

pair

Класс pair (пара) стандартной библиотеки C++ позволяет нам определить одним объектом пару значений, если между ними есть какая-либо семантическая связь. Эти значения могут быть одинакового или разного типа. Для использования данного класса необходимо включить заголовочный файл:

```
#include <utility>
```

Например, инструкция

```
pair< string, string > author( "James", "Joyce" );
```

создает объект author типа pair, состоящий из двух строковых значений. Отдельные части пары могут быть получены с помощью членов first и second:

```
string firstBook;  
if ( author.first == "James" &&  
    author.second == "Joyce" )  
    firstBook = "Stephen Hero";
```

Если нужно определить несколько однотипных объектов этого класса, удобно использовать директиву typedef:

```
typedef pair< string, string > Authors;  
Authors proust( "marcel", "proust" );  
Authors joyce( "James", "Joyce" );  
Authors musil( "robert", "musil" );
```

Вот другой пример употребления пары. Первое значение содержит имя некоторого объекта, второе – указатель на соответствующий этому объекту элемент таблицы.

```
class EntrySlot;  
extern EntrySlot* look_up( string );  
typedef pair< string, EntrySlot* > SymbolEntry;  
  
SymbolEntry current_entry( "author", look_up( "author" ));  
// ...  
if ( EntrySlot *it = look_up( "editor" ))  
{  
    current_entry.first = "editor";  
    current_entry.second = it;  
}
```

Пример.

```
#include <utility>
#include <iostream>

using namespace std;

/* STL pair data type containing int and float
*/

typedef struct pair<int,float> PAIR_IF;

int main(void)
{
    PAIR_IF pair1=make_pair(18,3.14f);

    cout << pair1.first << " " << pair1.second << endl;
    pair1.first=10;
    pair1.second=1.0f;
    cout << pair1.first << " " << pair1.second << endl;
}
```