**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-2-21

Камиев Абулхаир

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

#include <iostream>

using namespace std;

double power(double n, int p = 2);

int main() {

double base;

int exponent;

cout << "Введите основание числа: ";

cin >> base;

cout << "Введите степень числа (больше 2): ";

cin >> exponent;

double result = power(base, (exponent > 2) ? exponent : 2);

cout << "Результат возведения в степень: " << result << std::endl;

return 0;

}

**Задание №2**

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

#include <iostream>

using namespace std;

// Определение перегруженной функции power для типа char

double power(char n, int p) {

double result = 1.0;

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

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

}

return result;

}

// Определение перегруженной функции power для типа short int

double power(short int n, int p) {

double result = 1.0;

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

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

}

return result;

}

// Определение перегруженной функции power для типа long int

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;

}

// Определение перегруженной функции power для типа float

double power(float n, int p) {

double result = 1.0;

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

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

}

return result;

}

int main() {

// Примеры вызова функции power() с разными типами аргументов

cout << "power('A', 3): " << power('A', 3) << endl;

cout << "power(5, 4): " << power(static\_cast<float>(5), 4) << endl;

cout << "power(123456L, 2): " << power(123456L, 2) << endl;

cout << "power(3.14f, 3): " << power(3.14f, 3) << endl;

return 0;

}

**Задание №3**

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.

//3

#include <iostream>

#include <vector>

#include <string>

using namespace std;

// Структура для хранения информации о работнике

struct Employee {

string lastName;

int age;

string specialty;

double averageSalary;

};

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

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

// Количество заводов

int numFactories;

cout << "Введите количество заводов: ";

cin >> numFactories;

// Инициализация переменных с общим количеством слесарей и токарей

totalLocksmiths = 0;

totalTurners = 0;

// Ввод информации по каждому заводу

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

int numEmployees;

cout << "Введите количество работников на заводе " << i + 1 << ": ";

cin >> numEmployees;

// Вектор для хранения информации о работниках на текущем заводе

vector<Employee> employees;

// Ввод информации о каждом работнике

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

Employee employee;

cout << "Введите фамилию работника " << j + 1 << ": ";

cin >> employee.lastName;

cout << "Введите возраст работника " << j + 1 << ": ";

cin >> employee.age;

cout << "Введите специальность работника " << j + 1 << ": ";

cin >> employee.specialty;

cout << "Введите средний оклад работника " << j + 1 << ": ";

cin >> employee.averageSalary;

// Подсчет слесарей и токарей

if (employee.specialty == "слесарь") {

totalLocksmiths++;

} else if (employee.specialty == "токарь") {

totalTurners++;

}

// Добавление работника в вектор

employees.push\_back(employee);

}

// Добавление вектора с информацией о работниках на текущем заводе

factories.push\_back(employees);

}

}

int main() {

// Вектор векторов для хранения информации о работниках на каждом заводе

vector<vector<Employee>> factories;

// Общее количество слесарей и токарей

int totalLocksmiths, totalTurners;

// Вызов функции для ввода информации по заводам и подсчета слесарей и токарей

processFactories(factories, totalLocksmiths, totalTurners);

// Вывод результатов

cout << "Общее количество слесарей: " << totalLocksmiths << endl;

cout << "Общее количество токарей: " << totalTurners << endl;

return 0;

}

//