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»

Дисциплина: « Объектно-ориентрованное проектирование / Объектно-ориентрованное программирование»

**ОТЧЕТ**

По лабораторной работе №1

Тема: «Повторение»

Выполнила: студентка группы

ПИ(б)-3-21 Пак Ксения

Проверил: Мусабаев Э. Б.

Бишкек – 2024

Задания

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 ().

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

int main() {

double n;

int p;

double result;

cout << "---------- Task 1 ----------\n";

cout << "Enter base: \n";

cin >> n;

cout << "Do you want to enter the exponent? (Enter 0 for default, any other number for custom): ";

cin >> p;

if (p != 0) {

cout << "Enter exponent: ";

cin >> p;

result = power(n, p);

}

else result = power(n);

cout<<"Result: "<<result<<"\n\n";

return 0;

}

double power(double n, int p) {

double result = 1.0;

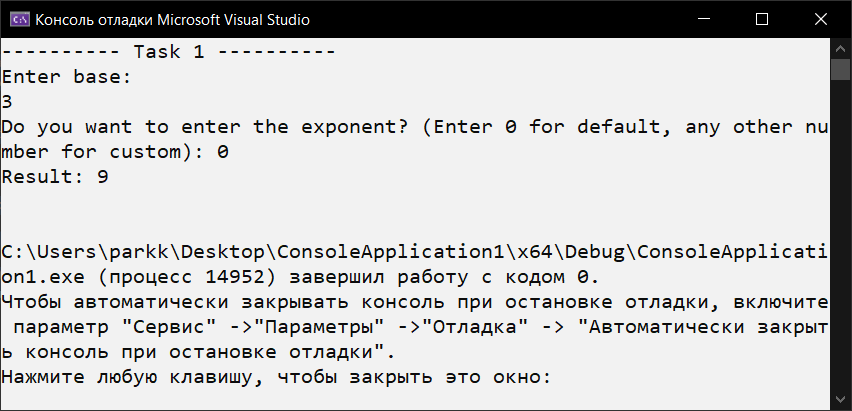
for (int i = 0; i < p; i++) {

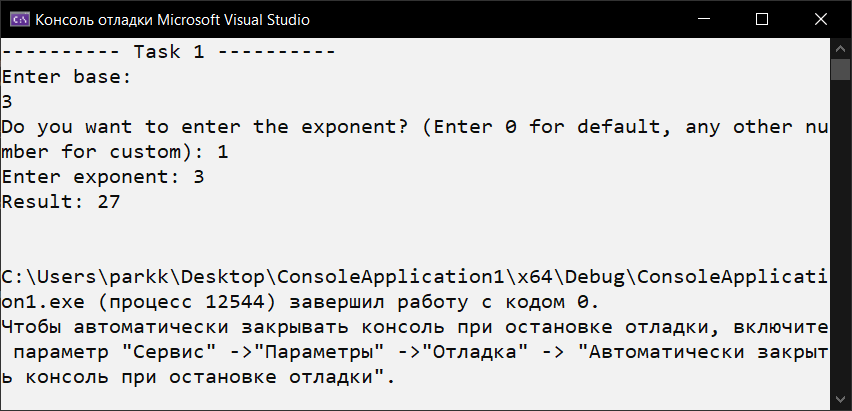
result \*= n;

}

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 ().

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

int main() {

double n;

char n2;

int n3;

long n4;

float n5;

short n6;

int p;

double result;

cout << "---------- Task 1 ----------\n";

cout << "Enter base: \n";

cin >> n;

cout << "Do you want to enter the exponent? (Enter 0 for default, any other number for custom): ";

cin >> p;

if (p != 0) {

cout << "Enter exponent: ";

cin >> p;

result = power(n, p);

}

else result = power(n);

cout<<"Result: "<<result<<"\n\n";

cout << "---------- Task 2 ----------\n";

cout << "char\n";

cout << "Enter base: \n";

cin >> n2;

cout << "ASCII value of entered char: " << static\_cast<int>(n2) << "\n";

cout << "Enter exponent: \n";

cin >> p;

result = power(n2, p);

cout << "Result: " << result << "\n\n";

cout << "int\n";

cout << "Enter base: \n";

cin >> n3;

cout << "Enter exponent: \n";

cin >> p;

result = power(n3, p);

cout << "Result: " << result << "\n\n";

cout << "short\n";

cout << "Enter base: \n";

cin >> n6;

cout << "Enter exponent: \n";

cin >> p;

result = power(n6, p);

cout << "Result: " << result << "\n\n";

cout << "long\n";

cout << "Enter base: \n";

cin >> n4;

cout << "Enter exponent: \n";

cin >> p;

result = power(n4, p);

cout << "Result: " << result << "\n\n";

cout << "float\n";

cout << "Enter base: \n";

cin >> n5;

cout << "Enter exponent: \n";

cin >> p;

result = power(n5, p);

cout << "Result: " << result << "\n\n";

return 0;

}

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=2) {

double result = 1.0;

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

result \*= n;

}

return result;

}

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

double result = 1.0;

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

result \*= n;

}

return result;

}

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

double result = 1.0;

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

result \*= n;

}

return result;

}

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

double result = 1.0;

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

result \*= n;

}

return result;

}

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

double result = 1.0;

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

result \*= n;

}

return result;

}

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.

struct Employee {

public:

int id;

string surname;

int age;

string position;

double averageSalary;

void print() {

cout << "id: " << id << "\n";

cout << "surname: " << surname<<"\n";

cout << "age: " << age << "\n";

cout << "position: " << position << "\n";

cout << "average salary: " << averageSalary << "\n";

}

};

struct Factory {

string name;

vector<Employee> employees;

void inputEmployees() {

int n;

cout << "Enter the number of employees at the " << name << " factory: ";

cin >> n;

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

Employee employee;

cout << "Enter information for employee " << i + 1 << ":\n";

cout << "ID: ";

cin >> employee.id;

cout << "Surname: ";

cin >> employee.surname;

cout << "Age: ";

cin >> employee.age;

cout << "Position: ";

cin >> employee.position;

cout << "Average Salary: ";

cin >> employee.averageSalary;

employees.push\_back(employee);

}

}

pair<int, int> countEmployees() const {

int countLocksmith = 0;

int countTurner = 0;

for (const auto& employee : employees) {

if (employee.position == "locksmith") {

countLocksmith++;

}

else if (employee.position == "turner") {

countTurner++;

}

}

cout << "Number of locksmiths at the " << name << " factory: " << countLocksmith << endl;

cout << "Number of turners at the " << name << " factory: " << countTurner << endl;

return { countLocksmith, countTurner };

}

};

int main() {

setlocale(LC\_ALL, "English");

const int N = 2;

vector<Factory> factories(N);

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

cout << "Enter the name of factory " << i + 1 << ": ";

cin >> factories[i].name;

factories[i].inputEmployees();

cout << endl;

}

int locksmith=0, turner=0;

for (const auto& factory : factories) {

pair<int, int> counts = factory.countEmployees();

locksmith += counts.first;

turner += counts.second;

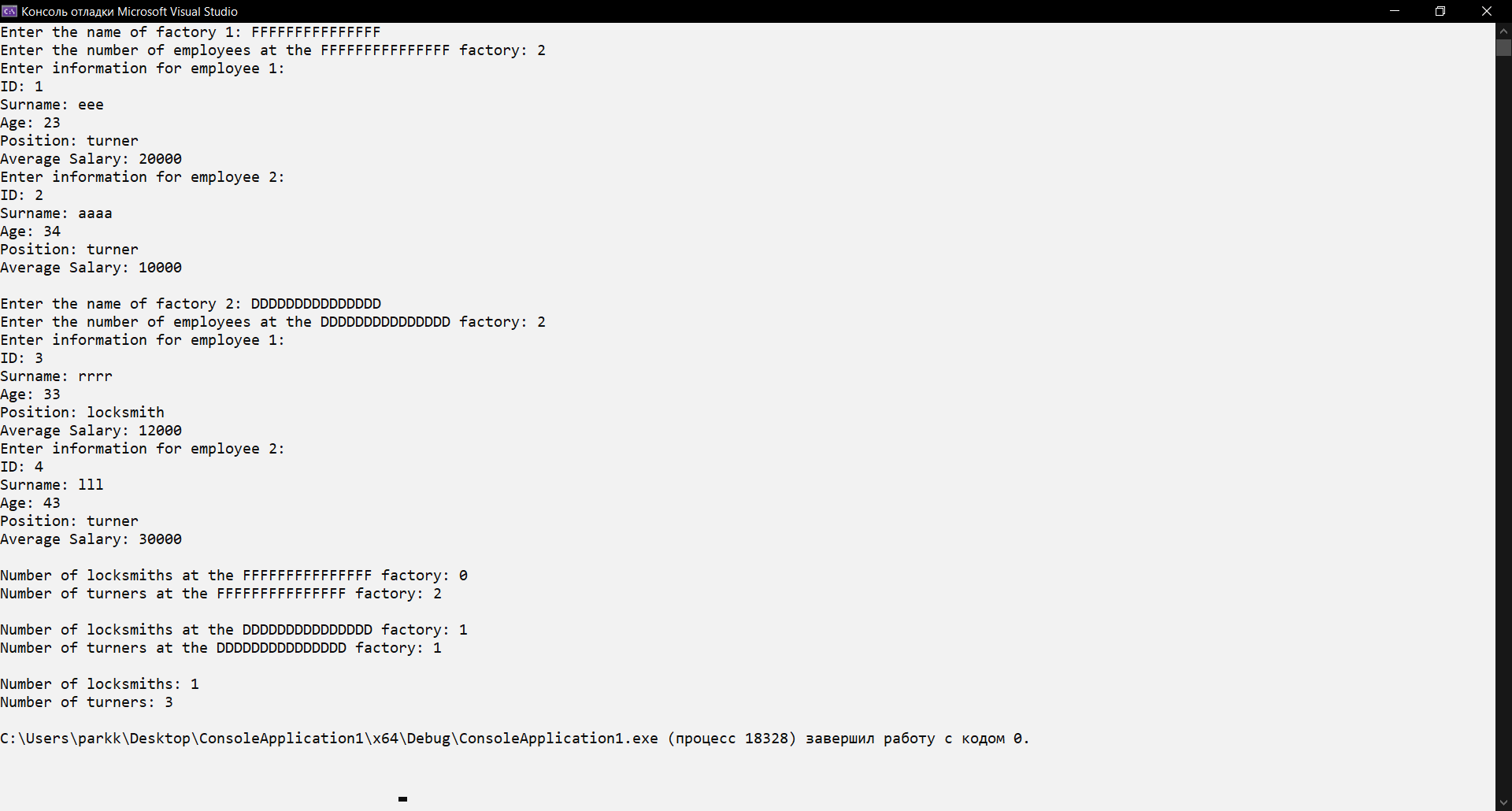
cout << endl;

}

cout << "Number of locksmiths: " << locksmith << endl;

cout << "Number of turners: " << turner << endl;

return 0;

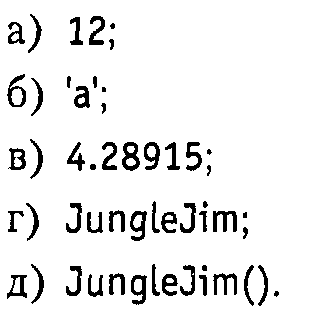
}

Ответы на вопросы

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

В языке C++ определены следующие базовые типы данных: логический тип bool, целочисленные типы, типа чисел с плавающей точкой, символьные типы.

1. К каким элементам программы относятся следующие:



Int, char, double, string, функция

1. Что такое функция? Какова роль функций в языке С++? Функция - это блок кода, который выполняет конкретную задачу. Она может быть вызвана из других частей программы.
2. Напишите синтаксис функции.

<тип возвращаемого значения> <имя функции>(<параметры>) {

// Тело функции

}

1. Опишите способы использования функций в программах с объявлением функций и без объявления функций.

Без объявления функций (Forward Declaration):

Определение перед использованием: Функции могут быть определены до их использования в коде. Компилятор просматривает весь код и знает о существовании функции до ее вызова.

// Определение функции перед использованием

void myFunction() {

// Тело функции

}

int main() {

// Вызов функции

myFunction();

return 0;

}

С объявлением функций:

Прототипы функций: Объявления (прототипы) функций помещаются в начало файла или в отдельный заголовочный файл. Это позволяет компилятору знать о функции до ее определения.

// Прототип функции

void myFunction();

int main() {

// Вызов функции

myFunction();

return 0;

}

// Определение функции после использования

void myFunction() {

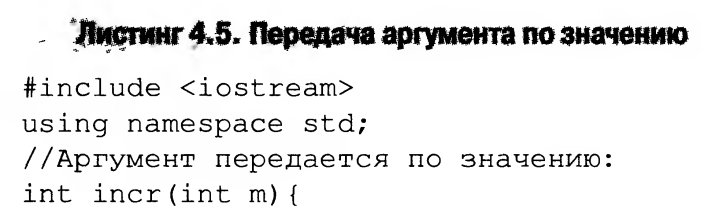
// Тело функции

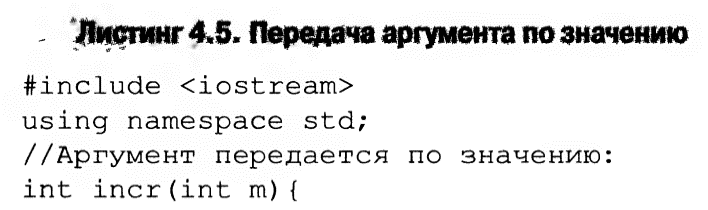
}

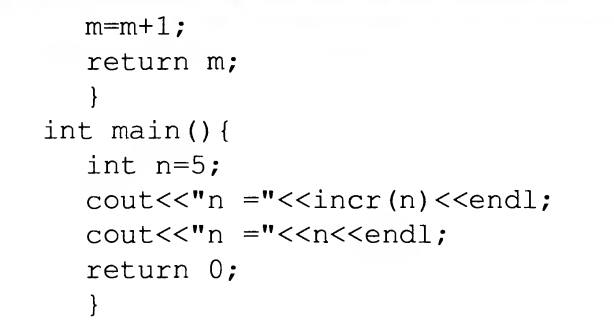
1. Объясните механизмы передачи аргументов по значению и по ссылке в функцию.

Объясните результаты работы программ.

1)







2))





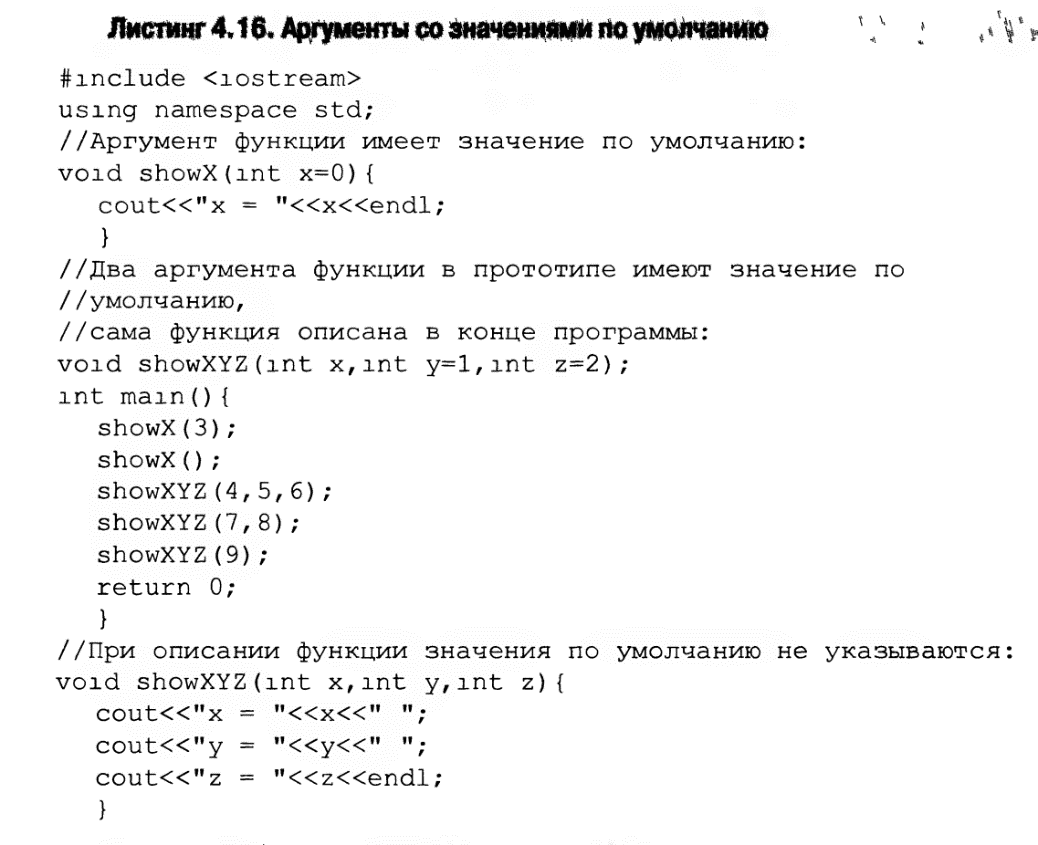
1. По значению: Копия значения аргумента передается в функцию.
2. По ссылке: Ссылка на переменную передается в функцию, позволяя функции изменять оригинальное значение.
3. Назовите разновидности аргументов, которые могут быть переданы параметрам функции?

Обязательные аргументы: Требуются для вызова функции.

Необязательные аргументы: Могут иметь значения по умолчанию.

1. Что такое аргументы по умолчанию?

Объясните результат работы программы:

,

Аргументы, для которых предоставлены значения по умолчанию. Если значение не передано, используется значение по умолчанию. В примере аргумент функции showX() имеет значение по умолчанию x=0, а аргументы функции showXYZ() имеют значение по умолчанию y=1, z=2.

1. Что такое перегрузка функций? Возможность определения нескольких функций с одним именем, но разными параметрами.
2. Чем отличается структура (struct) от массива?

struct: Пользовательский тип данных, который объединяет различные типы данных под одним именем.

Массив: Упорядоченный набор однотипных элементов.