**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ**

**КЫРГЫЗСКОЙ РЕСПУБЛИКИ**

**КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. И. Раззакова**

**ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Кафедра: **Программное обеспечение компьютерных систем**

Дисциплина «**Объектно-Ориентированное Программирование**»

Отчет

Лабораторная работа №1

Выполнил: студент группы ПИ-3-21

Сагыналиев Таалайбек

Проверил: Мусабаев Э. Б.

**Бишкек 2024**

**Задание:**

1. Возведение числа **n** в степень **p** – это умножение числа **n** на себя **p** раз.

Напишите функцию (не библиотечную) с именем **power** (), которая в качестве аргументов принимает значение типа **double** для **n** и значение типа **int** для **p** и возвращает значение типа **double**.

Для аргумента, соответствующего степени числа, задать значение по умолчанию, равное **2**, чтобы при отсутствии показателя степени при вызове функции число **n** возводилось в квадрат.

Сделайте проверку, будет ли пользователь вводить степень числа. Если будет, то эта степень – число, большее 2. Напишите функцию **main**(), которая запрашивает у пользователя ввод аргументов для функции **power** () и отображает на экране результаты её работы. Напишите код первого задания с *объявлением* (прототипом) функции power ().

1. Взяв в качестве основы функцию **power** () из первого задания, работающую только со значением типа **double**. Создайте перегруженные функции с этим же именем, принимающими в качестве аргумента значения типа **char**, short **int**, **long int, float**. Напишите программу, вызывающую функцию **power** () со всеми возможными типами аргументов. Напишите код второго задания без *объявления* функции power ().
2. Информация по работникам N заводов задается структурой с полями: фамилия, возраст, специальность, средний оклад. Ввести информацию по заводам, посчитать количество слесарей и токарей. Вывести эти значения на консоль.

**Задание 1**

#include <iostream>

using namespace std;

double power(double n, int p) {

double result = 1;

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

result \*= n;

}

return result;

}

Эта функция принимает два аргумента: число n и степень p. Затем она использует цикл для умножения числа n на само себя p раз и возвращает результат.

int main() {

double number;

int exp;

cout << "Put number = ";

cin >> number;

cout << "Put number, number must be > 2 = ";

cin >> exp;

if (exp < 2) {

cout << "Exp is < 2" << endl;

exp = 2;

}

double result = power(number, exp);

cout << "Result = " << result;

return 0;

}

**Задание 2**

double power(char n, int p);

double power(short int n, int p);

double power(long int n, int p);

double power(float n, int p);

int main() {

cout << "power('S', 2): " << power('S', 2) << endl; // with char

cout << "power(100, 2): " << power(static\_cast<short int>(100), 2) << endl; //short int

cout << "power(100000L, 2): " << power(static\_cast <long int>(100000L), 2) << endl; //long int

cout << "power(1.23f, 2): " << power(static\_cast < float>(1.23f), 2) << endl; // float

return 0;

}

double power(char n, int p) {

double result = 1.0;

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

result \*= static\_cast<int>(n);

}

return result;

}

double power(short int n, int p) {

double result = 1;

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

result \*= n;

}

return result;

}

double power(long int n, int p) {

double result = 1.0;

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

result \*= static\_cast<double>(n);

}

return result;

}

double power(float n, int p) {

double result = 1.0;

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

result \*= static\_cast<double>(n);

}

return result;

}

**Задание 3**

#include <vector>

struct Employee {

string lastName;

int age;

int specialty;

double averageSalary;

};

// Функция для ввода информации по заводам и подсчета слесарей и токарей

void processFactories(vector<vector<Employee>>& factories, int& totalLocksmiths, int& totalTurners) {

int numFactories;

cout << "Put number of factories: ";

cin >> numFactories;

totalLocksmiths = 0;

totalTurners = 0;

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

int numEmployees;

cout << "Put number of eployee " << i + 1 << ": ";

cin >> numEmployees;

vector<Employee> employees;

// Ввод информации о каждом работнике

for (int j = 0; j < numEmployees; ++j) {

Employee employee;

cout << "Put surname " << j + 1 << ": ";

cin >> employee.lastName;

cout << "Put age " << j + 1 << ": ";

cin >> employee.age;

cout << "Put specialty Lockmith = 1, Turners = 2; j = " << j + 1 << ": ";

cin >> employee.specialty;

cout << "Put salary " << j + 1 << ": ";

cin >> employee.averageSalary;

if (employee.specialty == 1) {

totalLocksmiths++;

}

else if (employee.specialty == 2) {

totalTurners++;

}

// Добавление работника в вектор

employees.push\_back(employee);

}

factories.push\_back(employees);

}

}

int main() {

vector<vector<Employee>> factories;

int totalLocksmiths, totalTurners;

processFactories(factories, totalLocksmiths, totalTurners);

cout << "Total locksmith: " << totalLocksmiths << endl;

cout << "Total turners: " << totalTurners << endl;

return 0;

}

**Контрольные вопросы**

1Какие стандартные типы используются в С++? Сколько под них резервируется памяти?

int (целочисленный тип):

Размер: обычно 4 байта (32 бита), но может быть разным в зависимости от компилятора и системы.

char (символьный тип):

Размер: обычно 1 байт.

float (вещественный тип с плавающей запятой):

Размер: обычно 4 байта.

double (вещественный тип с плавающей запятой с повышенной точностью):

Размер: обычно 8 байт.

bool (логический тип):

Размер: часто 1 байт, но это может варьироваться в зависимости от реализации.

short (короткий целочисленный тип):

Размер: обычно 2 байта.

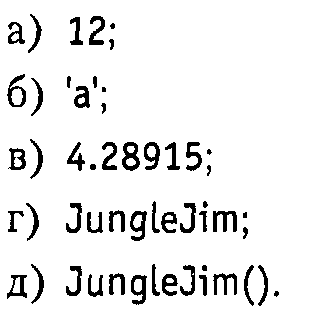
long (длинный целочисленный тип):

Размер: обычно 4 байта на 32-битных системах и 8 байт на 64-битных системах.

long long (очень длинный целочисленный тип):

Размер: обычно 8 байт.

2К каким элементам программы относятся следующие:



12 –числовая константа

‘a’ - символьная константа

4,28915 – вещественная константа

JungleJim - переменная

JungleJim()-функция

3Что такое функция? Какова роль функций в языке С++?

Функция представляет собой блок кода, который выполняет определенную задачу. Функции позволяют разбивать программу на более мелкие, логические части, делая код более читаемым, поддерживаемым и многократно используемым. Каждая функция в C++ имеет имя и может принимать аргументы (входные параметры), а также возвращать значение (выходной результат).

4 Напишите синтаксис функции

возвращаемый\_тип имя\_функции(параметры) {

// Тело функции

// ... код функции ...

return результат; // возвращаемое значение

}

возвращаемый\_тип: это тип данных, который функция возвращает. Может быть любым допустимым типом данных в C++ или ключевым словом void, если функция не возвращает значения.

имя\_функции: это уникальное имя функции, которое используется для ее вызова.

параметры: это переменные, которые функция принимает в качестве входных данных. Параметры указываются в круглых скобках и могут быть пустыми, если функция не принимает аргументы.

Тело функции: это блок кода, который выполняется, когда функция вызывается.

return результат: это оператор return, который используется для возврата значения из функции. Он необязательный, и его можно опустить в функциях с типом возвращаемого значения void

6. Объясните механизмы передачи аргументов по значению и по ссылке в функцию.

Объясните результаты работы программ

int incr(int m) {

m = m + 1;

return m;

}

int main() {

int n = 5;

cout << "n = " << incr(n) << endl;

cout << "n = " << n << endl;

return 0;

}

int incr(int &m) {

m = m + 1;

return m;

}

int main() {

int n = 5;

cout << "n = " << incr(n) << endl;

cout << "n = " << n << endl;

return 0;

}

В первом случаи — это способ передачи аргументов по значению, при котором функция создает копии передаваемых переменных, и после завершения, данные очищаются, а сами переменные, которые передавались в функцию остаются не именными.

Во втором случаи передача аргументов по ссылке, в этом случае в функцию адрес переменной аргумента в памяти, и она сохраняет изменения в самой переменной.

7Назовите разновидности аргументов, которые могут быть переданы параметрам функции?

Стандартные типы данных, структуры, ссылки и указатели.

8Что такое аргументы по умолчанию?

Это параметр функции, который имеет определённое (по умолчанию) значение. Если пользователь не передаёт в функцию значение для параметра по умолчанию, то используется значение по умолчанию.

9 Что такое перегрузка функций?  
Под перегрузкой функции понимается, определение нескольких функций (две или больше) с одинаковым именем, но различными параметрами.

10 Чем отличается структура (struct) от массива?

**Массив может хранить коллекцию элементов данных одного типа, а структура может создать пользовательские типы данных (которые создаются за счет комбинации других типов данных) и хранить как единое целое.**