

# Лабораторная работа №1 «Создание классов и объектов»

## Цель

---

Научиться создавать классы и объекты этих классов, вызывать методы и обращаться к данным через методы этих объектов, передавать объекты других классов через методы, понимать влияние модификаторов доступа на видимость данных и методов.

## Теоретическая информация

---

### Понятия класса и объекта

**Объект** – это сущность, выделяемая из словаря предметной области, которая характеризуется набором свойств и событийным поведением. Последнее обычно представляется в виде множества событий, из которых образуется жизненный цикл объекта. В более распространенной терминологии для именования этих двух аспектов описания объекта используют термины **атрибут** и **метод**. Атрибутами могут выступать как переменные стандартных типов данных, так и объекты других классов.

**Класс** – фундаментальное понятие объектно-ориентированного метода. Это абстракция, обозначающее некоторое подмножество объектов, имеющих сходные черты как в наборе свойств, так и в поведении. На C++ создание (описание) класса выполняется с помощью следующей конструкции:

```
тип_класса имя_класса{список_членов_класса};
```

где тип\_класса – одно из служебных слов class, struct, union;

имя\_класса – идентификатор;

список\_членов\_класса – определения атрибутов и методов класса.

### Пример:

```
class Window {  
    int x;                //координаты левого верхнего угла
```

```

        int y;                                //
        int Width, Height;                    //ширина и высота
public:
        void Hide() { /*реализация*/ };        //скрыть
        void Show() { /*реализация*/ };        //показать
        void Paint() { /*реализация*/ };        //прорисовать
    } ;

```

## Создание объектов

Для создания объекта используется конструкция, аналогичная определению переменной:

```
имя_класса имя_объекта;
```

### Пример:

```

car my_car, car2;
car *carp = &car2;           // указатель на объект
car cars[30];                // массив объектов
car &c = my_car;              // ссылка на объект

```

Второй способ создавать указатель на объект с помощью операции new:

```
Имя_класса * имя_указателя = new Имя_конструктора(параметры_конструктора) ;
```

### Пример:

```
Window *w1 = new Window();
```

Обращаться к данным (атрибутам) объекта и вызывать методы объекта, если они отмечены как public, можно двумя способами. Первый с помощью конструкций:

```

имя_объекта.имя_данного
имя_объекта.имя_функции

```

### Пример:

```

Window w1, w2;
w1.x = 3.4;
w1.y = 0.3;
w1.Show();
w2.Paint();

```

Второй способ – используя указатель на объект:

указатель\_на\_объект->имя\_данного  
указатель\_на\_объект->имя\_функции

### Пример:

```
Window *w1 = new Window();  
w1->x = 3.4;           //возможно, если x public  
w1->y = 0.3;           //возможно, если y public  
w1->Show();
```

## **Видимость атрибутов и методов**

Для изменения видимости атрибутов и методов в определении класса можно использовать спецификаторы доступа: **public**, **private**, **protected**.

Общедоступные (**public**) – доступны в любой части программы. Они могут использоваться любой функцией как внутри данного класса, так и вне.

Собственные (**private**) атрибуты и методы могут использоваться только в классе и не доступны извне. Они могут использоваться также функциями – “друзьями” данного класса. Если модификатор доступа не указан, то члены класса по умолчанию считаются **private**.

Защищенные (**protected**) компоненты доступны внутри класса и в классах наследниках.

Считается, что данные (атрибуты) класса должны быть скрыты от внешнего доступа, т.е. описаны как **private**. Доступ к таким данным осуществляется через методы (**get**, **set**), которые описываются в классе как **public**.

### Пример:

```
class Text {  
    AnsiString s;           // private по умолчанию  
public:  
    AnsiString& getText(void); //доступно из вне  
};
```

## **Перегрузка методов**

В одном классе могут быть реализованы **несколько методов с одинаковым именем** (перегрузка). Такое возможно при соблюдении одного требования: набор параметров методов должен быть разным.

## Задание

---

1. Создать приложение согласно выбранного варианта.
2. В приложении должно быть описано 2 класса (основной и дополнительный), в каждом из которых должны быть закрытые (private) данные и открытые (public) методы.
3. Функции, которые должны быть реализованы в приложении: добавление, удаление, редактирование и просмотр.
4. Хотя бы в одном из классов должно быть несколько методов с одинаковыми именами, но с разным набором параметров для инициализации сразу нескольких атрибутов класса.
5. В основном классе должен быть динамический массив объектов дополнительного класса (рекомендуется использовать list).
6. Для добавления в этот массив должен использоваться метод, в параметрах которому должен передаваться объект дополнительного класса.
7. Реализовать все необходимые проверки на ввод данных.
8. В теле основной программы необходимо реализовать меню, через которое можно использовать весь функционал программы.
9. Сделать выводы.

## Варианты

---

### Вариант 1

Тема проекта: приложение «Журнал регистрации корреспонденции».

В приложении должно быть реализовано 2 класса: Журнал регистрации и Корреспонденция. Класс Журнал содержит номер, имя. Класс Корреспонденция содержит дату, тему, отправителя.

## **Вариант 2**

Тема проекта: приложение «Учет денежных средств».

В приложении должно быть реализовано 2 класса: Журнал учета и Платеж. Класс Журнал содержит карта, имя. Класс Платеж содержит дату, сумму, описание.

## **Вариант 3**

Тема проекта: приложение «Энциклопедия «История в лицах».

В приложении должно быть реализовано 2 класса: Энциклопедия и Личность. Класс Энциклопедия содержит название, год издания. Класс Личность содержит имя, род деятельности, описание.

## **Вариант 4**

Тема проекта: приложение «Касса кинотеатра».

В приложении должно быть реализовано 2 класса: Кинотеатр и Сеанс. Класс Кинотеатр содержит название, адрес. Класс Сеанс содержит дату, время, название.

## **Вариант 5**

Тема проекта: приложение «Календарь планирования мероприятий».

В приложении должно быть реализовано 2 класса: Календарь и Мероприятие. Класс Календарь содержит название помещения, адрес. Класс Мероприятие содержит дату, Название, Описание.

## **Вариант 6**

Тема проекта: приложение «Каталог кинофильмов».

В приложении должно быть реализовано 2 класса: Каталог и Фильм. Класс Каталог содержит название, год создания. Класс Фильм содержит название, жанр, описание.

### **Вариант 7**

Тема проекта: приложение «Библиотека».

В приложении должно быть реализовано 2 класса: Библиотека и Книга. Класс Библиотека содержит название, адрес. Класс Книга содержит название, жанр, автор.

### **Вариант 8**

Тема проекта: приложение «Отдел кадров».

В приложении должно быть реализовано 2 класса: Отдел и Работник. Класс Отдел содержит название, телефон. Класс Работник содержит имя, должность, дату приема.

### **Вариант 9**

Тема проекта: приложение «Управление пользователями».

В приложении должно быть реализовано 2 класса: Приложение и Пользователь. Класс Приложение содержит название, требуем объем. Класс Пользователь содержит логин, пароль, роль.

### **Вариант 10**

Тема проекта: приложение «Расписание занятий».

В приложении должно быть реализовано 2 класса: Расписание и Занятие. Класс Расписание содержит номер учебного заведения, адрес. Класс Занятие содержит название, день недели, время.

### **Вариант 11**

Тема проекта: приложение «Журнал учителя».

В приложении должно быть реализовано 2 класса: Журнал и Урок. Класс Журнал содержит название предмета, имя учителя. Класс Урок содержит дату, тему, номер класса.

### **Вариант 12**

Тема проекта: приложение «Учет клиентов».

В приложении должно быть реализовано 2 класса: Журнал регистрации и Клиент. Класс Журнал содержит название организации, телефон. Класс Клиент содержит имя, адрес, номер договора.

### **Вариант 13**

Тема проекта: приложение «Тестирование знаний».

В приложении должно быть реализовано 2 класса: Тест и Вопрос. Класс Тест содержит название предмета, сложность. Класс Вопрос содержит текст вопроса, ответ, количество баллов.

### **Вариант 14**

Тема проекта: приложение «Организация экскурсий».

В приложении должно быть реализовано 2 класса: Журнал регистрации и Экскурсия. Класс Журнал содержит организацию, контактный телефон. Класс Корреспонденция содержит дату, экскурсовод, описание.

### **Вариант 15**

Тема проекта: приложение «Телефонный справочник».

В приложении должно быть реализовано 2 класса: Справочник и Контакт. Класс Справочник содержит название, имя владельца. Класс Контакт содержит имя, телефон, адрес.

### **Итоговый отчет**

---

Укажите в отчете:

1. Тему лабораторной работы, свою фамилию и имя, группу, вариант задания;
2. Листинг программы;
3. Скриншоты результатов выполнения программы;
4. Краткие пояснения к алгоритму работы;
5. Выводы.

## Дополнительные вопросы

---

1. Можно ли определить несколько методов с одинаковым именем в одном классе?
2. Какие модификаторы доступа существуют и как они влияют на видимость данных и методов?
3. Как правильно организовать доступ к данным объекта?
4. Как обращаться к данным и методам объекта?
5. Можно ли объекту одного класса передать объект другого класса?