**8 Перечисления. Структуры**

Задание 1. Описать структуру с именем WORKER, содержащую следующие поля: фамилия и инициалы работника; название занимаемой должности; год поступления на работу. Написать программу, выполняющую следующие действия: ввод с клавиатуры данных в массив, состоящий из десяти структур типа WORKER (записи должны быть упорядочены по алфавиту); вывод на экран фамилий работников, стаж работы которых превышает значение, введенное с клавиатуры (если таких работников нет, вывести соответствующее сообщение).

Листинг программы:

const int NUM\_WORKERS = 3;

WORKER[] workers = new WORKER[NUM\_WORKERS];

for (int i = 0; i < NUM\_WORKERS; i++) {

Console.WriteLine("Введите данные работника " + (i + 1) + ":");

Console.Write("Фамилия Имя: "); workers[i].Name = Console.ReadLine();

Console.Write("Должность: "); workers[i].post = Console.ReadLine();

Console.Write("Год поступления на работу: ");

workers[i].YearOfEmployment = int.Parse(Console.ReadLine()); Console.WriteLine();

Array.Sort(workers, delegate (WORKER x, WORKER y)

{return x.Name.CompareTo(y.Name);});

Console.WriteLine("Сортированные данные:");

for (int i = 0; i < workers.Length; i++) // или (int i = 0; i < 3; i++){

Console.WriteLine("Фамилия Имя: " + workers[i].Name + ", Должность: " + workers[i].post + ", Год поступления: " + workers[i].YearOfEmployment);}

Console.WriteLine();

Console.Write("Введите стаж для просмотра превышающего значения: ");

int minYears = int.Parse(Console.ReadLine()); Console.WriteLine();

bool workerFound = false; for (int i = 0; i < workers.Length; i++)

{ int yearsOfEmployment = DateTime.Now.Year - workers[i].YearOfEmployment;

if (yearsOfEmployment > minYears)

{ Console.WriteLine(workers[i].Name + " имеет стаж работы " + yearsOfEmployment + " лет.");

workerFound = true;}}

if (!workerFound)

{Console.WriteLine("Нет работников со стажем больше, чем " + minYears + " лет.");}

Таблица 8.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Иванов ИИ, Дизайнер, 2010;  Варченя АС, Фотограф, 2019;  Букас КД, Исполнитель, 2022.  5 | Сортированные данные:  Фамилия Имя: Букас КД, Должность: Исполнитель, Год поступления: 2022  Фамилия Имя: Варченя АС, Должность: Фотограф, Год поступления: 2019  Фамилия Имя: Иванов ИИ, Должность: Дизайнер , Год поступления: 2010  Введите стаж для просмотра превышающего значения: 5  Иванов ИИ имеет стаж работы 13 лет. |

Источник: собственная разработка

Анализ результатов:

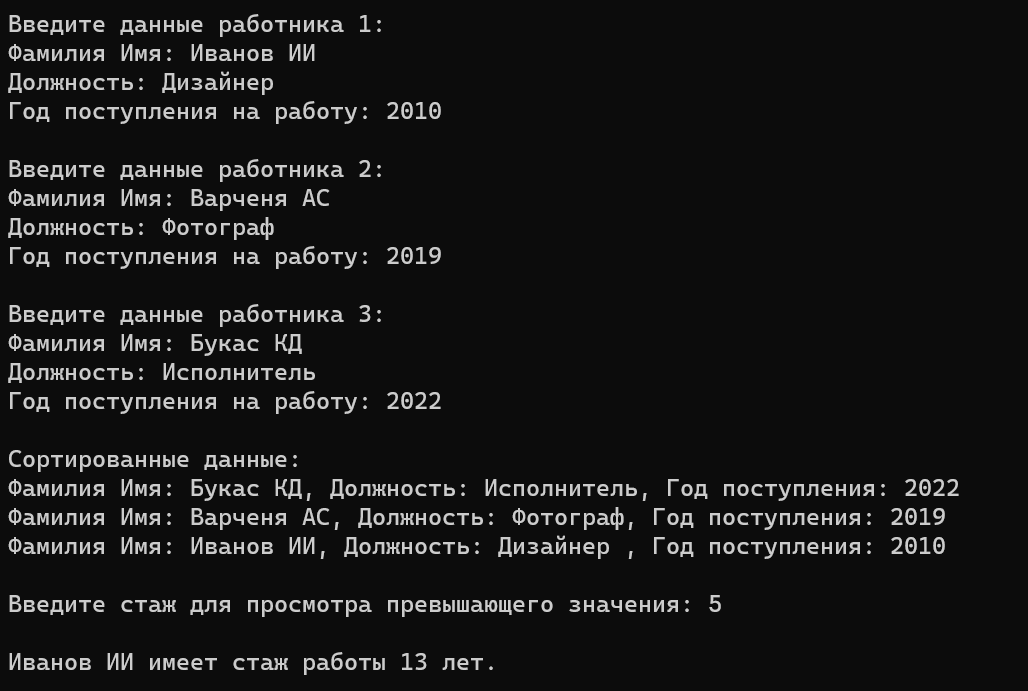


Рисунок 8.1 – Результат работы программы

Источник: собственная разработка

Задание 2.

1. Создайте перечисление, в котором будут содержаться должности сотрудников как имена констант. Присвойте каждой константе значение, задающее количество часов, которые должен отработать сотрудник за месяц.

2. Создайте класс Accauntant с методом bool AskForBonus (Post worker, int hours), отражающее давать или нет сотруднику премию. Если сотрудник отработал больше положеных часов в месяц, то ему положена премия.

Листинг программы:

enum Post{Менеджер=16,Программист=20,Дизайнер=28,Фотограф=39 }

class Accauntant{ public bool AskForBonus(Post post, int hours) => (int)post > hours; }

class Program{ static void Main(string[] args) {

Console.WriteLine("Введите количество часов, начиная с которого сотруднику положена премия: "); int hours = int.Parse(Console.ReadLine());

Accauntant accauntant = new Accauntant(); bool bonusIsGiven = false;

foreach (Post post in Enum.GetValues(typeof(Post))){

if (accauntant.AskForBonus(post, hours)){

Console.WriteLine($"Премия положена для должности {post}");

bonusIsGiven = true;}}

if (!bonusIsGiven){

Console.WriteLine("Премия не положена для ни одной должности");}}}

Таблица 8.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 25 | Премия положена для должности Дизайнер  Премия положена для должности Фотограф |

Источник: собственная разработка

Анализ результатов:

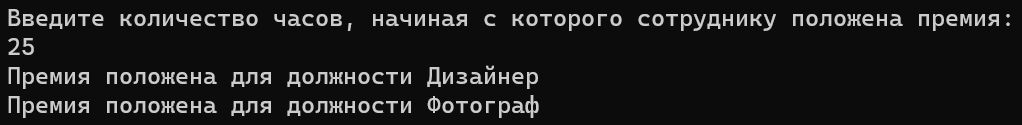


Рисунок 8.2 – Результат работы программы

Источник: собственная разработка