

Обработка структурных данных

Структура – это одна или несколько переменных (возможно, различных типов), которых для удобства работы с ними сгруппировали под одним именем.

Структуры помогают в организации сложных данных, поскольку позволяют группу связанных между собой переменных трактовать не как множество отдельных элементов, а как единое целое

Данные

Строка таблицы – запись

Столбец таблицы – поле

	Страна	Средняя продолжительность жизни	Население, чел.	Площадь, км ²
	Австралия	80.7	22450000	7686850
	Австрия	79.3	8373000	83871
	Азербайджан	66.3	8997400	86600
	Албания	77.75	3195000	28748
	Алжир	73.45	35423000	2381740
Тип	string	float	int	int
Имя переменной	Nazv	PrG	Nas	Pl

Для определения структурного типа используется ключевое слово **struct**:

struct имя_структурного_типа

{ описание поля 1; ... описание поля n; } [одна или более переменных];

Поле описывается как переменная стандартного типа.

Переменная структурного типа

Объявление переменной

Переменные структурного типа объявляются так же, как и переменные стандартных типов:

```
struct Strana { string Nazv; float PrG; int Nas; int Pl; };  
Strana L;
```

или

```
struct Strana { string Nazv; float PrG; int Nas; int Pl ;} L;
```

Инициализация переменной

Переменную структурного типа можно инициализировать одновременно с объявлением.

```
Strana L={«Алжир», 73.45, 35423000, 2381740};
```

Доступ к значениям полей

Доступ к значениям полей структурного типа осуществляется с помощью операции **точка**.

Имя_переменной_структурного_типа . имя_поля	L.Nazv L.Pl
---------------------------------------------	----------------

Массивы структурного типа

Массивы структурного типа определяются так же, как и массивы стандартных типов.

Имя_структурного_типа имя_переменной [количество_элементов_массива];

```
struct Strana { string Nazv; float PrG; int Nas; int Pl; };  
Strana M[200];
```

или

```
struct Strana { string Nazv; float PrG; int Nas; int Pl; } M[200];
```

Нумерация элементов массива начинается с нулевой компоненты. Доступ к элементам массива структурного типа также осуществляется с использованием операции точка. Например:

M[3].Nazv

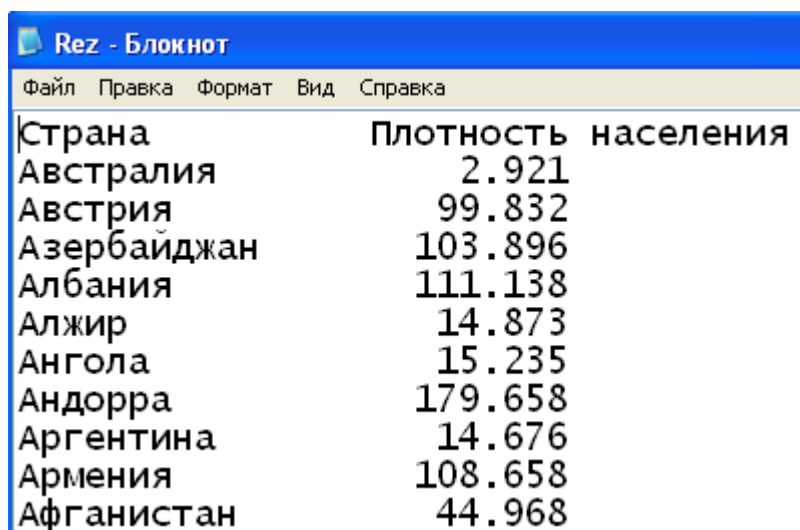
M[3].Pl

Задание

Входные данные для программы содержатся в файле **strana.txt**

Составить программу, которая

- создает структуру данных Strana с переменными соответствующего типа;
- заполняет эту структуру данными из файла;
- рассчитывает для всех стран плотность населения (чел. / км²);
- выводит результаты вычислений в файл Plotn.txt



Страна	Плотность населения
Австралия	2.921
Австрия	99.832
Азербайджан	103.896
Албания	111.138
Алжир	14.873
Ангола	15.235
Андорра	179.658
Аргентина	14.676
Армения	108.658
Афганистан	44.968

- выводит на экран название, площадь и население страны, имеющей наибольшую продолжительность жизни населения