Объектно-ориентированное программирование на алгоритмическом языке C++

МИРЭА, Институт Информационных технологий,
кафедра Вычислительной техники

Автор: доцент, канд. физ.-мат. наук,
Путуридзе Зураб Шотаевич

Организация преподавания ООП

- Количество лекций - 8
- Самостоятельное опережающее освоение материала (ссылки в новостях АСО) + перечень рекомендуемых учебников.
- Методические пособия.
- Практические занятия – понимание и освоение возможностей алгоритмического языка (виртуальный объект, инкапсуляция, наследование, полиморфизм).
- Курсовая работа – версионность разработки, конструктивное построение программы как системы, работа с иерархией объектов, реализация сигналов и обработчиков.

Профессиональное программирование

- Производство программного продукта.
- Программный продукт – программный код и сопроводительная документация.
- Документация разработки.
- Документация эксплуатации.
- Производство всегда организуется посредством определенной технологии (способа производства).
- Готовым конструкторов, разработчиков.

Этапы разработки программного продукта

Процедурное программирование	Объектно-ориентированное программирование
Постановка задачи (Что?)	Постановка задачи. Определение цели. Требования. (Что?)
Метод решения (Чем?)	Набор объектов. (Чем?) Используемые методы решения.
Алгоритм решения задачи (Как?)	Архитектура программы-системы. Взаимодействие объектов. Алгоритм функционирования, решения задачи. (Как?)
Блок-схема алгоритма	Схема архитектуры системы, иерархии объектов. Схема взаимодействия объектов. Схема алгоритма решения задачи.
Код программы	Код описания объектов - классы. Код конструирования системы, иерархии объектов. Код построения схемы взаимодействия объектов. Код алгоритма решения задачи.
Тестирование и отладка	Тестирование и отладка
Доработка документации	Доработка документации
Сдача программы и сопроводительной документации	Сдача программы и сопроводительной документации

Автоматизированная система обучения «Аврора»

- Учебно-технологическая среда производства программного продукта.
- Повышение производительности труда студента и преподавателя.
- Автоматизация всех рутинных операции.
- Поддержка всех этапов разработки.
- Экономия (эффективное использование) основного ресурса – времени.
- Непрерывное совершенствование.

Жизненный цикл виртуального объекта

Реализация жизненного цикла виртуального объекта на языке C++

Описание	Класс
Создание	Отработка конструктора объекта. Выделение памяти и исходная инициализация.
Объект	Завершение работы конструктора объекта.
Старт	Начало функционирования.
Функционирование	Участие объекта в работе (в алгоритме) приложения
Остановка	Начало отработки деструктора объекта.
Демонтаж	Отработка деструктора объекта.
Завершение	Завершение работы деструктора объекта. Освобождение выделенной памяти.

Описание заголовочной части класса

```
class «имя класса» {
[private:]
        «список скрытых элементов класса»
public:
        «список доступных элементов класса»
protected:
        «список защищенных элементов класса»
};
```

- имя класса ::= идентификатор
- элемент класса ::= описание свойства (поля, переменной)
- ::= описание заголовка метода

Описание части реализации класса

«тип возвращаемого значения» «имя класса» :: «имя метода» ([список параметров])

{

// тело метода (код алгоритма метода)

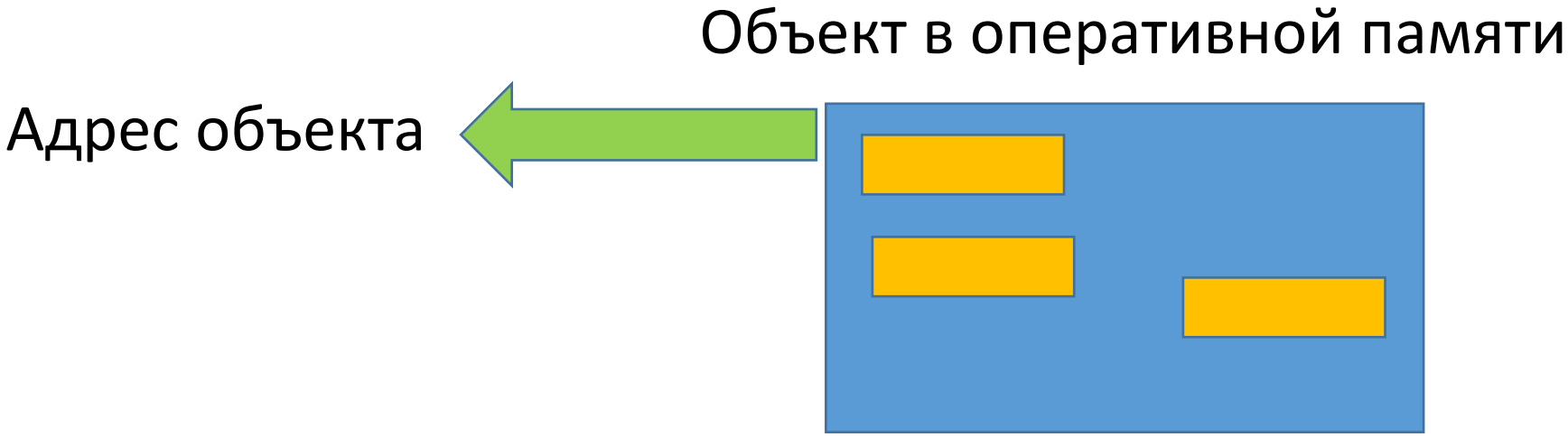
}

- имя класса – класс принадлежности метода

Отработка конструктора объекта

Конструктор

«имя класса» ([список параметров]);



Создание объекта

Объявление

«имя класса» «имя объекта» [,«имя объекта 1» …];

имя объекта ::= идентификатор

```
my_class  obj_1;
my_class  obj_2 ( “text” ), obj_3;
```

Динамическое создание объекта

```
my_class * p_obj_4 = new my_class ( );
```

Объявление объекта и доступ к его элементам

Доступ

«имя объекта».«имя элемента объекта» [([список аргументов])]

```
obj_1.i_numb = 15;
obj_1.m_vvod ( );
obj_1.m_calc ( i_n, s_oper );
```

Уничтожение объекта

Деструктор

~«имя класса» ( );

```
{

        return 0;
}

delete  p_obj_4;
```

Основная функция

```
#include <iostream>
using namespace std;

int main  ( )      // Точка входа в программу
{


   return  0;
}
```

Стандартный ввод и вывод в C++

```
#include <iostream>
using namespace std;

cout << выражение;          //   printf ( )

cin   >> переменная;        //   scanf  ( )
```

Пример 1

```
#include <iostream>
using namespace std;
 // ----- Заголовочная часть.
class   myclass {
   int a;
public:
      myclass ( ); // конструктор
      void show ( );
};
// ----- Часть реализации.
myclass :: myclass ( )
{
   cout << "В конструкторе \n";
   a = 10;
}
void  myclass :: show ( )
{
   cout << "a = " << a;
}
```

Пример 1

```
#include <iostream>
#include <locale>
using namespace std;
int main ( )
{
   setlocale ( LC_ALL, "Russian" );
   myclass  ob;  // объявление объекта, отработка конструктора
   ob.show ( );  // вызов открытого метода.
   return  0;
}
```

```
В конструкторе
10
```

Указатели и ссылки на объекты

```
#include <iostream>
using namespace std;
 // ----- Заголовочная часть.
class cl_1 {
private:
     int i;
public:
          cl_1   ( ); // конструктор
     void show_i ( );
     int  n;
};
 // ----- Часть реализации.
cl_1 :: cl_1 ()
{
    cout << "В конструкторе \n";
    i = 10;
}
void  cl_1 :: show_i ( ) {
    cout << “i = “ << i << “\n”;
}
```

Указатели и ссылки на объекты

```
#include <iostream>
#include <locale>
using namespace std;
int main ( )
{
        setlocale ( LC_ALL, "Russian" );
        cl_1   ob;              // объявление объекта
        cl_1 * p_ob;            // объявление указателя объекта заданного класса
        p_ob = & ob;            // присвоение к указателю объекта адреса объекта
        p_ob -> show_i ( );     // вывод значения свойства объекта
        return  0;
}
```

Программа выведет на консоль следующее:

```
В конструкторе
i = 10
```

Указатели и ссылки на объекты

В C++ ссылка – это простой ссылочный тип.

Объявление ссылки означает задание альтернативного идентификатора. По сути, ссылка это указатель, который привязан к определенной области памяти

```
cl_1   ob;          //объявление объекта, 0xdd000075
cl_1 & r_ob = ob; //альтернативное имя объекта, 0xdd000075
                    //объявление и инициализация
cout  << "\n" << & ob << "\n" << & r_ob;
```

Программа выведет на консоль следующее:

```
0xdd000075
0xdd000075
```

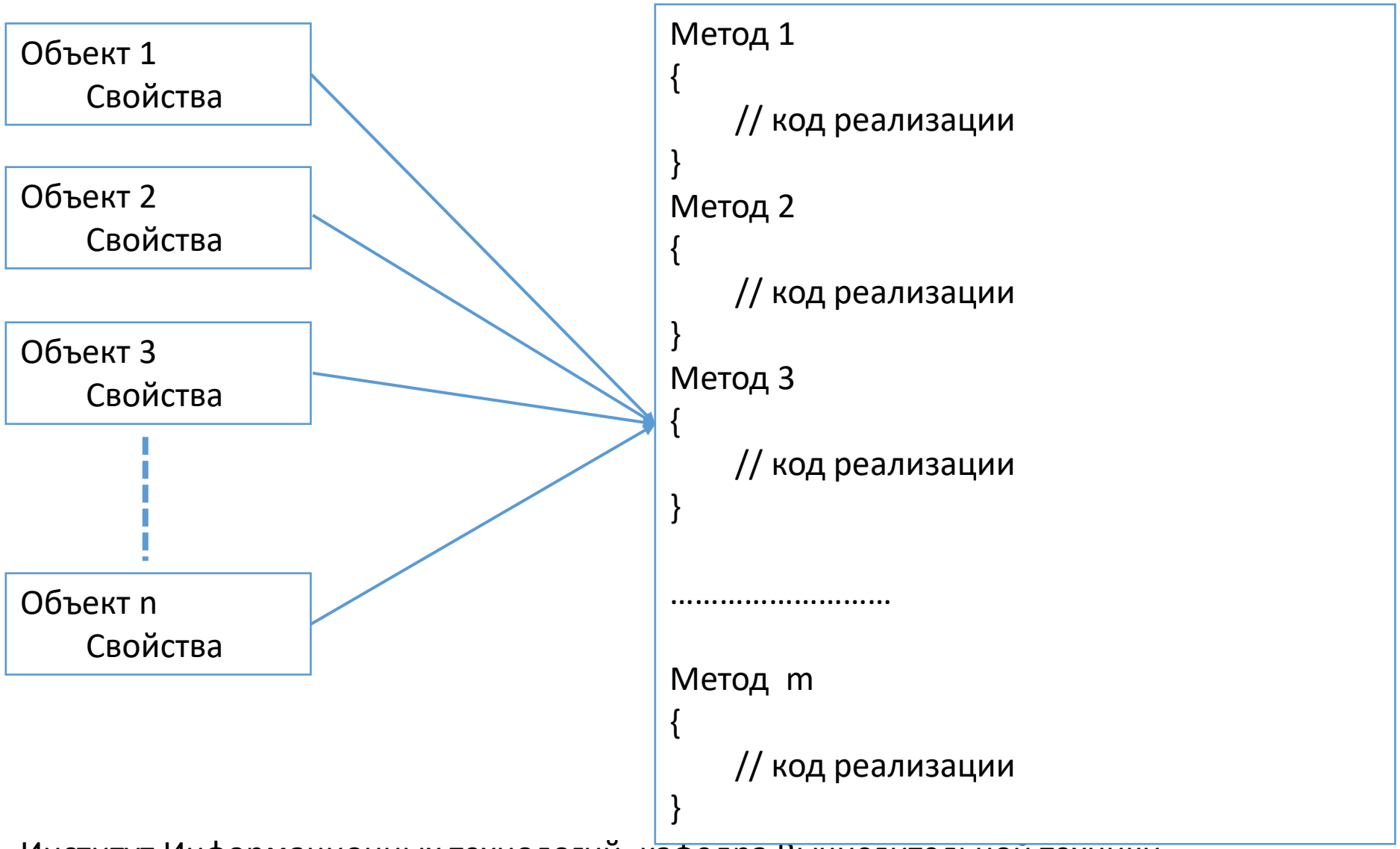
Указатели и ссылки на объекты

Оператор взятия адреса «&» используется для уже созданного объекта с целью получить его адрес, а ссылка это только задание альтернативного имени объекта.

```
cl_1   a;          // объявление объекта класса cl_1
cl_1   b;          // объявление объекта класса cl_1
cl_1 & ra =   a;   // объявление альтернативной имени объекта a и инициализация
cl_1 * p  = & ra; // объявление и инициализация указателя
a.n =   3;
b.n =   5;
cout << p -> n << "    " << ra.n << "\n";
p   = & b;
cout << p -> n << "    " << ra.n << "\n";

Программа выведет на консоль следующее:
В конструкторе
В конструкторе
3     3
5     3
```

Указатель this



Указатель this

```
#include <iostream>
using namespace std;
class cl_1 {
     int val;
public:
    void load_val ( int val );
    int  get_val  ( );
};
void cl_1 :: load_val ( int val )    {
     this -> val = val;
}
int cl_1 :: get_val ( )      {
     return this -> val; // тоже самое, что и return val;
}
```

Заветы Александра Невского

1. Уметь ладить с людьми любой национальности, не принижая своего достоинства, храня верность русскому миропониманию и обычаям.
2. Не преступать чужих границ, жить в ладу с соседями, а при необходимости принуждать их к миру.
3. Сберегать Святую Русь: народ, территорию и богатства души. Свято хранить связь времен и поколений.
4. Крепить духовную мощь Державы. Не в силе Бог, а в Правде – в праведной вере в особую духовную мощь русского народа.
5. Не преклоняться перед Западом. Не принимать его веры, его учения об искусстве наживать деньги, его безумного и безмерного стремления к потреблению.
6. Жить по Правде: в единстве слова и дела, в ладу с совестью, в разумном достатке.
7. Давать решительный отпор захватчикам, беспощадно карать предателей Отечества. Кто с мечом к нам придет – от меча и погибнет. На том стояла и стоять будет Русская земля!