

Лабораторная работа №21

Оптимизация кода

1 Цель работы

1.1 Изучить методы оптимизации программного кода.

2 Литература

2.1 Фленов, М. Е. Библия С#. – 3 изд. – Санкт-Петербург: БХВ-Петербург, 2016.
– URL: <https://ibooks.ru/bookshelf/353561/reading>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – п.8.9.

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 Удаление лишнего кода

Выполнить оптимизацию программного кода, убрав лишние переменные и неиспользуемый код:

```
int Sum(int a, int b)
{
    int number1 = a, number2 = b;
    int result = 0;
    result = number1 + number2;
    return result;
}
```

5.2 Использование формул для оптимизации

Выполнить оптимизацию программного кода, упростив следующие выражения и заменив вызов стандартных математических функций:

```
// вычисление площади кольца
double s = Math.Abs(Math.PI*Math.Pow(r1, 2) - Math.PI*Math.Pow(r2, 2));
```

```
// сумма чисел от 1 до n (заменить на сумму ряда натуральных чисел)
```

```
int i, sum = 0;
for (i = 1; i <= n; i++)
    sum += i;
```

5.3 Использование сдвигов на степени двойки

Написать метод, переводящий количество байт в строку с размером и единицей измерения Б / КБ / МБ / ГБ (выбрать максимально подходящий размер, как в проводнике)

```
string GetSize(int bytes)
{
    return $"{bytes} Б";
}
```

5.4 Оптимизация условий

Выполнить оптимизацию программного кода, уменьшив количество проверок условия и оптимизировав порядок условий:

```
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;
    }
}
```

5.5 Сохранение промежуточных результатов

Написать и протестировать рекурсивную функцию быстрого вычисления x^n , где n неотрицательное целое, используя возведение в квадрат.

Для ускорения работы рекурсия должна вызываться в ветке алгоритма не более одного раза.

Пример использования алгоритма быстрого возведения в степень (вместо 15 операций умножения будет 6 операций умножения):

$$a^{15} = a*(a^7)^2 = a*(a*(a^3)^2)^2 = a*(a*(a*(a^2)^2)^2)^2$$

Для некорректных данных возвращать -1.

Стандартный метод возведения в степень не использовать.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C#. Выполнить все задания из п.5 в одном решении LabWork21.

При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать.

При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

6.2 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 «оптимизация программного кода»?

8.2 Какова цель оптимизации программного кода?

8.3 Какие методы оптимизации программного кода применяются?