Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №8.**

**«Программа, управляемая событиями»**

Выполнил студент гр. РИС-24-2б

Молочко Артём Анатольевич

Проверил:

Доц. Каф. ИТАС

Ольга Андреевна Полякова

(оценка) (подпись)

(дата)

г. Пермь, 2025

**Постановка задачи**

Реализовать программу, управляемую событиями, для работы с печатными изданиями и журналами, используя класс-группу "Дерево".

Базовые классы:

PRINT (Печатное издание):

* + Название – string
  + Автор – string

MAGAZIN (Журнал) - производный от PRINT:

* + Количество страниц – int

### Команды:

* Создать группу (формат команды: m количество элементов группы)
* Добавить элемент в группу (формат команды: +)
* Удалить элемент из группы (формат команды: -)
* Вывести информацию об элементах группы (формат команды: s)
* Вывести информацию о названиях всех элементов группы (формат команды: z)
* Конец работы (формат команды: q)

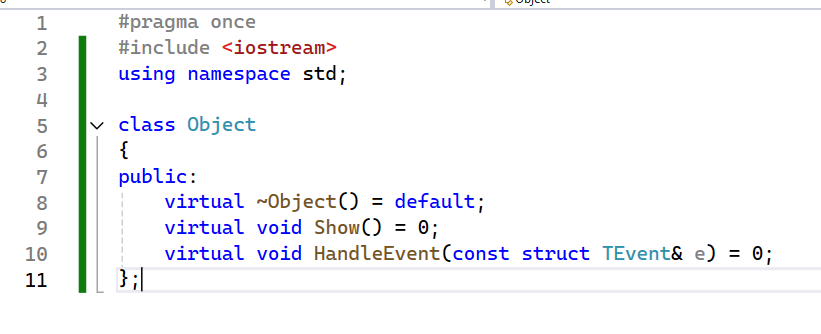
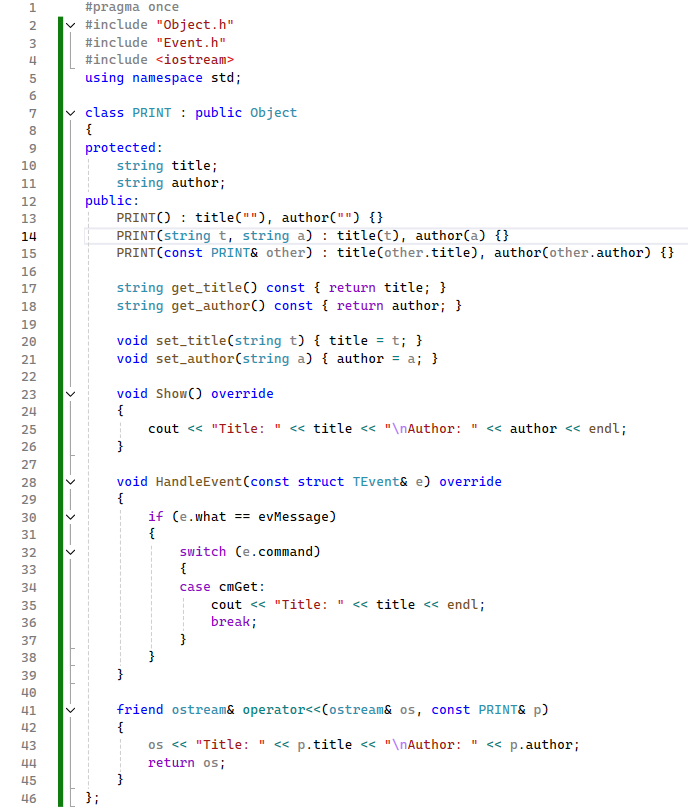
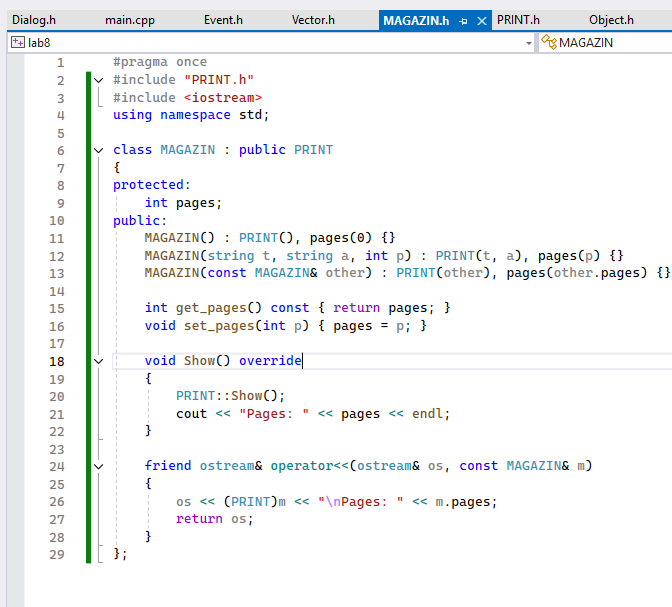
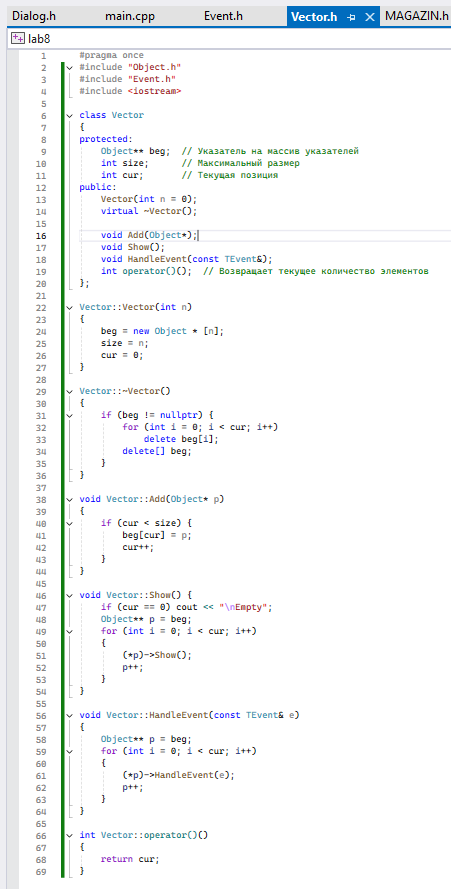
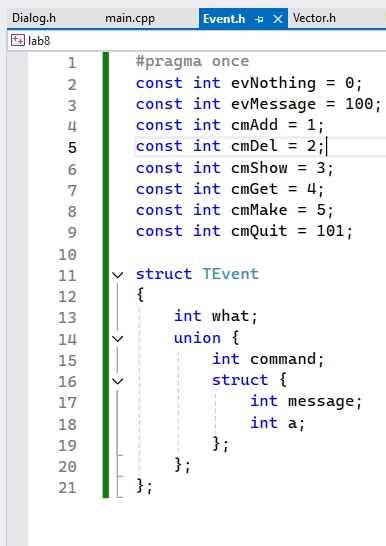
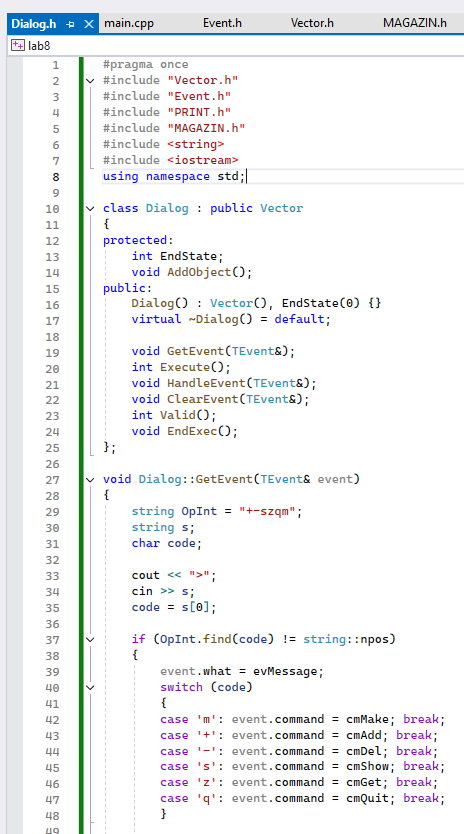
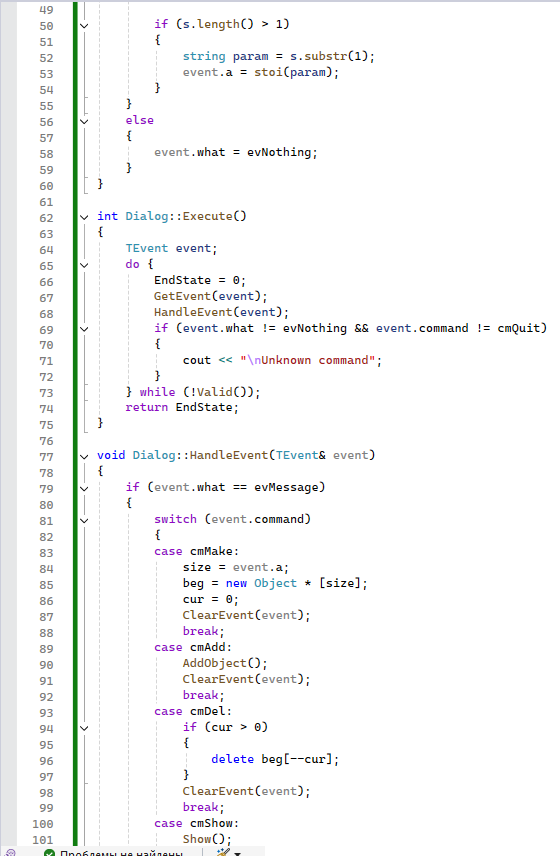
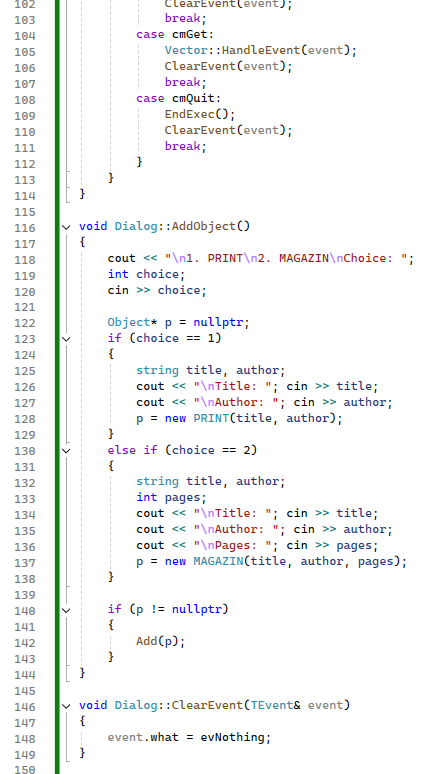
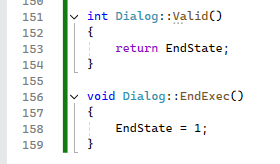
**Анализ задачи**

**Базовый класс**Object с виртуальными методами для полиморфной работы

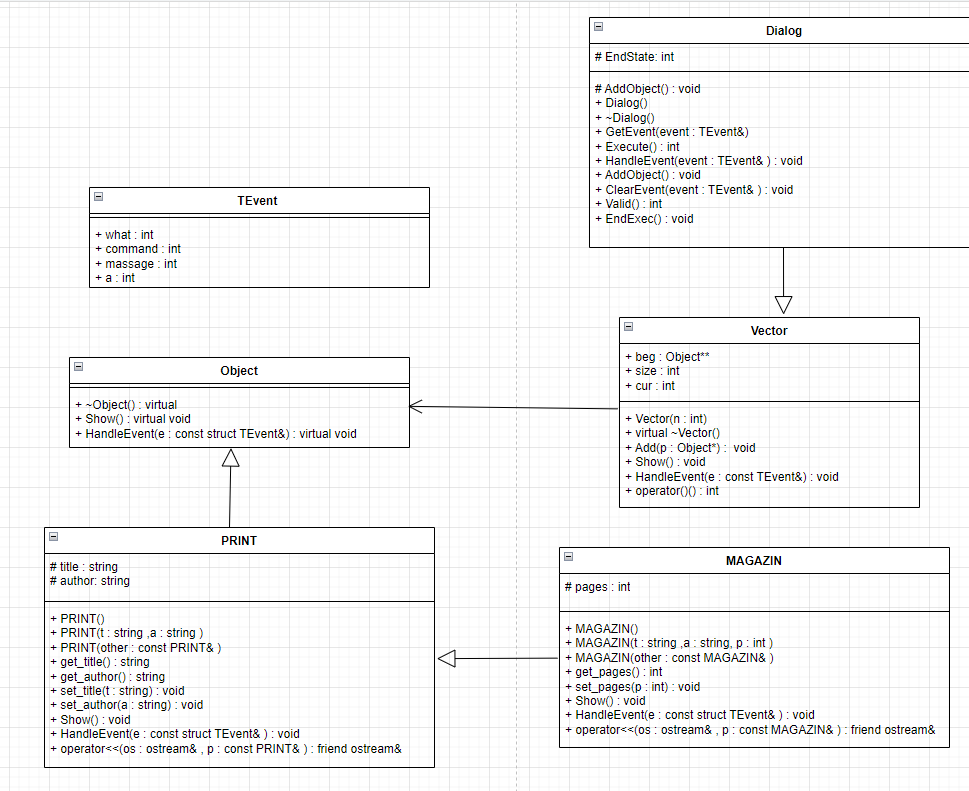
* **Производные классы**PRINT**и**MAGAZIN для хранения данных о печатных изданиях и журналах
* **Класс**Vector как контейнер для объектов (реализация группы)
* **Класс**Dialog для обработки событий и управления программой

**Обработка событий**  
Реализована через:

* **Структуру**TEvent (тип события + параметры)
* **Цикл**Execute() в Dialog, обрабатывающий команды пользователя
* **Методы**HandleEvent() в каждом классе для реакции на события

**Код программы**        

**Диаграмма классов**



**Контрольные вопросы**

**1. Что такое класс-группа? Привести примеры таких классов.**  
**Ответ:**  
Класс-группа — это класс, который содержит и управляет коллекцией других объектов (элементов). Примеры: Vector, List, Dialog из лабораторной работы.

**2. Привести пример описания класса-группы Список (List).**  
**Ответ:**

class List {

Node\* head;

Node\* tail;

int size;

public:

void Add(int value);

void Remove(int value);

void Print();

};

**3. Привести пример конструктора (с параметром, без параметров, копирования) для класса-группы Список.**  
**Ответ:**

List(); // Без параметров

List(int initialSize); // С параметром

List(const List& otherList); // Копирования

**4. Привести пример деструктора для класса-группы Список.**

~List() {

while (head != nullptr) {

Node\* temp = head;

head = head->next;

delete temp;

}

}

**5. Привести пример метода для просмотра элементов для класса-группы Список.**

void Print() {

Node\* current = head;

while (current != nullptr) {

cout << current->data << " ";

current = current->next;

}

}

**6. Какой вид иерархии дает группа?**  
**Ответ:**  
Группа создает иерархию "часть-целое" (агрегация/композиция), где класс-группа является целым, а элементы — частями.

**7. Почему во главе иерархии классов, содержащихся в группе объектов должен находиться абстрактный класс?**  
**Ответ:**  
Абстрактный класс (Object) обеспечивает единый интерфейс для работы с разными типами объектов через полиморфизм.

**8. Что такое событие? Для чего используются события?**  
**Ответ:**  
Событие — это объект, содержащий информацию о произошедшем действии. Используются для организации взаимодействия между объектами.

**9. Какие характеристики должно иметь событие-сообщение?**  
**Ответ:**

1. Тип события
2. Код команды
3. Дополнительные параметры

**10. Привести пример структуры, описывающей событие.**

struct TEvent {

int what;

union {

int command;

struct { int param1, param2; };

};

};

**11. Задана структура события. Какие значения, и в каких случаях присваиваются полю what?**  
**Ответ:**

what принимает значения:

* evNothing (0) — пустое событие
* evMessage (100) — событие-сообщение
* Другие коды для специфичных событий

**12. Задана структура события. Какие значения, и в каких случаях присваиваются полю command?**  
**Ответ:**  
command принимает значения:

* cmAdd (1) — добавить элемент
* cmDel (2) — удалить элемент
* cmShow (3) — показать элементы
* Другие коды команд

**13. Задана структура события [...] Для чего используются поля а и message?**  
**Ответ:**

* message — код дополнительной информации
* a — числовой параметр команды

**14. Какие методы необходимы для организации обработки сообщений?**  
**Ответ:**

1. GetEvent() — получение события
2. HandleEvent() — обработка события
3. ClearEvent() — очистка события

**15. Какой вид имеет главный цикл обработки событий-сообщений?**  
**Ответ:**

while (!EndState) {

GetEvent(event);

HandleEvent(event);

if (event.what != evNothing) ClearEvent(event);

}

**16. Какую функцию выполняет метод ClearEvent()? Каким образом?**  
**Ответ:**  
ClearEvent() обнуляет событие, присваивая полю what значение evNothing.

**17. Какую функцию выполняет метод HandleEvent()? Каким образом?**  
**Ответ:**  
HandleEvent() обрабатывает событие через switch-case по значению command.

**18. Какую функцию выполняет метод GetEvent()?**  
**Ответ:**  
GetEvent() получает ввод пользователя и преобразует его в структуру TEvent.

**19. Для чего используется поле EndState? Какой класс (объект) содержит это поле?**  
**Ответ:**  
EndState — флаг завершения программы. Содержится в классе Dialog.

**20. Для чего используется функция Valid()?**  
**Ответ:**  
Valid() проверяет условие завершения программы (возвращает true, если EndState != 0).