МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Бариев Эмин Юсуфович

Институт информационных технологий и управления в технических системах

курс 4 группа ИС/б-16-2

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №3

По дисциплине: «Рефакторинг программного обеспечения»

По теме: «Рефакторинг программного кода. Упрощение условных выражений»

|  |  |  |
| --- | --- | --- |
| Отметка о зачете | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_ |
|  |  | (дата) |
|  | Руководитель практикума |  |
| ст. преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Строганов В.А. |
| (должность) | (подпись) | (инициалы, фамилия) |

Севастополь 2019

**1 Цель работы**

Исследовать эффективность рефакторинга программного кода путем упрощения условных выражений. Получить практические навыки применения приемов рефакторинга объектно-ориентированных программ.

**2 Постановка задачи**

2.1. Выбрать фрагмент программного кода для рефакторинга.

2.2. Выполнить рефакторинг программного кода, применив не менее 5 приемов.

2.3. Составить отчет, содержащий подробное описание каждого модифицированного фрагмента программы и описание использованного метода рефакторинга.

**3 Ход работы**

3.1 Декомпозиция условного оператора (Decompose Conditional)

На рисунке 1 выделен условный оператор, который можно декомпозировать.

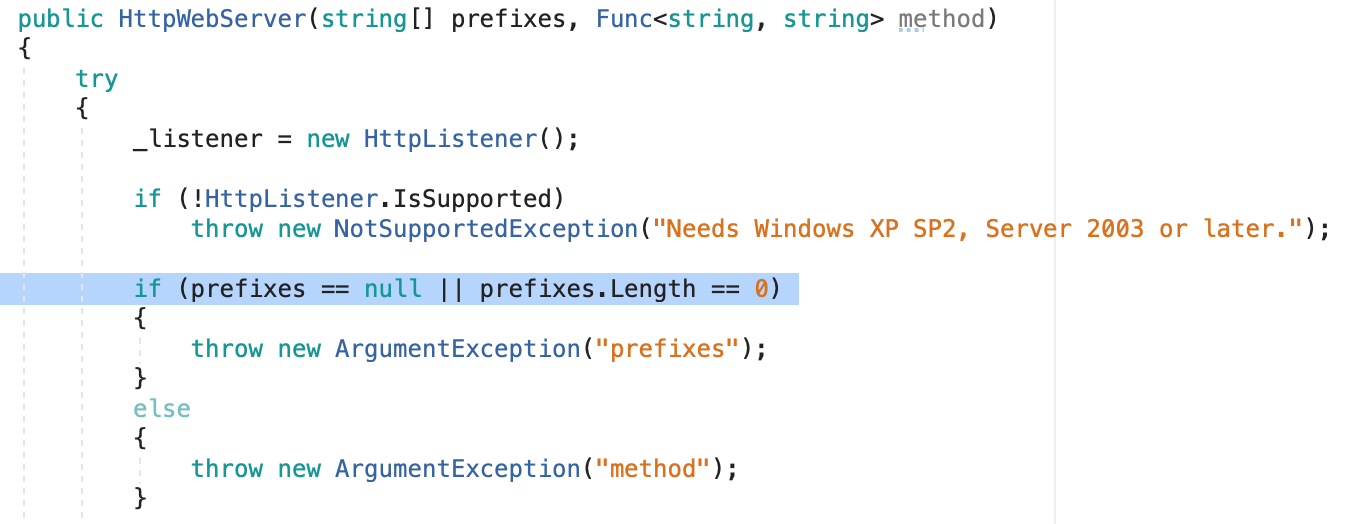


Рисунок 1 – Программный код до декомпозиции условного оператора



Рисунок 2 – Программный код после декомпозиции условного оператора

3.2 Консолидация дублируемых фрагментов в условных выражениях (Consolidate Duplicate Conditional Fragments)

На рисунке 3 показан программный, в котором можно объединить дублируемые фрагменты кода в условных операторах.

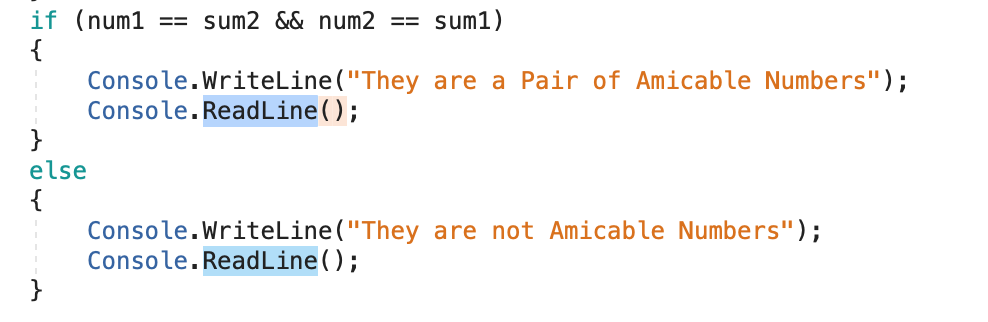


Рисунок 3 – Программный код до объединения дублируемого фрагмента кода

На рисунке 4 показан программы код после объединения дублируемого фрагмента кода

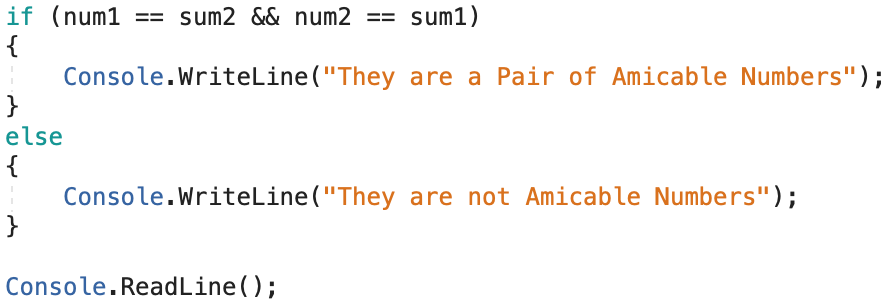


Рисунок 4 – Программный код до объединения дублируемого фрагмента кода

3.3 Удаление управляющего флага (Remove Control Flag)

Имеется переменная flag, действующая как управляющий флаг для ряда выражений. Выполним замену места, где управляющему флагу задается значение, приводящее к выходу из цикла, на оператор break.

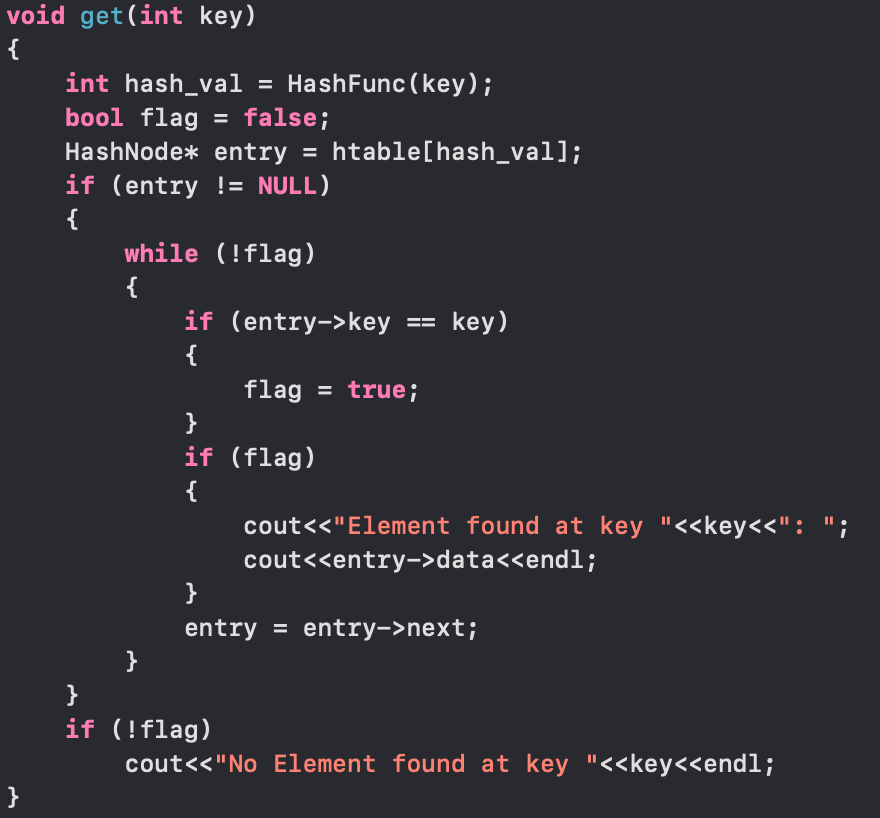


Рисунок 5 – Программный код до удаления управляющего флага

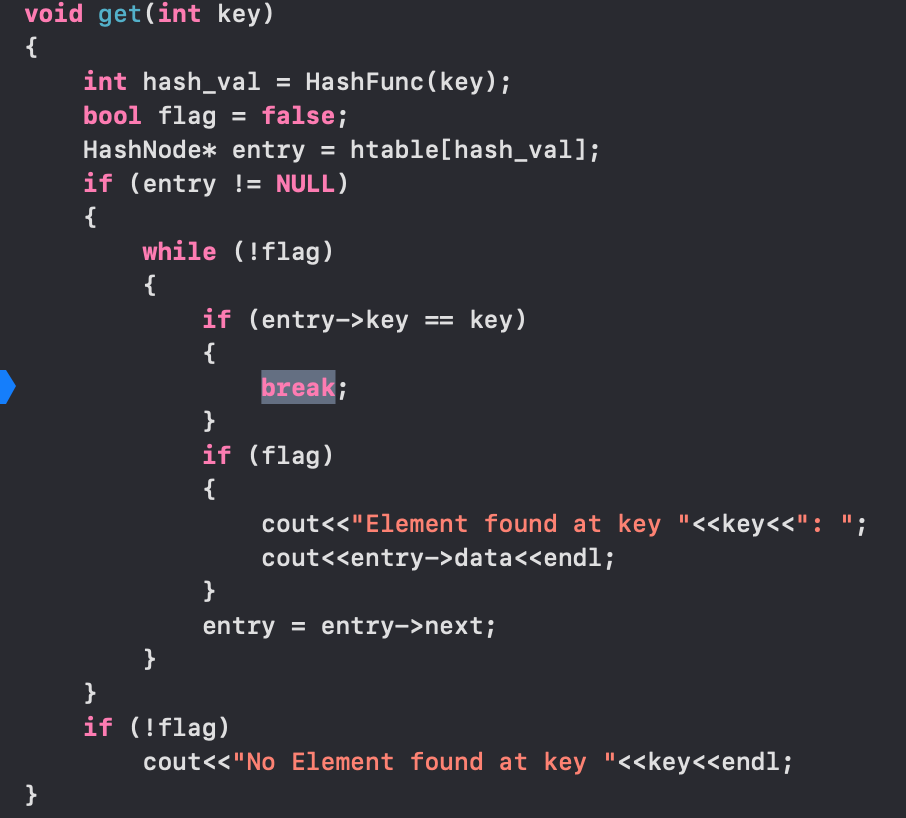
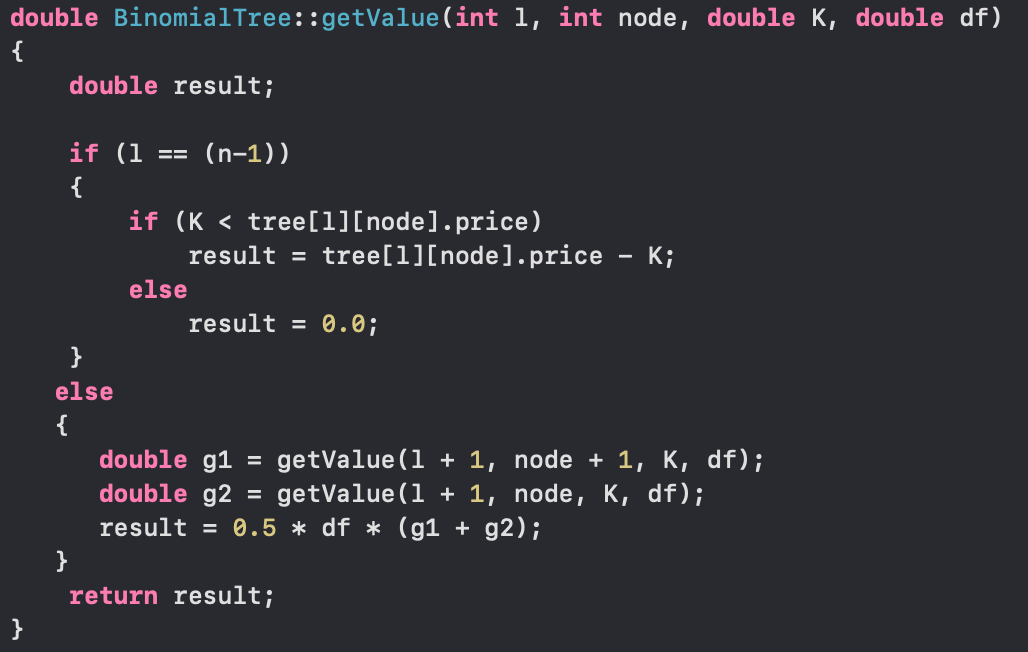
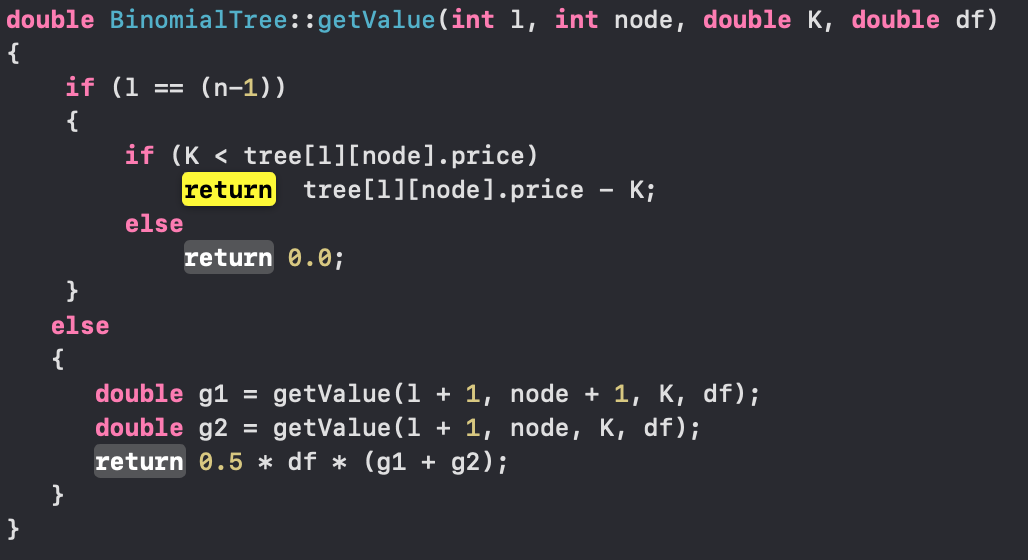


Рисунок 6 – Программный код после удаления управляющего флага

4. Замена вложенных условных операторов граничным оператором (Replace Nested Conditional with Guard Clauses)





**Выводы**

В ходе выполнения лабораторной работы была исследована эффективность рефакторинга программного кода путем упрощения условных выражений. Так же были получены практические навыки применения приемов рефакторинга объектно-ориентированных программ. При выполнении данной лабораторной работы были применены следующие техники рефакторинга:

* Декомпозиция условного оператора (Decompose Conditional)
* Консолидация дублируемых фрагментов в условных выражениях (Consolidate Duplicate Conditional Fragments)
* Удаление управляющего флага (Remove Control Flag)
* Замена вложенных условных операторов граничным оператором (Replace Nested Conditional with Guard Clauses)