Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Информационных технологий и программирования

Программирование на C++

Работа:

Лабораторная работа №2, Классы.

Вариант 5

Выполнил: Гаджиев С. И.,
M3115

Санкт-Петербург

2023 г.

ТЗ:
Согласно варианту описать указанные классы (варианты распределяются преподавателем лично). Написать программу, использующую описанные классы: инициализация переменных (ввод пользователя), выполнение действий с экземплярами класса (в зависимости от дальнейшего ввода пользователя).

Описания и реализация должны находиться в разных файлах. Доступ к полям класса – только через методы. Внешние функции для работы с данными класса не допускаются. Перегрузка стандартных арифметических операций для класса (только в виде методов класса) – в зависимости от задания и здравого смысла (уместна консультация с преподавателем практики).

Вариант 5:

Очередь строк.

Примечание: Максимальная длина очереди задается при инициализации и не может быть изменена позднее. Каждая строка ограничена по длине 255 символами.

Конструкторы: По умолчанию. Копирования. Максимальная длина. Очередь строк.

Функциональность: Длина очереди. Добавление строки в очередь. Изъятие строки из очереди. Просмотр последнего элемента. Просмотр первого элемента. Вывод на экран.

File “func.h”:

```
#ifndef OP2V2_FUNC_H
#define OP2V2_FUNC_H

class Queue {
    private:
        int maxLength;
        std::vector<std::string> queue;

    public:
        Queue() {
            maxLength = 10;
        }
        Queue(const Queue &other) {
            maxLength = other.maxLength;
            queue = other.queue;
        }
        Queue(int maxSize) {
            this->maxLength = maxSize;
        }

        int length() const;
        void enqueue(std::string str);
        std::string dequeue();
        std::string peek() const;
        void print() const;
};

#endif //OP2V2_FUNC_H
```

Класс Queue имеет три конструктора: конструктор по умолчанию, который создает очередь с максимальной длиной 10; конструктор копирования, который копирует содержимое одной очереди в другую; и конструктор с параметром, который задает максимальную длину очереди.

Методы класса включают:

length() - возвращает длину очереди.

enqueue(std::string str) - добавляет элемент в конец очереди. Если длина строки больше 255 символов, генерируется исключение std::length_error. Если очередь уже имеет максимальную длину, генерируется исключение std::overflow_error.

dequeue() - удаляет элемент из начала очереди и возвращает его значение. Если очередь пуста, генерируется исключение std::underflow_error.

peek() - возвращает значение элемента в начале очереди без его удаления. Если очередь пуста, генерируется исключение std::underflow_error.

print() - выводит все элементы очереди в стандартный вывод.

File “func.cpp”:

```cpp
#include <iostream>
#include <string>
#include <vector>
#include "func.h"

int Queue::length() const {
    return queue.size();
}

void Queue::enqueue(std::string str) {
    if (str.length() > 255) {
        throw std::length_error("STRING IS TOO LONG");
    }
    if (queue.size() >= maxLength) {
        throw std::overflow_error("QUEUE IS FULL");
    }
    queue.push_back(str);
}

std::string Queue::dequeue() {
    if (queue.empty()) {
        throw std::underflow_error("QUEUE IS EMPTY");
    }
    std::string str = queue.front();
    queue.erase(queue.begin());
    return str;
}

std::string Queue::peek() const {
    if (queue.empty()) {
        throw std::underflow_error("QUEUE IS EMPTY");
    }
    return queue.front();
}

void Queue::print() const {
    std::cout << "QUEUE:" << "\n";
```

```
    for (int i = 0; i < queue.size(); i++) {
        std::cout << queue[i] << "\n";
    }
}
```

Содержит определения функций-членов класса Queue, которые были объявлены в заголовочном файле func.h. Эти функции реализуют основные операции, которые можно выполнять с очередью: добавление элемента в конец очереди, удаление элемента из начала очереди, просмотр первого элемента и вывод всех элементов очереди.

File “main.cpp”:

```
#include <iostream>
#include <string>
#include <vector>
#include "func.h"
using namespace std;

int main() {
    cout << "Welcome! Nice to meet you in my console application :)" << "\n";
    cout << "Here I created a queue of strings." << "\n";

    cout << "Please, enter QUEUE size: ";
    int SIZE;
    cin >> SIZE;

    Queue queue(SIZE);
    int choice;
    string input;

    do {
        cout << "\n" << "QUEUE MENU:" << "\n";
        cout << "1. Add item to queue" << "\n";
        cout << "2. Remove item from queue" << "\n";
        cout << "3. View first item in queue" << "\n";
        cout << "4. View length of queue" << "\n";
        cout << "5. Print queue" << "\n";
        cout << "0. Exit" << "\n";

        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter item to add to queue: ";
                cin >> input;
                queue.enqueue(input);
                break;
            case 2:
                cout << "Item removed from queue: " << queue.dequeue() <<
"\n";
                break;
            case 3:
                cout << "First item in queue: " << queue.peek() << "\n";
                break;
            case 4:
                cout << "Queue length: " << queue.length() << "\n";
                break;
            case 5:
                queue.print();
                break;
```

```
            case 0:
                cout << "Goodbye!" << "\n";
                break;
            default:
                cout << "Invalid choice. Please try again." << "\n";
                break;
        }
    } while (choice != 0);

    return 0;
}
```

Тут я не поленился и сделал консольную менюшку.