

Министерство образования и науки Российской Федерации  
Уфимский государственный авиационный технический университет

Кафедра информатики

Пояснительная записка по дисциплине «Информатика»

К курсовой работе:

**“Исследование методов сортировки массивов”**

Выполнил: ст.гр.АД-127.Хакимов А.Р

Проверил: Лесков А.А.

Уфа – 2007

# Введение

Очень часто в жизни, а также на серьезных производствах приходится сортировать разные виды и типы данных. В основном этими данными являются числами. Самый простой и наиболее удобный способ сортировки чисел является использование персонального компьютера. В этой курсовой работе показано как работают 2 метода сортировки, а также наглядна видна их эффективность.

## *метод простых вставок*

*Последовательно просматриваем  $a_1, \dots, a_{i-1}$  и каждый новый элемент  $a_i$  вставляем на подходящее место в уже упорядоченную совокупность  $a_0, \dots, a_{i-1}$ . Это место определяется последовательным сравнением  $a_i$  с упорядоченными элементами  $a_0, \dots, a_{i-1}$ .*

листинг данного метода в программе:

```
Public Sub InsertionSort(ByRef Arr() As Double, ByVal N As Long)
    Dim I As Long
    Dim J As Long
    Dim K As Long
    Dim Tmp As Double

    N = N-1
    i = 1
    Do
        j = 0
        Do
            If Arr(i)<=Arr(j) then
                k = i
                Tmp = Arr(i)
                Do
                    Arr(k) = Arr(k-1)
                    k = k-1
                Loop Until Not k>j
                Arr(j) = Tmp
                j = i
            Else
                j = j+1
            End If
        Loop Until Not j<i
        i = i+1
    Loop Until Not i<=n
End Sub
```

## метод бинарных деревьев

Алгоритм основан на представлении массива в виде бинарного дерева, обладающего особыми свойствами. В памяти компьютера все элементы массива расположены последовательно, структура же дерева определяется следующим образом: будем считать, что  $i$ -ый элемент массива ("предок") имеет два элемента потомка с номерами  $2i+1$  и  $2i+2$ . Дерево имеет нормальную форму, если любой элемент предок больше своих потомков.

Для понимания алгоритма рассмотрите приведенную блок-схему. Из свойств алгоритма стоит заметить, что он дает стабильно хорошую скорость упорядочивания (порядка  $n \cdot \log(n)$ ), вне зависимости от того с каким массивом работает, и поэтому используется в случаях когда необходимо гарантировано упорядочить массив за короткое время.

листинг данного метода в программе:

```
Public Sub HeapSort(ByRef Arr() As Double, ByVal N As Long)
```

```
    Dim I As Long
```

```
    Dim J As Long
```

```
    Dim K As Long
```

```
    Dim T As Long
```

```
    Dim Tmp As Double
```

```
    If N=1 then
```

```
        Exit Sub
```

```
    End If
```

```
    i = 2
```

```
    Do
```

```
        t = i
```

```
        Do While t <> 1
```

```
            k = t \ 2
```

```
            If Arr(k-1) >= Arr(t-1) then
```

```
                t = 1
```

```
            Else
```

```
                Tmp = Arr(k-1)
```

```
                Arr(k-1) = Arr(t-1)
```

```
                Arr(t-1) = Tmp
```

```
                t = k
```

```
            End If
```

```
        Loop
```

```
        i = i+1
```

```
    Loop Until Not i <= n
```

```
    i = n-1
```

- Do
- Tmp = Arr(i)
- Arr(i) = Arr(0)
- Arr(0) = Tmp
- t = 1
- Do While t<>0
- k = 2\*t
- If k>i then
- t = 0
- Else
- If k<i then
- If Arr(k)>Arr(k-1) then
- k = k+1
- End If
- End If
- If Arr(t-1)>=Arr(k-1) then
- t = 0
- Else
- Tmp = Arr(k-1)
- Arr(k-1) = Arr(t-1)
- Arr(t-1) = Tmp
- t = k
- End If
- End If
- Loop
- i = i-1
- Loop Until Not i>=1
- End Sub
  
- g = g\2
- Loop Until Not g>0
- End Sub

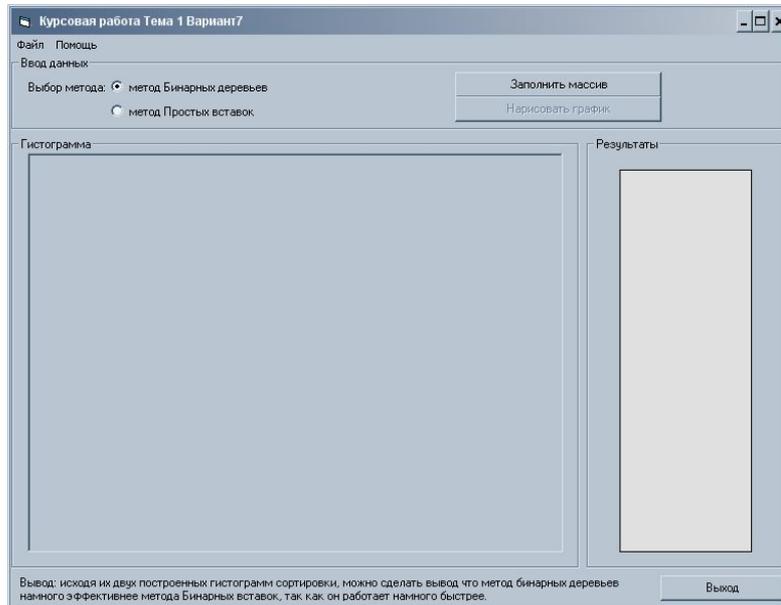
## Создание приложения

При запуске Microsoft Visual Basic 6.0 автоматически выскакивает окно с предложением создать новый проект Standard EXE. Нажимаем ОК. Начнём создание проекта с титульного листа. Для этого в меню Project выберем команду Add Form. В появившемся диалоговом окне выберем Dialog и нажмем ОК. В открывшейся форме расставим объекты: Label1, Label2, ... , Label8, Image1

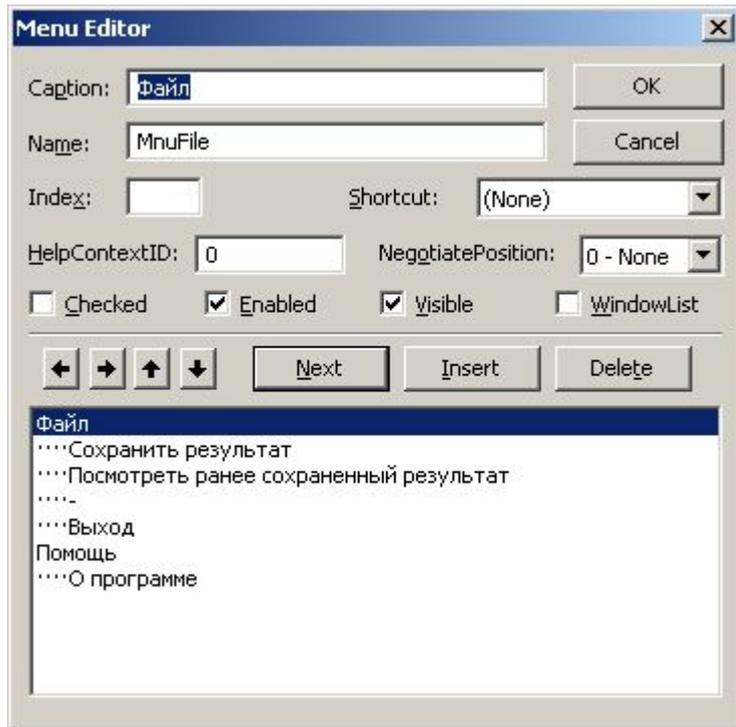
Присвоим свойству Caption значение “Титульный лист”. На кнопку Вход напишем следующий программный код:

```
Private Sub EnterButton_Click()  
    Unload Dialog  
    Form1.Show  
End Sub
```

При щелчке на кнопке ОК из памяти выгружается форма Dialog и загружается базовая форма, служащая начальной точкой для всех операций с программой, форма Form1. Для того, чтобы создать форму Form1, в меню Project выполним команду Add Form. В появившемся диалоговом окне выберем Form и нажмем ОК. Присвоим свойству Caption формы Form1 значение “ Курсовой проект - Тема 1 вариант 8”.



Создадим меню для проекта. Для этого в меню Tools выполним команду Menu Editor. В появившемся диалоговом окне напишем названия пунктов меню и названия процедур, которые будут запускаться при выполнении команд меню. Редактор меню с введёнными именами представлен на рисунке .



**Опишем какие процедуры выполняются в каждом пункте меню.**

1. MnuFileSave\_rez – осуществляет сохранение результата эксперимента. Результат сохраняется в файле log.txt, куда записывается система линейных уравнений, полученные результаты.

---

```
Private Sub MnuFileSave_rez_Click()  
Open App.Path & "\log.txt" For Output As #1  
Print #1, "" & Chr(13) & Chr(10) & "(" & Chr(13) & Chr(10) & Text1.Text  
Close #1  
End Sub
```

3. MnuFileLook\_rez – осуществляет вывод на экран ранее сохранённого результата, если предварительно его сохранили.

---

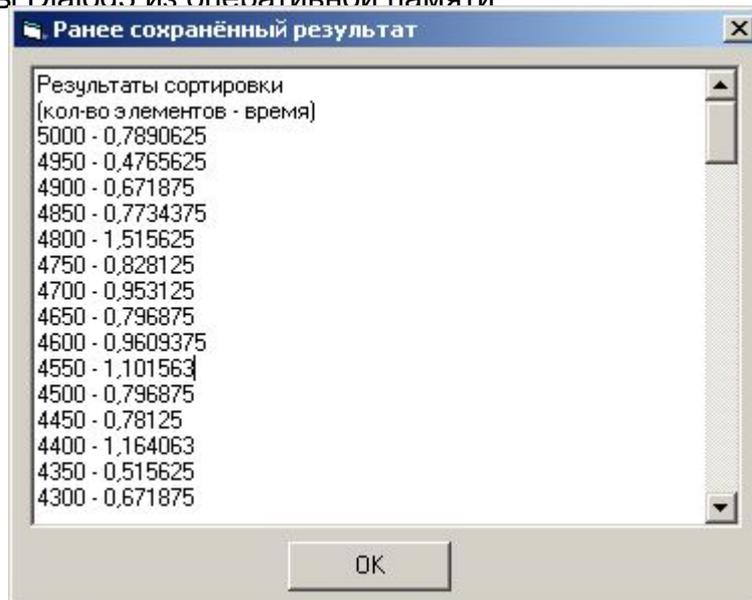
```
Private Sub MnuFileLook_rez_Click()  
Load Dialog3  
Dialog3.Show  
End Sub
```

---

При выполнении этой процедуры загружается форма Dialog3, на которой находится объект Text1, в который выводится сохраненный отчет. Отчет загружается при загрузке формы. Программный код Dialog3:

```
Private Sub Form_Load()  
Open App.Path & "\log.txt" For Input As #2  
Do Until EOF(2)  
    Line Input #2, LineOfText$  
    Text1.Text = Text1.Text & LineOfText$ & Chr(13) & Chr(10)  
Loop  
Close #2  
End Sub  
Private Sub OKButton_Click()  
    Dialog3.Hide  
    Unload Dialog3  
End Sub
```

В процедуре Form\_Load (), открывает файл, производит построчное считывание из него и вывод в объект Text1. Готовая форма Dialog3 изображена на рисунке (см. рис. №8). При щелчке на кнопке ОК происходит выгрузка формы Dialog3 из оперативной памяти



4. MnuHelpAbout – выводит на экран информацию о программе. Вывод осуществляется с помощью формы Dialog2.

```
Private Sub MnuHelpAbout_Click()  
    Dialog2.Show  
End Sub
```

При щелчке на кнопке ОК (объект Command1) происходит выгрузка формы Dialog2 из оперативной памяти.

Программный код формы Dialog2:

```
Private Sub OKButton_Click()  
    Unload Dialog2  
End Sub
```

5. MnuFileExit– производит выход из приложения.

```
Private Sub MnuFileExit ()  
    End  
End Sub
```

## **Список используемой литературы**

1. MS Visual Basic 6.0 для профессионалов. Микаэл Хальворсон. ЭКОМ 2001 г.
2. Информатика. Высшая школа. Острейковский В. А. 2000 г.
3. [www.informatic.ugatu.ic.ru](http://www.informatic.ugatu.ic.ru) – официальный сайт кафедры информатики УГАТУ, отдел дистанционной помощи студентам по написанию курсовых работ.