ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Лабораторная работа

Программа, управляемая событиями

Выполнил студент группы РИС-23-3Б

Мазунин М.А.

Проверила доцент кафедры ИТАС

О. А. Полякова

2024 г.

**1. Постановка задачи**

1. Определить иерархию пользовательских классов (см. лабораторную работу №5). Во главе иерархии должен стоять абстрактный класс с чисто виртуальными методами для ввода и вывода информации об атрибутах объектов.

2. Реализовать конструкторы, деструктор, операцию присваивания, селекторы и модификаторы.

3. Определить класс-группу на основе структуры, указанной в варианте.

4. Для группы реализовать конструкторы, деструктор, методы для добавления и удаления элементов в группу, метод для просмотра группы, перегрузить операцию для получения информации о размере группы.

5. Определить класс Диалог — наследника группы, в котором реализовать методы для обработки событий.

6. Добавить методы для обработки событий группой и объектами пользовательских классов.

7. Написать тестирующую программу.

8. Нарисовать диаграмму классов и диаграмму объектов.

**2. Анализ задачи**

Класс TEvent - Класс, описывающий событие

Класс Object - Абстрактный класс, стоящий во главе иерархии наследования

Класс Print - Класс, описывающий печатное издание. Наследуется от Object

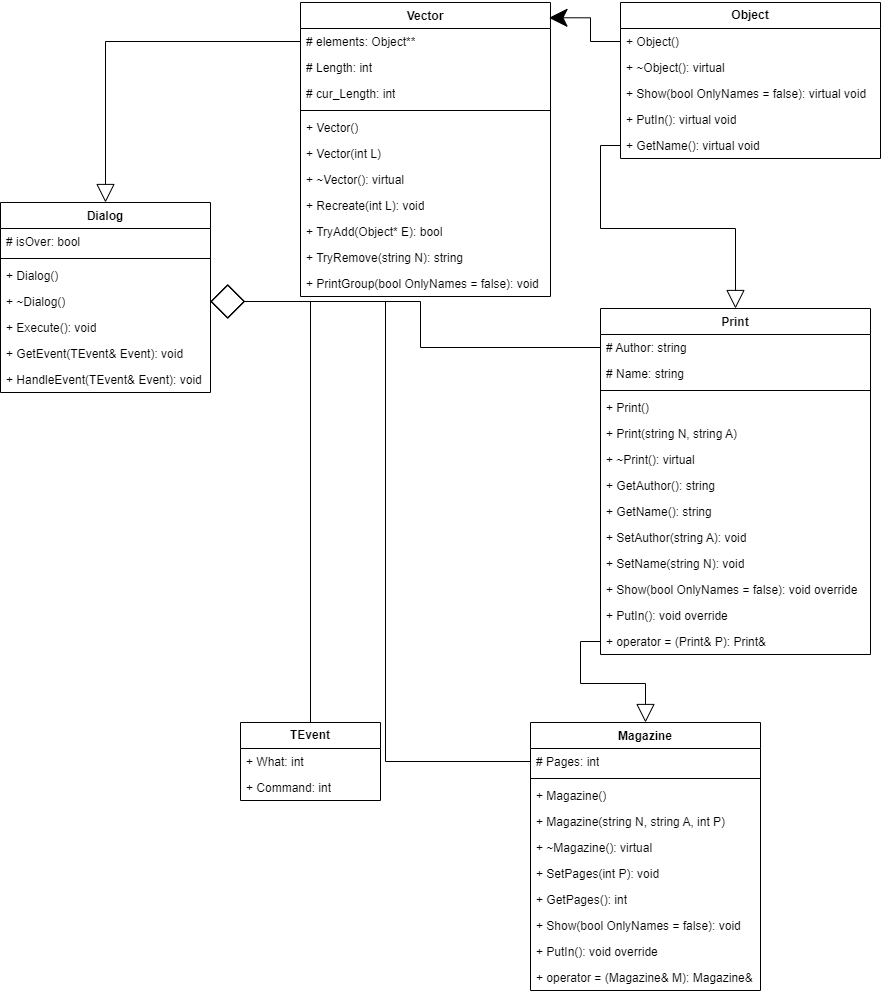
Класс Magazine - Класс, описывающий журнал. Наследуется от Print

Класс Vector - Класс-контейнер

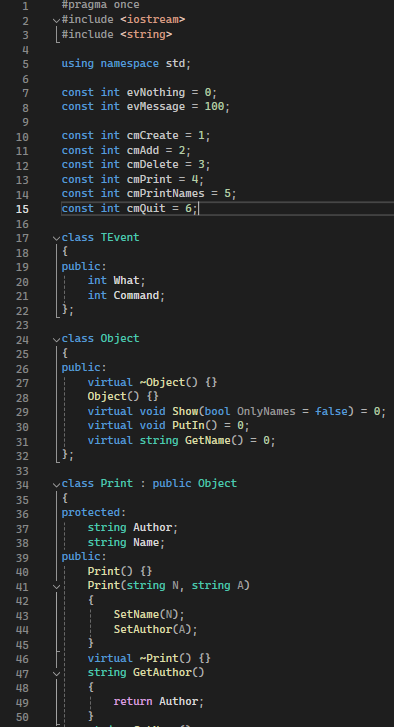
Класс Dialog - Класс, обеспечивающий интерфейс программы. Наследуется от Vector

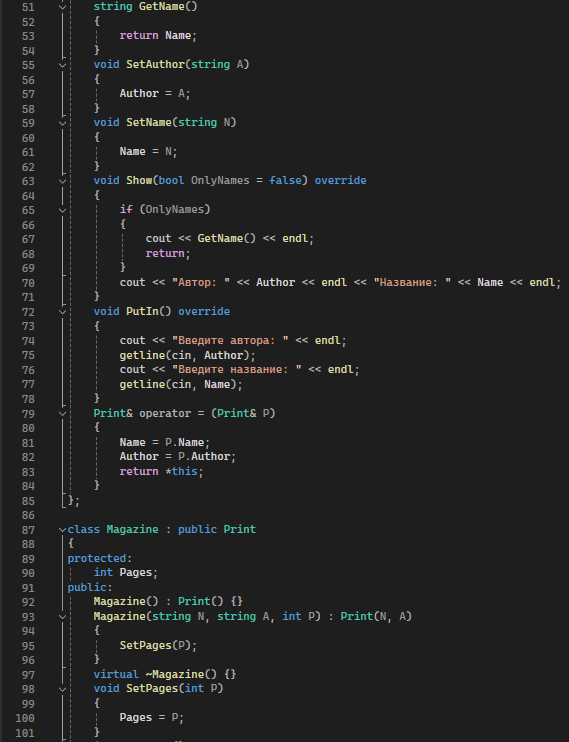
**3. Разбор работы алгоритма**

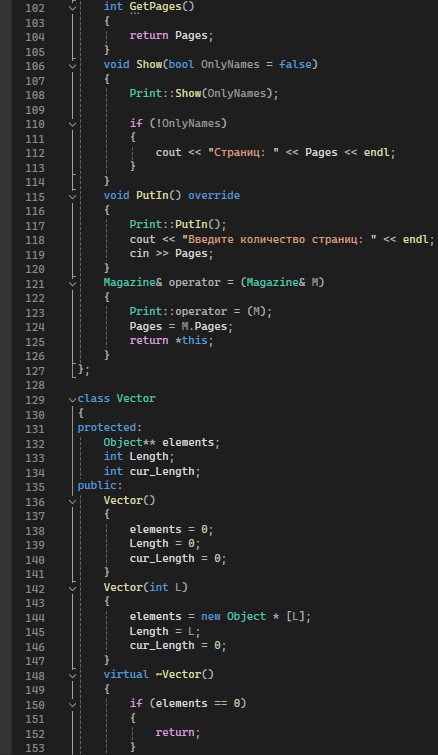
UML диаграмма

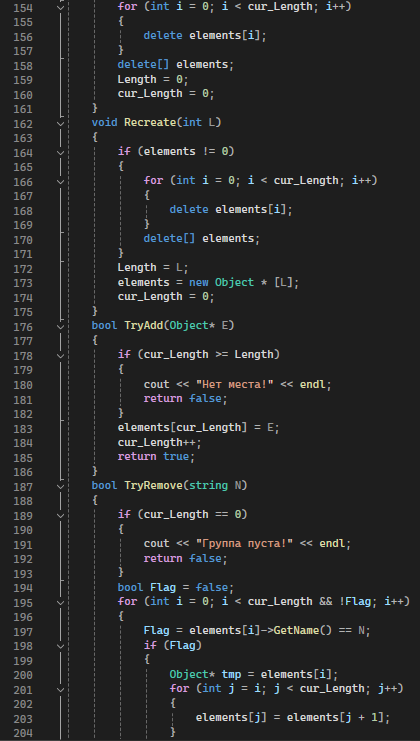


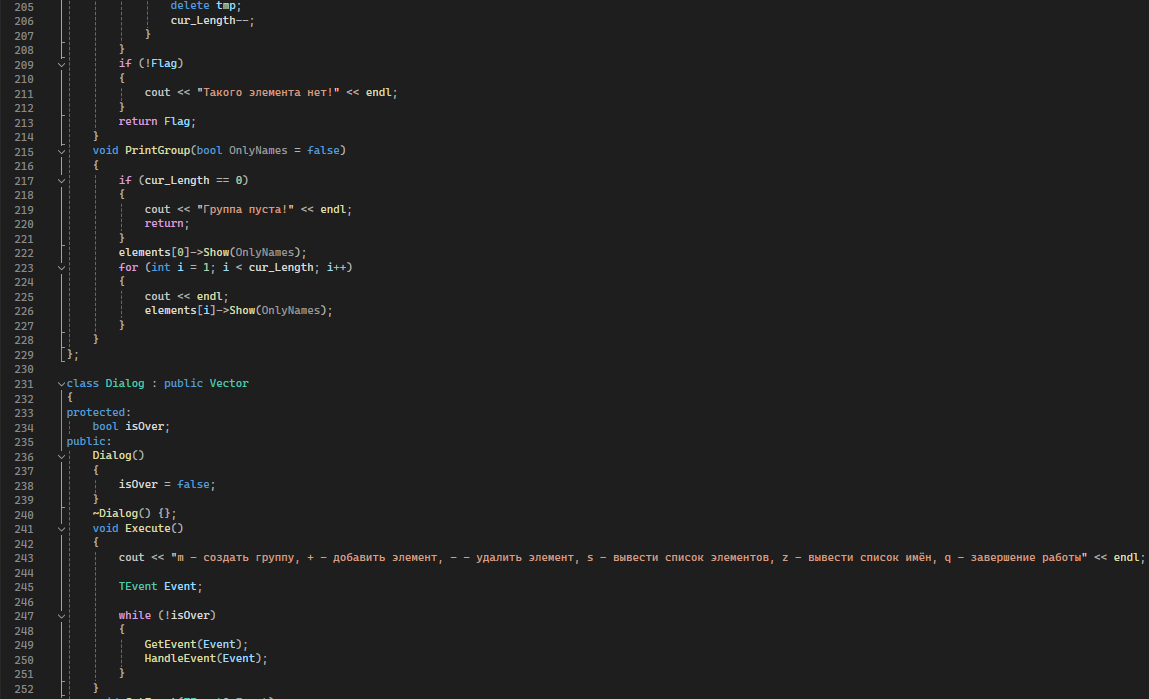
Код программы

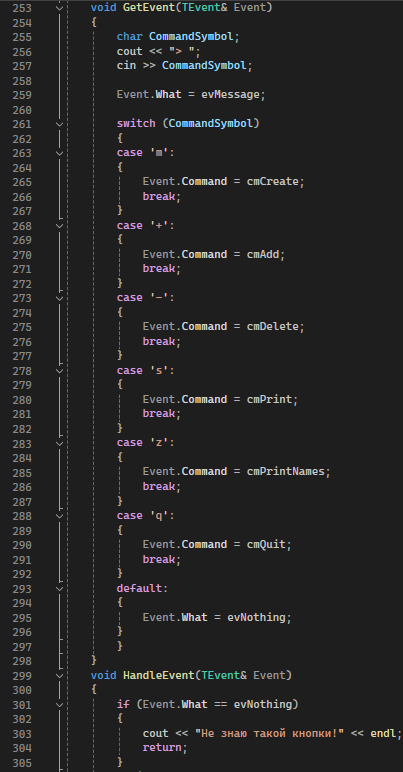


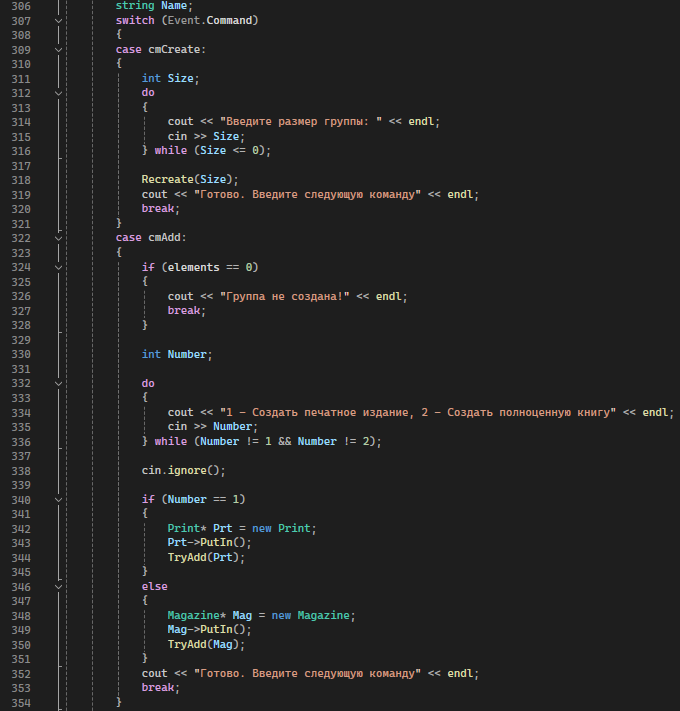


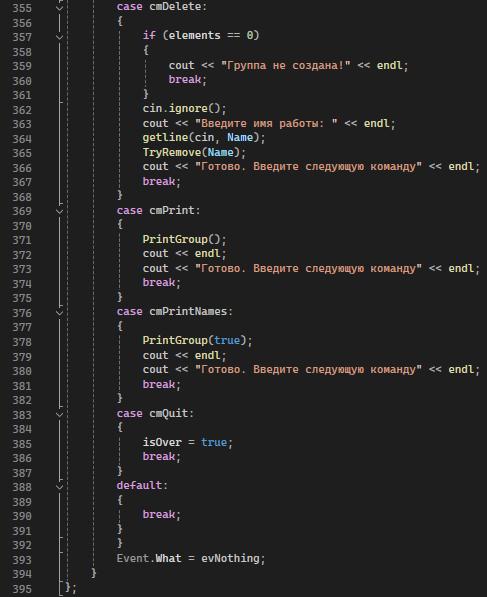




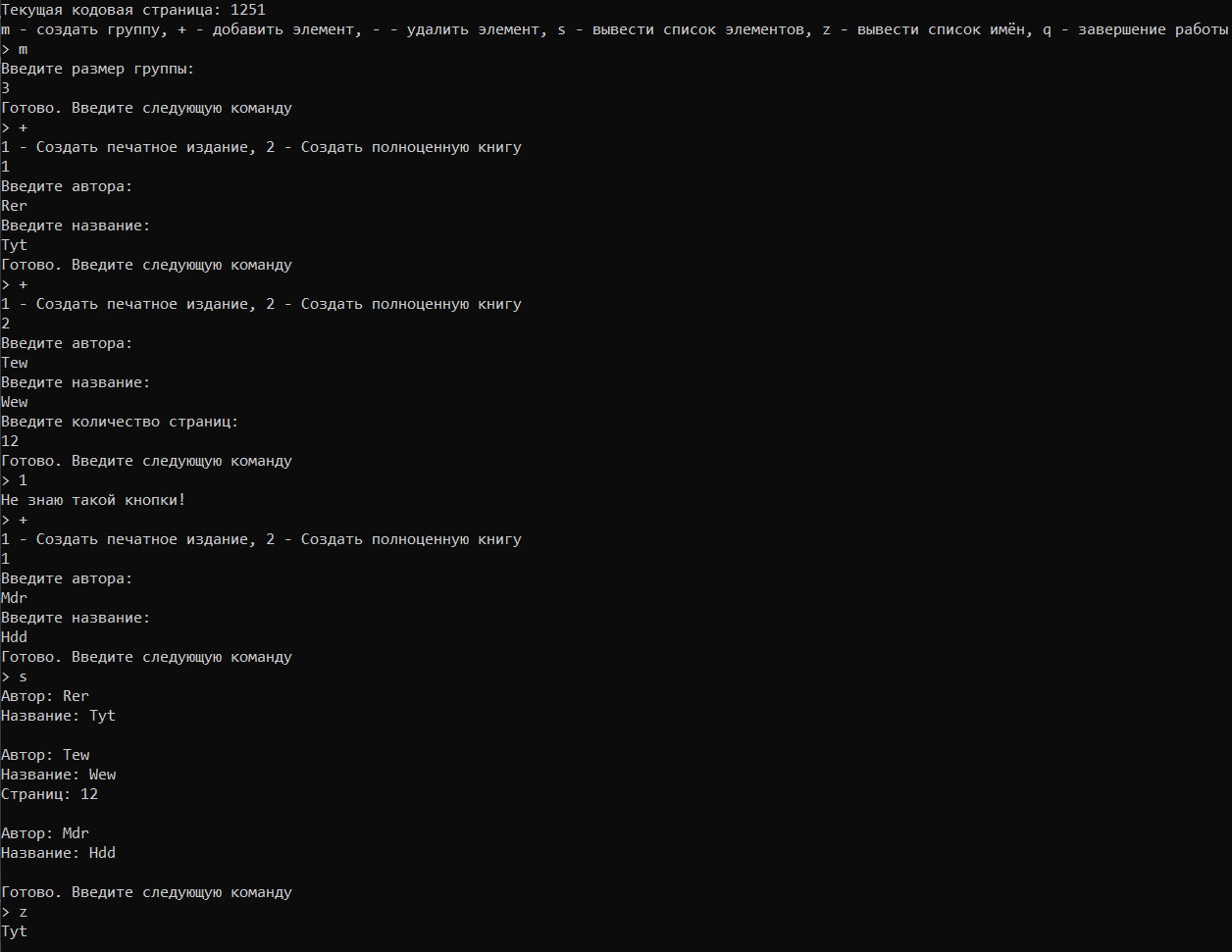








**4. Результаты работы программы**





**5. Контрольные вопросы**

1. Класс-группа – это класс, содержащий в себе набор элементов. Прим.: класс STL vector.

2. List<int> data;

3. List(){}

List(int count, int num …){for (int I = 0; i<count; i++){int\*ptr = &num; data.push\_back(\*ptr); ptr++;}}

List(List&lst){data = lst.data;}

4. ~List () {}

5. Проход итератором.

6. иерархию объектов.

7. Для обеспечения принципа подстановки.

8. Событие – это объект, содержащий в себе информацию от внешних источников (ввод пользователя, показатели датчиков) для ответа от программы.

9. Событие-сообщение должно содержать корректный код события и некоторую информацию.

10.

#pragma once

#include "Vector.h"

const bool ev\_empty = true;

const bool new\_message = false;

const char createGroup = 'm';

const char addElem = '+';

const char delElem = '-';

const char printData = 's';

const char printNames = 'z';

const char stopExec = 'q';

class Event

{

public:

bool isEmpty;

struct {

char command; //command with argument

int arg;

};

void clear() {

isEmpty = ev\_empty;

command = ' ';

arg = 0;

}

};

11. целочисленные.

12. целочисленные.

13. Определяют тип сообщения и аргумент.

14. getEvent(), response(), execute()

15. бесконечного цикла приёма сообщения и генерации ответа.

16. ClearEvent() очищает объект события, сбрасывая все значения к первоначальным.

17. HandleEvent() – набор switch-case-ов, отвечающий за действия программы.

18. GetEvent() – Метод, отвечающий за ввод события и решение обрабатывать его или нет.

19. EndState – Boolean переменная, отвечающая за работу программы. Если сменить её значения, то цикл прервётся.

20. Valid() проверяет введённые данные на правильность.

**6. Вывод**

Алгоритм выполняет поставленную задачу.

Данный проект можно найти на GitHub



