**8 СТРУКТУРЫ, ПЕРЕЧИСЛЕНИЯ**

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 знакам зодиака); вывод на экран информации о людях, родившихся в месяц, значение которого введено с клавиатуры (если таких нет, вывести соответствующее сообщение).

Листинг программы:

void MounthU(SortedSet<Znak> set) // метод для вывода записи соответствующего месяца, который введен с клавиатуры

{

Console.WriteLine("Введите месяц:");

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

List<Znak> result = (from n in set where n.dateOfBirth[1] == month select n).ToList<Znak>(); // выборка записей, соответствующих дате, введенной с клавиатуры

if (result.Count != 0) //проверка и переборка записей

{

foreach (var n in result)

{

Console.WriteLine(n.ToString());

}

}

else { Console.WriteLine("Данные отсутствуют!"); }

}

void Print(SortedSet<Znak> set) //Вывод всех записей

{

foreach (Znak n in set)

{

Console.WriteLine(n.ToString());

}

}

try

{

SortedSet<Znak> set = new SortedSet<Znak>(); //создание сортированного массива(множества)

Console.Write("Введите имя = ");

string fnameU = Console.ReadLine();

Console.Write("Введите фамилию = ");

string lnameU = Console.ReadLine();

Console.Write("Введите знак зодиака = ");

string zodiakU = Console.ReadLine();

Console.Write("Введите день рождения = ");

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

Console.Write("Введите месяц рождения = ");

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

Console.Write("Введите год рождения = ");

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

int[] dateU = new int[] { dayU, monthU, yearU };

set.Add(new Znak(fnameU, lnameU, zodiakU, dateU));

set.Add(new Znak("Alex", "Barly", "Scorpion", new int[] { 12, 11, 1981 })); //добавление записей

set.Add(new Znak("Nick", "Simpson", "Libra", new int[] { 10, 10, 1995 }));

set.Add(new Znak("Mary", "Douglas", "Scorpion", new int[] { 25, 11, 2005 }));

Console.WriteLine("------------------------------------------------------");

Print(set);

Console.WriteLine("------------------------------------------------------");

MounthU(set);

Console.ReadLine();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

public struct Znak:IComparable

{

public string firstName;

public string lastName;

public string zodiak;

public int[] dateOfBirth;

public Znak(string fName, string lName, string z, int[] date)

{

firstName = fName;

lastName = lName;

zodiak = z;

//dateOfBirth = new int[date.Length];

dateOfBirth = date;

}

public int CompareTo(object? obj) //Сортировка

{

var compareZnak=(Znak) obj;

if (zodiak.CompareTo(compareZnak.zodiak) == 0)

{

var dateX = new DateTime(dateOfBirth[2], dateOfBirth[1], dateOfBirth[0]);

var datey = new DateTime(compareZnak.dateOfBirth[2], compareZnak.dateOfBirth[1], compareZnak.dateOfBirth[0]);

return dateX.CompareTo(datey);

}

else

{

return zodiak.CompareTo(compareZnak.zodiak); //Сортировка по знаку зодиака

}

}

public override string ToString() //для корректного вывода информации о записи

{

return string.Format("Имя: {0}, фамилия: {1}, знак Зодиака: {2}, дата рождения: {3}.{4}.{5}",

firstName, lastName, zodiak, dateOfBirth[0], dateOfBirth[1], dateOfBirth[2]);

}

}

Таблица 8.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Введите имя = n  Введите фамилию = n  Введите знак зодиака = a  Введите день рождения = 1  Введите месяц рождения = 1  Введите год рождения = 2000  Введите месяц:  1 | ------------------------------------------------------  Имя: n, фамилия: n, знак Зодиака: a, дата рождения: 1.1.2000  Имя: Nick, фамилия: Simpson, знак Зодиака: Libra, дата рождения: 10.10.1995  Имя: Alex, фамилия: Barly, знак Зодиака: Scorpion, дата рождения: 12.11.1981  Имя: Mary, фамилия: Douglas, знак Зодиака: Scorpion, дата рождения: 25.11.2005  ------------------------------------------------------  Имя: n, фамилия: n, знак Зодиака: a, дата рождения: 1.1.2000 |

Анализ результатов:

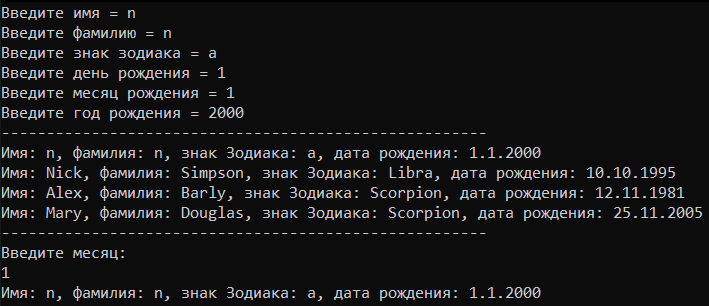


Рисунок 8.1 – Результаты работы программы

Источник: собственная разработка

Задание №2. Создайте класс Accauntant с методом bool AskForBonus (Post worker, int hours), отражающее давать или нет сотруднику премию. Если сотрудник отработал больше положенных часов в месяц, то ему положена премия.

Листинг программы:

try

{

int value = 1;

while (value == 1)

{

Console.WriteLine("Выберите работника:" +

'\n' + "1. Admin" +

'\n' + "2. Worker" +

'\n' + "3. Manager" +

'\n' + "4. Accountant");

int choice = Int32.Parse(Console.ReadLine());

Accountant acc = new Accountant();

switch (choice)

{

case 1:

acc.worker = Post.Admin;

break;

case 2:

acc.worker = Post.Worker;

break;

case 3:

acc.worker = Post.Manager;

break;

case 4:

acc.worker = Post.Accountant;

break;

}

Console.WriteLine("Введите количество отработанных часов: ");

int hours = Int32.Parse(Console.ReadLine());

if (Accountant.AskForBonus(acc.worker, hours))

Console.WriteLine("Сотруднику положена премия");

else Console.WriteLine("Сотруднику не положена премия");

Console.WriteLine("Хотите продолжить?" +

'\n' + "1. Да" +

'\n' + "2. Нет");

value = Int32.Parse(Console.ReadLine());

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

enum Post

{

Admin = 50,

Worker = 40,

Manager = 44,

Accountant = 45

}

class Accountant

{

public Post worker;

public static bool AskForBonus(Post worker, int hours)

{

if ((int)worker <= hours)

return true;

else return false;

}

}

Таблица 8.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Работник: 4  Введите количество отработанных часов:  100 | Сотруднику положена премия |

Анализ результатов:

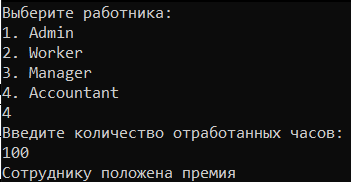


Рисунок 8.2 – Результаты работы программы

Источник: собственная разработка