

Программирование

Объектно-ориентированное программирование на языке C++

Лабораторная работа № 2

НАСЛЕДОВАНИЕ И ВИРТУАЛЬНЫЕ ФУНКЦИИ

Цель. Получить практические навыки создания иерархии классов, использования статических компонентов класса и виртуальных функций.

Теоретический материал.

Перед выполнением работы необходимо проработать раздел 2 и §1.4.4 пособия [1].

Основное содержание работы.

В одном решении (рабочей области) создать два проекта.

Проект № 1

Используя массив указателей типа базового класса, просмотреть полиморфные объекты с помощью виртуальной функции show().

Порядок выполнения проекта.

1. Определить иерархию классов (в соответствии с вариантом).
2. Реализовать классы.
4. Создать объекты разных классов.
5. Создать массив указателей типа базового класса.
5. Инициализировать эти указатели указателями на созданные объекты.
6. Пройдя по массиву, просмотреть объекты, вызывая виртуальную функцию show().
7. Сделать функцию show не виртуальной и посмотреть к чему это приведет.

Проект № 2

Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

Порядок выполнения проекта.

1. Определить иерархию классов (в соответствии с вариантом).
2. Определить в абстрактном классе абстрактную функцию show() для просмотра объекта
3. Определить класс типа item для создания связанного списка. Определить в классе статическую компоненту - указатель на начало связанного списка объектов, функцию для добавления объекта и статическую функцию для просмотра списка.
4. Реализовать классы.

5. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается.

6. Сделать соответствующие методы не виртуальными и посмотреть, что будет.

Методические указания.

1. Для определения иерархии классов связать отношением наследования классы, приведенные в приложении (для заданного варианта). Из перечисленных классов выбрать один, который будет стоять во главе иерархии. Это абстрактный класс.

2. Определить в классах все необходимые конструкторы и деструктор.

3. Строковые поля определять как `string`.

4. Компонентные данные класса специфицировать как **protected**.

5. Пример определения статических компонентов:

```
static person* begin; // указатель на начало списка
```

```
static void print(void); // просмотр списка
```

6. Статическую компоненту - данное инициализировать вне определения класса, в глобальной области.

7. Пример определения класса `item` :

```
class item
{
    static item* begin;
    item* next;
    person* p;
public:
    item(person* p);
    void add();
    static void show();
};
```

Функция `add()` добавляет элемент в начало списка.

8. Список просматривать путем вызова виртуального метода **show** каждого объекта.

9. Статический метод просмотра списка вызывать не через объект, а через класс.

10. Определение классов, их реализацию, демонстрационную программу поместить в отдельные файлы.

Содержание отчета.

1. Титульный лист: название дисциплины; номер и наименование работы; номер варианта; фамилия, имя, отчество студента; дата выполнения.

2. Постановка задачи. Следует дать конкретную постановку, т.е. указать, какие классы должны быть реализованы, какие должны быть в них конструкторы, компоненты-функции и т.д.
3. Иерархия классов в виде графа.
4. Определение пользовательских классов с комментариями.
5. Реализация конструкторов с параметрами и деструктора с пояснением.
6. Реализация методов для добавления объектов в список с пояснением.
7. Реализация методов для просмотра списка с пояснением.
8. Листинги демонстрационной программы (функции `main()`) каждого проекта с комментарием.
9. Объяснение необходимости виртуальных функций. Следует показать, какие результаты будут в случае виртуальных и не виртуальных функций.

Литература

1. Ноткин А.М. Технология объектно-ориентированного программирования на языке C++: Электронное учебное пособие – Пермь: Изд-во Перм. гос. техн. ун-та, 2007.

Приложение. Варианты заданий.

Перечень классов:

- 1) студент, преподаватель, персона, завкафедрой;
- 2) служащий, персона, рабочий, инженер;
- 3) рабочий, персона, инженер, администрация;
- 4) деталь, механизм, изделие, узел;
- 5) журнал, книга, печатное издание, учебник.
- 6) тест, экзамен, выпускной экзамен, испытание;
- 7) место, область, город, мегаполис;
- 8) игрушка, продукт, товар, молочный продукт;
- 9) квитанция, накладная, документ, чек;
- 10) цех, мастерская, фабрика, производство;
- 11) персона, студент, школьник, студент-заочник;
- 12) автомобиль, поезд, транспортное средство, экспресс;
- 13) республика, монархия, королевство, страна;
- 14) млекопитающие, парнокопытные, птицы, животное;
- 15) корабль, пароход, парусник, корвет,