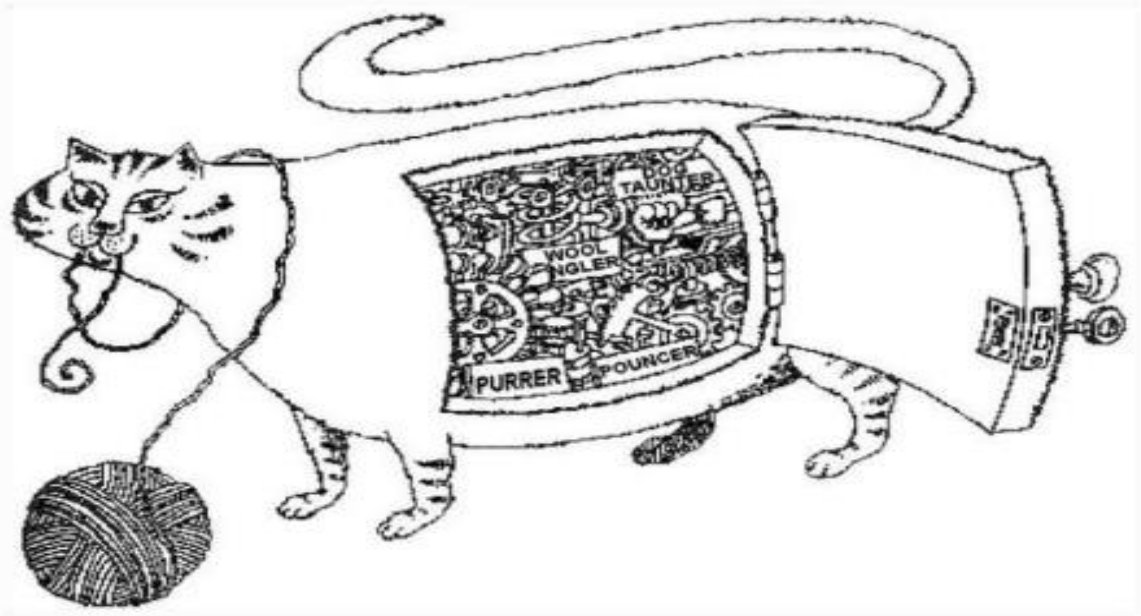
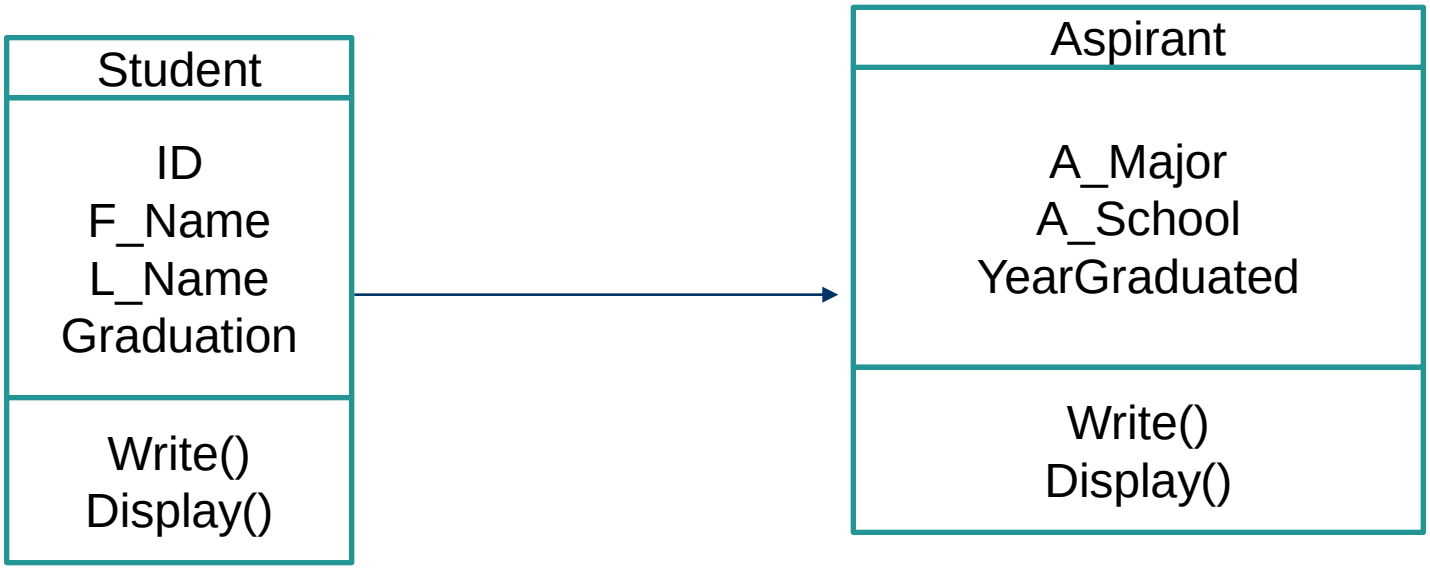
Инкапсуляция в C++

Тема 2.2 Пакетирование, наследование и полиморфизм

Инкапсуляция - способность объекта скрывать внутреннее устройство своих свойств и методов



Следование принципу инкапсуляции может уменьшить число связей между классами и упростить их независимую реализацию, модификацию и тестирование



Инкапсуляция позволяет

- **упростить интерфейс** класса, показав наиболее существенные для внешнего пользователя данные и методы.
- обеспечить **возможность внесения изменений** в реализацию класса без изменения других классов (важно для дальнейшего сопровождения и модернизации программного кода).

Инкапсуляция

```
class A
{
 public:
   int a, b; //данные открытого интерфейса
   int ReturnSomething(); //метод открытого интерфейса
 private:
   int Aa, Ab; //скрытые данные
   void Do_Something(); //скрытый метод
};
```

Объявление класса

- **Класс** – это определённый пользователем ***тип***. Определение класса задаёт *представление объектов этого класса* и *набор операций*, которые можно применять к таким объектам.

- Для объявления классов в C++ служит ключевое слово **class**
- Определение класса выглядит следующим образом:

```
class <имя класса>
{ <список членов класса> };
```

Private, protected, public

- **public, private и protected** — это модификаторы доступа
- **private** (закрытый, внутренний член класса) — обращения к члену допускаются только из методов того класса, в котором этот член определён.
- **protected** (защищённый, внутренний член иерархии классов) — обращения к члену допускаются из методов того класса, в котором этот член определён, а также из любых методов его классов-наследников.
- **public** (открытый член класса) — обращения к члену допускаются из любого кода.

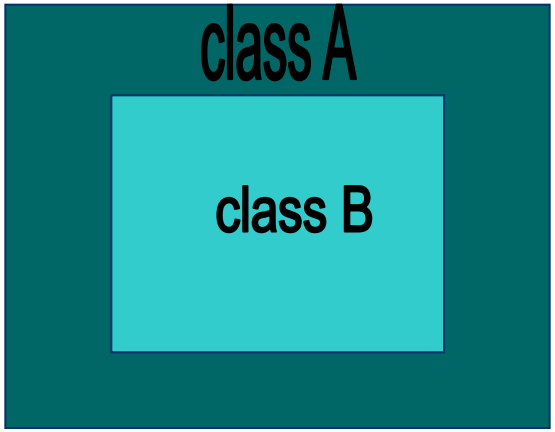
Private, protected, public

```
class student
  {
  private:
  /* список свойств и
  методов для использования
  внутри класса */
  public:
  /* список методов
  доступных другим функциям
  и объектам программы */
  protected:
  /*список средств,
  доступных при
  наследовании*/
  };
```

Доступ	private	protected	public
Сам класс	да	да	да
Друзья	да	да	да
Наследники	нет	да	да
Извне	нет	нет	да

Вложенные классы

- **Вложенный класс** (англ. *inner class*) — это класс целиком определённый внутри другого класса.




```
class A {
public:
    class B {
        int dd;
        int mm;
        int yy;
    public:
        B() { dd = 01; mm
  = 01; yy = 2000; };
        B(int d, int m,
  int y) { dd = d; mm =
  m; yy = y; };
    };
};
```

Неполное объявление класса

- Класс должен быть объявлен до использования его членов. Однако иногда в классе должен быть объявлен указатель или ссылка на объект другого класса еще до того, как он определен. В этом случае приходится прибегать к неполному объявлению класса

Неполное объявление класса

```
class PrevDecl ; // неполное объявление класса PrevDecl
   class AnyClass
   {
     int x;
     PrevDecl* obPtr;
   public:
     AnyClass ( int X ) { x = X ; } // конструктор класса
   AnyClass
   };
class PrevDecl // полное объявление класса PrevDecl
   {
     int y;
   public :
   PrevDecl ( ) ; // конструктор класса PrevDecl
   };
```