# **Практическое занятие №1**

*Дата*: 17.04.2023

*Тема:* «Принципы объектно-ориентированного программирования»

*Цель:*Изучить принципы объектно-ориентированного программирования и их применение.

**Ход работы**

**Задание 1.**

**Условие:** 1) Даны стороны прямоугольника a и b. Найти его площадь S = a·b и периметр P = 2· (a + b).

**Исходный код:**

using System;

class Program {

static void Main(string[] args) {

double a, b, s, p;

Console.Write("Введите сторону a: ");

a = double.Parse(Console.ReadLine());

Console.Write("Введите сторону b: ");

b = double.Parse(Console.ReadLine());

s = a \* b;

p = 2 \* (a + b);

Console.WriteLine("Площадь прямоугольника: {0}", s);

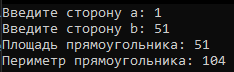
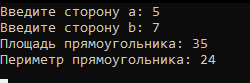
Console.WriteLine("Периметр прямоугольника: {0}", p);

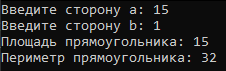
Console.ReadKey();

}

}

**Результат:**





**Задание 2.**

**Условие:** 2) Даны переменные A, B, C. Изменить их значения, переместив содержимое A в B, B — в C, C — в A, и вывести новые значения переменных A, B, C.

**Исходный код:**

using System;

class Program {

static void Main(string[] args) {

double a, b, c, temp;

Console.Write("Введите значение a: ");

a = double.Parse(Console.ReadLine());

Console.Write("Введите значение b: ");

b = double.Parse(Console.ReadLine());

Console.Write("Введите значение c: ");

c = double.Parse(Console.ReadLine());

temp = a;

a = b;

b = c;

c = temp;

Console.WriteLine("Новое значение a: {0}", a);

Console.WriteLine("Новое значение b: {0}", b);

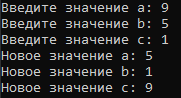
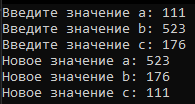
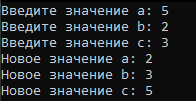
Console.WriteLine("Новое значение c: {0}", c);

Console.ReadKey();

}

}

**Результат:**



**Задание 3.**

**Условие:** 3) Дана масса M в килограммах. Используя операцию деления нацело, найти количество полных тонн в ней (1 тонна = 1000 кг).

**Исходный код:**

using System;

class Program {

static void Main(string[] args) {

double m, ton;

Console.Write("Введите массу в килограммах: ");

m = double.Parse(Console.ReadLine());

ton = m / 1000;

Console.WriteLine("Количество полных тонн: {0}", (int)ton);

Console.ReadKey();

}

}

**Результат:**



**Задание 4.**

**Условие:** 4) С начала суток прошло N секунд (N — целое). Найти количество полных минут, прошедших с начала суток.

**Исходный код:**

using System;

class Program {

static void Main(string[] args) {

int n, minutes;

Console.Write("Введите количество секунд: ");

n = int.Parse(Console.ReadLine());

minutes = n / 60;

Console.WriteLine("Количество полных минут: {0}", minutes);

Console.ReadKey();

}

}

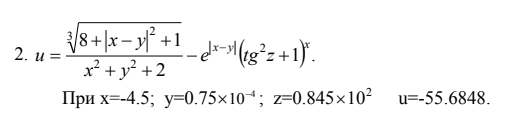
**Результат:**





**Задание 5.**

**Условие:**



**Исходный код:**

class Program

{

static void Main(string[] args)

{

double x = -4.5;

double y = 0.75 \* Math.Pow(10, -4);

double z = 0.845 \* Math.Pow(10, 2);

double u = -55.6848;

double result = ((Math.Pow(8 + Math.Pow(Math.Abs(x - y), 2) + 1, 1/3)) / (Math.Pow(x, 2) + Math.Pow(y, 2) + 2)) - Math.Exp(Math.Abs(x - y)) \* Math.Pow(Math.Tan(z), 2 \* x + 1);

Console.WriteLine("Результат: " + result);

Console.WriteLine("Правильный ответ: " + u);

Console.ReadKey();

}

}

**Результат:**



**Задание 6.**

**Условие:** Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае вычесть из него 2. Вывести полученное число.

**Исходный код:**

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите целое число:"); // выводим сообщение пользователю

int num = Convert.ToInt32(Console.ReadLine()); // считываем число, введенное пользователем

if (num > 0)

{

num += 1; // прибавляем 1, если число положительное

}

else

{

num -= 2; // вычитаем 2, если число отрицательное или равно 0

}

Console.WriteLine($"Результат: {num}"); // выводим результат

Console.ReadKey(); // ожидаем нажатия клавиши для завершения программы

}

}

**Результат:**

**Задание 7.**

**Условие:** Мастям игральных карт присвоены порядковые номера: 1 — пики, 2 — трефы, 3 — бубны, 4 — червы. Достоинству карт, старших десятки, присвоены номера: 11 — валет, 12 — дама, 13 — король, 14 — туз. Даны два целых числа: N — достоинство (6 ▯ N ▯ 14) и M — масть карты (1 ▯ M ▯ 4). Вывести название соответствующей карты вида «шестерка бубен», «дама червей», «туз треф» и т. п.

**Исходный код:**

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите достоинство карты (6-14):");

int n = Convert.ToInt32(Console.ReadLine()); // считываем достоинство карты

Console.WriteLine("Введите масть карты (1-4):");

int m = Convert.ToInt32(Console.ReadLine()); // считываем масть карты

string rank = ""; // переменная для хранения названия достоинства карты

switch (n)

{

case 6:

rank = "шестерка";

break;

case 7:

rank = "семерка";

break;

case 8:

rank = "восьмерка";

break;

case 9:

rank = "девятка";

break;

case 10:

rank = "десятка";

break;

case 11:

rank = "валет";

break;

case 12:

rank = "дама";

break;

case 13:

rank = "король";

break;

case 14:

rank = "туз";

break;

default:

Console.WriteLine("Ошибка: введено некорректное значение достоинства карты.");

Console.ReadKey();

return;

}

string suit = ""; // переменная для хранения названия масти карты

switch (m)

{

case 1:

suit = "пик";

break;

case 2:

suit = "треф";

break;

case 3:

suit = "бубен";

break;

case 4:

suit = "червей";

break;

default:

Console.WriteLine("Ошибка: введено некорректное значение масти карты.");

Console.ReadKey();

return;

}

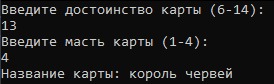
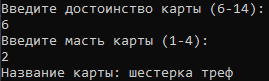
Console.WriteLine($"Название карты: {rank} {suit}"); // выводим результат

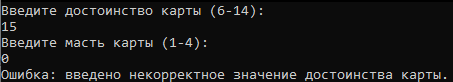
Console.ReadKey(); // ожидаем нажатия клавиши для завершения программы

}

}

**Результат:**



**Задание 8.**

**Условие:** Даны два целых числа A и B (A < B). Вывести в порядке возрастания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество N этих чисел.

**Исходный код:**

using System;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите целое число A:");

int a = Convert.ToInt32(Console.ReadLine()); // считываем число A

Console.WriteLine("Введите целое число B (B > A):");

int b = Convert.ToInt32(Console.ReadLine()); // считываем число B

int count = 0; // переменная для хранения количества чисел

for (int i = a; i <= b; i++) // перебираем все числа от A до B

{

Console.WriteLine(i); // выводим число i

count++; // увеличиваем счетчик чисел

}

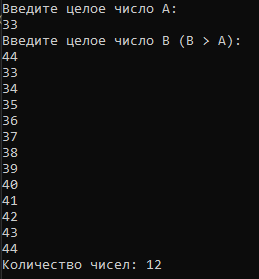
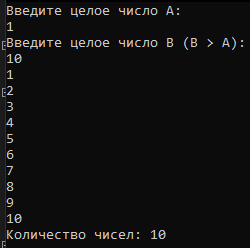
Console.WriteLine($"Количество чисел: {count}"); // выводим количество чисел

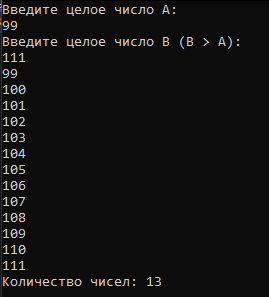
Console.ReadKey(); // ожидаем нажатия клавиши для завершения программы

}

}

**Результат:**



**Задание 9.**

**Условие:** Спортсмен-лыжник начал тренировки, пробежав в первый день 10 км. Каждый следующий день он увеличивал длину пробега на P процентов от пробега предыдущего дня (P — вещественное, 0 < P < 50). По данному P определить, после какого дня суммарный пробег лыжника за все дни превысит 200 км, и вывести найденное количество дней K (целое) и суммарный пробег S (вещественное число).

**Исходный код:**

using System;

class Program

{

static void Main(string[] args)

{

double p; // процент увеличения пробега

double distance = 10.0; // начальный пробег

double totalDistance = distance; // суммарный пробег

int days = 1; // количество дней

Console.WriteLine("Введите процент увеличения пробега (0 < P < 50):");

p = Convert.ToDouble(Console.ReadLine()); // считываем процент увеличения пробега

while (totalDistance <= 200.0) // пока суммарный пробег не превысит 200 км

{

distance \*= (1.0 + p / 100.0); // увеличиваем пробег на P процентов

totalDistance += distance; // увеличиваем суммарный пробег на пробег за текущий день

days++; // увеличиваем количество дней

}

Console.WriteLine($"Количество дней: {days}, суммарный пробег: {totalDistance} км");

Console.ReadKey(); // ожидаем нажатия клавиши для завершения программы

}

}

**Результат:**





