

Лабораторная работа №24

Рефакторинг кода

1 Цель работы

1.1 Изучить техники рефакторинга программного кода.

2 Литература

2.1 <https://zoomcode.ru/>

2.2 <https://abap-blog.ru/osnovy-abap/refactoring/>

2.3 <https://refactoring.guru/ru/refactoring/techniques>

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание лабораторной работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Извлечение переменной

Извлечение переменной помогает упростить выражение, которое сложно для восприятия. Например, условие имеет много сравнений и непонятно с первого взгляда.

5.1.1 Создать консольное приложение. Добавить в приложение закрытые поля, хранящие логин и пароль пользователя и метод IsCorrectUserData.

5.1.2 Реализовать в основной программе ввод следующих данных:

- логина,
- пароля,
- подтверждения пароля,

После ввода проверять в методе IsCorrectUserData введенные данные (передаются через параметры) на корректность (логин и пароль не пустые строки, пароль и подтверждение совпадают). Все проверки выполнить в одном условии в разделе if.

Если метод вернет истину, сообщить в методе Main об успешной регистрации и присвоить корректные значения полям класса, иначе — сообщить о том, что не удалось зарегистрироваться.

5.1.3 Для упрощения условия в методе IsCorrectUserData извлечь из него переменные isLoginCorrect, isPasswordCorrect, isConfirmCorrect, в которые записать результат каждого сравнения, и использовать эти переменные в условии в if.

5.2 Извлечение класса

Извлечение класса – техника рефакторинга, которая предполагает перенос группы связанных методов и полей из одного класса в новый, отдельный класс. Это может облегчить понимание и сопровождение кода, разбив его на более мелкие, более целенаправленные части.

Создать отдельный класс для хранения данных пользователя (логина и пароля). Объявить в нем метод IsCorrectUserData с теми же параметрами и логикой поведения, что в приложении из п.5.1. Использовать объект этого класса в приложении из п.5.1.

5.3 Замена сложного алгоритма

Вы могли найти алгоритм, который куда проще и эффективнее текущего. В этом случае надо заменить старый алгоритм новым.

Изменить алгоритм вычисления количества дней в месяце, добавив в метод GetDaysCount массив со списком значений для каждого месяца.

Возвращать по номеру месяца значение элемента массива с нужным индексом. Если год високосный, к февралю прибавлять день.

```
private int GetDaysCount(int month, int year)
{
    switch (month)
    {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            return 31;
        case 4: case 6: case 9: case 11:
            return 30;
        case 2:
            return (year % 400 == 0 || year % 100 != 0 && year % 4 == 0) ? 29 : 28;
        default:
            return -1;
    }
}
```

5.4 Замена кода ошибки исключением

Возвращение кодов ошибок — устаревшая практика процедурного программирования. В современном программировании для обработки ошибок используются исключения. При возникновении проблемы вы «выбрасываете» исключение и оно впоследствии «ловится» одним из обработчиков исключений.

Предусмотреть в методе, вычисляющем количество дней в месяце, генерацию исключения вместо возврата -1.

5.5 Замена условного оператора полиморфизмом

<https://zoomcode.ru/polymorphism-method.html>

5.5.1 Создать метод с параметром типФигуры, возвращающим площадь фигуры в зависимости от значения поля типФигуры (прямоугольник, круг и кольцо). Требуемые стороны/радиусы запрашивать у пользователя в методе в зависимости от выбранной фигуры.

5.5.2 Заменить условный оператор полиморфизмом:

- создать абстрактный класс Фигура с методом получения площади
- создать дочерние классы Круг, Прямоугольник, Кольцо
- в методе из п.5.5.1 реализовать вычисление, создав объект требуемого типа и вызвав метод вычисления площади.

5.6 Извлечение метода

<https://zoomcode.ru/extract-method.html>

Добавить в класс Фигура метод вычисления площади круга. Заменить выражение для вычисления площади круга на вызов метода и использовать этот метод для избавления от дублирования кода в классах Круг и Кольцо из п.5.5.2.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать приложение C#. Выполнить все задания из п.5 в проекте LabWork24. При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать.

6.2 При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода. Все наименования должны быть на английском языке в PascalCase/camelCase.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «рефакторинг»?

8.2 Какие группы техник рефакторинга существуют?

8.3 Как выполнить рефакторинг в Visual Studio?,