**8 Перечисления. Структуры.**

Задание №1. Описать структуру с именем ZNAK, содержащую следующие поля:

фамилия;

имя;

знак Зодиака;

дата рождения (массив из трех чисел).

Написать программу, выполняющую следующие действия:

ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ZNAK (записи должны быть упорядочены по дате рождения);

вывод на экран информации о людях, родившихся под знаком, название которого введено с клавиатуры (если таких нет, вывести соответствующее сообщение).

Листинг программы:

class Program

{

static void Main(string[] args)

{

SortedSet<Znak> set = new SortedSet<Znak>(new ZnakComparer());

for (int i = 0; i < 3; i++)

{

Console.Write("Enter name: ");

string name = Console.ReadLine();

Console.Write("Enter surname: ");

string surname = Console.ReadLine();

Console.Write("Enter Zodiac sign: ");

string sign = Console.ReadLine();

Console.WriteLine("Enter information about birthday: ");

Console.Write("Enter day of birth: ");

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

Console.Write("Enter month of birth: ");

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

Console.Write("Enter year of birth: ");

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

int[] bornDate = new int[] { day, month, year };

set.Add(new Znak(name, surname, sign, bornDate));

}

while (true)

{

Console.WriteLine("Enter Zodiac sign (enter '0' to exit): ");

string zodiac = Console.ReadLine();

List<Znak> result = (from n in set where n.zodiac == zodiac select n).ToList<Znak>();

if (result.Count != 0)

{

foreach (var n in result)

{

Console.WriteLine(n.ToString());

}

}

else if (zodiac == "0")

{

break;

}

else { Console.WriteLine($"No data found with sign '{zodiac}''!"); }

}

}

}

public struct Znak

{

public string zName;

public string zSurname;

public string zodiac;

public int[] Birth;

public Znak(string name, string surname, string zod, int[] date)

{

zName = name;

zSurname = surname;

zodiac = zod;

Birth = new int[date.Length];

Birth = date;

}

public override string ToString()

{

return string.Format("Name: {0}, Surname: {1}, Zodiac sign: {2}, Date of birth: {3}.{4}.{5}",

zName, zSurname, zodiac, Birth[0], Birth[1], Birth[2]);

}

}

class ZnakComparer : IComparer<Znak>

{

public int Compare(Znak x, Znak y)

{

if (x.Birth[0] > y.Birth[0])

return 1;

if (x.Birth[0] < y.Birth[0])

return -1;

else

return 0;

}

}

Таблица 8.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Gemini | Name: Violet, Surname^ Evergarden, Zodiac sign: Gemini, Date of birth: 14.10.2001 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 8.1.

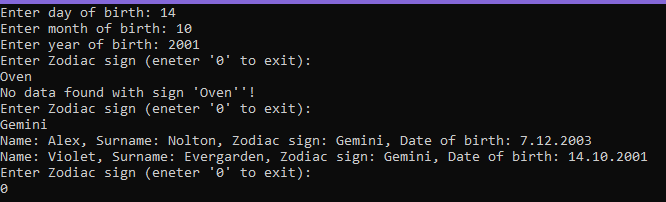


Рисунок 8.1 – Результат работы программы

Источник: собственная разработка

Задание №2. Создайте перечисление, в котором будут содержаться должности сотрудников как имена констант. Присвойте каждой константе значение, задающее количество часов, которые должен отработать сотрудник за месяц.

Листинг программы:

public enum PositionWorkTime : int

{

CEO = 101,

Accountant = 45,

Designer = 66,

Art\_Manager = 69,

Admin = 89,

Merchandiser = 55,

Ceo\_Deputy = 90,

}

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Employee work time: ");

foreach (int i in Enum.GetValues(typeof(PositionWorkTime)))

Console.WriteLine($"{Enum.GetName(typeof(PositionWorkTime), i)}, {i}" + " hrs");

}

}

Таблица 8.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Ceo, accountant, designer, art-manager, Admin, Merchandiser, Ceo-Deputy | 101, 45, 66, 69, 89, 55, 90 |

Источник: собственная разработка

Анализ результатов представлен на рисунке 8.2.

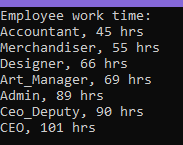


Рисунок 8.2 – Результат работы программы

Источник: собственная разработка