Министерство образования и науки Кыргызской Республики

Кыргызский государственный технический университет

им. И.Раззакова

Институт Совместных Образовательных Программ

Кафедра «Программное обеспечение компьютерных систем»

Направление: 710400 «Программная инженерия»

Дисциплина: «Объектно-ориентированное проектирование и программирование ПО»

ОТЧЕТ

По лабораторной работе №1

Выполнил: студент группы

ПИ(б)-ИСОП 1-21 Именов Алишер

Проверила: ст. пр. Мусабаев Э.Б.

Бишкек – 2024

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 ().

#include <iostream>

// Прототип функции power

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

int main() {

setlocale(LC\_ALL, "rus");

double base;

int exponent;

// Ввод базового числа

std::cout << "Введите число: ";

std::cin >> base;

// Проверка на ввод степени

std::cout << "Будете вводить степень числа? (1 - да, 0 - нет): ";

int userChoice;

std::cin >> userChoice;

if (userChoice == 1) {

// Ввод степени

std::cout << "Введите степень (целое число больше 2): ";

std::cin >> exponent;

if (exponent <= 2) {

std::cout << "Ошибка: степень должна быть целым числом больше 2." << std::endl;

return 1;

}

}

else {

// Используем значение по умолчанию (степень 2)

exponent = 2;

}

// Вызов функции power и вывод результата

double result = power(base, exponent);

std::cout << "Результат: " << result << std::endl;

return 0;

}

// Определение функции power

double power(double n, int p) {

double result = 1.0;

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

result \*= n;

}

return result;

}

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 ().

#include <iostream>

// Перегруженные функции power

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

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

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

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

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

int main() {

setlocale(LC\_ALL, "rus");

// Тестирование функции power с разными типами аргументов

std::cout << "double: " << power(2.0) << std::endl;

std::cout << "char: " << power('A') << std::endl;

std::cout << "short int: " << power(static\_cast <short>(5)) << std::endl;

std::cout << "long int: " << power(1000000L) << std::endl;

std::cout << "float: " << power(3.14f) << std::endl;

return 0;

}

// Определение функций power

double power(double n, int p) {

double result = 1.0;

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

result \*= n;

}

return result;

}

double power(char n, int p) {

// Просто вызываем функцию power для типа double

return power(static\_cast<double>(n), p);

}

double power(short int n, int p) {

// Просто вызываем функцию power для типа double

return power(static\_cast<double>(n), p);

}

double power(long int n, int p) {

// Просто вызываем функцию power для типа double

return power(static\_cast<double>(n), p);

}

double power(float n, int p) {

// Просто вызываем функцию power для типа double

return power(static\_cast<double>(n), p);

}

1. 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.

#include <iostream>

#include <vector>

struct Worker {

std::string surname;

int age;

std::string specialization;

double averageSalary;

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int numFactories;

std::cout << "Введите количество заводов: ";

std::cin >> numFactories;

// Создаем вектор для хранения информации по работникам на всех заводах

std::vector<Worker> workers;

// Ввод информации по заводам

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

int numWorkers;

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

std::cin >> numWorkers;

// Ввод информации по работникам на текущем заводе

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

Worker worker;

std::cout << "Введите фамилию работника: ";

std::cin >> worker.surname;

std::cout << "Введите возраст работника: ";

std::cin >> worker.age;

std::cout << "Введите специальность работника: ";

std::cin >> worker.specialization;

std::cout << "Введите средний оклад работника: ";

std::cin >> worker.averageSalary;

// Добавляем работника в вектор

workers.push\_back(worker);

}

}

// Подсчет количества слесарей и токарей

int numLocksmiths = 0;

int numTurners = 0;

for (const auto& worker : workers) {

if (worker.specialization == "слесарь") {

numLocksmiths++;

}

else if (worker.specialization == "токарь") {

numTurners++;

}

}

// Вывод результатов

std::cout << "Количество слесарей: " << numLocksmiths << std::endl;

std::cout << "Количество токарей: " << numTurners << std::endl;

return 0;

}