

Лабораторная работа №12 по дисциплине "Программирование"

ТЕМА: Динамические массивы.

СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ПАПОК

1. В Вашей папке создайте папку с именем П_лр12, совпадающим с именем файла с этим заданием, и затем скопируйте файл с заданием в эту папку.
2. При выполнении задания создавайте в папке П_лр12 для каждой программы папку с именем, по смыслу соответствующим действиям, выполняемым составляемой программой, и сохраняйте все файлы, имеющие отношение к этой программе, в этой созданной папке (имена сохраняемых файлов должны отличаться от имен, присваиваемых по умолчанию, и должны по смыслу соответствовать действиям, выполняемым составляемой программой, причем рекомендуется, чтобы имя модуля отличалось от имени проекта цифрой, соответствующей номеру версии модуля).
3. ПРИ ВЫПОЛНЕНИИ СЛЕДУЮЩИХ ЛАБОРАТОРНЫХ РАБОТ ДЕЙСТВУЙТЕ АНАЛОГИЧНО ПРЕДЫДУЩЕМУ.

ДИНАМИЧЕСКИЕ МАССИВЫ

4. Динамические массивы отличаются от статических тем, что при их объявлении (описании) не указывается число элементов массива и место для их размещения не отводится. В связи с этим динамические массивы следует использовать тогда, когда объемы массивов заранее неизвестны и определяются в процессе выполнения программы.
Формат объявления (описания) типа динамического массива:
<Имя> **Array of** <Тип элементов>
5. Примеры объявлений (описаний) динамических массивов:
Var a: **Array of** Integer; //Одномерный динамический массив вещественных чисел
b: **Array of Array of** Real; //Двумерный динамический массив целых чисел
6. Перед использованием динамического массива следует задать его размер процедурой **SetLength**, в качестве аргументов которой указываются имя массива и целые значения, определяющие размеры массива.
Например, при выполнении оператора процедуры
SetLength(a,10);
для одномерного динамического массива a будет выделена память для размещения 10 элементов этого массива и заданы нулевые значения этих элементов.
Индексы динамического массива являются целыми числами, начинающимися с 0.
В приведенном выше примере динамический массив a содержит 10 элементов от a[0] до a[9] включительно.
При выполнении оператора процедуры
SetLength(b, i, k);
для двумерного динамического массива b будет выделена память для размещения i·k элементов этого массива и заданы нулевые значения этих элементов.
7. При записи операторов элементы динамических массивов имеют такой же вид, как и элементы статических массивов.
8. При повторном применении процедуры **SetLength** к уже существующему в памяти динамическому массиву его размер изменяется.
Если новое значение размера больше предыдущего, то все значения элементов

сохраняются и в конце массива добавляются новые нулевые элементы. Если новое значение размера меньше предыдущего, то в массиве остаются только значения первых элементов (массив усекается).

9. Удалить из памяти динамический массив (например, массив *a*) можно:
или присвоением ему значения **nil**:
 `a:=nil;`
или использованием функции **Finalize**:
 `Finalize(a);`
или использованием процедуры **SetLength** для задания нулевого размера массива:
 `SetLength(a,0);`

ПРОГРАММА 12-1

10. Создайте консольное приложение, при выполнении которого происходит то же, что и при выполнении ПРОГРАММЫ 9-1, но с использованием одномерного динамического массива (разумеется, с нижней границей индексов, равной нулю).

ПРОГРАММА 12-2

11. Создайте консольное приложение, при выполнении которого происходит то же, что и при выполнении ПРОГРАММЫ 12-1, но с вводом размера динамического массива.

ПРОГРАММА 12-3

12. Создайте консольное приложение, при выполнении которого происходит то же, что и при выполнении ПРОГРАММЫ 11-1, но с использованием двумерного динамического массива (разумеется, с нижними границами индексов, равными нулю) и с вводом размеров массива.

ОТЧЕТ О ВЫПОЛНЕНИИ РАБОТЫ

13. Продемонстрируйте тексты созданных программ, а также их выполнение в среде программирования и независимо от нее.
-