Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЁТ**

**по дисциплине «Информатика»**

Семестр: 2

На тему: «Программа управления событиями»

Выполнил студент ИВТ-22-2б:

Коняев Александр Сергеевич

Проверил доцент кафедры ИТАС:

Полякова Ольга Андреевна

Пермь 2023

**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.

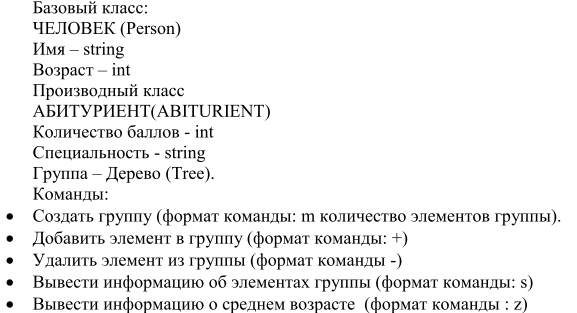
1. Определить класс-группу на основе структуры, указанной в варианте.
2. Для группы реализовать конструкторы, деструктор, методы для добавления и

удаления элементов в группу, метод для просмотра группы, перегрузить операцию

для получения информации о размере группы.

1. Определить класс Диалог – наследника группы, в котором реализовать методы дляобработки событий.
2. Добавить методы для обработки событий группой и объектами пользовательских классов.
3. Написать тестирующую программу.
4. Нарисовать диаграмму классов и диаграмму объектов.

**Вариант 13:**



**Описание классов**

Класс Object.

* Конструктор без параметров
* Метод Input – ввод значений
* Метод Show – вывод значений
* Метод HandleEvent

Класс Person.

* Конструкторы
* Метод Show
* Метод Input
* Метод Get\_name
* Метод Set\_name
* Метод Get\_age
* Метод Set\_age
* Метод HandleEvent
* Перегруженный оператор =

Класс Abiturient.

* Конструкторы
* Метод Show
* Метод Input
* Метод Get\_count
* Метод Set\_count
* Метод Get\_spec
* Метод Set\_spec
* Перегруженный оператор =

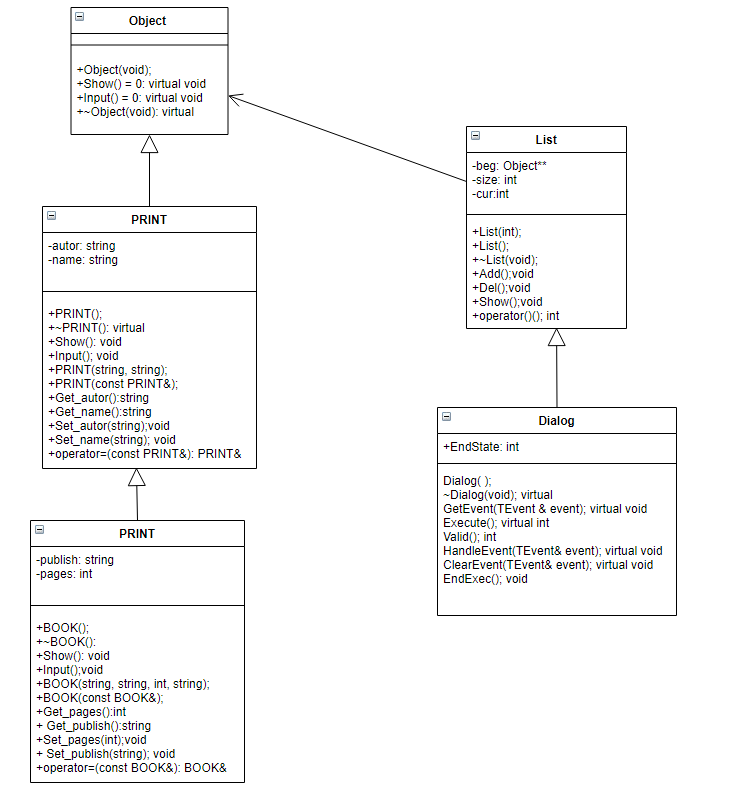
Класс Vector.

* Конструкторы
* Метод Add
* Метод Del
* Метод Show
* Метод HandleEvents

Класс Dialog.

Методы типа public:деструктор, конструктор без параметров, виртуальные функции: GetEvent(TEvent & event), Execute(); HandleEvent(TEvent& event), ClearEvent(TEvent& event), Valid(); EndExec().

Также описание классов представлено на UML-диаграмме.



**Основная функция**

int main() {

setlocale(LC\_ALL, "rus");

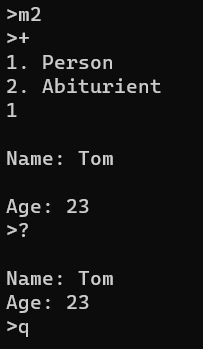
Dialog D;

D.Execute();

return 0;

}}

**Результат работы программы**

****

**Контрольные вопросы**

1. *Что такое класс-группа? Привести примеры таких классов.*

Группа − это объект, в который включены другие объекты. Объекты, входящие в

группу, называются элементами группы. Элементы группы, в свою очередь, могут быть группой.

Примеры:

1. Дом, состоящий из окон дверей и т.д.
2. Огород, состоящий из растений, системы полива и плана выращивания.
3. Человек, состоящий из внутренних органов, внешних данных.
4. *Привести пример описания класса-группы Список (List).*

class List {

private:

Object\* c1;//Указатель на голову списка

int size;//Размер списка

};

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

//Конструктор без параметров

List::List() {

c1 = NULL;

size = 0;

}

//Конструктор с параметрами

List::List(int n) {

if (n < 1) {

return;

}

else {

size = n;

Object\* buff = new Object;//Первый элемент списка

buff->v1 = NULL;

buff->cc = 0;

c1 = buff;

for (int i = 1; i < n; i++) {//СОЗДАНИЕ СЛЕДУЮЩИХ УЗЛОВ СПИСКА

Object\* buff = new Object;

buff->v1 = c1;

buff->cc = 0;

c1 = buff;

}

}

}

//Конструктор копирования

List::List(const List& v1) {

size = v1.size;

c1 = v1.c1;

}

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

List::~List() {

for (int s = 0; s < size; s++) {

Object\* b1 = c1;

c1 = c1->v1;

delete b1;

}

size = 0;

}

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

int List::check(int vv) {

if (vv >= size) {

return 0;

}

else {

Object\* b1 = c1;//Буферный указатель на узел списка

for (int i = 0; i < vv; i++) {//Перевод указателя на нужный элемент

b1->v1;

}

return b1->cc;//Вызвращает поле найденного объекта

}

}

1. *Какой вид иерархии дает группа?*

Группа иерархию объектов(иерархию типацелое/часть), построенную на основе агрегации, первый вид иерархия классов,построенная на основе наследования.

1. *Почему во главе иерархии классов, содержащихся в группе объектов должен*

*находиться абстрактный класс?*

Абстрактный класс позволяет определять методы индивидуально для каждого производного класса.

1. *Что такое событие? Для чего используются события?*

Событие с точки зрения языка С++ – это объект, отдельные поля которого

характеризуют те или иные свойства передаваемой информации.

События нужны для более интерактивного взаимодействия программы и пользователя(зоздание окон, меню и пр.), независимо от действий, которые будут связаны с ним.

1. *Какие характеристики должно иметь событие-сообщение?*

Command – код команды, которую необходимо выполнить при появлении

данного события;

Передаваемая с событием информация (сообщение).

1. *Привести пример структуры, описывающей событие.*

struct TEvent

{

int what;//тип события

union

{

int command;//код команды

int message;//Сообщение события

};

};

*11. Задана структура события*

struct TEvent

{

int what;

union

{

MouseEventType mouse;

KeyDownEvent keyDown;

MessageEvent message;

}

};

*Какие значения, и в каких случаях присваиваются полю what?*

What задаёт тип события. Существует два типа событий evNothing – это пустое событие и evMessage - это событие-сообщение от объекта.

evNothing присваеивается what, после обработки очередного события, а также what изначально присвоено evNothing.

evMessage присваеивается what, когда в обработку поступает событие.

1. *Задана структура события*

struct TEvent

{

int what;//тип события

union

{

int command;//код комманды

struct//параметры команды

{

int message;

int a;

};

};

};

*Какие значения, и в каких случаях присваиваются полю command?*

Command будет иметь значение равное какому-нибудь событию, то есть ,например, программист задал три события(Eventzero=0, Eventone=1, Eventtwo=2), то command может иметь значение 1,2,3.

*13. Задана структура события*

struct TEvent

{

int what;//тип события

union

{

int command;//код комманды

struct//параметры команды

{

int message;

int a;

};

};

};

*Для чего используются поля a и message?*

Message нужно для определения сообщения определённого события, а “a” это параметр определённого события.

1. *Какие методы необходимы для организации обработки сообщений?*
2. Нужен метод формирования событий событий.
3. Главный цикл обработки событий. Он постоянно получает

событие путем вызова метода формирования событий и обрабатывает их с помощью обработчика событий. Этот цикл завершается, когда поступит событие «конец».

1. Обработчик событий. Обрабатывает каждое событие нужным для

него образом. Если объект должен обрабатывать определенное событие (сообщение), то его «обработчик» должен распознавать это событие и реагировать на него должным образом. Событие может распознаваться, например, по коду команды (поле command).

1. Метод очистки событий. Когда событие обработано,то его нужно очистить, чтобы оно не обрабатывалось далее.
2. *Какой вид имеет главный цикл обработки событий-сообщений?*

int TMyApp::Execute()

{

do

{

endState=0;

GetEvent(event); //получить событие

HandleEvent(event); //обработать событие

//событие осталось не обработано

if(event.what!=evNothing)

EventError(event);

}

while(!Valid());

return endState;

}

1. *Какую функцию выполняет метод ClearEvent()? Каким образом?*

ClearEvent очищает событие, когда оно обработано, чтобы оно не обрабатывалось далее. Присваивает what пустое событие.

1. *Какую функцию выполняет метод HandleEvent ()?Каким образом?*

Обрабатывает каждое событие нужным для него образом. HandleEvent распознает, например, по коду команды (поле command), какое действие ему нужно совершить и выполняет его.

1. *Какую функцию выполняет метод GetEvent ()?*

Формирование событий.

1. *Для чего используется поле EndState? Какой класс (объект) содержит это поле?*

Используется, как флаг окончания диалога. Его содержит класс Dialog.

1. *Для чего используется функция Valid()?*

Для проверки поля EndState, или же проверки конца диалога.