,color);
6. **this**.length = length;
7. }
9. **public** double getSalePrice() {
10. **return** (length > 20) ? regularPrice \* 0.95 : regularPrice;
11. }
12. }

carshop.interfaces.Admin.java

1. package carshop.interfaces;
2. **public** **interface** Admin {
3. double getIncome();
4. }

carshop.interfaces.Customer.java

1. package carshop.interfaces;
2. **public** **interface** Customer {
3. double[] getCarsPrice();
4. String[] getCarsColor();
5. double getCarPrice(int id);
6. String getCarColor(int id);
7. boolean purchaseCar(int id);
8. }

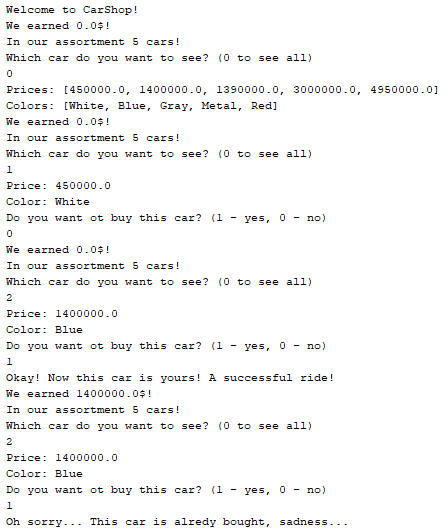
carshop.impl.MuOwnAutoShop.java

1. package carshop.impl;
2. import carshop.interfaces.\*;
3. import carshop.cars.\*;
4. **public** **class** **MyOwnAutoShop** implements Admin, Customer {
5. **private** double income;
6. **private** Car[] cars;
8. **public** MyOwnAutoShop() {
9. cars = **new** Car[5];
10. cars[0] = **new** Sedan(190, 450000, "White", 5);
11. cars[1] = **new** Ford(210, 1430000 ,"Blue", 2020, 30000);
12. cars[2] = **new** Ford(290, 1400000, "Gray", 2015, 10000);
13. cars[3] = **new** Truck(177, 3000000, "Metal", 2000);
14. cars[4] = **new** Truck(180, 5500000, "Red", 7200);
15. }
17. **public** double getIncome() {
18. **return** income;
19. }
21. **public** int getCarsCount() {
22. **return** cars.length;
23. }
25. **public** double[] getCarsPrice() {
26. double[] prices = **new** double[cars.length];
27. **for**(int i = 0; i < cars.length; i++)
28. {
29. prices[i] = getCarPrice(i);
30. }
31. **return** prices;
32. }
34. **public** String[] getCarsColor() {
35. String[] colors = **new** String[cars.length];
36. **for**(int i = 0; i < cars.length; i++)
37. {
38. colors[i] = getCarColor(i);
39. }
40. **return** colors;
41. }
43. **public** double getCarPrice(int id) {
44. **if**(id >= 0 && id < cars.length)
45. **return** cars[id].getSalePrice();
46. **else**
47. **return** 0;
48. }
50. **public** String getCarColor(int id) {
51. **if**(id >= 0 && id < cars.length)
52. **return** cars[id].color;
53. **else**
54. **return** "none";
55. }
57. **public** boolean purchaseCar(int id) {
58. **if**(id >= 0 && id < cars.length)
59. **if**(!cars[id].isSellOut)
60. {
61. cars[id].isSellOut = **true**;
62. income += getCarPrice(id);
63. **return** **true**;
64. }
65. **return** **false**;
66. }
67. }

carshop.CarShop.java

1. package carsop;
2. import carshop.impl.MyOwnAutoShop;
3. import java.util.Scanner;
4. import java.util.Arrays;
5. **public** **class** **CarShop** {
6. **public** **static** **void** main(String[] args) {
7. MyOwnAutoShop autoShop = **new** MyOwnAutoShop();
8. Scanner console = **new** Scanner(System.**in**);
10. System.**out**.println("Welcome to CarShop!");
12. int carChoice = 0;
13. **do** {
14. System.**out**.println("We earned " + autoShop.getIncome() + "$!");
15. System.**out**.println("In our assortment " + autoShop.getCarsCount() + " cars!");
16. System.**out**.println("Which car do you want to see? (0 to see all)");
17. carChoice = console.nextInt();
18. **if**(carChoice == 0)
19. {
20. System.**out**.println("Prices: " + Arrays.toString(autoShop.getCarsPrice()));
21. System.**out**.println("Colors: " + Arrays.toString(autoShop.getCarsColor()));
22. }
23. **else**
24. {
25. carChoice -= 1;
26. System.**out**.println("Price: " + autoShop.getCarPrice(carChoice));
27. System.**out**.println("Color: " + autoShop.getCarColor(carChoice));
28. System.**out**.println("Do you want ot buy this car? (1 - yes, 0 - no)");
29. int buyChice = console.nextInt();
30. **if**(buyChice == 1)
31. {
32. **if**(autoShop.purchaseCar(carChoice))
33. {
34. System.**out**.println("Okay! Now this car is yours! A successful ride!");
35. }
36. **else**
37. {
38. System.**out**.println("Oh sorry... This car is alredy bought, sadness...");
39. }
40. }
41. }
42. }
43. **while** (carChoice != -1);
44. }
45. }

Тестирование и отладка:



*Рисунок 1. Рейзльтат работы программы*

**Задание 2.** (вариант 2) Необходимо описать интерфейс, содержащий одну функцию:

int count(int number);

Данный интерфейс позволяет вычислять характеристики чисел. Необходимо реализовать два класса для этого интерфейса. Первый класс должен вычислить количество символов в десятичной записи для этого числа. Второй класс должен вычислить количество различных простых чисел, являющихся множителями данного числа. Необходимо, чтобы приложение запросило у пользователя число и выдало статистические сведения для обоих классов.

Исходный код:

counter.interfaces.Counter.java

1. package counter.interfaces;
2. **public** **interface** Counter {
3. int count(int number);
4. }

counter.impl. DecimalDigitsCount.java

1. package counter.impl;
2. import counter.interfaces.\*;
3. **public** **class** **DecimalDigitsCount** implements Counter {
5. **public** int count(int number) {
6. int result = (number == 0) ? 1 : 0;
7. **while** (number != 0) {
8. result++;
9. number /= 10;
10. }
11. **return** result;
12. }
13. }

counter.impl.PrimeFactorsCount.java

1. package counter.impl;
2. import counter.interfaces.Counter;
3. **public** **class** **PrimeFactorsCount** implements Counter {
4. **public** int count(int number) {
5. int result = 0;
6. **for**(int i = 2; i <= number; i++) {
7. **if**(number % i == 0) {
8. **if**(isPrime(i)) {
9. result++;
10. }
11. }
12. }
13. **return** result;
14. }
16. **public** **static** boolean isPrime(int number) {
17. **for** (int i = 2; i < number; i++) {
18. **if** (number % i == 0) {
19. **return** **false**;
20. }
21. }
22. **return** **true**;
23. }
24. }

counter.MainClass.java

1. package counter;
2. import counter.interfaces.Counter;
3. import counter.impl.\*;
4. import java.util.Scanner;
5. **public** **class** **MainClass**
6. {
7. **public** **static** **void** main(String[] args)
8. {
9. Counter counter;
10. Scanner console = **new** Scanner(System.**in**);
12. int x = console.nextInt();
13. counter = **new** DecimalDigitsCount();
14. System.**out**.println("Decimal digits count: " + counter.count(x));
16. counter = **new** PrimeFactorsCount();
17. System.**out**.println("Prime factors count: " + counter.count(x));
18. }
19. }

Тестирование и отладка:



*Риснуок 2. Результа работы программы*



*Рисунок 3. Результат работы программы*

**Вывод**

В результа выполнения лабораторной раоты были освоены базовые принципы ООП. Получены практические навыки по работе с классами и объектами на платформе Java. Получены навыки использования наследование, абстрактные классы, интерфейсы.