**Лабораторная работа №4 Черкасов Александр Андреевич 1 вариант А-08-19**

**Содержание**

[**Задание (4.1)** 1](#_Toc53335864)

[**Задание (4.2)** 1](#_Toc53335865)

[**Задание (4.3)** 1](#_Toc53335866)

[**1. Постановка задачи** 1](#_Toc53335867)

[**2. Разработка программы** 1](#_Toc53335868)

[2.1 Разработка функций и методов классов 1](#_Toc53335869)

[2.2 Разработка интерфейса 2](#_Toc53335870)

[**3. Реализация и тестирование программы** 2](#_Toc53335871)

[3.1 Описание разработанной программы 2](#_Toc53335872)

[3.2 Тестирование программы 2](#_Toc53335873)

[**Вывод** 2](#_Toc53335874)

[**Приложение. Код программы** 2](#_Toc53335875)

# **Задание (4.1)**

Создать класс целых чисел. Определить оператор ++, как функцию-член и -- как дружественную функцию

# **Задание (4.2)**

Создать объект очередь с перегруженными +, +=, добавление элемента в очередь и сложение очередей, -- для извлечения из очереди, - для вычитания очередей.

# **Задание (4.3)**

Создать класс вектор, содержащий ссылку на long, размерность вектора и переменную ошибки. Класс имеет конструкторы по умолчанию, конструктор с одним и двумя параметрами, конструктор копирования и деструктор. Определить оператор +, - , \* , - как дружественные функции, =, +=, -=, \*=,[] - как функции-члены. Определить операторы =, +, - , \*, +=, -=, \*= с числом типа long, операторы ++ и --. Определить функцию печати. Перегрузить операторы вывода и ввода в поток.

# **1. Постановка задачи**

Разработать объектно-ориентрованную программу в соответствии с заданиями 4.1, 4.2 и 4.3

**Функции**:

**Входные данные**:

**Выходные данные**:

**Ограничения**:

**Вид приложения** - консольное приложение на языке C++.

**Среда разработки** – CLion

# 

# **2. Разработка программы**

2.1 Разработка функций и методов классов

|  |  |
| --- | --- |
| **tInt** | |
| **Конструкторы** | **Описание** |
| tInt(); |  |
| explicit tInt(const int&); |  |
| **Поля** |  |
| int\* value; |  |
| **Деструктор** |  |
| ~tInt() |  |
| **Методы** |  |
| tInt operator++(); |  |
| tInt operator++(int); |  |
| **Дружественные методы** |  |
| friend tInt& operator--(tInt&); |  |
| friend tInt& operator--(tInt&,int); |  |
| friend std::ostream& operator<<(std::ostream&,const tInt&); |  |

|  |  |
| --- | --- |
| **queue** | |
| **Конструкторы** | **Описание** |
| queue() |  |
| **Деструктор** |  |
| ~queue() |  |
| **Поля** |  |
| std::vector<T> array; |  |
| **Методы** |  |
| void push(const T&); |  |
| T pop(); |  |
| queue& operator+=(const T&); |  |
| queue& operator--(); |  |
| queue operator--(int); |  |
| queue operator+(const queue&); |  |
| queue operator-(const queue&); |  |
| queue operator=(const queue&); |  |
| **Дружественные методы** |  |
| friend std::ostream& operator<<(std::ostream&,const queue&); |  |

|  |  |
| --- | --- |
| **vector** | |
| **Конструкторы** | **Описание** |
| vector(); |  |
| explicit vector(const size\_t&); |  |
| vector(long\*,const size\_t&); |  |
| vector(const vector&); |  |
| **Деструктор** |  |
| ~ vector() |  |
| **Поля** |  |
| long\* array; |  |
| size\_t length; |  |
| size\_t element\_count; |  |
| static const size\_t length\_step = 10; |  |
| static std::invalid\_argument no\_element; |  |
| **Методы** |  |
| void push\_back(const long&); |  |
| long pop(); |  |
| vector& operator=(const long&); |  |
| vector& operator+=(const long&); |  |
| vector& operator-=(const long&); |  |
| long& operator[](const size\_t&); |  |
| vector& operator++(); |  |
| vector operator++(int); |  |
| vector& operator--(); |  |
| vector operator--(int); |  |
| vector& operator=(const vector&); |  |
| void reallocate\_memory(const size\_t&); |  |
| static void copy(const long\*,const size\_t&,long\*); |  |
| **Дружественные методы** |  |
| friend vector operator+(const vector&,const long&); |  |
| friend vector operator-(const vector&,const long&); |  |
| friend vector operator\*(const vector&,const long&); |  |
| friend std::ostream& operator<<(std::ostream&,const vector&); |  |
| friend std::istream& operator>>(std::istream&,vector&); |  |

2.2 Разработка интерфейса

# **3. Реализация и тестирование программы**

3.1 Описание разработанной программы

3.2 Тестирование программы

|  |  |  |
| --- | --- | --- |
| **№ теста** | **Смысл теста** | **Результат** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# **Вывод**

# **Приложение. Код программы**