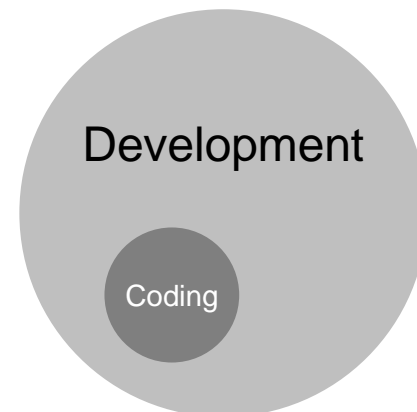


Разработка ПО систем управления

Весенний семестр 2018 г.

Кафедра управления и информатики НИУ «МЭИ»

В чем дело?



- Разработка ПО \gg написания кода.
 - Мышление:
 - **Общая задача** \rightarrow **конкретные задачи** \rightarrow алгоритмы \rightarrow код.
 - Разбираться: в коде, с библиотеками, с технологиями.
 - **Проектирование**: рационально связать части программы.
 - Инструментарий, не только для написания кода.
 - Механизмы работы программы и инструментов.
 - Язык: Pascal удобно учить, но нужен C++.
- Программирование — вторая грамотность! ([Ершов](#))
- ПО АС имеют специфику:
 - Жесткие требования к эффективности.
 - Высокая надежность и легкое сопровождение.

Что нужно*, кроме языка?

- Решать задачи самостоятельно и качественно.
 - Декомпозировать (разбивать на части).
 - Отлаживать и тестировать.
- Пользоваться популярными библиотеками.
 - Знать стандартную библиотеку.
 - Подключать библиотеки к программе.
- Работать в команде над большим проектом.
 - Код в **системе контроля версий** (Git и т. п.).
 - **Процесс сборки** сложный, нужно понимать его.
- Изучать конкретные технологии и библиотеки.
 - Привычка, навык искать и **читать документацию**.
 - Актуальная информация — **на английском**.

* Источники:



План курса

- Основы языка C++
- Системы контроля версий (Git)
- Декомпозиция задачи
- Динамическая память
- Процесс и инструменты для сборки (make, CMake)
- Подключение библиотек
- Качество кода, отладка, тестирование
- Работа с двоичными данными
- Объектно-ориентированное программирование

Нужное

мышление

инструментарий

механизмы

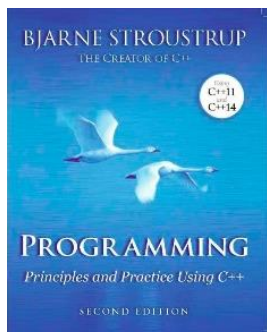
язык

проектирование

Учебный процесс

- Лекции: раз в 2 недели, пропадает 09.05, итого 7 шт.
- Лабораторные работы: 4 шт. + защиты, раз в 2 недели
- Зачет с оценкой.
- <http://uii.mpei.ru/study/courses/cs> — всё,
 - но потом :-)
- Преподаватели:
 - Мохов Андрей Сергеевич (Лекции, лабораторные работы)
 - Яблоков Иван Олегович (Лабораторные работы)

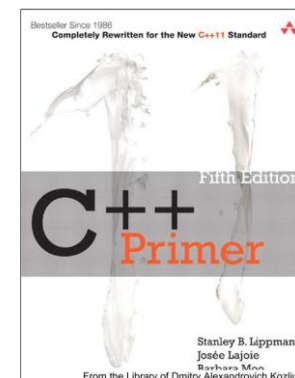
Литература (больше на сайте)



- ❑ Bjarne Stroustrup. *Programming: Principles and Practices using C++* (2014 или 2011):

Страуструп Б. Программирование: принципы и практика использования C++. — СПб., «Вильямс». — 2012 г.

- ❑ Stanley B. Lippman. *C++ Primer*, 5th Ed. Липман С. C++: базовый курс, 5 изд.



[стандарт]

- ❑ ISO/IEC 14882—2014 (Draft)
<http://open-std.org/JTC1/SC22/WG21>
 - Сайт «C++ Reference»

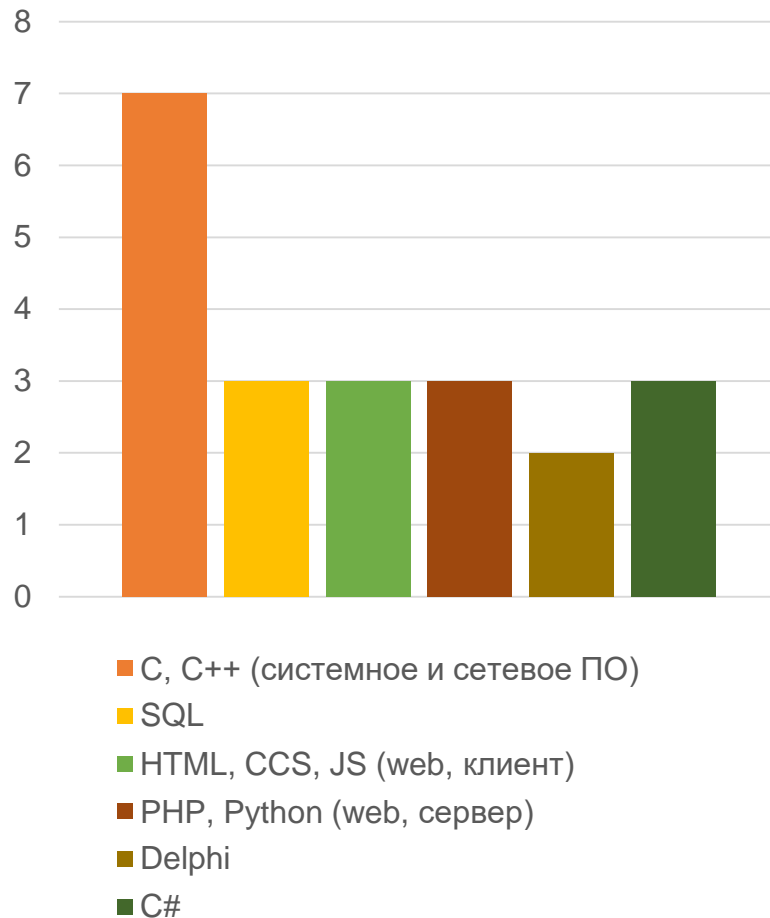
[K&R]

- ❑ Brian Kernighan & Dennis Ritchie. *The C Programming Language*, 2nd Edition.
Керниган Б., Ритчи Д. Язык программирования C.

Основы языка C++

Почему C++?

Выпускники используют
(2015)



Почему не Python/SQL/?..

Системы промышленной автоматизации:

- низкого уровня:
 - встраиваемые (спец. устройства);
 - системное ПО (драйверы, ...);
- реального времени (производительные);
- сложные, но надежные.

Привлекательность C++

1. Zero-overhead abstractions

2. Maps directly to hardware

- 1 + 2 = программы одновременно понятны и быстры.
- Совместимость:
 - высокая совместимость с языком C;
 - обратная совместимость (backwards compatibility).
- Гибкое управление ресурсами:
 - доступ к выделению памяти и т. п. вручную;
 - возможность автоматического управления ресурсами.
- Солидная поддержка, стандартизация.


Структура программы на C++

```
1.  #include <iostream>
2.  using namespace std;
3.  int main()
4.  {
5.      // Вывод строки на экран.
6.      cout << "Hello, world!\n";
7.      return 0;
8.  }
```

Типы данных и переменные

Тип данных определяет формат данных и операции над значением.

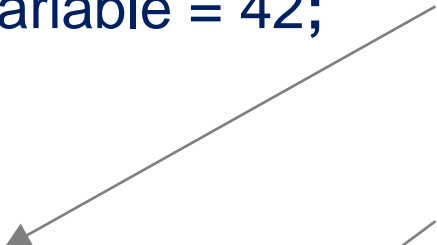
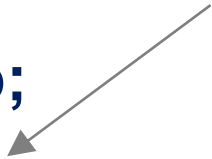
Можно и желательно указывать начальное значение.



```
int negative = -42;           // целое со знаком
unsigned int count = 66;      // целое без знака
double pi = 3.14159265358979; // действительное
char letter = 'A';           // символ
string name = "Dmitry";      // строка
bool condition = true;       // логическое значение
```

См. подробности в раздаточном материале.

Неизменяемые переменные

- Переменные, которые нельзя изменить.
 - Значение должно быть задано сразу.
- **const int variable = 42;**
 - **int const variable = 42;**  Может зависеть от переменных.
- **int a = 10;**
int b = 20;
int const sum = a + b; 
 - Не может зависеть от переменных и **const**.
 - Может зависеть от **constexpr**.
- **double constexpr PI = 3.14;**
 - Только для констант этапа компиляции (π , e , ...).

ВВОД И ВЫВОД

1. `#include "sdt.h"`

2.

3. `int main()`

4. `{`

5. `cout << "Enter your name and age: ";`

6. `string name;`

7. `unsigned int age;`

8. `cin >> name >> age;`

9. `const unsigned int next = age + 1;`

10. `cout << "Hello, " << name << ", next year "`

11. `<< "you will be " << next << " years old.\n";`

12. `}`

dmitry@lab: ~/mpei/c++2014

Enter your name and age: **Shepard 34**

Hello, Shepard, next year you will be 35 years old.

Операторы и выражения

- Присваивание: `n = 5;` `k = p = 0;`
 - До присваивания значение не определено!
- Сокращенное присваивание:
`q += 5; // q = q + 5`
`v *= 2; // v = v * 2`
- Инкремент и декремент:
`int i = 5;`
`++i; // 6, i == 6`
`i--; // 6, i == 5`
- Остаток от деления: `a % b`
- Проверка равенства: `x == y`
 - неравенства: `x != y`
- Логические операции:
 - «И»: `&&`
 - «ИЛИ»: `||`
 - «НЕ» `!`
 - `n < 20 && (n == 2 || n > 9)`



`9 / 5 == 1`
`9.0 / 5 == 1.8`

Проверки условий

❑ **if** (*условие*)
 действие-1
else
 действие-2

❑ **if** (*условие-1*)
 действие-1
else if (*условие-2*)
 действие-2
else
 действие-3

❑ **if** (*t == 100*) {
 c++;
} **else**
 *t *= 2*;



❑ **if** (*x % 2*) {
 cout << "odd";
} **else if** (*x % 3*) {
 q++;
} **else** {
 cout << x << '\n';
}

Циклы **while** и **do...while**

C++

- ☐ **while** (*условие*)
тело цикла
- ☐ **do**
тело цикла
while (*условие*);
- ☐ **break**;
- ☐ **continue**;

Pascal, Delphi

- ☐ **while** *условие* **do**
тело цикла
- ☐ **repeat**
тело цикла
until not *условие*;
- ☐ **Break**;
- ☐ **Continue**;

Цикл **for**

for (*инициализация; условие; действие*)
тело цикла

```
for (int i = 0; i < 10; ++i)  
    cout << i << '\n';
```

Подобно коду на C++...

```
int i = 0;  
while (i < 10) {  
    cout << i << '\n';  
    ++i;  
}
```

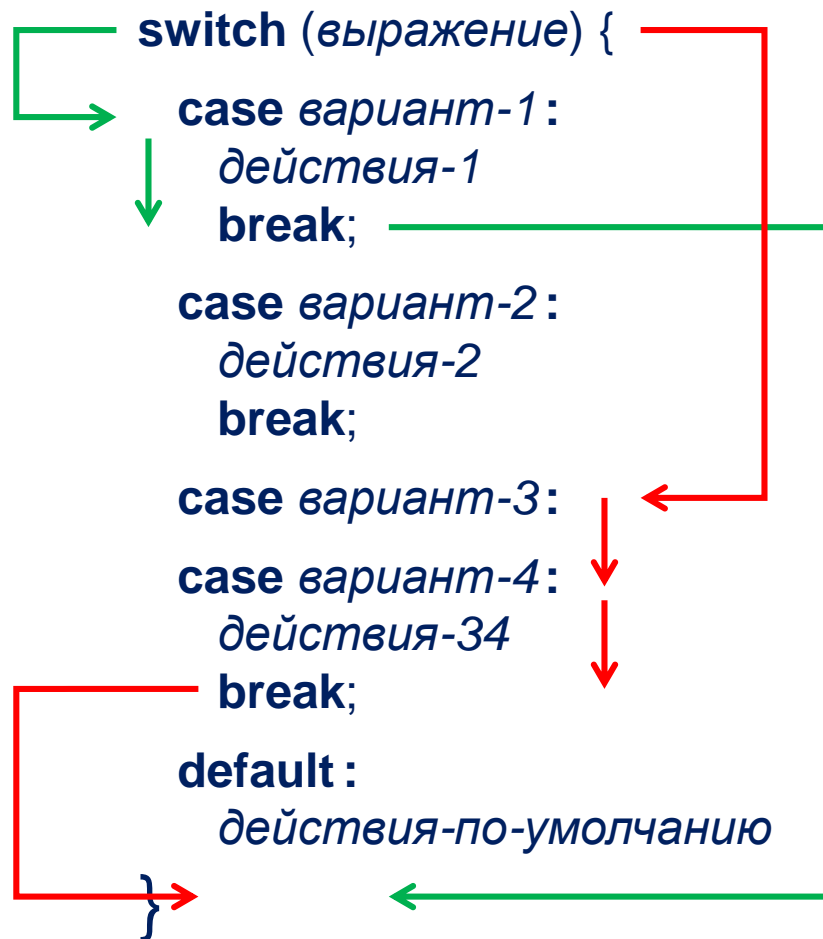
Подобно коду:

```
инициализация;  
while (условие) {  
    тело цикла;  
    действие;  
}
```

...и на Pascal:

```
for I := 0 to 9 do  
    WriteLn(I);
```

Переключатель (switch)

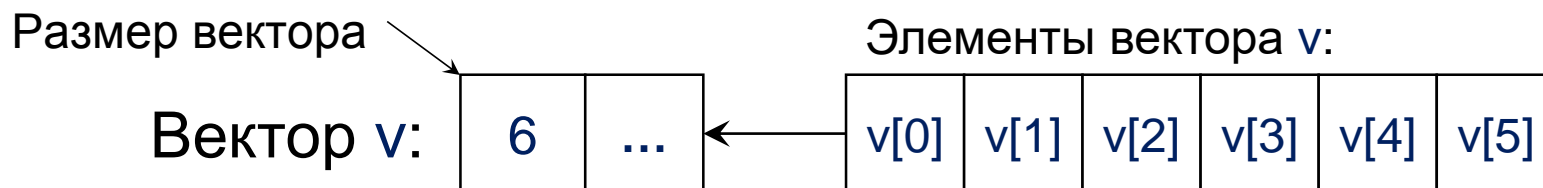


```
char answer;  
cout << "Yes or No? ";  
cin >> answer;  
  
switch (answer) {  
    case 'y':  
    case 'Y':  
        // Ответ «Да».  
        break;  
    case 'n':  
    case 'N':  
        // Ответ «Нет».  
        break;  
    default:  
        cout << "Type Y or N!";  
}
```

Тип `vector<T>`, «вектор»

«Динамический массив» элементов типа `T`.

Для использования подключаем заголовочный файл `<vector>`



В круглых скобках передаются значения.

```
vector<double> v(10);
```

В угловых скобках «передаются» типы.

Смысл:

«выполнить *операцию* с вектором `v`, используя *аргументы*».

`v.операция(аргументы);`

Использование `vector<T>`

- Вектор из чисел:

```
vector<double> numbers;
```

- Вектор из 5 чисел:

```
vector<double> numbers(5);
```

```
vector<double> numbers {1, 2, 3, 4, 5 };
```

- Вектор из 5 нулей:

```
vector<double> zeros(5, 0);
```

- Третье число (нумерация элементов с 0):

```
double third = numbers[2];
```

- Узнать размер:

```
unsigned int size = numbers.size();
```

Изменение `vector<T>`

- Изменить элемент:
`numbers[3] = 9;`
- Добавить элемент в конец:
`numbers.push_back(42.99);`
- Удалить последний элемент:
`numbers.pop_back();`
- Изменить размер до 10 элементов:
`numbers.resize(10);`
- Очистить вектор:
`numbers.clear();`

Статистические расчеты

```
vector<double> xs;
```

```
int n;
```

```
cin >> n;
```

```
xs.resize(n);
```

```
for (int i = 0; i < n; ++i)
```

```
    cin >> xs[i];
```

```
double mean = 0;
```

```
for (double x : xs)
```

```
    mean += x;
```

```
mean /= xs.size();
```

```
double min = xs[0];
```

```
double max = xs[0];
```

```
for (double x : xs) {
```

```
    if (x < min) {
```

```
        min = x;
```

```
    } else if (max < x) {
```

```
        max = x;
```

```
    }
```

```
}
```

Литература к лекции

- *Programming Principles and Practices Using C++:*
 - глава 1: о программировании в целом;
 - главы 2—4: типы данных, выражения, конструкции;
 - упражнения (drills).
- *C++ Primer:*
 - глава 1: пошаговое написание программы;
 - главы 2—5: типы данных, выражения, конструкции;
 - упражнения.
- *C++ Reference* (<http://cppreference.com>)