

Лабораторная работа №2

По курсу «Программирование»

«СОЗДАНИЕ СПИСКА ОБЪЕКТОВ»

Цель: Получить практические навыки создания связного списка объектов.

Порядок выполнения работы:

1. Показать в программе использование указателя на объект и указателя на компоненту-функцию.
2. Определить в классе статическую компоненту – указатель на начало связанного списка объектов и статическую функцию для просмотра списка.
3. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается.
4. Сделать соответствующие методы неvirtуальными и посмотреть, что будет.
5. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе (смотри пункт 5 следующего раздела).

Содержание отчета:

1. Титульный лист.
2. Постановка задачи.
3. Иерархия классов.
4. Определение классов (добавленных или измененных по сравнению с лабораторной работой №1).
5. Фрагмент программы, показывающий использование указателя на объект и указателя на функцию с объяснением.
6. Реализация методов для добавления объектов в список.
7. Реализация методов для просмотра списка.
8. Листинг демонстрационной программы.

Методические указания:

1. Можно определить указатель на компоненты-функции.
*тип_возвр_значения (имя_класса :: *имя_указателя_на_функцию)*
(специф_параметров_функции);

Пример.

```
double (complex::*ptcom) (); // Определение указателя
ptcom = &complex::real; // Настройка указателя
// Теперь для объекта А можно вызвать его функцию
complex A(5.2, 2.7);
cout<<(A.*ptcom) ();
```

Можно определить также тип указателя на функцию

```
typedef double (complex::*PF) ();
а затем определить и сам указатель
PF ptcom=&complex::real;
```

Лабораторная работа №2

Срок сдачи – 4 неделя

2. В программе необходимо предусмотреть размещение объектов как в статической, так и в динамической памяти, а также создание массивов объектов.

Примеры.

а) массив студентов размещается в статической памяти

```
STUDENT группа[3];
группа[0].Set("Иванов", 19, 50);
```

и т.д.

или

```
STUDENT группа[3]={STUDENT("Иванов", 19, 50),
STUDENT("Петрова", 18, 25.5),
STUDENT("Сидоров", 18, 45.5)};
```

б) массив студентов размещается в динамической памяти

```
STUDENT *p;
p=new STUDENT [3];
p->Set("Иванов", 19, 50);
```

и т.д.

2. Пример использования указателя на компонентную функцию

```
void (STUDENT::*pf)();
pf=&STUDENT::Show;
(p[1].*pf)();
```

3. Пример определения статических компонентов:

```
static person *begin;           // указатель на начало списка
static void print(void);       // просмотр списка
```

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

5. Для добавления объекта в список предусмотреть метод класса, т.е. объект сам добавляет себя в список. Например, `a.Add()` – объект `a` добавляет себя в список.

6. Включение объекта в список можно выполнять при создании объекта, т.е. поместить операторы включения в конструктор. В случае иерархии классов, включение объекта в список должен выполнять **только** конструктор базового класса. Вы должны продемонстрировать оба этих способа.

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

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