Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И. Раззакова

Факультет информационных технологий

Кафедра «Программное обеспечение компьютерных систем»

Направление:710400 «Программная инженерия»

ОТЧЕТ

По дисциплине: «Функционально-ориентированное проектирование»

Лабораторная работа №4

# Тема: «Перегрузка функций. Шаблоны функции. Рекурсивные функции.»

Выполнила: студентка группы

ПИ(б)-5-19 Ажиходжоева Каныкей

Проверил: доцент Искаков Р.Т

Бишкек – 2020

1. Напишите программу, которая использует шаблон функции maximum для поиска максимального из трех целых чисел, трех чисел с плавающей запятой и трех символов.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

template<class Type>

Type maximum(Type a, Type b,Type c) {

Type k = a;

if (k < b)k = b;

if (k < c)k = c;

return k;

}

int main() {

setlocale(LC\_ALL, "rus");

int a, b, c;

double a1, b1, c1;

char a2, b2, c2;

cout << "Введите 3 числа(Целые):";

cin >> a >> b >> c;

cout << "Введите 3 числа(Дробные):";

cin >> a1 >> b1 >> c1;

cout << "Введите 3 символа:";

cin >> a2 >> b2 >> c2;

cout << "Максимальное число(Целое):" << maximum(a, b, c) << endl;

cout << "Максимальное число(Дробное):" << maximum(a1, b1, c1) << endl;

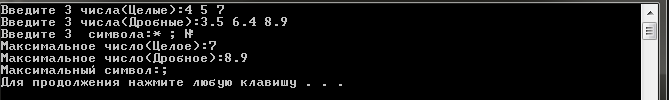
cout << "Максимальный символ:" << maximum(a2, b2, c2) << endl;

system("pause");

return 0;

}

* Результат вычислений



1. Напишите программу, которая использует шаблон функции по имени min для определения наименьшего из трех аргументов. Проверьте программу, используя пары целых чисел, символов и чисел с плавающей запятой.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

template<class T>

T min(T, T);

int main() {

setlocale(LC\_ALL, "rus");

int a, b;

double a1, b1;

char a2, b2;

cout << "Введите 2 числа(Целые):";

cin >> a >> b;

cout << "Минимальное число(Целое):" << min(a, b) << endl;

cout << endl;

cout << "Введите 2 числа(Дробные):";

cin >> a1 >> b1;

cout << "Минимальное число(Дробное):" << min(a1, b1) << endl;

cout << endl;

cout << "Введите 2 символа:";

cin >> a2 >> b2;

cout << "Минимальный символ:" << min(a2, b2) << endl;

cout << endl;

system("pause");

return 0;

}

template<class T>

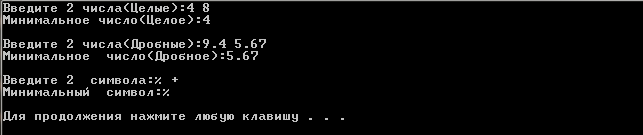
T min(T a, T b) {

if (a < b)return a;

else return b;

}

* Результат вычислений



1. Возведение числа n в степень р — это умножение числа n на себя р раз. Напишите функцию с именем power(), которая в качестве аргументов принимает значение типа double для n и значение типа int для р и возвращает значение типа double. Для аргумента, соответствующего степени числа, задайте значение по умолчанию, равное 2, чтобы при отсутствии показателя степени при вызове функции число n возводилось в квадрат. Напишите функцию main(), которая запрашивает у пользователя ввод аргументов для функции power(), и отобразите на экране результаты ее работы.

Взяв в качестве основы функцию power() из предыдущего упражнения, работающую только со значением типа double, создайте перегруженные функции с этим же именем, принимающими в качестве аргумента значения типа char, int, long и float. Напишите программу, вызывающую функцию power() со всеми возможными типами аргументов.

* Блок-схема





* Код программы

#include<iostream>

using namespace std;

double power(double, int p = 2);

int power(int, int p = 2);

float power(float, int p = 2);

long power(long, int p = 2);

char power(char, int p = 2);

int main() {

setlocale(LC\_ALL, "rus");

double a1, res1;

int p, a2, res2;

char r, a3, res3;

long a4, res4;

float a5, res5;

cout << "Введите число (дробное): ";

cin >> a1;

cout << "Будет ли степень + или - ( по умолчанию в квадрат): ";

cin >> r;

if (r == '+') {

skip:cout << "Введите степень: ";

cin >> p;

cout << endl;

if (p <= 0) {

cout << "Введите положительную степень\n";

goto skip;

}

else

res1 = power(a1, p);

}

else

res1 = power(a1);

cout << "Pезультат: " << endl;

cout << "Введённое вещественное число - " << a1 << ", в степени равно: " << res1 << endl;

cout << endl;

cout << "Введите число (целое): ";

cin >> a2;

cout << "Будет ли степень + или - ( по умолчанию в квадрат): ";

cin >> r;

if (r == '+') {

q:cout << "Введите степень: ";

cin >> p;

cout << endl;

if (p <= 0) {

cout << "Введите положительную степень\n";

goto q;

}

else

res2 = power(a2, p);

}

else

res2 = power(a2);

cout << "Pезультат: " << endl;

cout << "Введённое целое число- " << a2 << ", в степени равно: " << res2 << endl;

cout << endl;

cout << "Введите символ (Char): ";

cin >> a3;

cout << "Будет ли степень + или - ( по умолчанию в квадрат): ";

cin >> r;

if (r == '+') {

t:cout << "Введите степень: ";

cin >> p;

cout << endl;

if (p <= 0) {

cout << "Введите положительную степень\n";

goto t;

}

else

res3 = power(a3, p);

}

else

res3 = power(a3);

cout << "Pезультат: " << endl;

cout << "Введённый символ- " << a3 << ", в степени равно: " << res3 << endl;

cout << endl;

cout << "Введите число (long): ";

cin >> a4;

cout << "Будет ли степень + или - ( по умолчанию в квадрат): ";

cin >> r;

if (r == '+') {

m:cout << "Введите степень: ";

cin >> p;

cout << endl;

if (p <= 0) {

cout << "Введите положительную степень\n";

goto m;

}

else

res4 = power(a4, p);

}

else

res4 = power(a4);

cout << "Pезультат: " << endl;

cout << "Введённое число (long)- " << a4 << ", в степени равно: " << res4 << endl;

cout << endl;

cout << "Введите число (float): ";

cin >> a5;

cout << "Будет ли степень + или - ( по умолчанию в квадрат): ";

cin >> r;

if (r == '+') {

e:cout << "Введите степень: ";

cin >> p;

cout << endl;

if (p <= 0) {

cout << "Введите положительную степень\n";

goto e;

}

else

res5 = power(a5, p);

}

else

res5 = power(a5);

cout << "Pезультат: " << endl;

cout << "Введённое вещественное число (float)- " << a5 << ", в степени равно: " << res5 << endl;

system("pause");

return 0;

}

double power(double n, int p) {

double result = 1;

for (int i = 1; i <= p; i++) {

result \*= n;

}

return result;

}

int power(int n, int p)

{

int result = 1;

for (int i = 1; i <= p; i++) {

result \*= n;

}

return result;

}

float power(float n, int p)

{

float result = 1;

for (int i = 1; i <= p; i++) {

result \*= n;

}

return result;

}

long power(long n, int p)

{

long result = 1;

for (int i = 1; i <= p; i++) {

result \*= n;

}

return result;

}

char power(char n, int p)

{

char result = 1;

for (int i = 1; i <= p; i++) {

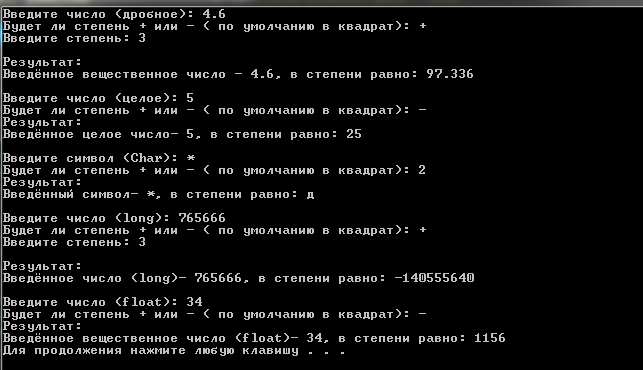
result \*= n;

}

return result;

}

* Результат вычислений



1. Ряд Фибонначи состоит из чисел, каждое из которых является суммой двух предыдущих (1, 1, 2, 3, 5, 8, 13, …). Найти n-ный элемент ряда, используя рекурсию.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

int Fib(int);

int main() {

setlocale(LC\_ALL, "rus");

int n, i;

cout << "Введите порядковый номер элемента: ";

cin >> n;

for (i = 1; i <= n; i++)

{

cout << i << "." << Fib(i) << endl;

}

system("pause");

return 0; }

int Fib(int a) {

if ((a == 1) || (a == 2))

return 1;

else

return (Fib(a - 1)) + (Fib(a - 2)); }

* Результат вычислений



1. Наибольший общий делитель (НОД) двух целых чисел х и у — это наибольшее целое, на которое без остатка делится каждое из двух чисел. Напишите рекурсивную функцию nod, которая возвращает наибольший общий делитель чисел х и у. НОД для х и у опреде­ляется рекурсивно следующим образом: если у равно 0, то nod (x, у) возвращает х; в противном случае nod(x, у) равняется nod(y, х % у), где % — это операция вычисления остатка.

* Блок-схема



* Код программы

#include<iostream>

using namespace std;

int nod(int, int);

int main() {

setlocale(LC\_ALL, "rus");

int a, b;

cout << "Введите первое число: ";

cin >> a;

cout << "Введите второе число: ";

cin >> b;

cout << "НОД данных чисел равен: " << nod(a, b) << endl;

system("pause");

return 0;

}

int nod(int x, int y) {

if (y == 0)

return x;

else

return nod(y, x % y);

}

* Результат вычислений

