

Концепты C++17 в их отношении к концептам C++0x

Пеленицын Артём Михайлович

ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ (Ростов-на-Дону)
Институт математики, механики и компьютерных наук им. И. И. Воровича
Кафедра информатики и вычислительного эксперимента

Современные информационные технологии
и ИТ-образование 2015



Мотивация к явным ограничениям в шаблонах

Стандартная сортировка стандартного списка

```
std::list l{2, 1, 3};  
std::sort(l.begin(), l.end());
```

Текст ошибки: 67 строк (GCC 4.9)

*no match for 'operator-' (operand types are
'std::_List_iterator<int>' and 'std::_List_iterator<int>')*

Причина: ISO/IEC 14882:1998, с. 555

```
template<class RandomAccessIterator>  
void sort(RandomAccessIterator first,  
          RandomAccessIterator last);
```



То же, но с концептами C++1z (Concepts Lite)

Заголовок функции сортировки с концептом `SortableIt`

```

template<SortableIt It>
void sort(It first, It last);

```

Текст ошибки (GCC 6.0, <http://melpon.org/wandbox/>)

```

error: cannot call function
      'void sort(It,It)
           [with It =
                std::_List_iterator<int>]'
      sort(l.begin(), l.end());

note: concept
      'SortableIt<std::_List_iterator<int>>',
      was not satisfied

```



Что включал проект концептов C++0x

- 1 Синтаксис для определения именованного набора ограничений – концепта.
- 2 Синтаксис для наложения ограничений (концепта) при объявлении шаблона класса или функции.
- 3 Способ проверки соответствия использования шаблонных параметров внутри шаблона класса или функции наложенным ограничениям.
- 4 Способ указания соответствия конкретных типов тем или иным концептам – так называемые модели концептов (concept map).



Что включал проект концептов C++0x

- 1 Синтаксис для определения именованного набора ограничений – концепта.
- 2 Синтаксис для наложения ограничений (концепта) при объявлении шаблона класса или функции.
- 3 Способ проверки соответствия использования шаблонных параметров внутри шаблона класса или функции наложенным ограничениям.
- 4 Способ указания соответствия конкретных типов тем или иным концептам – так называемые модели концептов (concept map).



Синтаксис для наложения ограничений при объявлении шаблона

Полная форма для (template<SortableIt It>)

```
template<class It> requires SortableIt<It>  
void sort(It first, It last);
```

Несколько ограничений

```
template<typename T, typename A>  
  requires Object<T>() && Allocator<A>()  
class vector;
```



Синтаксис определения концептов (1)

Ассоциированные типы и ограничения на них

```
template<typename A>
concept bool Allocator()
{
    return requires () {
        typename A::pointer;
        requires Pointer<typename A::pointer>;
        // ...
    };
}
```



Синтаксис определения концептов (2)

Использование допустимых выражений

```
template<typename T>  
concept bool Equality_comparable() {  
  
    return requires (T a, T b) {  
        {a == b} -> bool;  
        {a != b} -> bool;  
    };  
  
}
```



Использование C++14 (шаблоны переменных)

```
template<typename I>  
concept bool InputIterator  
    = WeakInputIterator<I>()  
    && EqualityComparable<I>()  
    && Derived<IteratorCategory<I>,  
              input_iterator_tag>();
```

