

# Структуры данных в C++

Материалы для занятия в кружке информатики ФИМЛИ.

Руководитель кружка Грицуляк Р.Т.  
21.09.2016

# Базовые типы

## Ключевые слова

- void, char, wchar\_t
- short, int, long
- float, double

## Сложные типы

- short int, unsigned short int, long long, etc

# Базовые типы

Type specifier	Equivalent type	Width in bits by data model				
		C++ standard	LP32	ILP32	LLP64	LP64
short	short int	at least 16	16	16	16	16
short int						
signed short						
signed short int						
unsigned short	unsigned short int					
unsigned short int						

# Базовые типы

int	int	at least 16	16	32	32	32
signed						
signed int						
unsigned	unsigned int					
unsigned int						

# Базовые типы

long	long int	at least 32	32	32	32	64
long int						
signed long						
signed long int						
unsign ed long	unsign ed long int					
unsign ed long int						

# Базовые типы

long long	long long int (C++11)	at least <b>64</b>	<b>64</b>	<b>64</b>	<b>64</b>	<b>64</b>
long long int						
signed long long						
signed long long int						
unsigned long long	unsigned long long int (C++11)					
unsigned long long int						

# Float

`float` — тип с плавающей точкой одинарной точности. Обычно 32-битный тип с плавающей точкой формата IEEE-754

`double` — тип с плавающей точкой двойной точности. Обычно 64-битный тип с плавающей точкой формата IEEE-754

`long double` — тип с плавающей точкой повышенной точности. Не обязательно отображается на типы IEEE-754. Обычно 80-битный тип с плавающей точкой формата x87 на архитектурах x86 и x86-64

# Базовые типы



floating point	32	IEEE-754	$\pm 3.4 \cdot 10^{\pm 38}$ (~7 digits)	min subnormal: $\pm$ <b>1.401,298,4</b> · <b><math>10^{-47}</math></b> min normal: $\pm$ <b>1.175,494,3</b> · <b><math>10^{-38}</math></b> max: $\pm$ <b>3.402,823,4</b> · <b><math>10^{38}</math></b>
	64	IEEE-754	$\pm 1.7 \cdot 10^{\pm 308}$ (~15 digits)	min subnormal: $\pm$ <b>4.940,656,45</b> <b>8,412</b> · <b><math>10^{-324}</math></b> min normal: $\pm$ <b>2.225,073,85</b> <b>8,507,201,4</b> · <b><math>10^{-308}</math></b> max: $\pm$ <b>1.797,693,13</b> <b>4,862,315,7</b> · <b><math>10^{308}</math></b>



# Базовые типы

- Типы для строк:
- Char — 8 bit , wchar\_t
- wchar\_t — windows 16 bit, linux 32
- char16\_t UTF-16
- char32\_t UTF-32

# Гарантии по базовым типам

```
1 == sizeof(char)
```

```
<= sizeof(short)
```

```
<= sizeof(int)
```

```
<= sizeof(long)
```

```
<= sizeof(long long)
```

# НОВЫЕ ТИПЫ

- Рекомендуется использовать из первой строки, тестировать с второй строкой

<code>int8_t</code> <code>int16_t</code> <code>int32_t</code> <code>int64_t</code> (optional)	signed integer type with width of exactly 8, 16, 32 and 64 bits respectively with no padding bits and using 2's complement for negative values (provided only if the implementation directly supports the type) (typedef)
<code>int_fast8_t</code> <code>int_fast16_t</code> <code>int_fast32_t</code> <code>int_fast64_t</code>	fastest signed integer type with width of at least 8, 16, 32 and 64 bits respectively (typedef)

# Больше информации

- Страуструп: язык программирования c++ (4 издание)

<http://en.cppreference.com/w/cpp/types/integer>

# Структуры

- Ключевые слова `struct`, `class` — задают тип данных.
- `struct a_struct { int a; int b;};`
- `class a_class {public: int a; int b;};`
- Параметризованные:
- `vector<int>`

# Параметризация: пример

```
struct Ptr2Pair2ndGT {  
    template<typename It>  
    bool operator()(It it1, It it2) const { return it1-  
        >second > it2->second; }  
};
```

# Библиотека c++

- Последовательные контейнеры
- `array<тип>`
- `vector<тип>`
- `deque<тип>`
- `forward_list<тип>`
- `list<тип>`

# Адаптеры

- `stack`
- `queue`
- `priority_queue`

«оболочки» вокруг типов «оборачиваемых» адаптером.

Имеют специальные требования к типам оборачиваемых элементов (должны поддерживать определенные операции)



# Ассоциативные контейнеры

- Set, unordered\_set
- Multiset, unordered\_multiset
- Map, unordered\_map
- Multimap, unordered\_multimap
- Доступ к значению по ключу.

# Задание

- Численный эксперимент и исследование стандартных контейнеров и адаптеров:
- Выбрать контейнер
- Использовать стандартные и быстрые типы.
  - Определить сколько памяти занимает при 10 м элементов.
  - Сколько времени занимает добавление 10м элементов.
  - Удаление элементов.
  - Поиск элементов.
  - Другие операции.
  - представить программу и таблицей результатов.