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 – ориентированное программирование»

**ОТЧЕТ**

По лабораторной работе №1.

Тема: «Повторение»

                                                                       Выполнила: студент группы

ПИ(б)-2-21 Абдылдаева Аэлита

Проверил: Мусабаев Э. Б.

**Бишкек – 2022**

**Тема:** повторить пройденный материал - типы данных, функции и структуры в программировании для лучшего понимания свойств классов в объектно-ориентированном программировании.

**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 ().

***Код:***

#include <iostream>

using namespace std;

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

int main() {

setlocale(LC\_ALL, "rus");

double x;

int exp;

cout << "Введите число: ";

cin >> x;

cout << "Желаете указать степень (1 - да, 0 - нет)? ";

int inputExp;

cin >> inputExp;

exp = (inputExp) ? (cout << "Введите степень: ", cin >> exp, exp) : 2;

cout << "Результат: " << power(x, exp) << endl;

return 0;

}

double power(double n, int p) {

double result = 1.0;

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

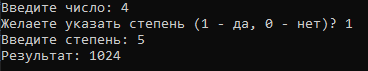
result \*= n;

}

return result;

}

***Результат:***





**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 ().

***Код:***

#include<iostream>

using namespace std;

double power(double n, int p);

char power(char n, int p);

short int power(short int n, int p);

long int power(long int n, int p);

float power(float n, int p);

int main() {

setlocale(LC\_ALL, "rus");

double x;

int exp;

cout << "Введите число: ";

cin >> x;

cout << "Желаете указать степень (1 - да, 0 - нет)? ";

int inputExp;

cin >> inputExp;

exp = (inputExp) ? (cout << "Введите степень: ", cin >> exp, exp) : 2;

cout << "Результаты: " << endl;

cout << "double: " << power(x, exp) << endl;

cout << "char: " << static\_cast<int>(power(static\_cast<char>(x), exp)) << endl;

cout << "short int: " << power(static\_cast<short int>(x), exp) << endl;

cout << "long int: " << power(static\_cast<long int>(x), exp) << endl;

cout << "float: " << power(static\_cast<float>(x), exp) << endl;

system("pause");

return 0;

}

double power(double n, int p) {

double result = 1.0;

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

result \*= n;

}

return result;

}

char power(char n, int p) {

char result = n;

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

result \*= n;

}

return result;

}

short int power(short int n, int p) {

short int result = n;

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

result \*= n;

}

return result;

}

long int power(long int n, int p) {

long int result = n;

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

result \*= n;

}

return result;

}

float power(float n, int p) {

float result = n;

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

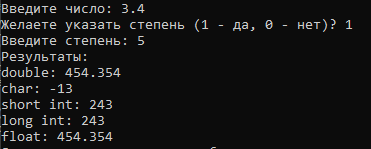
result \*= n;

}

return result;

}

***Результат:***



*Эти различия объясняются тем, как каждый тип обрабатывает числа и как происходит приведение типов, что может привести к потере точности или изменению значения.*

**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.

***Код:***

#include <iostream>

#include <vector>

using namespace std;

struct Worker {

string fio;

int age;

string special;

int oklad;

};

int main() {

setlocale(0, "");

int numFac;

cout << "Введите количество заводов: ";

cin >> numFac;

int numTokars = 0, numSlesars = 0;

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

cout << "Введите информацию по " << i + 1 << "-заводу:" << endl;

int numWorkers;

cout << "Введите количество работников: ";

cin >> numWorkers;

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

Worker worker;

cout << "ФИО: "; cin >> worker.fio;

cout << "Возраст: "; cin >> worker.age;

cout << "Специальность (slesar/tokar): "; cin >> worker.special;

cout << "Средний оклад: "; cin >> worker.oklad;

if (worker.special == "slesar") {

numSlesars++;

}

else if (worker.special == "tokar") {

numTokars++;

}

}

}

cout << "Количество токарей: " << numTokars << endl;

cout << "Количество слесарей: " << numSlesars << endl;

return 0;

}

***Результат:***

