**КУРСОВА РАБОТА**

Тема: Създаване на клас Workout за проследяване

на фитнес активност

Разработил: Ваня Тодорова Ванева

ф. номер: 22472128

курс: Първи

специалност: СКИ

Дата: ......................

# Теоретична част

## Разлика между методи и конструктори в C#

Метод - блок от код, който изпълнява определена задача. Той се дефинира вътре в клас и може да приема параметри, да връща стойности и да бъде извикван многократно. Ползва се за обработка на данни, изпълнение на действия, организиране на кода в по-малки логически части.

Конструктор - специален метод, който се извиква автоматично при създаване на нов обект от клас. Има същото име като класа. Няма тип на връщане. Може да има параметри. Основната му задача е да инициализира полетата на обекта.

|  |  |  |
| --- | --- | --- |
| Разлики: | Метод | Конструктор |
| Име | Произволно | Същото име като класа |
| Тип на връщане | Има (напр. void, int, …) | Няма |
| Извикване | Ръчно | Автоматично |
| Предназначение | Изпълнява действия или изчисления | Инициализира стойностите на полетата |
| Многократно извикване | Да | Да, но само при създаване на нов обект |

Таблица 1

## Обработка на масиви в C#

Масив в C# е структура от данни, която съдържа фиксиран брой елементи от един и същ тип. Масивите се използват за съхранение и обработка на множество стойности под общо име. Масивите в C# започват от индекс 0, т.е. първият елемент е достъпен чрез индекс 0, вторият чрез 1 и т.н.

### **Деклариране и инициализиране на масив**

int[] numbers = new int[5]; // масив с 5 елемента (стойности по подразбиране)

string[] names = { "Иван", "Мария", "Петър" }; // инициализиран с конкретни стойности

### **Обхождане на масив**

С for цикъл:

for (int i = 0; i < numbers.Length; i++)

{

Console.WriteLine(numbers[i]);

}

С foreach:

foreach (string name in names)

{

Console.WriteLine(name);

}

### **Типични операции върху масиви**

|  |  |
| --- | --- |
| Присвояване на стойност | numbers[0] = 42; |
| Достъп до елемент | Console.WriteLine(numbers[1]); |
| Дължина на масив | int len = numbers.Length; |
| Сумиране на елементи | int sum = numbers.Sum(); |
| Сортиране | Array.Sort(numbers); |
| Обръщане на реда | Array.Reverse(numbers); |

Таблица 2

## Използвана литература

Светлин Наков и колектив. Основи на програмирането със C#. Faber Publishing, София, май 2017 г.

Светлин Наков, Веселин Колев и колектив. Принципи на програмирането със C#. Фабер, Велико Търново, 2018 г.

Конструктор (обектно ориентирано програмиране). Уикипедия: Свободната енциклопедия. Последна редакция на 31 март 2025 г. Дата на достъп: 2 май 2025 г. Достъпно на: <https://bg.wikipedia.org/wiki/Конструктор_(обектно_ориентирано_програмиране)>

# Практическо изпълнение

## Програмен код

using System;

namespace courseWork

{

public class Workout

{

private string name, type;

private int duration, calories;

public Workout(string name, string type, int duration, int calories)

{

this.name = name;

this.type = type;

this.duration = duration;

this.calories = calories;

}

public string Name { get { return name; } set { name = value; } }

public int Calories { get { return calories; } set { calories = value; } }

public void Output()

{

Console.WriteLine($"\t{name} - {type} - {duration} mins - {calories} Kcals");

}

}

internal class Program

{

static void Main(string[] args)

{

Console.Write("Enter number of workouts for the week: ");

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

Workout[] workouts = new Workout[n];

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

{

Console.WriteLine($"\nEnter details for workout {i + 1}");

Console.Write("\tName: ");

string name = Console.ReadLine();

Console.Write("\tType: ");

string type = Console.ReadLine();

Console.Write("\tDuration(in mins): ");

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

Console.Write("\tKcals burned: ");

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

workouts[i] = new Workout(name, type, duration, calories);

Console.WriteLine();

}

CalsAsc(workouts);

Workout maxCals = workouts[workouts.Length - 1];

double avgCals = AvgCals(workouts);

Console.WriteLine($"Burned most Kcals during {maxCals.Name} - {maxCals.Calories}!");

Console.WriteLine($"Average Kcals burned - {avgCals:F2}\n");

Console.WriteLine("List of this week's workouts");

Console.WriteLine("\tname - type - duration - calories");

foreach(var w in workouts)

{

w.Output();

}

}

static void CalsAsc(Workout[] workouts)

{

Array.Sort(workouts, (w1, w2) => w1.Calories.CompareTo(w2.Calories));

}

static double AvgCals(Workout[] workouts)

{

double total = 0;

foreach(var w in workouts)

{

total += w.Calories;

}

return total / workouts.Length;

}

}

}

## Описание на решението

Клас Workoutсъдържа частните полета име, тип, продължителност и калории. Използва конструктор, който инициализира всички полета при създаване на нов обект. Има свойства само за име и калории и метод за извеждане на информация за тренировка.

В главния метод се създава масив от обекти Workout, като размерът се определя чрез вход от конзолата. За всяка тренировка се въвеждат необходимите данни (име, тип, продължителност и калории), които се използват за създаване на нов обект от тип Workout. Обектът се съхранява в масива.

Методът CalsAsc() сортира масива по броя изгорени калории във възходящ ред, използвайки Array.Sort() с lambda израз:

Array.Sort(workouts, (w1, w2) => w1.Calories.CompareTo(w2.Calories));

Този израз сравнява стойностите на калориите между два обекта Workout и ги подрежда по нарастваща стойност. Така тренировката с най-малко калории се оказва първа, а тази с най-много – последна.

Най-голямото количество изгорени калории се определя чрез последния елемент в сортирания вече масив (workouts[workouts.Length - 1).

Методът AvgCals изчислява средния брой изгорени калории за седмицата. Той използва цикъл foreach, с който се обхождат всички елементи от масива и се събират стойностите на калориите. След като се изчисли общата сума, тя се разделя на броя тренировки, за да се получи средната стойност.

След сортирането се отпечатва тренировката с най-много изгорени калории, средните калории за седмицата и списък с всички въведени тренировки.

## Екрани

