Уровень 1:

№1:

using System;

using static System.Console;

using static System.Math;

namespace L1\_task01

{

class Program

{

static void Main(string[] args)

{

double radius = 2d;

WriteLine($"Точка (0, 2) {IntoRussian(IsOnCircle(radius, 0, 2))} на окружности");

WriteLine($"Точка (1.5, 0.7) {IntoRussian(IsOnCircle(radius, 1.5, 0.7))} на окружности");

WriteLine($"Точка (1, 1) {IntoRussian(IsOnCircle(radius, 1, 1))} на окружности");

WriteLine($"Точка (3, 0) {IntoRussian(IsOnCircle(radius, 3, 0))} на окружности");

}

static bool IsOnCircle(double circleRadius, double x, double y)

{

bool isOnCircle = Abs(x \* x + y \* y - circleRadius \* circleRadius) <= 0.001;

return isOnCircle;

}

static string IntoRussian(bool isOnCircle)

{

if (isOnCircle)

{

return "Лежит";

}

else

{

return "НЕ лежит";

}

}

}

}

№2:

using System;

using static System.Console;

namespace L1\_task02

{

class Program

{

static void Main(string[] args)

{

Write("Введите x: ");

double x = double.Parse(ReadLine());

Write("Введите y: ");

double y = double.Parse(ReadLine());

bool isInsideTriangle = IsInsileTriangle(x, y);

WriteLine($"Точка ({x}, {y}) находится {IntoRussian(isInsideTriangle)} треугольника");

}

static bool IsInsileTriangle(double x, double y)

{

bool isUnderLeftEdge = y <= 1 + x;

bool isUnderRightEdge = y <= 1 - x;

bool isAboveAbcissa = y >= 0;

return isUnderLeftEdge && isUnderRightEdge && isAboveAbcissa;

}

static string IntoRussian(bool isOnCircle)

{

if (isOnCircle)

{

return "внутри";

}

else

{

return "ВНЕ";

}

}

}

}

№3:

using System;

using static System.Console;

namespace L1\_task03

{

class Program

{

static void Main(string[] args)

{

Write("Введите a: ");

int a = int.Parse(ReadLine());

Write("Введите b: ");

int b = int.Parse(ReadLine());

WriteLine("Ответ: " + GetMinOrMax(a, b).ToString());

}

static int GetMinOrMax(int a, int b)

{

if (a > 0)

{

if (a > b)

return a;

else

return b;

}

else

{

if (a < b)

return a;

else

return b;

}

}

}

}

Уровень 2:

№1:

using System;

using static System.Console;

namespace L2\_task01

{

class Program

{

static void Main(string[] args)

{

GetAverageAge();

}

static void GetAverageAge()

{

Write("Введите кол-во учеников: ");

int studentCount = int.Parse(ReadLine());

int ageSum = 0;

for (int i = 0; i < studentCount; i++)

{

Write($"Возраст {i + 1}-го ученика: ");

ageSum += int.Parse(ReadLine());

}

double averageAge = (double) ageSum / studentCount;

WriteLine($"Средний возраст учеников = {averageAge} лет.");

}

}

}

№2:

using System;

using static System.Console;

using static System.Math;

namespace L2\_task02

{

class Program

{

static void Main(string[] args)

{

GetPointsInsideCircle();

}

static void GetPointsInsideCircle()

{

WriteLine("Введите коодинаты центра круга: ");

Write("X = ");

double circleCenterX = double.Parse(ReadLine());

Write("Y = ");

double circleCenterY = double.Parse(ReadLine());

Write("Введите радиус круга: ");

double circleRadius = double.Parse(ReadLine());

Write("Введите кол-во точек: ");

int pointCount = int.Parse(ReadLine());

int suitPointCount = 0;

for (int i = 0; i < pointCount; i++)

{

WriteLine($"{1}-я точка:");

Write("X = ");

double x = double.Parse(ReadLine());

Write("Y = ");

double y = double.Parse(ReadLine());

bool isInsideCircle = IsInsideCircle(circleRadius, circleCenterX, circleCenterY, x, y);

WriteLine($"Точка ({x}, {y}) {IntoRussian(isInsideCircle)}");

if (isInsideCircle)

suitPointCount++;

}

WriteLine($"{suitPointCount} точек попадает в круг");

}

static bool IsInsideCircle(double circleRadius, double centerX, double centerY, double x, double y)

{

double xArgument = Pow(x - centerX, 2);

double yArgument = Pow(y - centerY, 2);

bool isInsideCircle = xArgument + yArgument <= Pow(circleRadius, 2);

return isInsideCircle;

}

static string IntoRussian(bool isInsideCircle)

{

if (isInsideCircle)

{

return "попадает в круг";

}

else

{

return "НЕ попадает в круг";

}

}

}

}

№4:

using System;

using static System.Console;

namespace L3\_task04

{

class Program

{

static void Main(string[] args)

{

GetPointsInsideRing();

}

static void GetPointsInsideRing()

{

Write("Введите внутренний радиус кольца: ");

double insideRingRadius = double.Parse(ReadLine());

Write("Введите внешний радиус кольца: ");

double outsideRingRadius = double.Parse(ReadLine());

Write("Введите кол-во точек: ");

int pointCount = int.Parse(ReadLine());

int suitPointCount = 0;

for (int i = 0; i < pointCount; i++)

{

WriteLine($"{1}-я точка:");

Write("X = ");

double x = double.Parse(ReadLine());

Write("Y = ");

double y = double.Parse(ReadLine());

bool isInRing = IsInRing(insideRingRadius, outsideRingRadius, x, y);

WriteLine($"Точка ({x}, {y}) {IntoRussian(isInRing)}");

if (isInRing)

suitPointCount++;

}

WriteLine($"{suitPointCount} точек попадает в кольцо");

}

static bool IsInRing(double insideRadius, double outsideRadius, double x, double y)

{

bool isInRing =

IsInsideCircle(outsideRadius, x, y) &&

!IsInsideCircle(insideRadius, x, y);

return isInRing;

}

static bool IsInsideCircle(double circleRadius, double x, double y)

{

bool isOnCircle = x \* x + y \* y <= circleRadius \* circleRadius;

return isOnCircle;

}

static string IntoRussian(bool isInsideCircle)

{

if (isInsideCircle)

{

return "попадает в кольцо";

}

else

{

return "НЕ попадает в кольцо";

}

}

}

}

Уровень 3:

№11:

using System;

using static System.Console;

namespace L3\_task11

{

class Program

{

static void Main(string[] args)

{

GetNotCatchStudents();

}

static void GetNotCatchStudents()

{

Write("Введите кол-во студентов: ");

int studentCount = int.Parse(ReadLine());

int notCatchStudentCount = 0;

int markSum = 0;

for (int i = 0; i < studentCount; i++)

{

WriteLine($"Студент #{i + 1}:");

Write("Оценка за 1-й экзамен: ");

int mark1 = int.Parse(ReadLine());

Write("Оценка за 2-й экзамен: ");

int mark2 = int.Parse(ReadLine());

Write("Оценка за 3-й экзамен: ");

int mark3 = int.Parse(ReadLine());

Write("Оценка за 4-й экзамен: ");

int mark4 = int.Parse(ReadLine());

bool isNotCatch = IsNotCatch(mark1, mark2, mark3, mark4);

WriteLine($"Данный студент {IntoRussian(isNotCatch)} за программой");

markSum += mark1 + mark2 + mark3 + mark4;

if (isNotCatch)

notCatchStudentCount++;

}

double averageMark = (double) markSum / studentCount / 4d;

WriteLine($"Кол-во неуспевающих студентов = {notCatchStudentCount}");

WriteLine($"Средний балл группы = {averageMark}");

}

static bool IsNotCatch(int mark1, int mark2, int mark3, int mark4)

{

return mark1 <= 2 || mark2 <= 2 || mark3 <= 2 || mark4 <= 2;

}

static string IntoRussian(bool isNotCatch)

{

if (isNotCatch)

return "НЕ успевает";

else

return "успевает";

}

}

}

№12:

using System;

using static System.Console;

using static System.Math;

namespace L3\_task12

{

class Program

{

static void Main(string[] args)

{

GetFigureAreas();

}

static void GetFigureAreas()

{

Write("Введите кол-во фигур: ");

int figureCount = int.Parse(ReadLine());

for (int i = 0; i < figureCount; i++)

{

Write("Введите R: ");

int r = int.Parse(ReadLine());

Write(

"Выберите фигуру, площать которой требуется найти:\n" +

"1 - Квадрат\n" +

"2 - Круг\n" +

"3 - Равносторонний треугольник\n\n> ");

int chosenFigure = int.Parse(ReadLine());

double area;

switch (chosenFigure)

{

case 1:

area = SquareArea(r);

break;

case 2:

area = CircleArea(r);

break;

case 3:

area = TriangleArea(r);

break;

default:

WriteLine("Ошибка: фигуры с таким номером нет");

continue;

}

WriteLine($"Площадь = {area}");

}

}

static double SquareArea(double a)

{

return a \* a;

}

static double CircleArea(double a)

{

return PI \* a \* a;

}

static double TriangleArea(double a)

{

return a \* a \* Sqrt(3) / 4;

}

}

}

№13:

using System;

using static System.Console;

using static System.Math;

namespace L3\_task13

{

class Program

{

static void Main(string[] args)

{

GetFigureAreas();

}

static void GetFigureAreas()

{

Write("Введите кол-во фигур: ");

int figureCount = int.Parse(ReadLine());

for (int i = 0; i < figureCount; i++)

{

Write("Введите A: ");

int a = int.Parse(ReadLine());

Write("Введите B: ");

int b = int.Parse(ReadLine());

Write(

"Выберите фигуру, площать которой требуется найти:\n" +

"1 - Прямоугольник\n" +

"2 - Кольцо\n" +

"3 - Равнобедренный треугольник\n\n> ");

int chosenFigure = int.Parse(ReadLine());

double area;

switch (chosenFigure)

{

case 1:

area = RectangleArea(a, b);

break;

case 2:

area = RingArea(a, b);

break;

case 3:

area = TriangleArea(a, b);

break;

default:

WriteLine("Ошибка: фигуры с таким номером нет");

continue;

}

WriteLine($"Площадь = {area}");

}

}

static double RectangleArea(double a, double b)

{

return a \* b;

}

static double RingArea(double a, double b)

{

return Abs(PI \* (a \* a - b \* b));

}

static double TriangleArea(double a, double b)

{

double p = (a + b + b) / 2;

return Sqrt(p \* (p - a) \* (p - b) \* (p - b));

}

}

}