Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное   
образовательное учреждение высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

ИНСТИТУТ ЭЛЕКТРОННЫХ И ИНФОРМАЦИОННЫХ СИСТЕМ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Лабораторная Работа №1. Рекурсия

Отчет по лабораторной работе №1 по учебной дисциплине «Алгоритмические языки и программирование»

по специальности 09.03.01 Информатика и вычислительная техника

|  |  |
| --- | --- |
|  | Руководитель  / И. Ю. Кулаков  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |
|  | Студент группы 3092  / Н. Д. Хорев  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

1. **Постановка задачи**

Разработать программу с интерфейсом для работы с рекурсиями.

* 1. **Наименование задачи** – Лабораторная работа №1. Рекурсия.
  2. **Словесное описание:**

Разработать программу с графическим интерфейсом для работы с рекурсиями:

1. По формулам 9 и 16:

9: F(n) = 1, при n = 1;

F(n) = n + 2 \* F(n - 1) , если n чётно;

F(n) = 1 + 3 \* F(n - 2), если n > 1 и при этом n нечётно.

16: F(n) = 1, при n = 1;

F(n) = 2, при n = 2;

F(n) = [(8\*n + F(n - 3))/9], если n > 2 и при этом n четно

F(n) = [(4\*n + F(n - 1) + F(n - 2))/7], если n > 2 и при этом n нечетно

1. Дополнительная программа (9 вариант): Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите первое, третье, пятое и т.д. из введенных чисел. Завершающий ноль выводить не надо.  
   В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции.

Программа должна содержать поля ввода, вывода значений, очистки полей, выбора рекурсивной формулы (1 программа), старта программы, информацию о программе.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Тип данных | Диапазоны представления | Простая/  структура | Вход/  выход/  Константа |
| Значения, введённые пользователем в поле ввода 1-й программы | N | unsigned int | 1 – 70 (рекурсия 1)  1 – 21187 (рекурсия 2) | Простая | Вход |
| Значения, введённые пользователем в поле ввода 2-й программы | num | int | 0-999999 | Простая | Вход |

**Внешние спецификации данных**

**1.3. Внешние спецификации функций программы**

Программа должна обрабатывать введённые значения и исключать ненужные (нечисловые значения, числа через пробел), ограничивать ввод количества символов во избежания ошибки, обеспечивать работу кнопок запуска, очистки полей, вывода окна с информацией о программе

Описываются функциональные требования к разрабатываемой программе и ее функциональные возможности.

**1.4. Математическая формулировка задачи**

Для вводимых значений используется тип данных unsigned int, так как отрицательные значения приводят к ошибке работы программы. Математической моделью являются формулы ниже, которые должны быть представлены в виде рекурсивных функций(1-2 в первой программе, 3 в дополнительной):  
1) F(n) = 1, при n = 1;

F(n) = n + 2 \* F(n - 1) , если n чётно;

F(n) = 1 + 3 \* F(n - 2), если n > 1 и при этом n нечётно.

2) F(n) = 1, при n = 1;

F(n) = 2, при n = 2;

F(n) = [(8\*n + F(n - 3))/9], если n > 2 и при этом n четно

F(n) = [(4\*n + F(n - 1) + F(n - 2))/7], если n > 2 и при этом n нечетно  
3) Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите первое, третье, пятое и т.д. из введенных чисел.

Первая программа:

* Ограничение для 1-й рекурсивной функции от 1 до 70 выбран для того, чтобы обеспечить достаточную скорость работы программы, поскольку значения выше 70 значительно замедляют работу программы вплоть до полной её остановки и закрытия.
* Ограничение для 2-й рекурсивной функции от 1 до 21187 обусловлена тем, что, превышая это значения программа принудительно закрывается.

Вторая программа:

* Ограничение для длины вводимых символов в 6 обусловлена тем, чтобы выводимые числа в поля вывода не заходили за рамки и для того, чтобы не выдавало ошибку «необработанное исключение».
* Ограничение для количества строк в 6 сделано по той же причине, что и для ввода символов в поле ввода.

**1.5. Спецификации интерфейса**

Для создания приложения была использована библиотека Windows Forms, которая позволила быстро создать графический интерфейс приложения и так же быстро написать поведения для расставленных элементов.

Реализован контролируемый ввод значений, который не даёт вводить некорректные значения для данной программы, возможность выбора рекурсивных функций, возможность очистки полей ввода и вывода.

Наличие кнопки «Информация» позволяет ознакомиться с особенностями программы и узнать ограничения на вводимые значения.

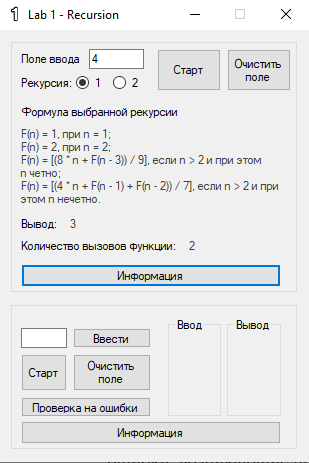


Рисунок Интерфейс приложения

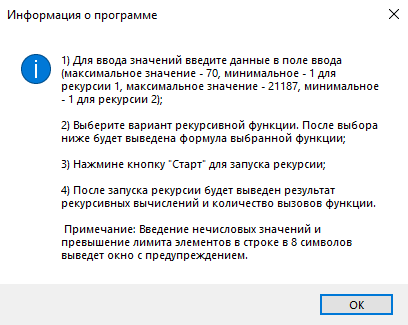


Рисунок Информация о программе 1

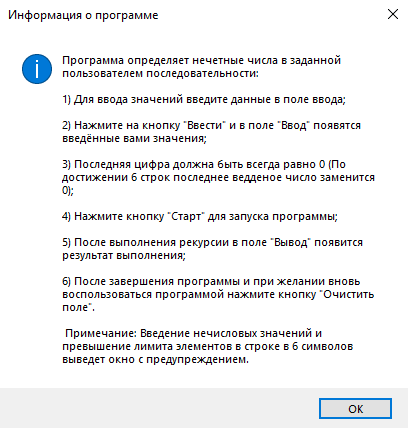


Рисунок Информация о программе 2

**Внешние данные тестирования**

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T01 |
| Заголовок | | | Проверка выхода из приложения |
| Предусловие | | | Приложение не запущено |
| Ожидаемый результат | | | Успешный выход из приложения |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Запуск 1lab.exe | Открытие главного окна приложения | Pass |
| 2 | Нажатие на кнопку «Закрыть» | Выход из приложения | Pass |
| Результат | | | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T02 |
| Заголовок | | | Проверка работы кнопки «Старт» |
| Предусловие | | | Приложение  запущено |
| Ожидаемый результат | | | Успешный вывод текста с формулой |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Выбрать одну из кнопок с рекурсией | Вывод текста с формулой рекурсии 1 | Pass |
| 2 | Ввести значение 4 в поле ввода | Ввод корректных данных | Pass |
| 3 | Нажатие на кнопку «Старт» | Вывод результата 3 и количества вызовов рекурсии 2 | Pass |
| Результат | | | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | T03 |
| Заголовок | | | Проверка работы кнопки «Очистить поле» |
| Предусловие | | | Приложение  запущено |
| Ожидаемый результат | | | Успешная очистка полей |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Выбрать одну из кнопок с рекурсией | Вывод текста с формулой рекурсии | Pass |
| 2 | Ввести значение в поле ввода | Ввод корректных данных | Pass |
| 3 | Нажатие на кнопку «Старт» | Вывод результата и количества вызовов рекурсии | Pass |
| 4 | Нажатие на кнопку «Очистить поле» | Очистка поля ввода, вывода результата, количества рекурсивных функций, формулы рекурсии | Pass |
| Результат | | | Pass |

| Идентификатор | | | T04 |
| --- | --- | --- | --- |
| Заголовок | | | Проверка контроля ввода |
| Предусловие | | | Приложение запущено |
| Ожидаемый результат | | | Выведена ошибка, возвращение к стандартным значениям |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Ввести некорректное значение | Введено некорректное значение | Pass |
| 2 | Выбрать вариант рекурсии | Выведена формула рекурсии | Pass |
| 3 | Нажатие на кнопку «Старт» | Вывод окна с ошибкой | Pass |
| 4 | Нажатие на кнопку «ОК» в всплывающем окне | Закрытие окна с ошибкой и очистка поля ввода с сброшенным выбором рекурсии | Pass |
| Результат | | | Pass |

| Идентификатор | | | T05 |
| --- | --- | --- | --- |
| Заголовок | | | Работоспособность второй программы |
| Предусловие | | | Приложение запущено |
| Ожидаемый результат | | | Корректное выполнение программы (см. п 1.2. - 2)) |
| Шаг | Действие | Предполагаемый результат | Pass/Fail |
| 1 | Ввести числа в поле ввода и нажать «Ввести» | Введены значения, ограниченные условием, список заканчивается 0 | Pass |
| 2 | Нажать «Старт» | В поле вывода выведены числа из списка через 1 | Pass |
| Результат | | | Pass |

Все возможные статусы программы приведены в приложении Б.

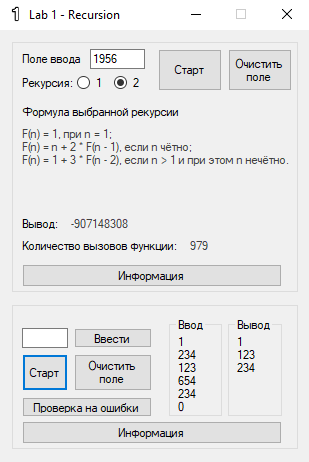
Если используем модуль ctest (рекомендую его), то вставляем код тестирования

с комментарием и объяснением что проверяет это тестирование.

Помните, что тестирование должно быть точечным, к примеру, проверять на границы диапазона или ввод пользователя.

**1.7. Пример работающей программы** (руководство пользователя)

Показывает скриншоты программы, если реализация может быть непонятной, то пишем руководство.



1. **Проектирование задачи** (заполняется от лица программиста)
   1. **Наименование программы**

1lab.exe

* 1. **Уточненное словесное описание задачи**

Разработать программу с графическим интерфейсом для работы с рекурсиями с заданными формулами. Программа должна содержать поля ввода

1. Дополнительная программа (9 вариант): Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите первое, третье, пятое и т.д. из введенных чисел. Завершающий ноль выводить не надо.  
   В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции.

Программа должна содержать поля ввода, вывода значений, очистки полей, выбора рекурсивной формулы (1 программа), старта программы, информацию о программе.

Если задача простая, то часто дублирует пункт 2.2, но там пишется простым языком от лица пользователя, а тут от лица программиста. Можно взять ТЗ из файла с заданием

* 1. **Выбор метода решения поставленной задачи**

Описываем методику вашего решения, обычно описываются уже существующие методы.

* 1. **Уточненные глобальные данные программы и пользовательские типы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект программы | Имя этого объекта в программе | Характеристики | | | Как используется в программе |
| Тип данных | Диапазоны представления | Простой/структура/ константа |
| MyForm.h | n | unsigned short int | 0-65535 | Простой | Параметр рекурсивных функций |
| MyForm.h | N | unsigned int | 0-4294967295 | Простой | Передача введённого числа в параметр рекурсивной функции |
| MyForm.h | count | unsigned int | 0-4294967295 | Простой | Значение вызовов рекурсивной функции |
| MyForm.h | choice\_check | bool | 0-1 | Простой | Обеспечивает выбор рекурсивной функции |

* 1. **Декомпозиция функций**

Описывается каждая функция и процедура в программе

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Назначение | Имя | Параметры | | Процедура/ функция | Тестируется/  не тестируется |
| Вход:  тип | Выход: тип |
| Зачем она |  | Данные на входе | Данные на выходе | Функция возвращает значения  Процедура ничего не возвращает | Можно ли ее тестировать, укажите номер теста или пункта из 2.6 |

* 1. **Блок-схема**

Блок-схема функций и процедур делается на основе ГОСТ 19.701-90, более подробно как это делать и частые ошибки разберем на занятиях. Файл прикреплен на Яндекс диске: https://disk.yandex.ru/d/SAM6E2OlpdCCVg



* 1. **Алгоритмизация**

Приводятся алгоритмы процедур/функций и программы

**Требования к исходному тексту программы**:

Программа должна быть разработана в стиле структурного программирования. Структурные уровни должны отделяться отступами. Текст программы должен содержать комментарии:

1. шапка (верхняя часть файла исходного текста):
   * название программы;
   * назначение программы (или данного файла в составе программы);
   * автор (фамилия, имя, группа, курс);
   * дата создания(если делали несколько дней, то интервал времени);
2. комментирование глобальных переменных и констант:
   * назначение переменных;
3. комментирование функций:
   * назначение функции;
   * входные параметры;
   * выходные параметры;
4. комментирование сложных для понимания участков программы:
   * назначение участка;
   * описание механизма работы.