Ćwiczenie 1:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_1

{

internal class Program

{

public struct Point2D

{

public double X { get; set; }

public double Y { get; set; }

public Point2D(double x, double y) {

X = x;

Y = y;

}

public void Reset() {

X = 0;

Y = 0;

}

public void IncrX(double dx) {

X += dx;

}

public void IncrY(double dy){

Y += dy;

}

public void Print2DPoint(){

Console.WriteLine($"(X={X} Y={Y})");

}

public double Dist(Point2D point){

double dist = Math.Sqrt(Math.Pow(point.X - X, 2) + Math.Pow(point.Y - Y, 2));

return dist;

}

}

static void Main(string[] args) {

double rad = 4;

List<Point2D> pointslist = new List<Point2D>();

for (int i = 0; i < 4; i++) {

Console.WriteLine("Wprowadz dwie wspolrzedne po spacji i nacisnij ENTER: ");

string[] parts = Console.ReadLine().Split(' ');

double x = double.Parse(parts[0]);

double y = double.Parse(parts[1]);

var point = new Point2D(x, y);

pointslist.Add(point);

}

while (true) {

Console.WriteLine("Wprowadz dwie wspolrzedne ostatniego punktu: ");

string[] parts\_5 = Console.ReadLine().Split(' ');

double x\_5 = double.Parse(parts\_5[0]);

double y\_5 = double.Parse(parts\_5[1]);

var point\_5 = new Point2D(x\_5, y\_5);

double min\_dist = double.MaxValue;

foreach (var p in pointslist) {

double dist = p.Dist(point\_5);

if (dist < min\_dist) {

min\_dist = dist;

}

}

if (point\_5.X < 0 ^ point\_5.Y < 0) {

Console.WriteLine("Współrzędne punktów to:");

foreach (var p in pointslist) {

p.Print2DPoint();

}

point\_5.Print2DPoint();

Console.WriteLine("KONIEC");

Console.ReadLine();

break;

}

else if (min\_dist < rad) {

Console.WriteLine("Współrzędne punktów to:");

foreach (var p in pointslist) {

p.Print2DPoint();

}

point\_5.Print2DPoint();

Console.WriteLine("KONIEC");

Console.ReadLine();

break;

}

else

{

Console.WriteLine($"Najblizszy punkt jest w odleglosci: {min\_dist}");

}

}

}

}

}

Ćwiczenie 2:

1. Wywołanie metody ToString() wypisuje „(0, 0)”.
2. Struktura zostaje utworzona
3. Wynik kompilacji:  
   Funkcja „Konstruktor struktury bez parametrów” nie jest dostępna w języku C# 7.3 , Użyj języka w wersji 10.0 lub noweszej.
4. Funkcja „Inicjatory pola struktury” nie jest dostępna w języku C# 7.3 …

Ćwiczenie 3:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_3

{

internal class Program

{

static void Main(string[] args)

{

int intCount = 0;

int floatCount = 0;

int stringCount = 0;

while (true)

{

Console.WriteLine("Wprowadź dane (lub -1 aby zakończyć): ");

string input = Console.ReadLine();

if (input == "-1")

{

break;

}

if (int.TryParse(input, out \_))

{

intCount++;

}

if (float.TryParse(input, out \_))

{

floatCount++;

}

else

{

stringCount++;

}

}

Console.WriteLine($"Int: {intCount}");

Console.WriteLine($"Float: {floatCount}");

Console.WriteLine($"String: {stringCount}");

Console.ReadLine();

}

}

}

Ćwiczenie 4:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_4

{

internal class Program

{

public enum DzienTygodnia

{

Poniedziałek = 1,

Wtorek = 2,

Środa = 3,

Czwartek = 4,

Piątek = 5,

Sobota = 6,

Niedziela = 7

}

public enum RozmiarLiczby

{

mała,

średnia,

duża

}

static void Main(string[] args)

{

Console.WriteLine("Wprowadź liczbę z zakresu 1-7:");

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

Console.WriteLine((DzienTygodnia)input);

Console.ReadLine();

Console.WriteLine("Część 2: \nWprowadź dowolną liczbę:");

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

RozmiarLiczby rozmiar;

if (liczba < 10)

{

rozmiar = RozmiarLiczby.mała;

}

else if (liczba < 100)

{

rozmiar = RozmiarLiczby.średnia;

}

else

{

rozmiar = RozmiarLiczby.duża;

}

Console.WriteLine($"Liczba {liczba} jest {rozmiar}.");

Console.ReadLine() ;

}

}

}

Ćwiczenie 5:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_5

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Wprowadź jeden znak: ");

char input = Console.ReadKey().KeyChar;

if ("aeąęiouy".IndexOf(input) >= 0)

{

Console.WriteLine("\nsamogłoska");

}

else if (char.IsDigit(input))

{

Console.WriteLine("\ncyfra");

}

else

{

Console.WriteLine("\ninny znak");

}

Console.ReadLine();

}

}

}

Ćwiczenie 6:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_6

{

internal class Program

{

static void Main(string[] args)

{

string input = Console.ReadLine();

Console.WriteLine(string.Join(" ", input.ToCharArray()));

Console.ReadLine();

}

}

}

Ćwiczenie 7:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_7

{

internal class Program

{

static void Main(string[] args)

{

//int liczba = 9999999999; // błąd przepełnienia

try

{

int liczba = int.MaxValue;

checked

{

liczba = liczba + 1;

}

}

catch (OverflowException ex)

{

Console.WriteLine("Błąd przepełnienia: " + ex.Message);

}

Console.ReadLine();

}

}

}

Ćwiczenie 8:

1. Dla Point (class) pola x i y zmieniły swoje wartości, natomiast dla Cord (struct) obiekt nie zmienił pól.
2. Wyniki wynoszą:
   1. False
   2. False
   3. False
   4. False

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_8

{

internal class Program

{

public struct Coords

{

public int x, y;

public Coords(int p1, int p2)

{

x = p1;

y = p2;

}

}

public class Point

{

public int x, y;

public Point(int p1, int p2)

{

x = p1;

y = p2;

}

}

static void Fun1(Point p)

{

p.x = 3;

p.y = 4;

}

static void Fun2(Coords c)

{

c.x = 3;

c.y = 4;

}

static void Main(string[] args)

{

var p = new Point(1, 2);

var c = new Coords(1, 2);

// --------- 1 -----------

//Fun1(p: p);

//Fun2(c: c);

//Console.WriteLine($"class: {p.x}, {p.y}");

//Console.WriteLine($"struct: {c.x}, {c.y}");

//Console.ReadLine();

// --------- 2 -----------

Console.WriteLine(Object.Equals(p, c));

Console.WriteLine(c.Equals(p));

Console.WriteLine(p.Equals(c));

Console.WriteLine(Object.ReferenceEquals(p, c));

Console.ReadLine();

}

}

}

Ćwiczenie 9:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace cw\_9 {

internal class Program

{

static bool IsValid(string input)

{

foreach (char c in input)

{

if (!char.IsDigit(c) && c != '+' && c != '-')

{

return false;

}

}

return true;

}

static int ApplyOperation(int result, char operation, int number)

{

if (operation == '+')

{

return result + number;

}

else if (operation == '-')

{

return result - number;

}

else

{

throw new InvalidOperationException("Nieznany operator.");

}

}

static int Calculate(string input)

{

int result = 0;

string number = string.Empty;

char sign = '+';

foreach (char c in input)

{

if (char.IsDigit(c))

{

number += c;

}

else

{

if (!string.IsNullOrEmpty(number))

{

result = ApplyOperation(result, sign, int.Parse(number));

number = string.Empty;

}

sign = c;

}

}

if (!string.IsNullOrEmpty(number))

{

result = ApplyOperation(result, sign, int.Parse(number));

}

return result;

}

static void Main(string[] args)

{

Console.WriteLine("Wprowadź wyrażenie do obliczenia (np. 12+2-3): ");

string input = Console.ReadLine();

if (!IsValid(input))

{

Console.WriteLine("Błąd: Wyrażenie zawiera niepoprawne znaki.");

Console.ReadLine();

return;

}

int result = Calculate(input);

Console.WriteLine($"Wynik: {result}");

Console.ReadLine();

}

}

}

Ćwiczenie 10:

Implicite:

* Z typu *int* na *long:*int a = 10;

long b = a;

* Z typu *float* na *double*:  
  float c = 3.14f;  
  double d = c;
* Z typu *char* na *int*:  
  char ch = 'A';  
  int asciiValue = ch;
* Z typu *byte* na *int*:  
  byte smallNumber = 100;  
  int biggerNumber = smallNumber;

Explicite:

* Z typu *double* na *int:*double e = 9.99;  
  int f = (int)e;
* Z typu *long* na *int:*  
  long largeNumber = 1000000;  
  int smallerNumber = (int)largeNumber;
* Z typu *int* na *byte:*  
  int g = 255;  
  byte h = (byte)g;
* Z typu *object* na *string:*  
  object obj = "Hello";  
  string str = (string)obj;