Курс лекций "Программирование"
Основы программирования на языках C и C++

Лекция 1. Вводная

Глухих Михаил Игоревич, к.т.н., доц.
mailto: glukhikh@mail.ru

Состав курса (4 семестра)

- I. Основы программирования на языках C и C++
- II. Объектно-ориентированное программирование на языке C++
- III. Программирование на языке Java
- IV. Java-технологии

Другие курсы по программированию

- III или IV? Алгоритмы и структуры данных
- VI. Объектно-ориентированный анализ и проектирование
- X. Проектирование архитектур ПО
- IX. Концепции языков программирования
- VII. Основы программной инженерии
- VIII. Технологии тестирования ПО
- IX-X. Технологии разработки ПО
- X. Методы анализа и обеспечения качества программных систем

Приложения программирования

- VII, VIII. Операционные системы
- IX. Системное программирование
- X. Проектирование ОС
- VIII. Базы данных
- VI, VII, VIII. Компьютерные сети и телекоммуникации
- IX. Параллельные вычисления
- IX. Разработка графических приложений

Занятия в 1-м семестре

- Лекции, 3 часа в неделю
  - Презентации в свободном доступе – http://kspt.icc.spbstu.ru
  - Контрольные работы и/или домашние задания
- Упражнения, 2 часа в неделю
  - 2-3 тренировочных задания + 3-4 индивидуальных задания
  - Зачет по итогам упражнений (зачет/незачет)
  - Возможность получить 5 или 4 за экзамен
  - При наличии успехов – возможность поучаствовать в реальных проектах
- Экзамен (5/4/3/2)
  - Самостоятельное решение тестового задания на компьютере
  - Работающая программа = Положительная оценка

Программа лекций 1-го семестра

1. Основные конструкции и операции языка С (лекции 1-3).
2. Методы хранения данных в ЭВМ (лекция 4).
3. Сложные типы данных в языке С (лекции 5-6).
4. Процедурное и модульное программирование на языке С (лекции 7-9).
5. Основы объектно-ориентированного программирования на языке С++ (лекции 10-15).

Литература

- Т.А. Павловская С/С++. Программирование на языке высокого уровня.
- Д. Кнут. Искусство программирования.
- http://cpp.hut2.ru/ - сайт "Программирование на С++"
- Б. Страуструп. Язык С++. 4-е издание (автор языка С++).
- Е. В. Пышкин. Структуры данных и алгоритмы: реализация на С/С++ - книга есть на http://intranet.ftk.spbstu.ru
- В. Г. Давыдов. Программирование и основы алгоритмизации. Учебное пособие.
- Г. Шилдт. С++. Самоучитель.

Программирование

- Искусство создания компьютерных программ с помощью языков программирования
  - Анализ задачи (понятия, подходы, приемы, существующие решения)
  - Проектирование (определение структуры программы), разработка алгоритмов
  - Написание программы (кодирование)
  - Тестирование, отладка, сопровождение

Программирование

- Прикладное ~ обычные программы
  - Разработка GUI
  - Разработка бизнес-логики
- Системное ~ операционные системы, драйверы, компиляторы...
- Программирование баз данных
- Встроенное (embedded) / Мобильное (mobile)
- Сетевое
  - Клиент-серверное
  - Web-программирование

Алгоритм

- Последовательность действий, необходимая для решения определенной задачи
- Формы записи (на примере алгоритма Евклида)
  - Словесная (пошаговая)
  - Блок-схема, другие варианты схем
  - Программа

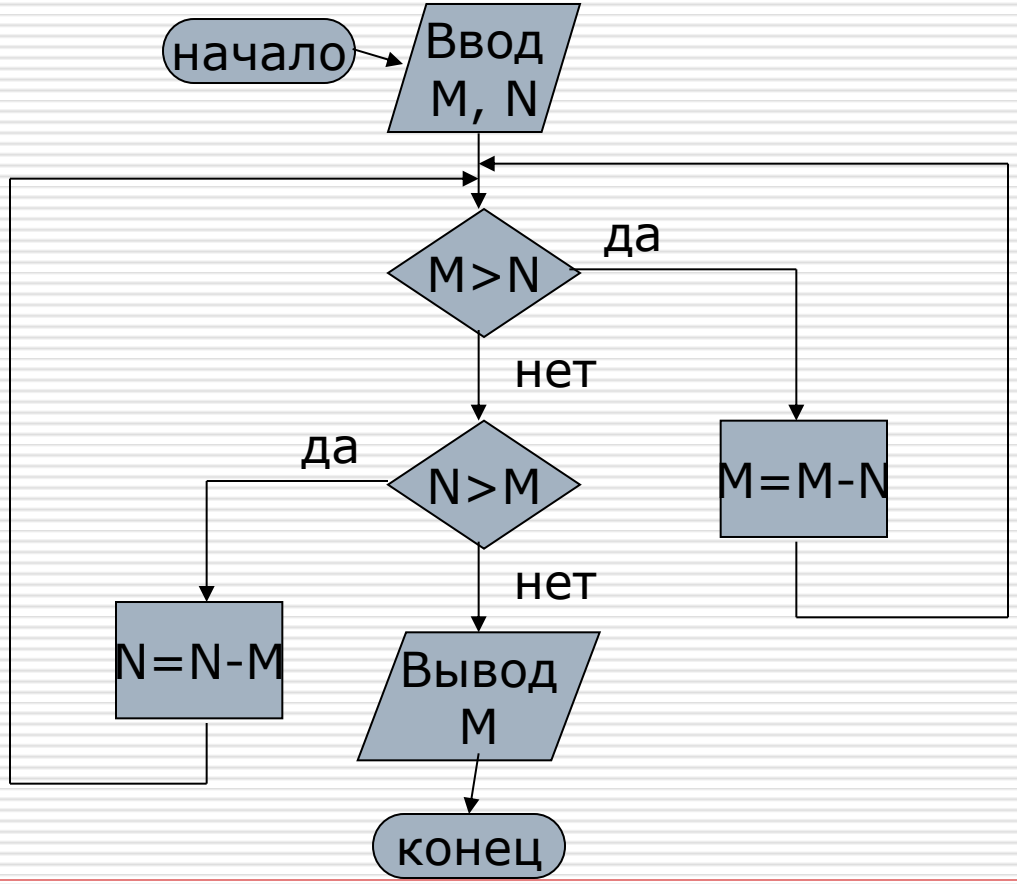
Пошаговая форма

1. Взять целые числа M и N.
2. Если M>N, уменьшить M на величину N и перейти к п. 5.
3. Если N>M, уменьшить N на величину M и перейти к п. 5.
4. Если M=N, перейти к п. 6.
5. Перейти к п. 2.
6. Результат (НОД) равен M.

Пример работы алгоритма

1. M=9, N=12.
2. 12>9. M=9, N=12-9=3.
3. 9>3. M=9-3=6, N=3.
4. 6>3. M=6-3=3, N=3.
5. 3=3. НОД=3.

Блок-схема



Программа

- Форма записи алгоритма, использующая правила языка программирования
- Языки программирования
  - Машинные коды
  - Языки низкого уровня (ассемблеры)
  - Языки высокого уровня

Языки высокого уровня

- Компилируемые и интерпретируемые
- Статическая и динамическая типизация
- Безопасные и небезопасные
- Императивные и декларативные
- Поддержка разных **парадигм** программирования
  - Процедурное программирование
  - Объектно-ориентированное программирование
  - Функциональное программирование
  - Логическое программирование
  - Обобщенное программирование

Рейтинг TIOBE (август 2014)

1. C – 16.4% (давно 1-й или 2-й)
2. Java – 15.0% (давно 1-й или 2-й)
3. Objective C – 9.6% (растёт)
4. C++ – 4.7% (падает)
5. Basic – 3.6% (~)
6. C# – 3.4% (падает)
7. Python – 3.1% (~)
8. PHP – 2.9% (падает)
9. Perl, Java Script, Visual Basic, Ruby, F#, Pascal, …

История языков программирования (вехи)

- Assembler ~1950
- Fortran ~ 1955
- Basic 1964 ➔ Visual Basic 1991
- Pascal 1970
- C 1973 ➔ C++, Objective C, C#
- Python 1991
- Java, Java Script, PHP 1995
- .NET языки 2001

История развития C/C++

- Создание C (лаборатория Bell Labs, 1973 год)
- Неформальная спецификация C (книга Кернигана и Ритчи, K&R C, 1978 год)
- Создание C++ Бьярном Страуструпом (1983 год)
- Создание Objective-C Бредом Коксом (1986 год)
- Формальная спецификация C (ANSI C или C89, 1989 год)
- Международный стандарт ISO для языка C++ (1998 год)
- Расширение языка C (стандарт C99, 1999 год)
- Второй стандарт ISO для языка C++ (2003 год)
- C++11 – новая версия стандарта (2011 год)
- C++14 – самая новая версия (18.08.2014)

Ключевые особенности C

- Императивный, компилируемый язык со статической типизацией
- Хорошо развитые низкоуровневые механизмы
  - Высокая эффективность (почти ассемблер)
  - Низкий уровень контроля (небезопасный язык)
- Часто используется в системном программировании (написание ОС, драйверов, компиляторов и пр.), встраиваемых системах (микроконтроллеры)
- Процедурное, структурное и модульное программирование
- Примеры приложений – ядро ОС Unix/Linux, сервер БД PostgreSQL

Ключевые особенности C++

- Поддерживает также объектно-ориентированный стиль, а в последних версиях и функциональный
- Почти полностью совместим с C
- Более удобен при написании программ среднего и большого размера
- При использовании ООП повышается уровень накладных расходов (однако, повышается и безопасность)
- Язык программирования общего назначения
- Примеры приложений – Mozilla Firefox, многие открытые программы для Unix/Linux

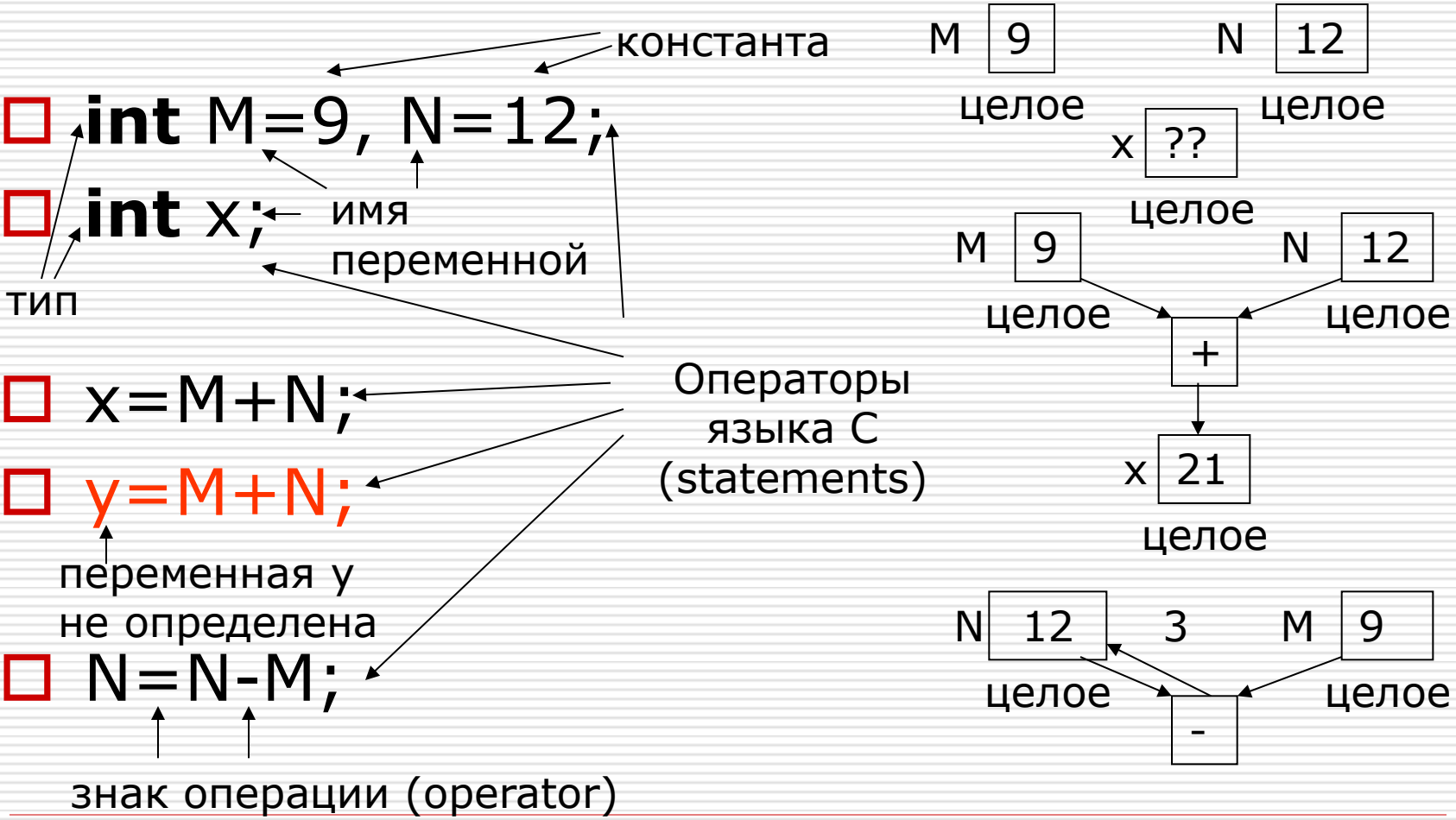
Основные реализации C и C++

- Microsoft Visual Studio (коммерческая)
  - версии 2005, 2008, 2010, 2013 годов
  - поддержка не только C++, но и других языков программирования (C#, Visual Basic, Java#, F#)
  - мощная система помощи MSDN
  - Free versions: Visual Studio Express 2010 / 2013
- Intel Parallel Studio (коммерческая)
- NetBeans (бесплатная)
- Eclipse (бесплатная)
- Qt Creator (бесплатная)
- GCC / MinGW (бесплатные компиляторы)
- Clang (бесплатный компилятор)

Структура программы на языках C и C++

- Программа состоит из одного или нескольких **файлов**, образующих **проект**
- Файлы содержат описания **данных** и **функций**
- Функция – участок программы, решающий часть задачи
- Главная функция – решает всю задачу (обычно с использованием других функций)
- Функции состоят из **операторов (statement)**, которые по умолчанию выполняются последовательно

Работа с данными, переменные



Имена (переменных)

- Имя в языках C и C++ всегда начинается с латинской буквы
- За ней может следовать неограниченное количество латинских букв, цифр и знаков подчеркивания
- Имя не может совпадать с ключевым словом (в частности, названия стандартных типов – такие как int – являются ключевыми словами)
- Языки C и C++ различают строчные и прописные буквы (alpha, ALPHA)

Примеры имен

- Корректные имена
  - alpha24
  - Red_Eye
  - canGo
  - b
- Некорректные имена
  - НОД (содержит русские буквы)
  - 3head (начинается с цифры)

Числовые типы C и C++, комментарии

```
int i, j; // Целые числа, от -2^31 до 2^31-1
/* Вещественные числа */
// Дв. точность, 13-14 знаков
double x;
// Од. точность, 6-7 знаков
float v, w;
// Логический, true(1)/false(0),
// только C++
bool ex;
```

Числовые константы C и C++

```
int i=1, j=-8, k=2562; // Целые
double x=0.0, y=45.9; // Вещественные
double b=-4e+12; // Вещ., -4*10^12
double s=3.2e-6; // Вещ., 3.2*10^-6
float z=-6.7f; // Вещ. одинарной точности
bool f1=true, f2=false; // Логические
```

Арифметические операции C и C++

```
int a=1, b, c=3, d; // Определение перемен-х
b=a+c; // Сложение, b=1+3=4
d=c-a; // Вычитание, d=3-1=2
a=b*d+2; // Умножение, a=4*2+2=10
a=b*(d+2); // Скобки, a=4*(2+2)=16
// !!! По правилам C/C++ int/int=int !!!
d=a/c; // Деление, d=10/3=3
d=d*(1/2); // !!! d=0 !!! т.к. 1/2=0
b=a%c; // Остаток от деления, b=10%3=1
```

Приоритеты

```
// Первыми выполняются операции в ()
// Затем выполняются *, /, % - слева направо
// Затем + и -, также слева направо
// Затем = (присваивание), справа налево
// Например
double x=3.6, y=2.8; // Определение переменных
x=y=(2.3+x)*(9.5-y)*(1.0/2);
// y=(2.3+3.6)*(9.5-2.8)*(1.0/2);
// y=5.9*6.7*0.5, (1.0/2) не 0,
// т.к. 1.0 – вещественное
// Итог: y=19.765, после чего x=y=19.765
```

Создание программы в среде Microsoft Visual Studio

1. Создание проекта (project) консольного приложения. При этом создается директория проекта, файл проекта и файл «решения» (solution).
2. Добавление нового C++ файла в группу исходных файлов (source files) проекта.
3. Написание текста программы.
4. Сборка проекта командой Build (осуществляется перевод программы на машинный язык).
5. Запуск программы командой Start without debugging (или командой Start debug).

Определение функции

```
// Главная функция всегда называется main
int main(void)
{
    // Результат равен 0, если программа
    // завершилась успешно, <0, если
    // произошла ошибка
    return 0;
}
```

тип исходных данных (аргументов)

тип результата

имя функции

Вывод на консоль средствами языка C++

```
// Подключение файла с описанием
// функций ввода-вывода C++
#include <iostream>
// Главная функция
int main(void) {
    // Вывод сообщения на экран (<< означает вывести)
    std::cout<<"Hello, world!"<<std::endl;
    return 0;
}
```

консоль

строковая константа

перевод строки

Windows-приложения – оконные и консольные

- Полноценное оконное приложение
  - Графический интерфейс (Окно, меню, рисунки, компоненты)
  - Классический ввод-вывод отсутствует
  - Более сложная организация (ответы на действия пользователя)
- Консольное приложение
  - Окно, в которое можно выводить текст (как пользователю, так и программе)
  - Графические возможности отсутствуют
  - Простая организация (непрерывное решение задачи)

Ввод с клавиатуры средствами языка C++

```
// Подключение файла с описанием
// функций ввода-вывода C++
#include <iostream>
// Главная функция
int main(void) {
   int a, b;
   // Вывод сообщения на экран (<< означает вывести)
   std::cout<<"Enter a, b: ";
   // Запрос a и b у пользователя (>> означает ввести)
   std::cin>>a>>b;
   // Вывод суммы на экран, в формате 2+3=5
   std::cout<<a<<"+"<<b<<"="<<a+b<<std::endl;
   return 0;
}
```

Для сравнения – та же программа средствами C

```
// Подключение файла с описанием
// функций ввода-вывода C
#include <stdio.h>
// Главная функция
int main(void) {
   int a, b;
   // Вызов функции вывода
   printf("Enter a, b: ");
   // Вызов функции ввода
   scanf("%d %d",&a,&b);
   // Вывод суммы на экран, в формате 2+3=5
   printf("%d+%d=%d",a,b,a+b);
   return 0;
}
```

Задача – определение времени прибытия поезда

- Поезд вышел со станции отправления 1-го числа в h1 часов m1 минут
- Время в пути h2 часов m2 минут
- Определить число и время (в часах и минутах) прибытия поезда на станцию назначения
- Ограничения: h1, m1, h2, m2 – целые неотрицательные числа, h1<24, m1<60, m2<60, поезд приходит на станцию назначения в том же месяце.

Анализ задачи – трудности

- Может выполняться m1+m2>59
  - Необходимо взять остаток от деления на 60. Результат деления нацело на 60 будет в часах (его придется добавить к сумме h1+h2).
  - Например: m1=35, m2=40.
  - (m1+m2)/60=1 час; (m1+m2)%60=15 минут.
- Может выполняться h1+h2>23
  - Аналогично, необходимо взять остаток от деления на 24. Результат деления нацело на 24 будет в днях, его нужно будет добавить к единице (поезд вышел в 1-й день).
  - Например: время отправления 14:35; время в пути 38:40; (h1+h2+1)/24=2 дня; (h1+h2+1)%24=5 часов.
  - Таким образом, поезд прибудет на 3-й день, в 5:15.

Написание программы

```
#include <iostream>
using namespace std; // чтобы не писать std:: каждый раз
int main(void) {
   int h1, m1, h2, m2;
   cout<<"Enter departure time (hh mm): ";
   cin>>h1>>m1;
   cout<<"Enter in-way time (hh mm): ";
   cin>>h2>>m2;
   int d3, h3, m3;
   m3=(m1+m2)%60;
   int hext=(m1+m2)/60;
   h3=(h1+h2+hext)%24;
   d3=1+(h1+h2+hext)/24;
   cout<<"Arrival time: day "<<d3<<", "<<h3<<":"<<m3<<endl;
   return 0;
}
```

Проверка работы

- Enter departure time (hh mm): 14 35
- Enter in-way time (hh mm): 38 40
- Arrival time: day 3, 5:15

Итоги

- Рассмотрели
  - числовые типы, переменные, константы
  - арифметические операции
  - главная функция
  - ввод-вывод
- Далее
  - операции, форматированный вывод
  - ветвление