**Министерством науки и высшего образования РФ российский государственный университет нефти и газа (национальный исследовательский университет) имени И. М. Губкина**

Кафедра управления безопасностью сложных систем

Отчет по лабораторной работе №4

Дисциплины ***Языки программирования***

Использование оператора ветвления SWITCH

**Выполнила:**

студентка группы КА-20-03

Батова Полина Александровна \_\_\_\_\_\_\_\_ (подпись)

**Проверил:**

Д.т.н. профессор

Корнеев Николай Владимирович \_\_\_\_\_\_\_\_ (подпись)

**Лабораторная работа №4. Использование оператора ветвления SWITCH.**

**Цель работы:** получить навыки использования оператора ветвления SWITCH при разработке программных решений на языке Java с использованием соответственно библиотеки Java Swing.

**Общее задание**

*Задание:* в качестве примера использования оператора SWITCH разрабатывается программа, которая переводит вес из фунтов в килограммы, учитывая, что в разных странах фунт «весит» по-разному. В диалоговом окне программы для выбора страны используется список.

Решение производится с помощью формулы: вес в килограммах = вес в фунтах \* коэффициент.

Для реализации поставленной задачи мы используем тип переменных double, объявляя коэффициент веса фунта (coef). Список стран отображается с помощью jList1. С помощью поля ввода (jTextField1) считываются входные данные, а именно количество фунтов. При нажатии кнопки (jButton1) начинается проверка, с коэффициентом какой страны программа будет работать далее, и подсчет веса в килограммах по приведенной выше формуле. Если выбранная страна - Россия, Англия, Австрия или Италии, то коэффициент (coef) принимает определенное, заранее заданное значение, в противном случае коэффициент устанавливается по умолчанию равным 0.5. Полученный результат выводится в окошко – message Box (jOptionPane1).

Блок-схема:

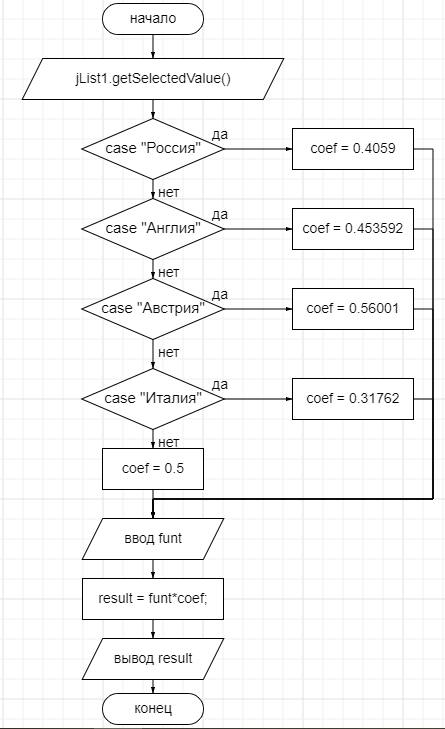


Рисунок 1 – блок-схема общего задания.

Эта простейшая задача позволяет получить базовые навыки работы с оператором ветвления SWITCH.

Результат:

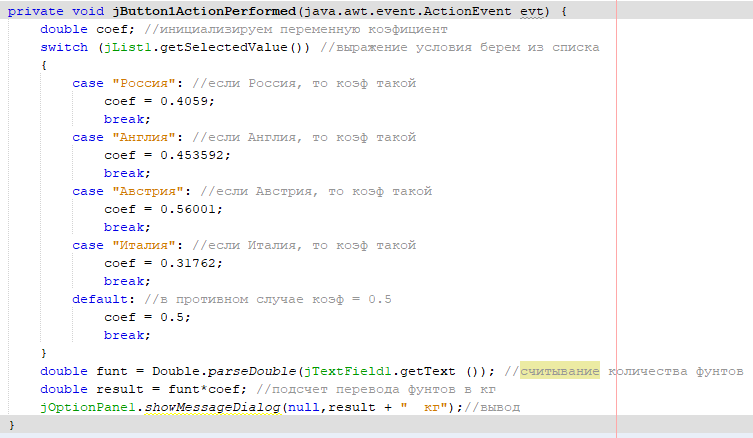


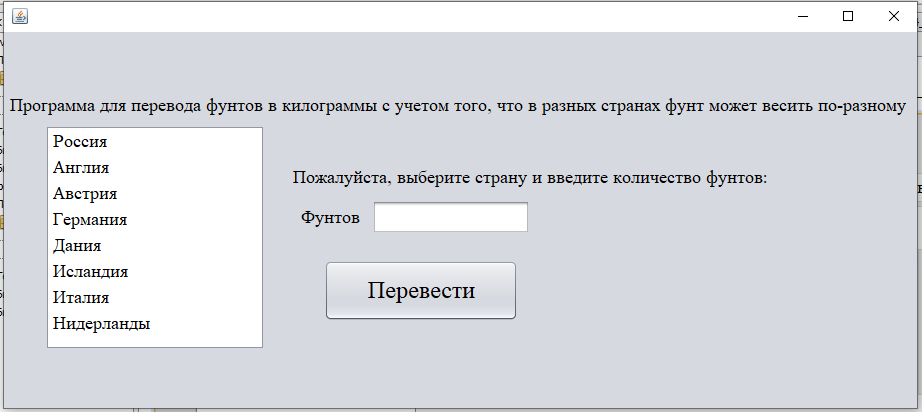
Рисунок 2 – код общего задания.

Рисунок 3 – запуск приложения (общее задание).

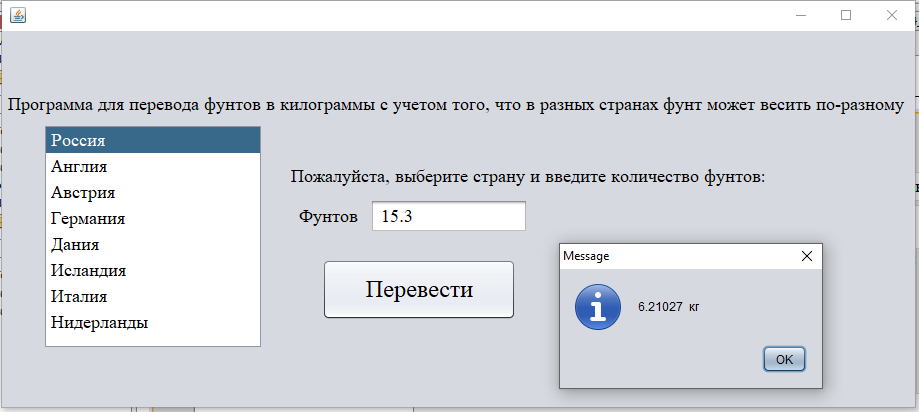


Рисунок 4 – пример работы программы общего задания (1).

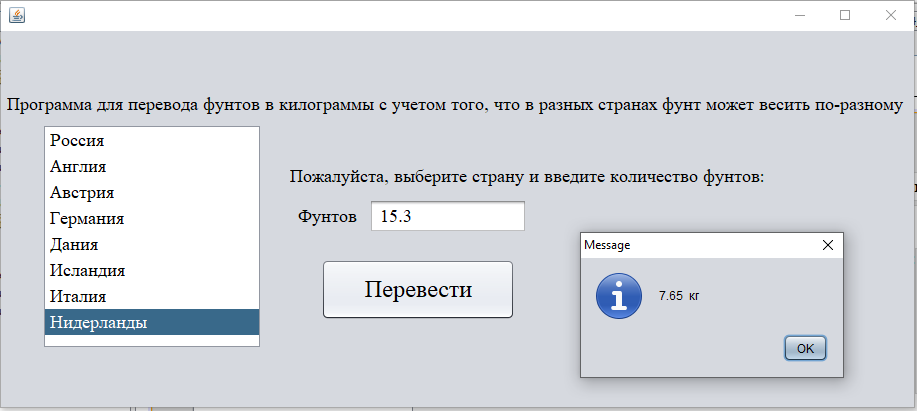


Рисунок 5 – пример работы программы общего задания (2).

**Индивидуальное задание**

*Формулировка задания:* разработать программу, позволяющую узнать, к какому подмножеству дестабилизирующих факторов (ДФ) относится выбранный пользователем ДФ. Перечень ДФ представлен с помощью списка.

Для реализации поставленной задачи создадим список (jList1), включающий в себя 8 ДФ, и создадим переменную nomer, которая будет обозначать номер подмножества ДФ. При нажатии кнопки jButton1 и выборе ДФ из списка программа проверяет, какому значению case соответствует данный ДФ, и в зависимости от этого присваивает переменной nomer значение от 1 до 3. Для того, чтобы не писать восемь раз команду присваивания по отношению к nomer, воспользуемся особенностью оператора перехода break. Их отсутствие создает одинаковую реакцию программы на подряд идущие разные значения case вплоть до ближайшего встреченного break. Таким образом, можно сократить количество команд присваивания до трёх.

Для удобства на панель приложения добавим краткую информацию о подмножествах ДФ из монографии д-р. техн. наук, проф. кафедры управления безопасностью сложных систем Н.В. Корнеева.

Для более легкого восприятия блок-схемы, дадим ДФ порядковые номера по соответствию с выше указанной монографией:

9. Неумышленное заражение вирусом

10. Неосторожные действия, приводящие к разглашению конфиденциальной информации или делающие ее общедоступной

13. Игнорирование организационных ограничений (установленных правил) в системе

17. Ввод ошибочных данных

8. Нелегальное внедрение и использование неучтенных программ с последующим необоснованным расходованием ресурсов

18. Неумышленное повреждение каналов связи

24. Вербовка персонала или отдельных пользователей, имеющих определенные полномочия

16. Пересылка данных по ошибочному адресу абонента (устройства)

Блок-схема:

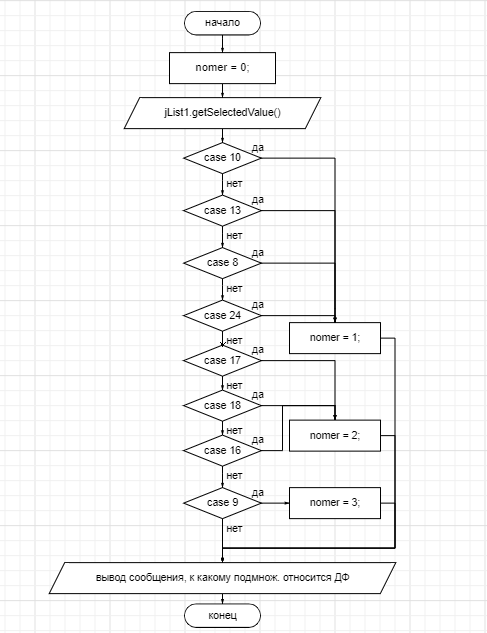


Рисунок 6 – блок-схема индивидуального задания.

Результат:

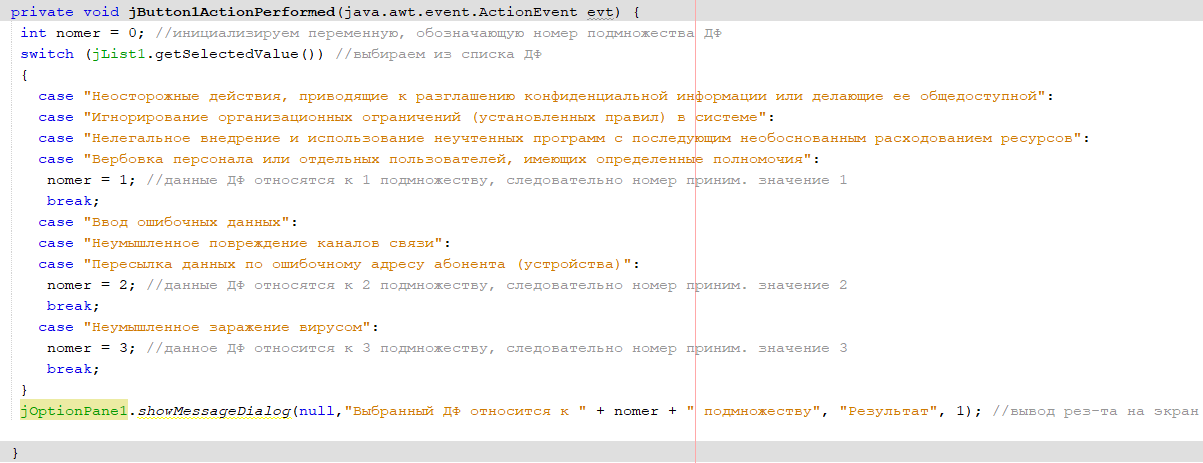


Рисунок 7 – код индивидуального задания.

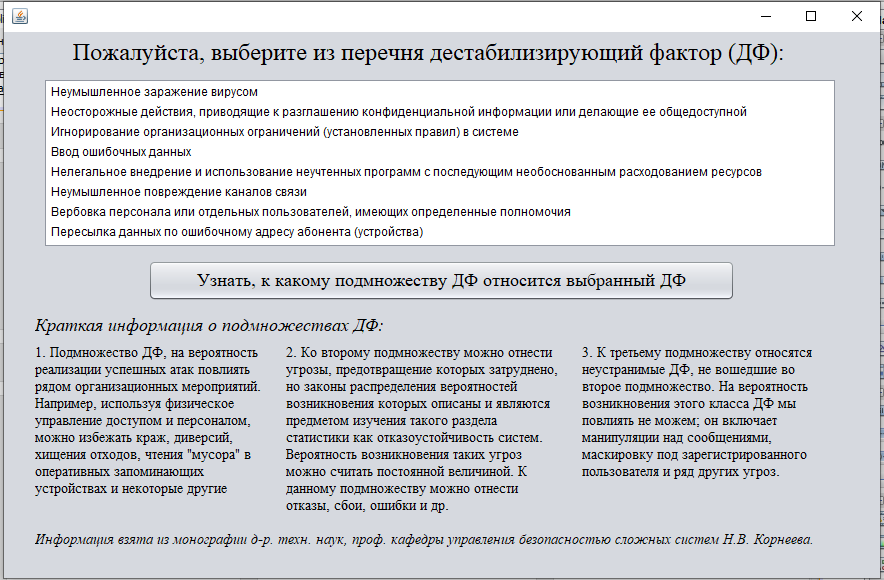


Рисунок 8 – запуск приложения (индивидуальное задание).

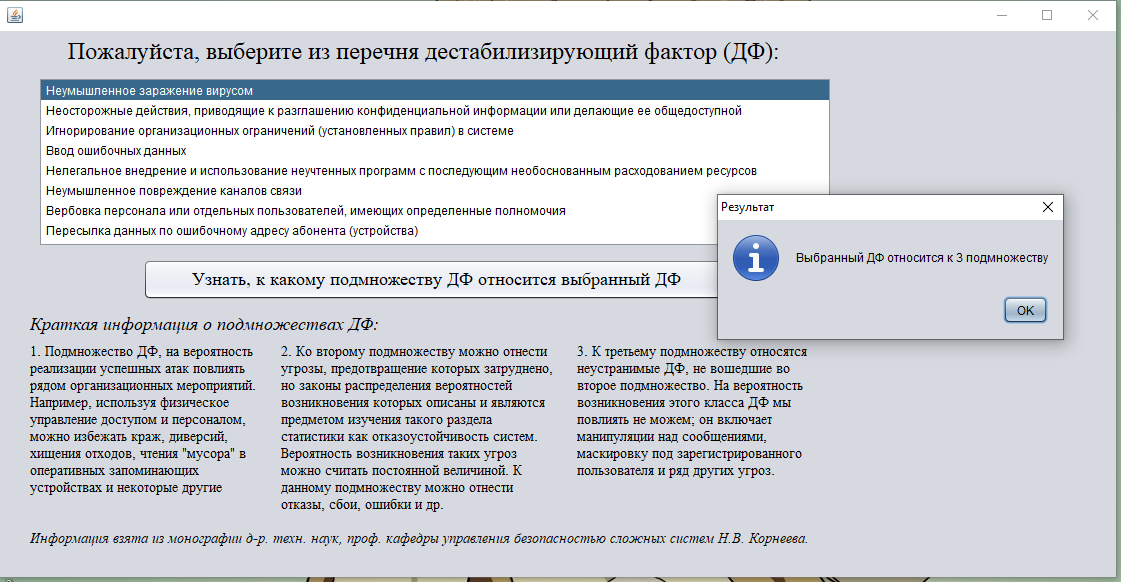


Рисунок 9 – Пример работы программы индивид. задания (1).

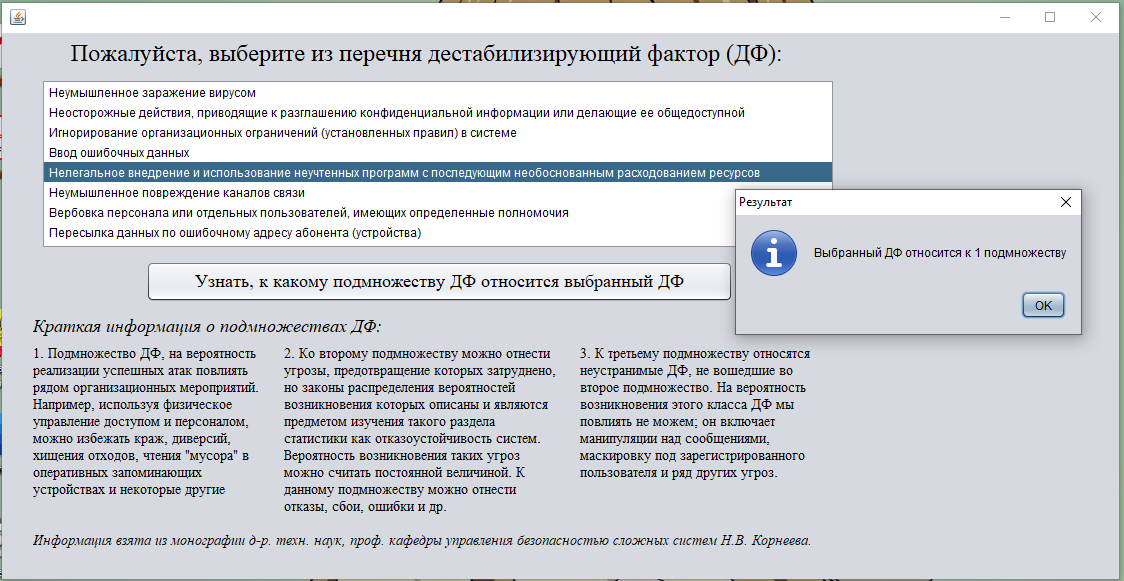


Рисунок 10 – Пример работы программы индивид. задания (2).

**Вывод**

В ходе решения поставленных задач мной были получены навыки использования оператора ветвления SWITCH при разработке программных решений на языке Java с использованием соответственно библиотеки Java Swing.