

# STL

Тема 2. Обобщенные алгоритмы.  
Часть 4. Обобщенные числовые алгоритмы

# Обобщенные числовые алгоритмы

- ❑ **accumulate**
- ❑ **partial\_sum**
- ❑ **adjacent\_difference**
- ❑ **inner\_product**

# accumulate

Задача: Суммировать (выполнить произвольную операцию заданную предикатом) со всеми элементами из заданного диапазона.

```
vector<int> v1, v2(20);  
  
for (int i = 1; i < 21; i++) v1.push_back(i);  
  
int total;  
total = accumulate(v1.begin(), v1.end(), 0);  
  
int ptotal;  
ptotal = accumulate(v1.begin(), v1.end(), 1, multiplies<int>());
```

# partial\_sum

Задача: Для последовательности  $\{X_n\}$  хранящейся в контейнере вычисляет последовательность  $\{\sum X_n\}$

```
vector<int> V1( 10 );  
list <int> L1;  
  
for (int t = 1 ; t <= 10 ; t++ ) { L1.push_back( t ); }  
  
vector<int>::iterator Viterend;  
Viterend = partial_sum ( L1.begin ( ) , L1.end ( ) , V1.begin ( ) );
```

# adjacent\_difference

Задача: Для последовательности  $\{X_n\}$  хранящейся в контейнере вычисляет последовательность  $\{X_n - X_{n-1}\}$

```
int x1[20], x2[20];  
for (int i = 0; i < N; ++i)  
    x1[i] = i;  
  
partial_sum(&x1[0], &x1[N], &x2[0]);  
  
adjacent_difference(&x2[0], &x2[N], &x2[0]);
```

# inner\_product

Задача: Получить скалярное произведение двух диапазонов

```
int x1[5], x2[5];
for (int i = 0; i < N; ++i)
{
    x1[i] = i + 1;
    x2[i] = i + 2;
}

int result = inner_product(&x1[0], &x1[N], &x2[0], 0);

result = inner_product(&x1[0], &x1[N], &x2[0], 1,
                      multiplies<int>(), plus<int>());
```

# Практическое задание

1. Вычислить факториал первых 20 чисел
2. Вычислить разность 20 первых соседних элементов ряда Фибоначчи
3. Найти сумму последовательность  $1, 1/2, 1/4, 1/8, 1/16 \dots 1/2^n$  используя алгоритм `inner_product`