Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Реферат

По дисциплине «Основы алгоритмизации и программирования»

На тему «Сравнение скорости передачи параметров в функцию разными способами»

Выполнила:

Студент(ка) 1 курса 7 группы

Подшиваленко Диана Игоревна

Проверил:

Белодед Николай Иванович

2023, Минск

**Содержание**

1. Предисловие … 3
2. Передача параметров по значению … 4
3. Передача параметров, используя указатель … 5
4. Передача параметров по ссылке … 6
5. Передача параметров по константной ссылке … 7
6. Передача параметров, использую класс памяти register … 8
7. Передача параметров, использую класс памяти static … 9
8. График … 10
9. Плюсы и минусы … 11

**Предисловие**

В данном реферате будут рассмотрены разные способы передачи параметров в функцию, а также их сравнение по скорости работы, плюсы и минусы каждого способа.

**1. Передача параметров по значению**

В первом способе будем передавать параметры по значению.

#include <iostream>

#include <ctime>

#include <string>

#include<Windows.h>

using namespace std;

void write\_num(int);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int num;

int n;

cin >> n;

clock\_t start = clock();

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

cin >> num;

write\_num(num);

}

clock\_t end = clock();

double duration = (float)(end - start) / CLOCKS\_PER\_SEC;

printf("%.5f ", duration);

}

void write\_num(int num) {

printf("%d ", num);

}

Результаты:

200000 чисел:

Среднее время: 0.816 секунды

500000 чисел:

Среднее время: 2.037 секунды

1000000 чисел:

Среднее время: 4.029 секунды

**2. Передача параметров, используя указатель**

Во втором способе будем передавать параметры, используя указатель.

#include <iostream>

#include <ctime>

#include <string>

#include<Windows.h>

using namespace std;

void write\_num(int \*);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int num;

int n;

cin >> n;

clock\_t start = clock();

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

cin >> num;

write\_num(&num);

}

clock\_t end = clock();

double duration = (float)(end - start) / CLOCKS\_PER\_SEC;

printf("%.5f ", duration);

}

void write\_num(int \*num) {

printf("%d ", \*num);

}

Результаты:

200000 чисел:

Среднее время: 0.802 секунды

500000 чисел:

Среднее время: 2.031 секунды

1000000 чисел:

Среднее время: 4.022 секунды

**3. Передача параметров по ссылке**

В третьем способе будем передавать параметры по ссылке.

#include <iostream>

#include <ctime>

#include <string>

#include<Windows.h>

using namespace std;

void write\_num(int &);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int num;

int n;

cin >> n;

clock\_t start = clock();

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

cin >> num;

write\_num(num);

}

clock\_t end = clock();

double duration = (float)(end - start) / CLOCKS\_PER\_SEC;

printf("%.5f ", duration);

}

void write\_num(int &num) {

printf("%d ", num);

}

Результаты:

200000 чисел:

Среднее время: 0.809 секунды

500000 чисел:

Среднее время: 2.043 секунды

1000000 чисел:

Среднее время: 4.098 секунды

**4. Передача параметров по константной ссылке**

В четвёртом способе будем передавать параметры по константной ссылке.

#include <iostream>

#include <ctime>

#include <string>

#include<Windows.h>

using namespace std;

void write\_num(const int &);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int num;

int n;

cin >> n;

clock\_t start = clock();

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

cin >> num;

write\_num(num);

}

clock\_t end = clock();

double duration = (float)(end - start) / CLOCKS\_PER\_SEC;

printf("%.5f ", duration);

}

void write\_num(const int &num) {

printf("%d ", num);

}

Результаты:

200000 чисел:

Среднее время: 0.846 секунды

500000 чисел:

Среднее время: 2.017 секунды

1000000 чисел:

Среднее время: 4.092 секунды

**5. Передача параметров, использую класс памяти register**

В пятом способе будем передавать параметры, использую класс памяти register.

#include <iostream>

#include <ctime>

#include <string>

#include<Windows.h>

using namespace std;

void write\_num(register int);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

register int num;

int n;

cin >> n;

clock\_t start = clock();

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

cin >> num;

write\_num(num);

}

clock\_t end = clock();

double duration = (float)(end - start) / CLOCKS\_PER\_SEC;

printf("%.5f ", duration);

}

void write\_num(register int num) {

printf("%d ", num);

}

Результаты:

200000 чисел:

Среднее время: 0.802 секунды

500000 чисел:

Среднее время: 2.026 секунды

1000000 чисел:

Среднее время: 4.049 секунды

**6. Передача параметров, использую класс памяти static**

В шестом способе будем передавать параметры, использую класс памяти static.

#include <iostream>

#include <ctime>

#include <string>

#include<Windows.h>

using namespace std;

void write\_num(static int);

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

static int num;

int n;

cin >> n;

clock\_t start = clock();

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

cin >> num;

write\_num(num);

}

clock\_t end = clock();

double duration = (float)(end - start) / CLOCKS\_PER\_SEC;

printf("%.5f ", duration);

}

void write\_num(static int num) {

printf("%d ", num);

}

Результаты:

200000 чисел:

Среднее время: 0.807 секунды

500000 чисел:

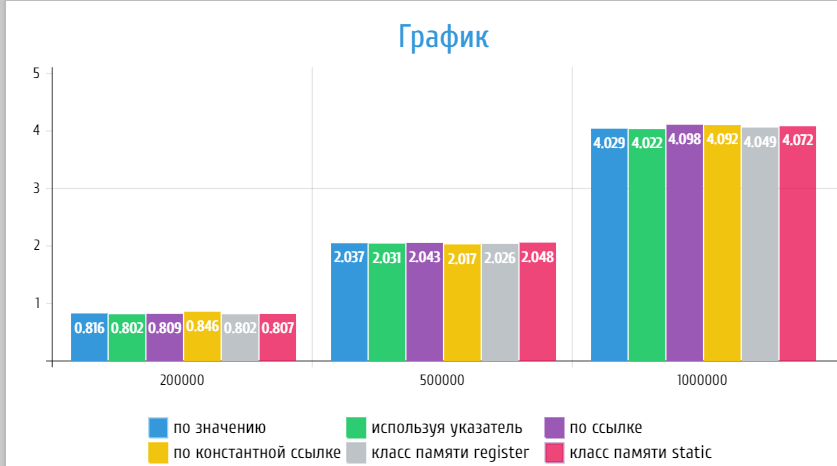
Среднее время: 2.048 секунды

1000000 чисел:

Среднее время: 4.072 секунды

**График**

Результаты исследований отразим на графике.



Как показывает график, все способы работают практически одинаково по времени. Отличия не существенны.

**Плюсы и минусы каждого способа**

1) по значению

+  Сами аргументы никогда не изменяются функцией, в которую передаются, что предотвращает возникновение побочных эффектов.

+ Аргументы могут быть переменными, литералами, выражениями, структурами, классами или перечислениями (т.е. почти всем, чем угодно).

**­­–** Копирование структур и классов может привести к значительному снижению производительности (особенно, когда функция вызывается много раз).

2) по ссылке и по константной ссылке

+ Ссылки позволяют функции изменять значение аргумента, что иногда полезно. В противном случае, для гарантии того, что функция не изменит значение аргумента, нужно использовать константные ссылки.

+ Поскольку при передаче по ссылке копирования аргументов не происходит, то этот способ гораздо эффективнее и быстрее, чем передача по значению, особенно при работе с большими структурами или классами.

+ Ссылки могут использоваться для возврата сразу нескольких значений из функции (через параметры вывода).

– Трудно определить, является ли параметр, переданный по не константной ссылке, параметром ввода, вывода или того и другого одновременно. Разумное использование const и суффикса Out для внешних переменных решает эту проблему.

– По вызову функции невозможно определить, будет аргумент изменен функцией или нет. Аргумент, переданный по значению или по ссылке, выглядит одинаково. Мы можем определить способ передачи аргумента только просмотрев объявление функции. Это может привести к ситуации, когда программист не сразу поймет, что функция изменяет значение аргумента.

3) по адресу

+ Передача по адресу позволяет функции изменить значение аргумента, что иногда полезно. В противном случае, используем const для гарантии того, что функция не изменит аргумент.

+ Поскольку копирования аргументов не происходит, то скорость передачи по адресу достаточно высокая, даже если передавать большие структуры или классы.

+ Мы можем вернуть сразу несколько значений из функции, используя параметры вывода.

– Все указатели нужно проверять, не являются ли они нулевыми. Попытка разыменовать нулевой указатель приведет к сбою в программе.

– Поскольку разыменование указателя выполняется медленнее, чем доступ к значению напрямую, то доступ к аргументам, переданным по адресу, выполняется также медленнее, чем доступ к аргументам, переданным по значению.