НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++ (ч.1 - основы)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com
http://berkut.homelinux.com

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Компьютеры – для чего?

Компьютер — это универсальный инструмент, который есть у человечества.

Компьютеры:

- выполняют вычисления
- позволяют хранить огромное количество информации
- позволяют обмениваться информацией, независимо от местонахождения…

Проблемы?

- Чтобы автоматизировать даже самый простой процесс, необходимо компьютеру сказать чётко и недвусмысленно, что именно он должен делать.
- К сожалению наш язык и язык компьютера совершенно не похожи.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Языки программирования

- Между машиной и человеком есть серьёзный языковой барьер, который необходимо как-то преодолеть, иначе компьютер нас не поймёт.
- Поскольку компьютеры нас не понимают, они самостоятельно ничего делать не будут.
- Как средство общения между человеком и компьютером, придумано огромное количество **языков программирования**.

Языки программирования

- С помощью языков программирования, мы создаём программы и компьютер уже непосредственно работает с программами.
- **Программы** представляют собой **наборы инструкций**, которые **компьютер может понимать и выполнять**.

Типы программ

- Для эффективного общения с компьютером существует широкий спектр языков программирования (ЯП).
- В зависимости от типа проекта, существует множество факторов, которые необходимо учитывать при выборе ЯП.

Компиляция, интерпретация и JIT-компиляция

- Процесс компиляции переводит код написанный на языке программирования в родной язык целевой машины. Программа, выполняющая этот процесс называется компилятор.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Компиляция, интерпретация и JIT-компиляция

- Интерпретируемые языки программирования читаются программой под названием интерпретатор и выполняются этой же программой.

Компиляция, интерпретация и JIT-компиляция

- Компиляция на лету (или JIT-компиляция). Такие языки быстро компилируются, в момент запуска программы.
- Программы, написанные на JIT-языках, как правило, сильно оптимизируются, тем самым восстанавливается баланс между производительностью и кроссплатформенностью.

Высокий или низкий уровни программирования

- Низкоуровневые языки, в основном, работают непосредственно с оборудованием, и, следовательно, больше всего подходят для написания драйверов устройств.
- Низкоуровневые языки программирования почти всегда компилируются.

Высокий или низкий уровни программирования

- В языках высокого уровня все внимание уделяется концепции языка.
- Язык высокого уровня обычно легче понять, чем язык низкого уровня.
- Как правило, разработать программу на языке высокого уровня намного проще и быстрее, чем на языке низкого уровня.

Системы типов данных языков программирования

- Сильная система вводит ограничения на различные типы переменных, которые могут быть преобразованы друг к другу без явного преобразования.
- Идеальная система данных должна запрещать неявное преобразование «склейку» типов данных, в особенности, если, это не имеет никакого смысла.

Системы типов данных языков программирования

- Слабая система ввода не ставит никаких ограничений, за этим должен следить программист.
- Например, в таких языках можно попытаться выполнить умножение числа на строку или символ, хотя и результат умножения теряет всякий смысл, так как строку на число умножать нельзя.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Поддерживаемые парадигмы в программировании

- **Декларативная парадигма**
- **Функциональная парадигма**
- **Обобщённая парадигма**
- **Императивная парадигма**
- **Структурная парадигма**
- **Процедурная парадигма**
- **Объектно-ориентированная парадигма**

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Язык программирования C++

- C++ является ISO-стандартизированным ЯП
  - В течение некоторого времени, C++ не имел официального стандарта, однако с 1998 года, C++ был стандартизирован комитетом ISO
- C++ компилируемый язык
  - C++ компилируется непосредственно в машинный код, что позволяет ему быть одним из самых быстрых в мире языков
- C++ является строго типизированным ЯП
  - C++ подразумевает, что программист знает, что делает, и позволяет невероятное количество возможностей, ограниченных только лишь фантазией
- C++ поддерживает множество парадигм
  - C++ поддерживает процедурную, обобщённую, и объектно-ориентированную парадигмы программирования…
- C++ является полностью совместимым с языком Си
  - В C++ можно использовать Си библиотеки и они будут исправно работать.

Компиляторы C++

Компилятор — программа, транслирующая исходный (высокоуровневый) код программы в конечный (низкоуровневый) код.

Компиляция — процесс преобразования высокоуровневого исходного текста программы, в эквивалентный текст программы, но уже на низкоуровневом языке.

Компоновщик (Линкер) — программа, которая генерирует исполнимый модуль путём связывания объектных файлов проекта.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

IDE (Интегрированная среда разработки)

- **IDE** — сочетание текстового редактора и компилятора.
- Разработка, компиляция и запуск своих программы осуществляется непосредственно в IDE.
- Интегрированные среды разработки упрощают процесс составления программ, так как написание кода компиляция и запуск программ выполняются в одной программе — IDE.
- Ещё одной важной особенностью IDE является то, что IDE помогает быстро найти и исправить ошибки компиляции.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

IDE (Интегрированная среда разработки)

- **Code::Blocks с Mingw** - рекомендуемая, бесплатная IDE.
  - Code::Blocks также доступна на Linux.
- **Microsoft Visual Studio** — это хорошая среда разработки приложений под ОС Windows.
- **NetBeans IDE** – многоязычная (Java, C++, PHP и др.) кроссплатформенная среда разработки

Введение в язык C++

- Язык C++ представляет собой набор команд, которые говорят компьютеру, что необходимо сделать.
- Этот набор команд, обычно называется исходный код или просто код.
- Командами являются или «**функции**» или «**ключевые слова**».

Введение в язык C++

- Ключевые слова C/C++ являются основными строительными блоками языка.
- Функции являются сложными строительными блоками, так как записаны они в терминах более простых функций.
- Такая структура функций напоминает содержание книги.

Введение в язык C++

- Каждая программа в C++ имеет одну функцию, её называют главная или **main**-функция, выполнение программы начинается именно с этой функции.
- Из главной функции, вы также можете вызывать любые другие функции, неважно, являются ли они написанными нами, или, как упоминалось ранее, предоставляются компилятором.
- Чтобы получить доступ к стандартным функциям, которые поставляются с компилятором, необходимо подключить заголовочный файл используя препроцессорную директиву - **`#include`**.

Первая программа на C++

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

int main()
{
  cout << "Моя первая программа на C++\n";
}
```

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Первая программа на C++

```
#include <iostream>
#include <windows.h>

using namespace std;

int main()
{
  SetConsoleCP(1251);
  SetConsoleOutputCP(1251);
  cout << "Моя первая программа на C++\n";
}
```

Комментарии

- Добавляйте комментарии к коду, чтобы сделать его понятнее не только для себя но и для других.
- Однострочный комментарий
  // - сообщает компилятору, что остальная часть строки является комментарием
- Многострочный комментарий
  /* и затем */

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Объявление переменных в C++

Чтобы объявить переменную используется синтаксис тип <имя>;

```
int num;
char character;
float num_float;
```

Допустимо объявление нескольких переменных одного и того же типа в одной строке

```
int x, y, z, d;
```

Распространенные ошибки при объявлении переменных в C++

- Если вы попытаетесь использовать переменную, которую не объявили, ваша программа не будет скомпилирована, и вы получите сообщение об ошибке.
- В C++, все ключевые слова языка, все функции и все переменные чувствительны к регистру.

Использование переменных

```
#include <iostream>
#include <windows.h>

using namespace std;

int main()
{
   int number;

   SetConsoleCP(1251);
   SetConsoleOutputCP(1251);
   cout << "Введите число: ";
   cin >> number;
   cout << "Вы ввели: "<< number <<"\n";
}
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Типы данных в C++

Тип	Размер	Диапазон значений
целочисленный (логический) тип данных		
bool	1	0 / 255
целочисленный (символьный) тип данных		
char	1	0 / 255
целочисленные типы данных		
short int	2	-32 768 / 32 767
unsigned short int	2	0 / 65 535
int	4	-2 147 483 648 / 2 147 483 647
unsigned int	4	0 / 4 294 967 295
long int	4	-2 147 483 648 / 2 147 483 647
unsigned long int	4	0 / 4 294 967 295
long long	8	-9223372036854775807 / 9223372036854775807
типы данных с плавающей точкой		
float	4	3.4e-38 / 3.4e+38
long float	8	1.7e-08 / 1.7e+308
double	8	1.7e-08 / 1.7e+308

Изменение и сравнение величин

Независимо от того, какой тип данных вы используете, переменные не представляют особого интереса без возможности изменения их значения

Операция	Описание	Операция	Описание
+	Сложение	==	Равенство
-	Вычитание	>	Больше
*	Умножение	<	Меньше
/	Деление	!=	Не равно
%	Остаток	>=	Больше или равно
=	Присваивание	<=	Меньше или равно

Изменение и сравнение величин

```
a = 4 * 6; // а равно 24
a = a + 5; // равно сумме исходного значения и пяти
a == 5     // не присваивается 5, выполняется проверка, а равно 5 или нет
```

```
a < 5  // Проверка, а менее пяти?
a > 5  // Проверка, а больше пяти?
a == 5 // Проверка, а равно пяти?
a != 5 // Проверка, а неравно пяти?
a >= 5 // Проверка, а больше или равно пяти?
a <= 5 // Проверка, а меньше или равно пяти?
```

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Арифметические операции C++

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

int main()
{
    double sum, razn, pow, div; // объявление переменных через запятую
    double a1; // отдельное объявление переменной a1
    double a2; // отдельное объявление переменной a2
    cout << "Vvedite pervoe chislo: ";
    cin >> a1;
    cout << "Vvedite vtoroe chislo: ";
    cin >> a2;
    sum  = a1 + a2;  // операция сложения
    razn = a1 - a2;  // операция вычитания
    pow  = a1 * a2;  // операция умножения
    div  = a1 / a2;  // операция деления
    cout << a1 << "+" << a2 << "=" << sum  << endl;
    cout << a1 << "-" << a2 << "=" << razn << endl;
    cout << a1 << "*" << a2 << "=" << pow  << endl;
    cout << a1 << "/" << a2 << "=" << div  << endl;
    return 0;
}
```

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Математические функции

Заголовочный файл <cmath>

Функция	Описание	Пример
abs(a)	модуль или абсолютное значение от **a**	abs(-3.0)= 3.0 abs(5.0)= 5.0
sqrt(a)	корень квадратный из **a,** причём **a** не отрицательно	sqrt(9.0)=3.0
pow(a, b)	возведение **a** в степень **b**	pow(2,3)=8
ceil(a)	округление **a** до наименьшего целого, но не меньше чем **a**	ceil(2.3)=3.0 ceil(-2.3)=-2.0
floor(a)	округление **a** до наибольшего целого, но не больше чем **a**	floor(12.4)=12 floor(-2.9)=-3
fmod(a, b)	вычисление остатка от a/b	fmod(4.4, 7.5) = 4.4 fmod(7.5, 4.4) = 3.1
exp(a)	вычисление экспоненты e^a	exp(0)=1
sin(a)	**a** задаётся в радианах	
cos(a)	**a** задаётся в радианах	
log(a)	натуральный логарифм **a**(основанием является экспонента)	log(1.0)=0.0
log10(a)	десятичный логарифм **a**	Log10(10)=1
asin(a)	арксинус **a**, где **-1.0 < a < 1.0**	asin(1)=1.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Математические функции

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

int main()
{
    cout << "log10(10)      = " << log10(10.0)    << endl;
    cout << "log10(1)       = " << log10(1.0)     << endl;
    cout << "log(2.718281)  = " << log(2.718281)  << endl;
    cout << "sqrt(9)        = " << sqrt(9.0)      << endl;
    cout << "pow(2,3)       = " << pow(2.0,3.0)   << endl;
    cout << "abs(0)         = " << abs(0.0)       << endl;
    cout << "abs(-5)        = " << abs(-5.0)      << endl;
    cout << "ceil(3.14)     = " << ceil(3.14)     << endl;
    cout << "ceil(-2.4)     = " << ceil(-2.4)     << endl;
    cout << "floor(3.14)    = " << floor(3.14)    << endl;
    cout << "floor(-2.4)    = " << floor(-2.4)    << endl;
    cout << "fmod(2.4/2.0)    = " << fmod(2.4,2.0) << endl;
    return 0;
}
```

Демонстрация

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Операции присваивания в C++

- Для сокращённой записи выражений в языке программирования C++ есть специальные операции, которые называются операциями присваивания.
- Рассмотрим фрагмент кода, с использованием операции присваивания.

```
int value = 256;
value = value + 256;
// обычное выражение с использованием двух операций:  = и +
value += 256;
// сокращённое эквивалентное выражение
```

Операции присваивания в C++

В C++ существует пять операций присваивания, не считая основную операцию присваивания: =

- `+=` операция присваивания-сложения;
- `-=` операция присваивания-вычитания;
- `*=` операция присваивания-умножения;
- `/=` операция присваивания-деления;
- `%=` операция присваивания-остатка от деления

Операция	Обозначение	Пример	Экв.пример
операция присваивания-сложения	+=	var += 16	var = var + 16
операция присваивания-вычитания	-=	var -= 16	var = var - 16
операция присваивания-умножения	*=	var *= 16	var = var * 16
операция присваивания-деления	/=	var /= 16	var = var / 16
операция присваивания-остатка от деления	%=	var %= 16	var = var % 16

Спасибо!

Вопросы?

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ ІМЕНІ АДМІРАЛА МАКАРОВА

Алгоритмизация и программирование

Программирование на C/C++

(ч.1 – основы)

Беркунский Е.Ю., кафедра ИУСТ, НУК
eugeny.berkunsky@gmail.com,
http://berkut.homelinux.com