

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики

Факультет инфокоммуникационных технологий

Направление подготовки 11.03.02

Лабораторная работа №5

«Выражения и управление программным потоком. Построение классов»

Выполнил:

Швалов Даниил Андреевич

Группа: К33211

Проверил:

Иванов Сергей Евгеньевич

Санкт-Петербург

2024

Введение

Цель работы:

- создание приложений, реализующих основные управляющие конструкции;
- проектирование классов, взаимодействие между классами;
- проектирование иерархии классов для задач из различных предметных областей;
- реализация наследования и полиморфизма.

Ход работы

Задание 1. Использование битовых операций

В данном задании необходимо выполнить умножение переменной на 10, используя операции сдвига и сложения.

Поскольку умножение с помощью битовой арифметики возможно только на степени двойки, то необходимо разложить умножение на 10 на степени двойки. Таким образом, умножение на 10 можно представить как сумму умножения на 8 и умножения на 2. Для умножения на 8 необходимо сдвинуть изначальное значение на 3 бита влево, а для умножения на 2 — на 1 бит влево. На рисунке 1 приведен исходный код получившейся программы. На рисунке 2 показан пример работы программы.

```

import java.util.Scanner;

public class Task1 {
    private static int multiplyByTen(int number) {
        //      number * 8    +   number * 2
        return (number << 3) + (number << 1);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Введите число: ");
        int number = scanner.nextInt();
        int result = multiplyByTen(number);
        System.out.printf(
            "Число %d, умноженное на десять, равно %d\n",
            number,
            result
        );
    }
}

```

Рисунок 1 – Исходный код программы

```

task-1 > java Task1
Введите число: 8
Число 8, умноженное на десять, равно 80
task-1 > java Task1
Введите число: 51
Число 51, умноженное на десять, равно 510
task-1 > java Task1
Введите число: -1
Число -1, умноженное на десять, равно -10

```

Рисунок 2 – Пример работы программы

Задание 2. Использование оператора switch

В данном задании необходимо создать класс `Date`, который хранит информацию о годе, месяце и дне. Также он должен обладать методом `displayMonth`, который возвращает наименование месяца.

В ходе выполнения задания был создан класс `Date`, который обладает геттерами и сеттерами для года, месяца и дня. Каждый из них проверяет корректность передаваемых данных. Если данные некорректны, то выбрасывается исключение. Также был создан конструктор, принимающий год, месяц и день. Внутри он использует те же сеттеры.

На рисунке 3 приведен исходный код получившейся функции `displayMonth`. В ней используется оператор `switch` для создания соответствия между числовым и строковым представлением месяца.

```
public static String displayMonth(int month) {  
    switch (month) {  
        case 1:  
            return "January";  
        case 2:  
            return "February";  
        case 3:  
            return "March";  
        case 4:  
            return "April";  
        case 5:  
            return "May";  
        case 6:  
            return "June";  
        case 7:  
            return "July";  
        case 8:  
            return "August";  
        case 9:  
            return "September";  
        case 10:  
            return "October";  
        case 11:  
            return "November";  
        case 12:  
            return "December";  
        default:  
            throw new IllegalArgumentException("Invalid month value: " + month);  
    }  
}
```

Рисунок 3 – Исходный код метода displayMonth

На рисунке 4 приведен исходный код функции main. В ней создается дата, после чего выводится текстовое представление месяца. На рисунке 5 приведен вывод данной программы.

```
public static void main(String[] args) {  
    Date date = new Date(2024, 7, 3);  
    System.out.println("Month: " + Date.displayMonth(date.getMonth()));  
}
```

Рисунок 4 – Исходный код функции main



```
task-2 > java Date  
Month: July
```

Рисунок 5 – Пример работы программы

Задание 3. Проектирование классов

В данном задании необходимо спроектировать иерархию классов для хранения информации о сделке между двумя участниками.

В ходе выполнения данного задания были разработаны следующие классы:

- Account — хранит информацию об участнике сделки: его фамилию, имя и количество денежных средств;
- Product — хранит информацию о товаре: его название и стоимость;
- ProductItem — хранит информацию о товаре и его количестве;
- Deal — хранит информацию о сделке.

Для тестирования работы классов был написан код, представленный на рисунке 6. В ней считывается из консоли информация об участниках сделки, а также информация о товарах. После этого сделка закрывается. Результат работы данной программы представлен на рисунке 7.

```

public static void main(String[] args) {
    System.out.println("Enter seller information");
    Account seller = Account.read();
    System.out.println("Enter buyer information");
    Account buyer = Account.read();

    Deal deal = new Deal(seller, buyer);

    System.out.print("Enter number of product types: ");
    Scanner scanner = new Scanner(System.in);
    int productItemsCount = scanner.nextInt();
    for (int i = 0; i < productItemsCount; ++i) {
        Product product = Product.read();
        System.out.print("Enter count: ");
        int count = scanner.nextInt();
        deal.addProduct(product, count);
    }

    deal.close();

    System.out.println("Seller money: " + seller.getMoney());
    System.out.println("Buyer money: " + buyer.getMoney());
}

```

Рисунок 6 – Исходный код функции main

```
task-3 > java Task3
Enter seller information
Enter first name: Ivan
Enter last name: Ivanov
Enter money: 100
Enter buyer information
Enter first name: Petr
Enter last name: Petrov
Enter money: 1000
Enter number of product types: 2
Enter name: Apple
Enter price: 10
Enter count of the product: 10
Enter name: Orange
Enter price: 20
Enter count of the product: 5
Seller money: 300.0
Buyer money: 800.0
```

Рисунок 7 – Пример работы программы

Задание 4. Проектирование иерархии классов организации

В данном задании необходимо спроектировать иерархию классов организации.

В ходе выполнения данного задания были спроектированы следующие классы:

- Employee — общий класс работника, содержит фамилию, имя и размер заработной платы;
- Engineer — класс, представляющий инженера и наследуемый от класса Employee, также содержит квалификацию инженера;
- Manager — класс, представляющий менеджера и наследуемый от класса Employee, также содержит список подчиненных сотрудников;
- Secretary — класс, представляющий секретаря и наследуемый от класса Employee, также содержит информацию о начальнике;

— `Directory` — класс, представляющий директора и наследуемый от класса `Employee`.

Все эти классы реализуют или переопределяют метод `display`, который возвращает информацию о сотруднике в виде строки. Вызовы этого метода являются полиморфными.

На рисунке 8 приведен код функции `main`, с помощью которого была протестирована работа программы. В ней создается экземпляр каждого класса, а потом вызывается функция `display`. На рисунке 9 приведен пример работы программы.

```

public static void main(String[] args) {
    Employee employee = new Employee("Ivan", "Ivanov", 10);
    Engineer engineer = new Engineer("Petr", "Petrov", 20, 1);

    List<Employee> subordinates = new ArrayList<>();
    subordinates.add(employee);
    subordinates.add(engineer);

    Manager manager = new Manager("Svetlana", "Tamarova", 30, subordinates);
    Director director = new Director("Bogdan", "Ibragimov", 100);
    Secretary secretary = new Secretary("Elena", "Skorohodova", 20, director);

    System.out.println(employee.display());
    System.out.println();

    System.out.println(engineer.display());
    System.out.println();

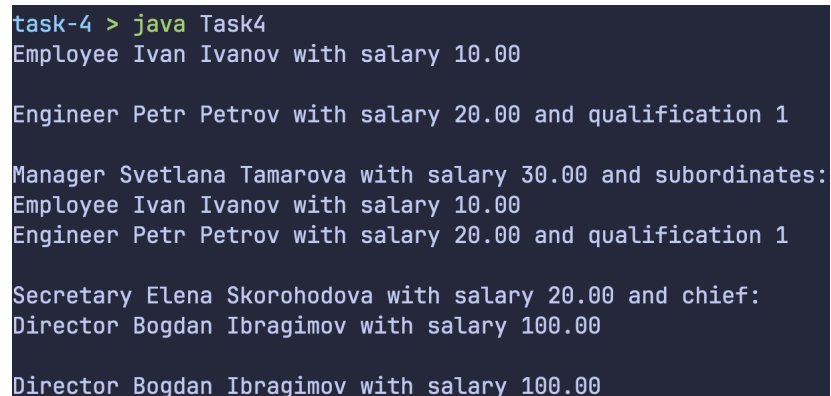
    System.out.println(manager.display());
    System.out.println();

    System.out.println(secretary.display());
    System.out.println();

    System.out.println(director.display());
    System.out.println();
}

```

Рисунок 8 – Исходный код функции main



```

task-4 > java Task4
Employee Ivan Ivanov with salary 10.00

Engineer Petr Petrov with salary 20.00 and qualification 1

Manager Svetlana Tamarova with salary 30.00 and subordinates:
Employee Ivan Ivanov with salary 10.00
Engineer Petr Petrov with salary 20.00 and qualification 1

Secretary Elena Skorohodova with salary 20.00 and chief:
Director Bogdan Ibragimov with salary 100.00

Director Bogdan Ibragimov with salary 100.00

```

Рисунок 9 – Пример работы программы

Заключение

В ходе выполнения лабораторной работы были созданы приложения, реализующие основные управляющие конструкции. Также была спроектирована иерархия классов и принципы взаимодействия между ними. В добавок ко всему, было реализовано наследование и использован полиморфизм.

Цель, поставленная в начале работы, достигнута, задачи выполнены.