

# Комплекс программ для моделирования процесса написания символов при чтении лекций (версия 2017 года)

Попов А.А. apopov@vvoi.ru

Целью комплекса программ является отображение в электронной форме процесса написания символов на доске при чтении лекций. В результате для преподавателя становится возможным излагать материал с многочисленными формулами или с кодами компьютерных программ, не прибегая к доске и мелу, но, используя традиционную форму изложения, когда к имеющейся последовательности добавляется по одному символу или малому графическому фрагменту [1, 2].

В новой версии, в сравнении с предыдущей версией [3], расширены функциональные возможности приложений Symbols, ReadPages, ReadSymbols, добавлены приложения ReadPages\_Bak, Delete\_All\_M, которые делают более удобной работу с комплексом программ. Добавлен класс **TwoSize.class**, позволяющий увеличивать размеры символов. В класс CommentFrame добавлена дополнительная информация о новых символах и реализована возможность инвертировать цвета в окне с комментариями. Приложение BatFiles, а также классы finFrame, WindowDestroyer и Pilot остались без изменений, неизменным остался также формат информационных файлов Symb\_\*.txt.

## 1. Файл **Symbols.class**

Программа предназначена для формирования файлов **Symb\_\*.txt** с символами и графическими фрагментами по специальному формату, позволяющему с помощью другой программы **ReadSymbols.class** выводить подготовленную информацию по одному символу или малому графическому фрагменту. Программа загружается с помощью командной строки

```
java Symbols
```

которая размещена в пакетном файле **Symbos.bat**.

В отличие от предыдущей версии комплекса программ здесь есть возможность, как размещать информацию в новом файле, так и добавлять информацию к ранее сформированному файлу. После загрузки программы появляется окно для выбора необходимого действия (рис. 1).

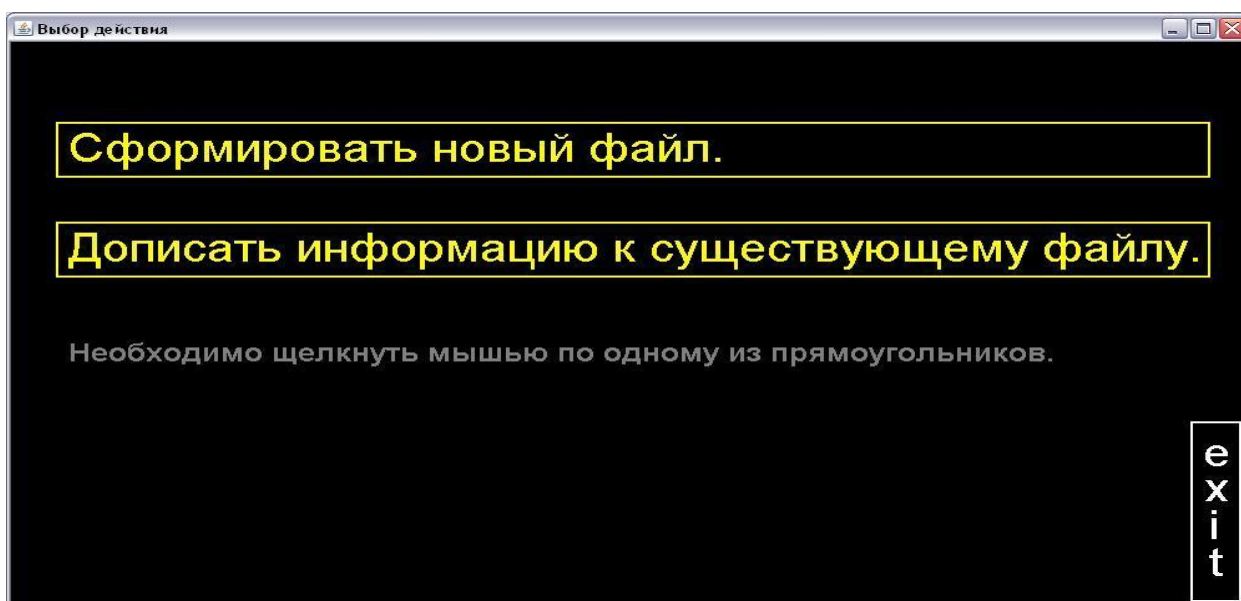


Рис. 1. Выбор действия при занесении информации в текущий файл вида Symb\_\*.txt.

При дополнении существующего файла необходимо щелкнуть мышью по нижнему прямоугольнику. В итоге появляется возможность выбрать конкретный файл для добавления к нему информации (рис.2).

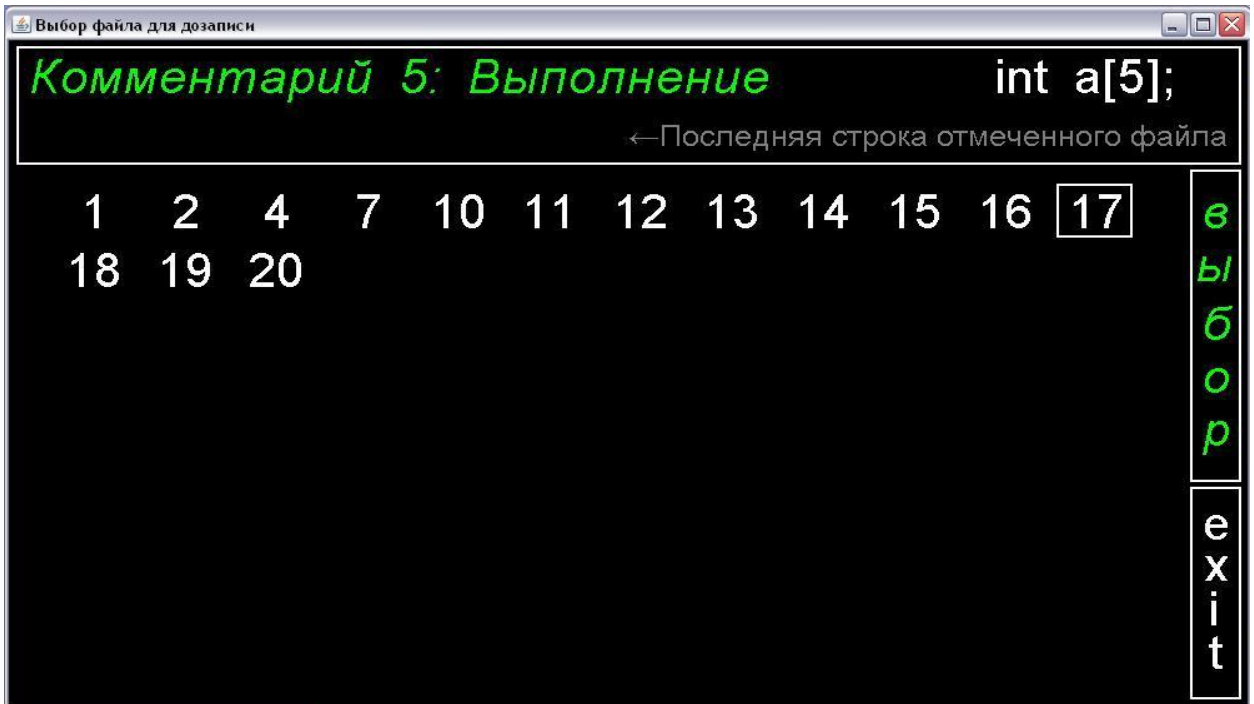


Рис.2. Выбор номера файла вида Symb\_\*.txt для добавления к нему информации.

2

Щелчком мыши отмечается номер файла, к которому необходимо добавить информацию. Затем необходимо щелкнуть мышью по прямоугольнику с надписью “выбор”. Для исключения ошибки при выборе файла в верхней рамке выводится последняя строка файла, номер которого обведен малым прямоугольником. После выбора появляется окно с последней страницей выбранного файла (рис. 3).

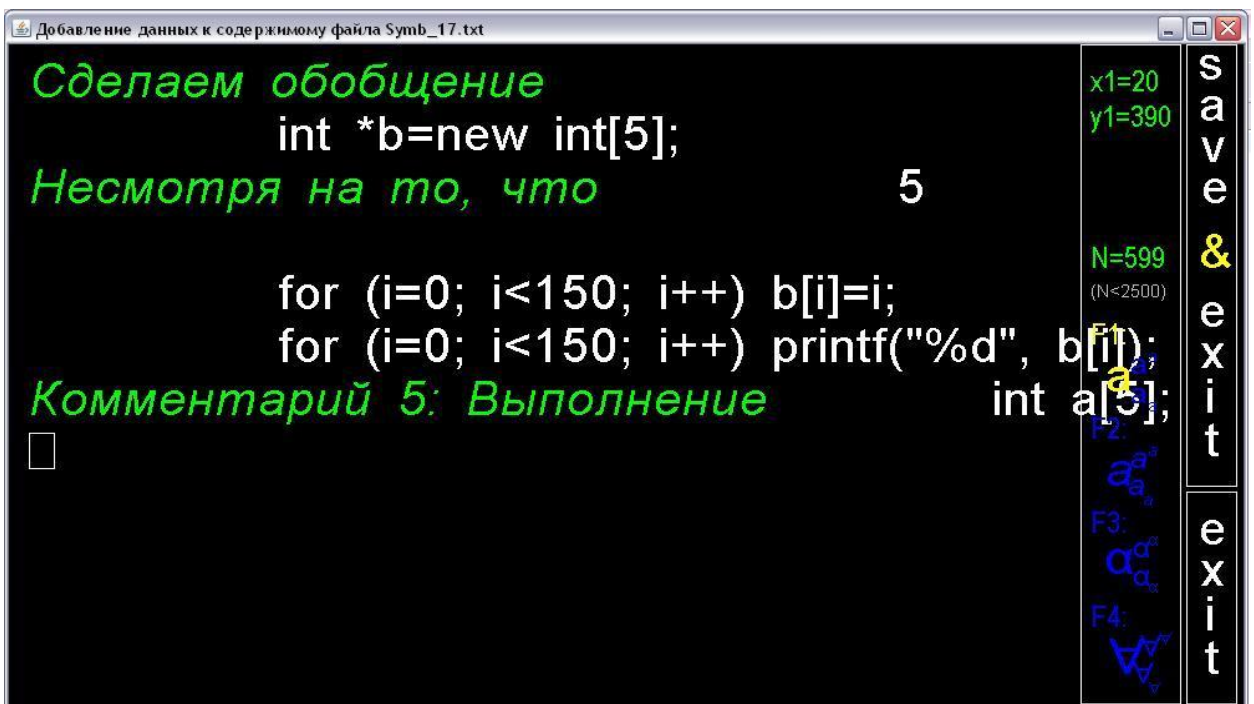


Рис.3. Окно для добавления информации к файлу Symb\_17.txt.

При формировании нового файла необходимо щелкнуть по верхнему прямоугольнику в исходном окне (рис. 1). В результате появляется окно для ввода информации в новый файл (рис. 4).

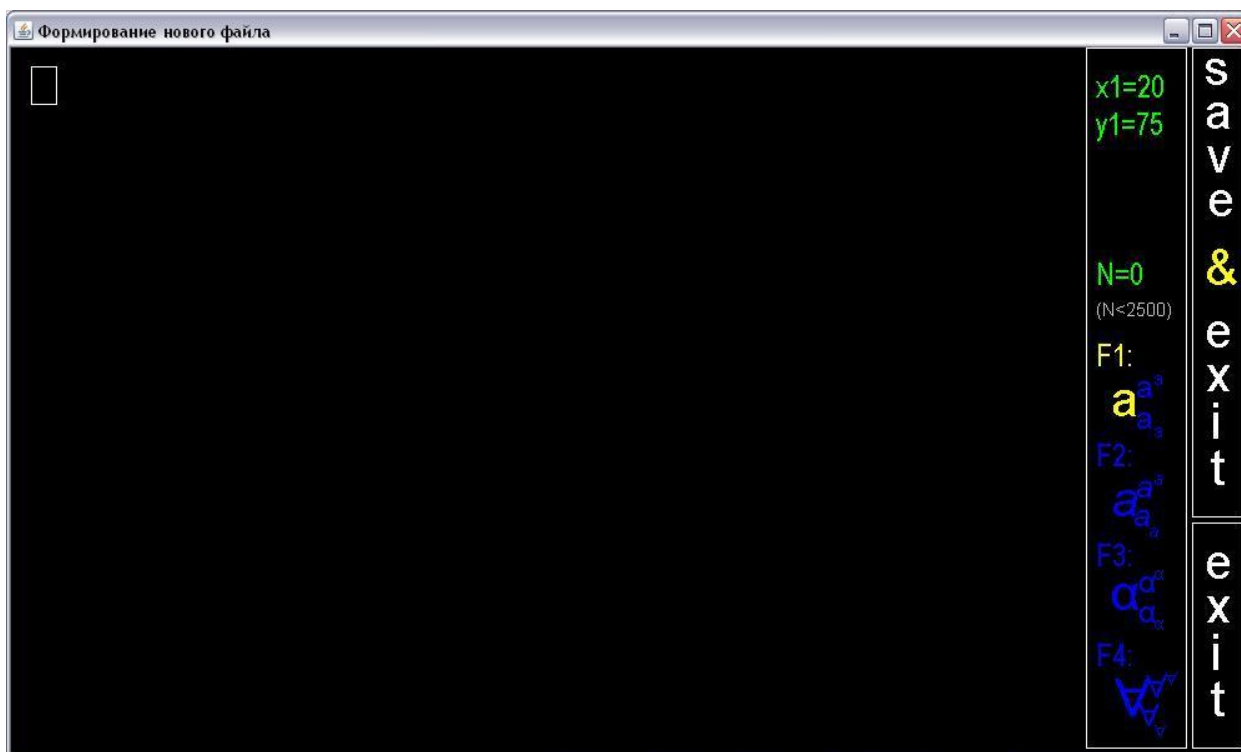


Рис. 4. Окно за занесения текстовой и графической информации в новый файл вида Symb\_\*.txt.

3

По окончании ввода полная информация обо всех символах записывается в файл вида **Symb\_\*.txt**. При каждой записи проверяется наличие файлов указанного вида и автоматически определяется порядковый номер очередного файла.

Место ввода отмечено курсором в виде прямоугольника (рис. 3-4) положение которого можно корректировать, смещая вверх, вниз, влево, вправо при нажатии клавиш  $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$ . Курсор можно разместить в любом месте окна, устанавливая указатель мыши в нужное место окна и щелкая левой кнопкой. При наборе символов в пределах одной строки можно смещать курсор на 1 символ при нажатии пробельной клавиши и на 5 символов при нажатии клавиши **Esc**.

При наборе текста из нескольких строк переход к очередной строке происходит после нажатия клавиши **Enter**. Если курсор расположен вблизи нижней границы окна, то после нажатия клавиши **Enter**, он переводится в верхнюю часть окна, которая автоматически очищается. Каждый введенный символ теперь будет принадлежать новой странице.

В информационной части окна приведены координаты в пикселях для левого нижнего угла курсора. Ниже представлена информация о количестве введенных символов. Символы могут быть двух видов: обычный латинский и русский, наклонный латинский и русский. Переход от одного вида к другому происходит после нажатия клавиш **F1** и **F2**.

После нажатия клавиши **F3** латинским буквам клавиатуры ставится в соответствие наиболее используемые в физике и математике греческие буквы. Здесь же приведены обозначения множеств натуральных, целых, рациональных, вещественных и комплексных чисел, обозначения окрестности точки и проколотой окрестности точки. Раскладку клавиатуры с греческими символами можно вывести на экран в отдельном окне, если указатель мыши разместить на символе **F3**, а затем нажать и удерживать в нажатом состоянии левую кнопку мыши (рис. 5). При освобождении кнопки мыши окно с дополнительной информацией закрывается.



В новой версии в режиме **F4** добавлено 5 символов, которые расположены в верхнем ряду. Из них символы следования и эквивалентности продублированы, поскольку ранее использованы в режиме **F5**.

Клавиши **PgUp** и **PgDn** позволяют перейти к шрифтам меньших размеров, уменьшить изображение курсора, а также поместить курсор в месте ввода индекса, субиндекса, показателей степени. Последний символ можно стереть, нажимая клавишу **BackSpace**.

Данная программа позволяет включать в информационный файл сплошную линию, пунктирную линию, линию со стрелкой, прямоугольник, эллипс и прямоугольник со скругленными углами. Переход в режим ввода графических примитивов производится после нажатия клавиши **F5** (рис.7).

В этом режиме курсор принимает вид малой окружности с зеленым заполнением только после щелчка левой кнопки мыши в том месте окна, где расположен ее указатель. Для вывода каждого примитива необходимо иметь в окне две таких “точки”. Щелкая левой кнопкой мыши еще раз можно получить изображения второго курсора. Координаты обоих курсоров выведены в информационной части окна. Курсоры можно двигать с помощью клавиш  $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$ . После нажатия клавиш “1”, “2”, “3”, “4”, “5”, “6” основной клавиатуры на экран выводится линии или фигуры. Таблица соответствия между клавишами 1-6 и графическими компонентами приведена в информационной части окна. При появлении в окне выбранной компоненты изображения курсоров исчезает. С помощью клавиши “L” можно выводить кусочно-ломанную линию. В этом случае точка конца последнего отрезка становится точкой начала нового отрезка, т.е. изображение последнего курсора сохраняется.

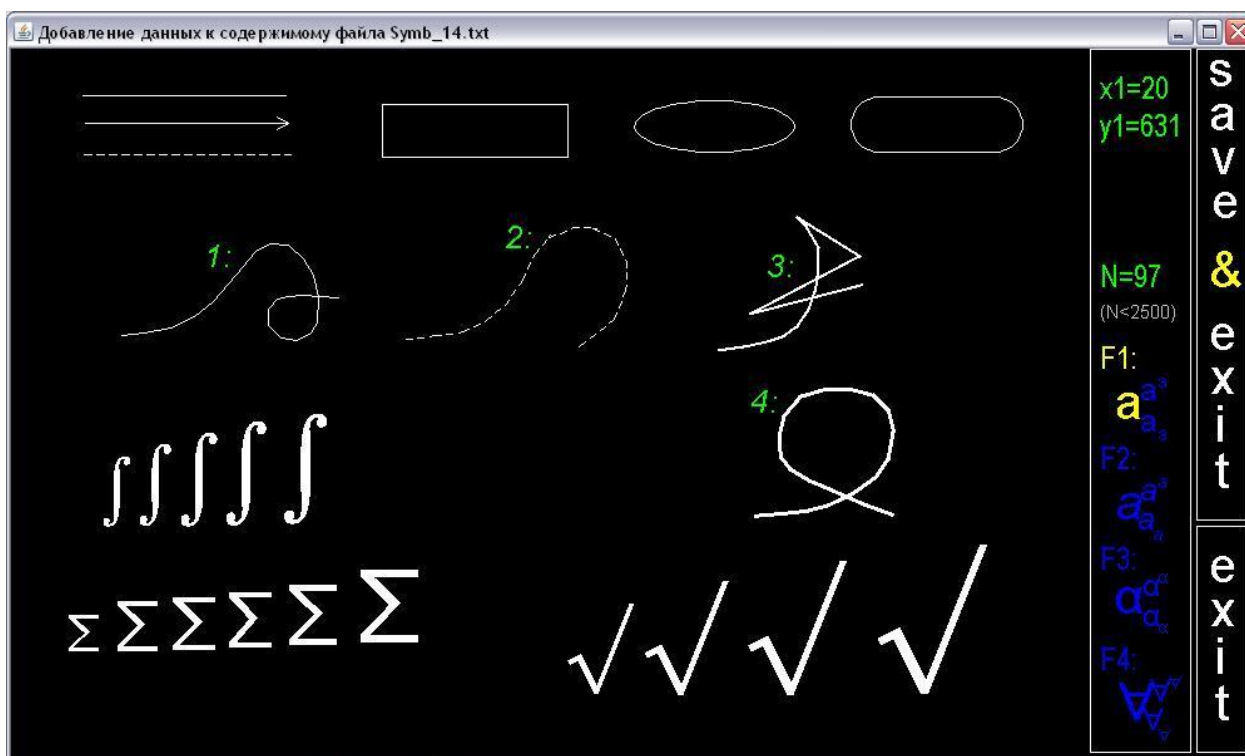


Рис. 7. Графические примитивы, вывод которых обеспечивается после нажатия клавиши F5.

Кривая 1 (рис. 7) составлена из малых отрезков с использованием клавиши “L”.

В новой версии добавлена возможность выводить кусочно-ломанные линии толщиной 2 и 3 пикселя (кривые 3 и 4) при нажатии клавиш “D” и ”T”, и в виде пунктирной линии (кривая 2) при нажатии клавиши “P”.

В графическом режиме, можно вывести большие круглые скобки, большие квадратные скобки и большие фигурные скобки. Данные символы рассматриваются здесь, как графические компоненты. Каждая из них вписывается в прямоугольную область, ограниченную по диагонали зелеными точечными курсорами. В этом же режиме можно вывести символы следования и эквивалентности, границы которых определяются горизонтальными координатами двух зеленых курсоров. В зависимости от разности этих координат возможен вывод символов следования и эквивалентности трех различных размеров, что в полной мере соответствует трем возможным размерам символов формируемых информационных файлов. Вывод указанных компонент происходит после нажатия клавиш “[“, “]”, “7”, “8”, “9”, “0”, “-“, “=”. Таблицу соответствия указанных клавиш с рассматриваемыми здесь графическими компонентами можно вывести в отдельном окне после нажатия и удерживания левой кнопки мыши, когда ее указатель наведен на символ **F5**.

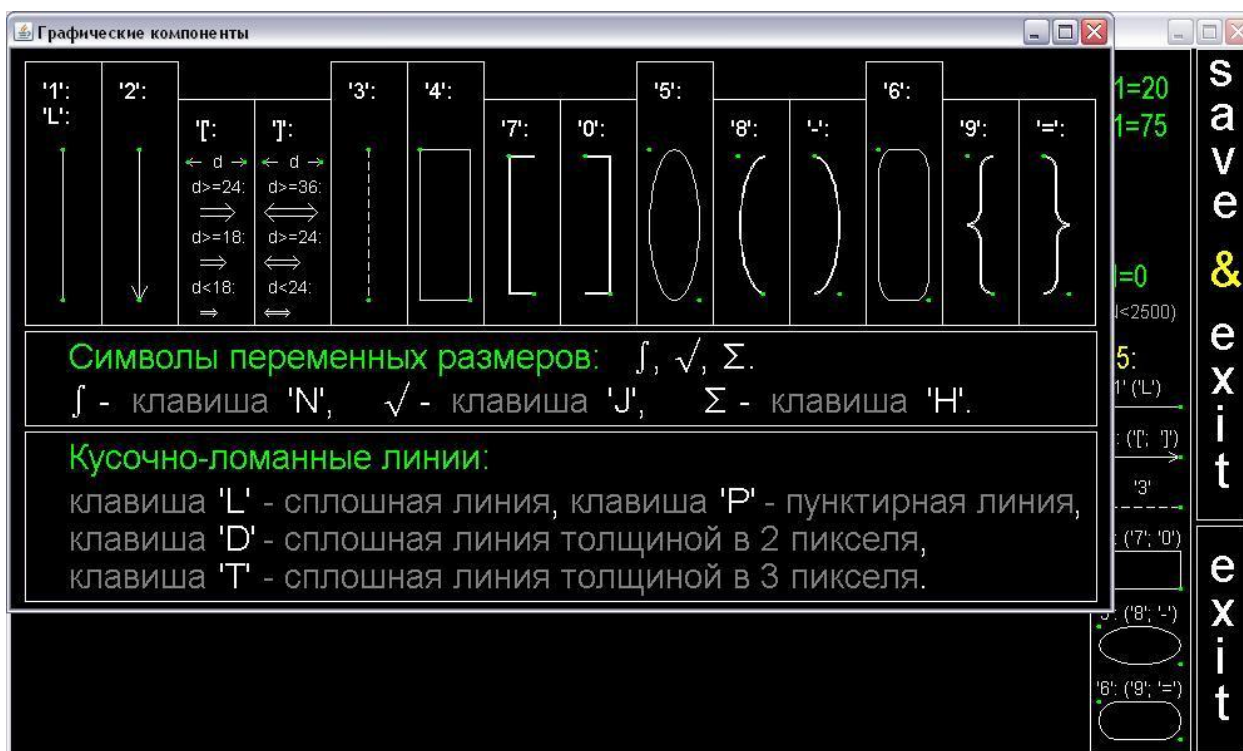


Рис. 8. Соответствие клавиш графическим примитивам в режиме F5.

В новой версии для режима **F5** добавлена возможность ввода знаков интеграла, радикала и суммы переменных размеров (рис. 8). Для их ввода используются латинские буквы “N”, “J” и “H”, такие же, как и в режиме **F4** для ввода тех же знаков. Размер шрифта определяется разностью ординат двух зеленых точечных курсоров и принадлежит отрезку [40, 120]. Из двух курсоров положение символа определяет нижний курсор. По отношению к нему символ будет располагаться справа и выше.

Для нанесения на экран экспериментальных точек они могут быть оформлены, как малые заполненные белыми пикселями прямоугольники или малые эллипсы. В частном случае, они могут стать малыми квадратами или малыми кругами. Если расстояния между пикселями двух курсоров, и в горизонтальном направлении, и в вертикальном направлении не превосходят 8 пикселей, то выводится прямоугольник или эллипс с заполнением, в противном случае без заполнения. Узкий прямоугольник шириной не более 4 пикселей воспринимается программой, как толстая линия, т.е. прямоугольник в данном случае выводится заполненный белым цветом.

Переход к режимам **F1-F4** из режима **F5** возможен только в том случае, когда на экране не осталось ни одного курсора. Эти курсоры можно убрать, нажимая клавишу **Escape**.

При завершении работы с программой следует иметь в виду, что кнопка закрытия окна заблокирована, но есть две области окна, щелкая по которым можно завершить программу. В одном случае программа завершается без записи, в другом случае с записью. При записи текущее окно закрывается и открывается окно с именем файла, в который занесена информация о символах и графических компонентах. Последнее окно закрывается традиционным способом, т.е. с помощью кнопки закрытия окна.

## 2. Файл **ReadSymbols.class**

Программа предназначена для посимвольного вывода на экран содержимого файлов **Symb\_\*.txt**. Загрузка производится с помощью командной строки

```
java ReadSymbols Symb_*.txt ,
```

которую размещена в пакетном файле **ReadSymbos.bat** для посимвольного просмотра текущего сформированного файла. После имени файла **ReadSymbols** в командной строке следует имя файла для просмотра **Symb\_\*.txt**. Вместо звездочки необходимо задать порядковый номер рассматриваемого файла.

Каждое нажатие любой латинской буквы сопровождается выводом в окно очередного символа в данном случае из файла **Symb\_15.txt** (рис. 9). Здесь приведен фрагмент лекции по теме “Интегрирующий множитель” дисциплины “Дифференциальные уравнения”.

При переходе на следующую страницу нижняя часть предыдущей страницы сохраняется на экране и исчезает только при заполнении текущей страницы на треть. К предыдущим страницам можно вернуться, нажимая клавишу **PgUp**. Переход к текущей странице производится с помощью клавиши **PgDn**. Клавиша **BackSpace** позволяет удалить последний символ в пределах текущей страницы.

При выводе символов с помощью программы **ReadSymbols.class** используются три цвета: белый, зеленый и желтый. Для инверсии цветов символов и фона достаточно нажать клавишу **Home** (рис. 10). В итоге та же информация представлена с помощью символов черного, синего и красного цвета на белом фоне. При нажатии клавиши **Enter** цветные символы в зависимости от цвета фона становятся белыми или черными. При повторном нажатии клавиши **Enter** символы вновь становятся трехцветными.

На месте очередного выводимого символа располагается курсор. Изображение курсора исчезает при отображении на экран последнего символа из текущего файла. Кроме того, в правой части появляется прямоугольная область с названием **Exit**. Щелкая по ней, можно завершить программу.

Данные из файла Symb\_15.txt

$$\mu(x,y)P(x,y)dx + \mu(x,y)Q(x,y)dy = 0$$

$$\frac{\partial}{\partial y} [\mu(x,y)P(x,y)] = \frac{\partial}{\partial x} [\mu(x,y)Q(x,y)]$$

$$\frac{\partial \mu}{\partial y} P + \frac{\partial P}{\partial y} \mu = \frac{\partial \mu}{\partial x} Q + \frac{\partial Q}{\partial x} \mu$$

$$\frac{\partial \mu}{\partial y} P - \frac{\partial \mu}{\partial x} Q = \mu \left[ \frac{\partial Q}{\partial y} - \frac{\partial P}{\partial x} \right]$$

$$P \frac{\partial}{\partial y} \ln \mu - Q \frac{\partial}{\partial x} \ln \mu = \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}$$

Рис. 9. Иллюстрация посимвольного вывода из текущего файла Symb\_15.txt.

Данные из файла Symb\_15.txt

$$\mu(x,y)P(x,y)dx + \mu(x,y)Q(x,y)dy = 0$$

$$\frac{\partial}{\partial y} [\mu(x,y)P(x,y)] = \frac{\partial}{\partial x} [\mu(x,y)Q(x,y)]$$

$$\frac{\partial \mu}{\partial y} P + \frac{\partial P}{\partial y} \mu = \frac{\partial \mu}{\partial x} Q + \frac{\partial Q}{\partial x} \mu$$

$$\frac{\partial \mu}{\partial y} P - \frac{\partial \mu}{\partial x} Q = \mu \left[ \frac{\partial Q}{\partial y} - \frac{\partial P}{\partial x} \right]$$

$$P \frac{\partial}{\partial y} \ln \mu - Q \frac{\partial}{\partial x} \ln \mu = \frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}$$

Рис. 10. Инверсия цветов после нажатия клавиши Home.

При завершении программы автоматически формируется файл **M\_\*.txt**. Номер файла **M\_\*.txt** соответствует номеру просматриваемого файла **Symb\_\*.txt**. Если файл **M\_\*.txt** удалить, то, возможен посимвольный просмотр текущего файла **Symb\_\*.txt**, начиная с первого символа.

Формулы могут иметь субиндексы, т.е. иметь более мелкие размеры в сравнении с индексами (рис.11).



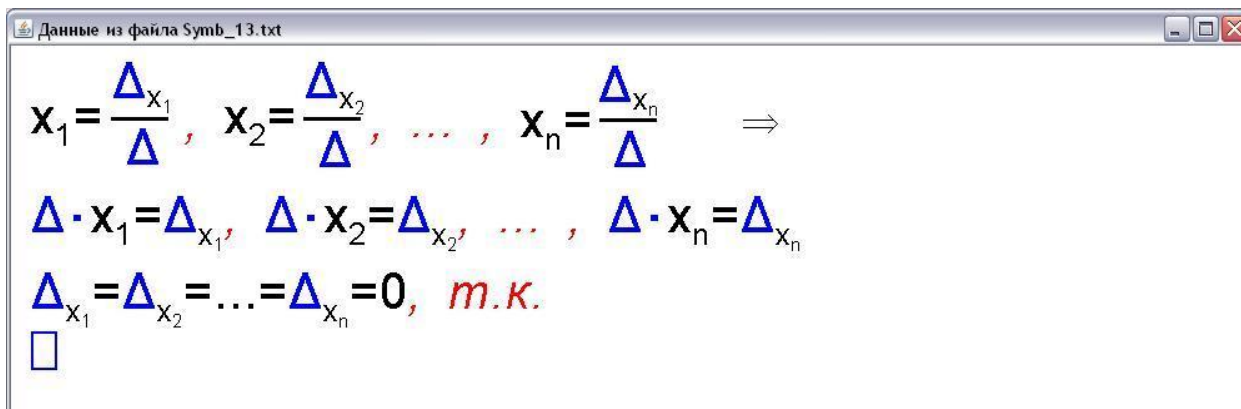


Рис. 11. Выводимый текст, содержащий индексы и субиндексы.

В новой версии программа **ReadSymbols.class** дополнена вспомогательным классом **TwoSize.class**, с помощью которого можно создавать копии символов и выводить их в увеличенном виде в отдельном окне. При малых расстояниях видеопроектора от экрана субиндексы издали могут быть плохо различимыми.

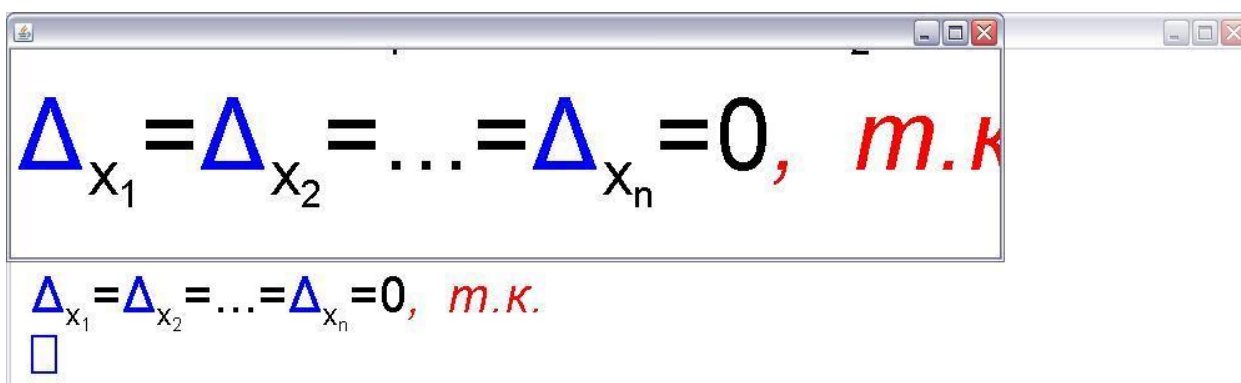


Рис. 12. В отдельном окне увеличена последняя строка.

Для того, чтобы увеличить тот или иной индекс и вывести его в отдельном окне с соответствующим окружением, достаточно нанести указатель мыши на мелкий символ, нажать левую кнопку мыши и удерживать ее в нажатом состоянии (рис 12). После освобождения кнопки мыши дополнительное окно закрывается, и программа возвращается в исходный режим.

### 3. Файл **ReadPages.class**

Файл загружается с помощью командной строки

**java ReadPages**

, которая размещена в пакетном файле **ReadPages.bat**. После загрузки этой программы становится возможным постраничный просмотр всех файлов вида **Symb\_\*.txt**. Переход от файла к файлу происходит после нажатия клавиш  $\uparrow$  и  $\downarrow$ . В правой части окна выводится номер текущего файла. Ниже в окне выведена область **Exit** для выхода из программы. Постраничный просмотр текущего файла производится с помощью клавиш **PgUp** и **PgDn**.

В данной программе предусмотрена возможность коррекции текущего файла **Symb\_\*.txt**. После нажатия клавиши **Enter** происходит переход программы в режим

коррекции. В новой версии можно редактировать не только символы, но и графические примитивы. Коррекция происходит в более сложном виде, поэтому для возможности возврата к моменту начала коррекции, автоматически создается резервная копия текущего файла, соответствующая общепринятой процедуре, т.е. создается файл с тем же именем, но с расширением "bak".

В данном режиме в правой части окна появляется информация, необходимая для ввода символов. В отличие от предыдущей версии для каждого графического примитива на экран выводятся его границы в виде двух синих малых кругов, охватывающие его по диагонали (рис. 13). Для вставки или удаления символов необходимо щелчком мыши между символами образовать курсор в виде прямого угла сине-зеленого цвета. При каждом нажатии клавиши **Delete** удаляется один символ, расположенный за курсором. Если удалить все символы в этом ряду, то происходит автоматический выход из режима коррекции с записью данных в новый файл.

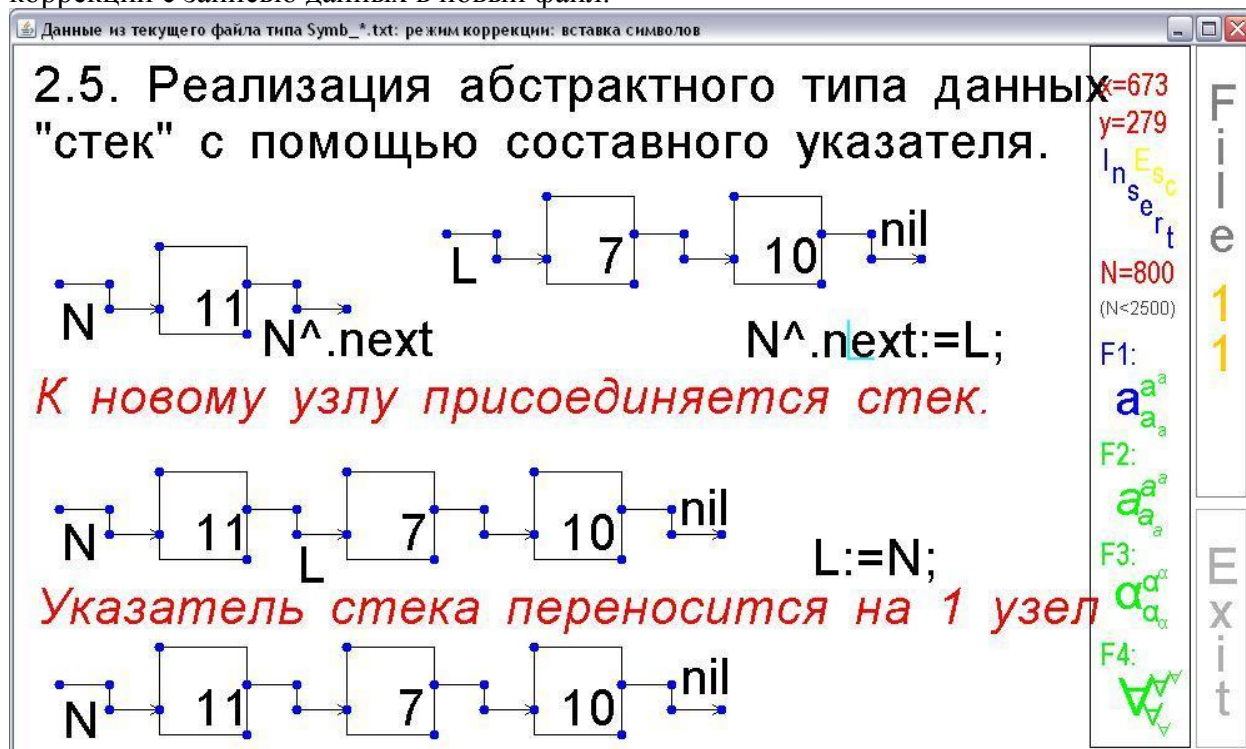


Рис.13. Режим коррекции с возможностью редактировать графические примитивы.

При нажатии клавиш с латинскими буквами в режимах **F1-F4** производится вставка символов после курсора. При нажатии клавиши **Insert** вместо вставки при вводе символов будет происходить режим замены. Также как и при удалении, после полной замены всех символов, расположенных за курсором производится автоматический выход из режима коррекции с сохранением измененных данных. При повторном нажатии клавиши **Insert** происходит возвращение программы в режим вставки.

Клавиши  $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$  позволяют двигать курсор. Причем вместе с курсором будет двигаться расположенная за ним часть строки, т.е. любую символьную строку можно разбить на части и разместить их в любой части окна. При нажатии клавиши **Escape** символы, следующие за курсором, больше не связаны с ним. Такой курсор можно продвинуть в любую часть окна и вывести за ним символ или группу символов. В данном случае положение курсора позволяет зафиксировать последовательность вывода символов при использовании программы **ReadSymbols.class**.

В новой версии в режиме вставки можно корректировать форму графических примитивов. Для этого необходимо щелкнуть мышью по синей "точке" того графического примитива, размеры которого необходимо изменить (рис. 14). Отмеченная "точка"

становится красной. Далее с помощью клавиш  $\uparrow$ ,  $\downarrow$ ,  $\leftarrow$ ,  $\rightarrow$  можно изменить размеры графического примитива, двигая красную точку.

При необходимости графический примитив с красной точкой можно удалить, нажимая клавишу **Delete** (рис. 15). После удаления в данном случае прямоугольника программа автоматически выходит из режима коррекции с сохранением последних данных.



11

Рис.14. Квадрат с элементом “7” при изменении координат красной точки преобразован в прямоугольник.



Рис.15. Удален прямоугольник, охватывающий элемент “7”.

Элемент “7” вновь можно охватить квадратом, если реализовать режим вставки графического элемента. В режиме коррекции необходимо выделить красной точкой элемент, перед которым производится вставка. В данном случае – это отрезок прямой. Затем перейти в режим **F5** для вставки графических примитивов. Далее необходимо выполнить процедуру включения графического элемента, как в программе **Symbols.class**: щелкнуть два раза по полю окна, при этом образуются два зеленых курсора (рис. 16) и нажать цифру “4” основной клавиатуры. В результате выводится квадрат, который вписывается в область, ограниченную по диагонали зелеными курсорами. После восстановления вначале измененного, а затем удаленного квадрата программа выходит из режима коррекции с сохранением последних изменений.



Рис.16. Порядок вставки графического примитива.

При работе с графическими примитивами переход в режим **F5** производится только для вставки. Во всех остальных случаях графические примитивы можно редактировать, находясь в режимах **F1-F4**.

#### 4. Файл **ReadPages\_Bak.class**

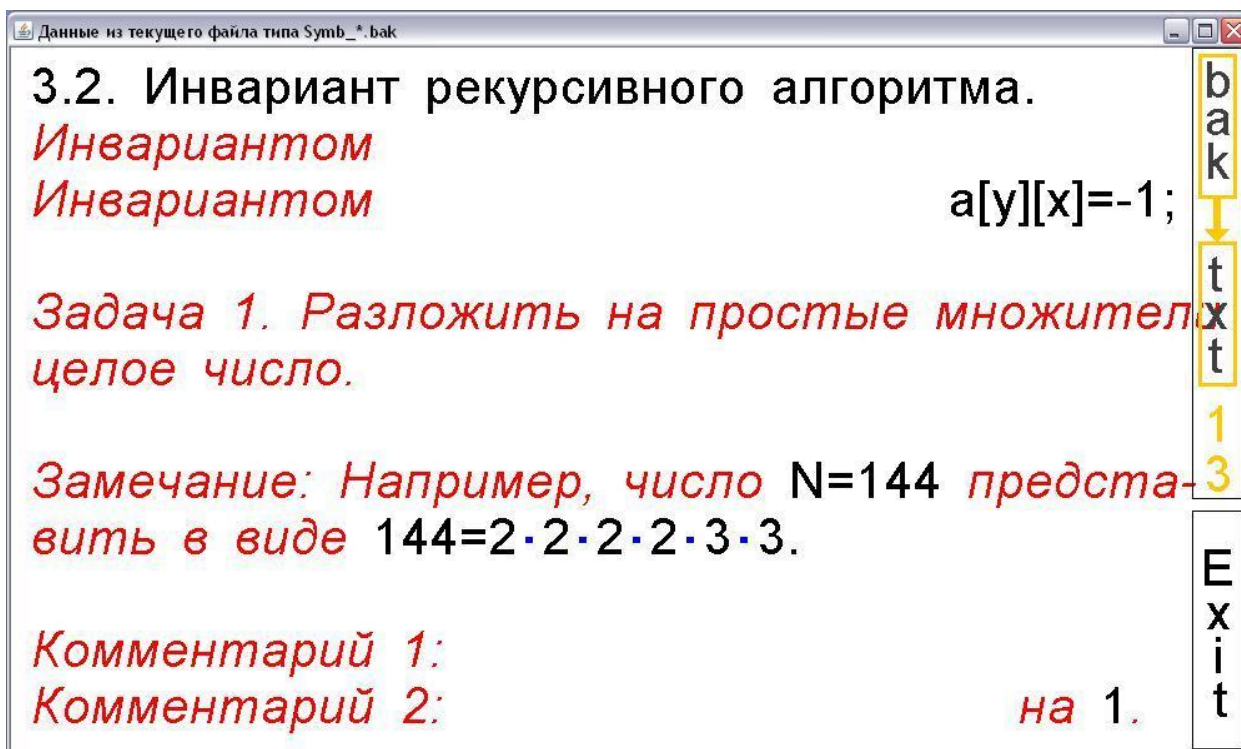
Файлы **Symb\_\*.txt** и **Symb\_\*.bak** являются двоичными, что требует для их просмотра использование специальных программ. В части 3 данного сообщения рассмотрена программа **ReadPages.class** для просмотра и изменения файлов **Symb\_\*.txt**. В данной части анализируется программа **ReadPages\_Bak.class**, необходимая для просмотра файлов **Symb\_\*.bak**. Программа загружается с помощью командной строки

```
java ReadPages_Bak,
```

которая размещена в пакетном файле **ReadPages\_Bak.bat**. После загрузки появляется окно для просмотра файлов с расширением **Symb\_\*.bak** (рис.17).

В данной программе определены функции просмотра текущего файла **Symb\_\*.bak** с помощью клавиш **PgUp** и **PgDn**, переход к следующему файлу или возврат к предыдущему при нажатии клавиш **↑** и **↓**.

Если у пользователя после просмотра того или иного файла возникнет необходимость перевести его в основные файлы, т.е. файл с расширением \*.txt, необходимо щелкнуть по соответствующему полю в правом верхнем углу окна (рис. 17). Далее необходимо следовать выводимым в окно комментариям. После щелчка по левому прямоугольнику (рис. 18) файл **Symb\_13.bak** переходит в разряд основных файлов.



13

Рис.17. Просмотр файлов Symb\_\*.bak.

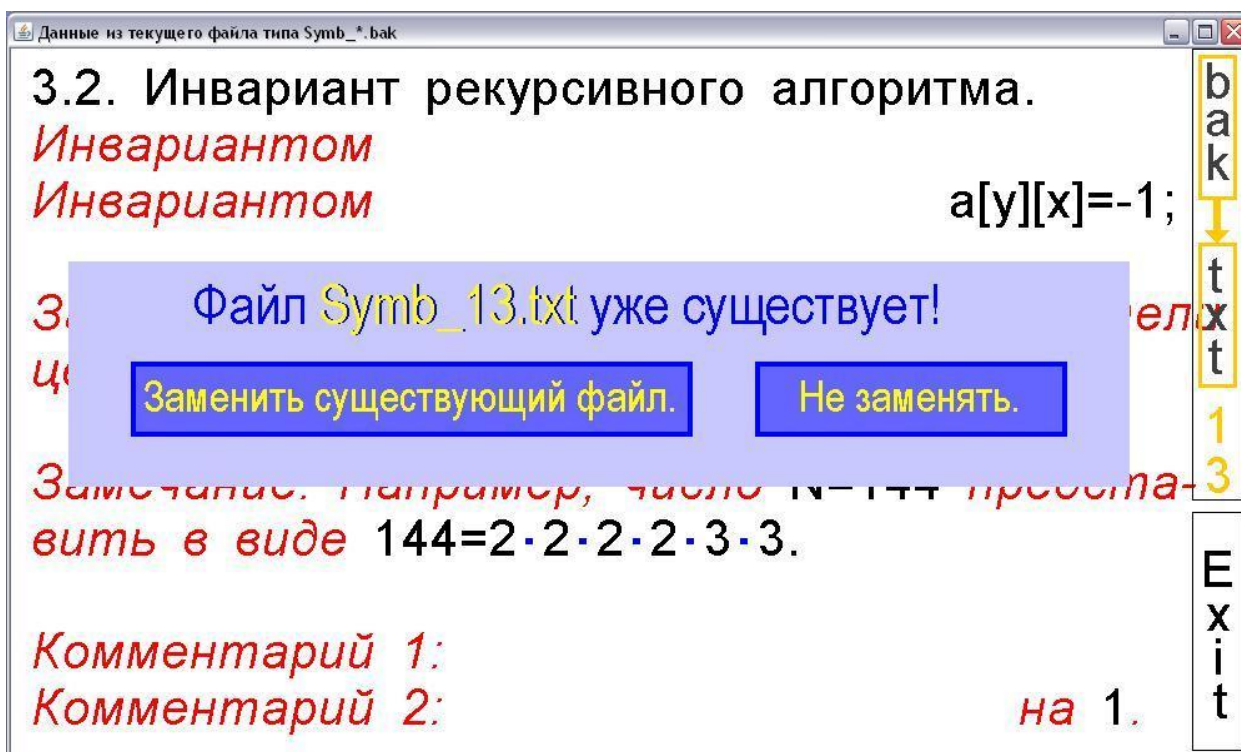


Рис.18. Процесс преобразования файла Symb\_13.bak в файл Symb\_13.txt.

## 5. Файл **Delete\_All\_M.class**

В процессе сопровождения лекций для каждого просмотренного файла вида **Symb\_\*.txt** образуется файл **M\_\*.txt** с тем же номером. В файл **M\_\*.txt** заносится информация о количестве просмотренных символов. Если количество записанных символов равно количеству символов в текущем файле **Symb\_\*.txt**, то этот файл не учитывается при дальнейшем чтении информационных файлов. Когда просмотрены все файлы вида **Symb\_\*.txt** образуется огромное количество вспомогательных файлов **M\_\*.txt**. При повторном чтении того же курса лекций, пусть даже в новой последовательности, все вспомогательные файлы необходимо удалить. Поскольку все файлы программного комплекса находятся в одной папке, то для исключения неприятностей, связанных со случайным удалением, все файлы **M\_\*.txt** удаляются программным путем с помощью командной строки

```
java Delete_All_M,
```

которая расположена в пакетном файле **Delete\_All\_M.bat**.

## 6. Файл **BatFiles.class**

В лекции может использоваться несколько файлов вида **Symb\_\*.txt**. Каждый из них загружается с помощью командной строки

```
java ReadSymbols Symb_*.txt
```

Причем необходимо предусмотреть варианты, когда следует вернуться к уже ранее посимвольно прочитанному файлу. Для реализации всех вариантов необходимо составить пакетный файл. Этот файл должен содержать специальную программу, которая позволяет переходить от одной командной строки к другой строке не только в прямом направлении, но и обратном. Например, если в файле **Lect\_1.bat** предполагается посимвольно просмотреть три файла вида **Symb\_\*.txt** с номерами 1, 2 и 3, то пакетный файл будет иметь вид:

```
@echo off  
:100  
java Pilot 1  
if errorlevel 4 goto 4  
if errorlevel 3 goto 3  
if errorlevel 2 goto 2  
java ReadSymbols Symb_1.txt  
goto 100  
:2  
java ReadSymbols Symb_2.txt  
goto 100  
:3  
java ReadSymbols Symb_3.txt  
goto 100  
:4
```

Файл **Pilot** является файлом-менеджером, обеспечивающим перемещение по пакетному файлу. Завершаясь с различными кодами, он обеспечивает переход к нужной командной строке вида

```
java ReadSymbols Symb_*.txt .
```

После выполнения этой командной строки происходит возврат к файлу **Pilot**. Пакетный файл может содержать до нескольких десятков строк, поэтому файлы указанного вида формируются автоматически с помощью программы **BatFiles**. Эта программа запускается при использовании командной строки

```
java BatFiles ,
```

которая располагается в пакетном файле **BatFiles.bat**.

После запуска программы появляется окно с двумя панелями (рис 19). На левой панели выведены номера всех файлов вида **Symb\_\*.txt**. На пустую правую панель необходимо перенести номера тех файлов **Symb\_\*.txt**, которые составят информацию об очередной лекции. В нижней части экрана для файла с отмеченным номером выведена его первая строка, что дает возможность сделать правильный выбор при включении этого файла в очередную лекцию. Для переноса номера с левой панели на правую панель необходимо отметить нужный номер, нажимая и не отпуская левую кнопку мыши, передвинуть указатель мыши на правую панель и освободить кнопку. Номера файлов на правой панели можно переставлять местами, а также возвращать их на левую панель. После щелчка мыши по полю **“Save & Exit”** программа завершает работу с формированием очередного пакетного лекционного файла **Lect\_\*.bat**. В этом файле содержатся командные строки вида **java ReadSymbols Symb\_\*.txt** в нужной последовательности.

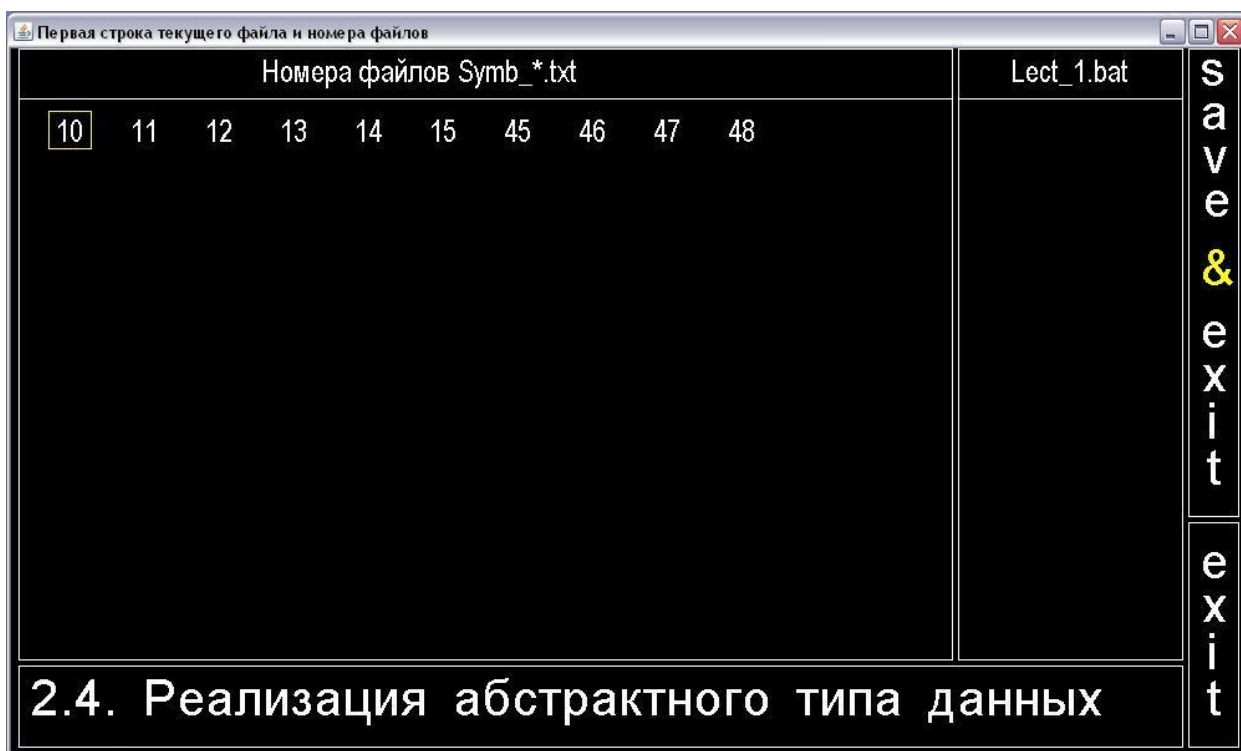


Рис. 19. Окно для визуального проектирования пакетного файла Lect\_1.bat.

При формировании нового пакетного файла номера уже использованных файлов **Symb\_\*.txt** не включаются в поля окна, формируемого программой **BatFiles**. Программа **BatFiles** предварительно анализирует все сформированные файлы вида **Lect\_\*.bat**, а затем блокирует все встретившиеся номера информационных файлов **Symb\_\*.txt**. Когда все файлы **Symb\_\*.txt** распределены по лекциям, то программа **BatFiles** выдает соответствующий комментарий.

#### 7. О дополнительных файлах программного комплекса (WindowDestroyer, CommentFrame, finFrame, TwoSize)

Для обеспечения работы программного комплекса используются четыре дополнительных файла, которые загружаются автоматически и неотделимы от рассмотренных выше классов. Рассмотренные ранее программы используют класс **Frame**, который дает возможность образовывать окна. Файл **WindowDestroyer.class** необходим для корректного закрытия окон, поэтому данный класс используется всеми файлами. Файл **CommentFrame.class** создает дополнительное окно, в которое помещается комментарий с

раскладкой клавиатуры для ввода греческих букв, математических символов и графических примитивов. Этот класс используется файлами **Symbols.class** и **ReadPages.class**. Файл **finFrame.class** создает дополнительное окно, в которое помещается информация о сформированных файлах **Symb\_\*.txt** и **Lect\_\*.bat**. Этот класс загружается программами **Symbols.class** и **BatFiles.class**. В новой версии добавлен класс **TwoSize.class**, который используется программой **ReadSymbols.class** и необходим для временного увеличения размеров символов.

## 8. Работа с файлами **Lect\_\*.bat** при чтении лекций

В пункте 6 приведен примерный вид пакетного лекционного файла. При его загрузке в первую очередь запускается программа **Pilot.class**, которая выводит меню в виде последовательности прямоугольников.

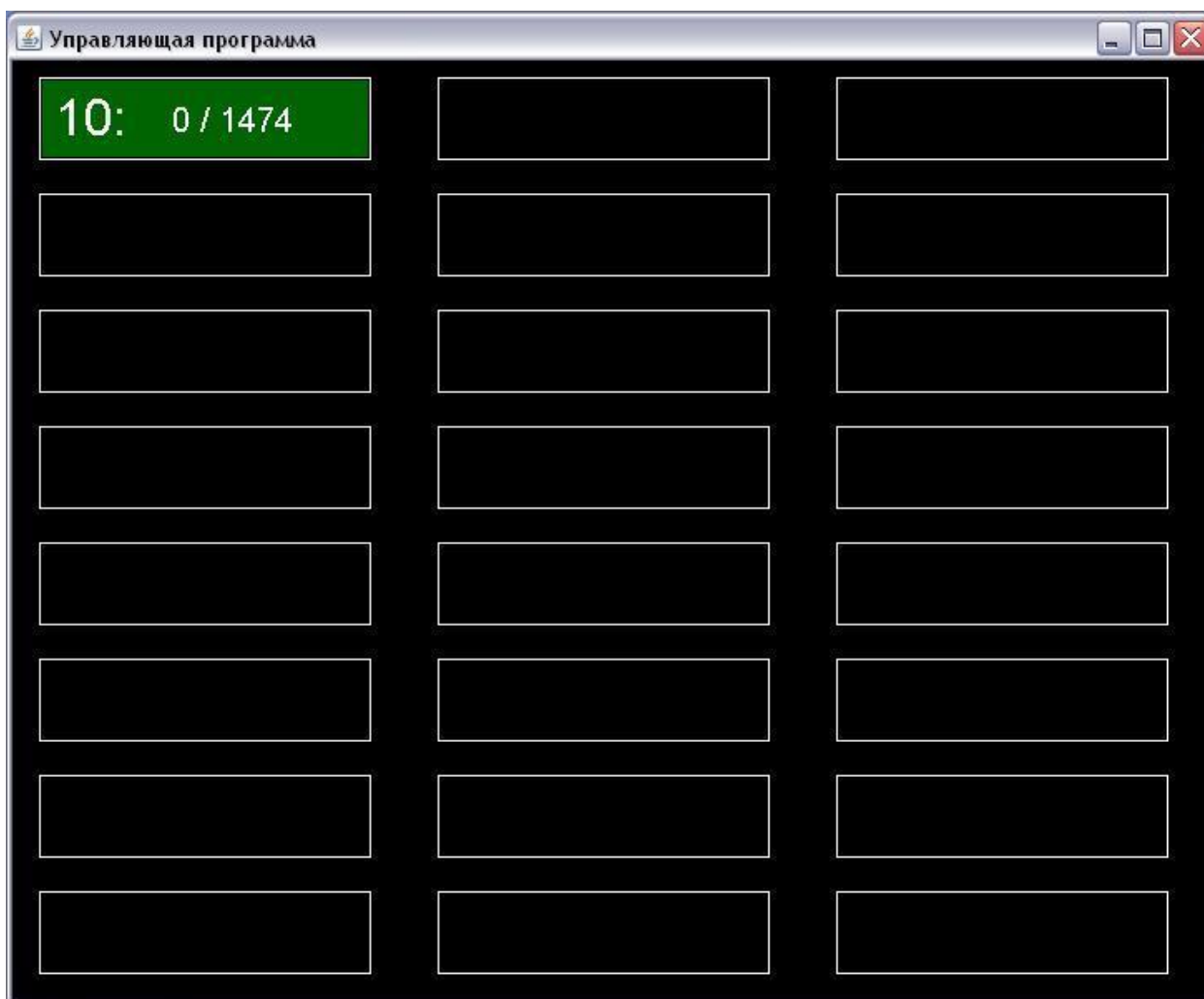


Рис. 20. Окно начальной загрузки лекционного пакетного файла **Lect\_1.bat**.

При первой загрузке файла **Lect\*.bat** меню содержит один прямоугольник (рис.20), т.е. остальные части пакетного файла недоступны. В прямоугольнике указаны три числа: номер файла **Symb\_\*.txt** (число в левой части прямоугольника) и два числа, заданные в виде дроби, в знаменателе которой указано количество символов в загружаемом файле **Symb\_\*.txt**, а в числителе количество прочитанных символов. При нажатии клавиши **Enter** программа **Pilot.class** завершается с кодом, равным 1. Этот код не может быть обработан условными операторами **if**. В результате запускается первая



командная строка, содержащая программу **ReadSymbols.class** и информационный файл **Symb\_\*.txt**. После посимвольного просмотра файла **Symb\_\*.txt** и завершения работы программы **ReadSymbols.class**, автоматически происходит повторная загрузка программы **Pilot.class**. В результате загрузки в окне появляется меню с двумя прямоугольниками. Меню в программе **Pilot.class** увеличивается на один прямоугольник после полного просмотра очередного информационного файла. В меню всегда выделен последний прямоугольник, который соответствует еще не просмотренному файлу. Для постраничного просмотра ранее прочитанных символов можно выбрать любой прямоугольник, двигаясь по меню с помощью клавиш **↑** и **↓**. Последний прямоугольник с надписью **Exit** появляется, когда полностью просмотрены все файлы **Symb\_\*.txt**, включенные в лекционный пакетный файл. В этом случае после нажатия клавиши **Enter** программа **Pilot.class** завершается с кодом, равным значению последней метки пакетного файла **Lect\_\*.bat**. В результате работа пакетного файла завершается. Код завершения пакетного файла генерируется также программой **Pilot.class** после нажатия клавиши **End**, что дает возможность прервать лекцию в любом месте.

### Литература

1. Попов А.А. Отображение в электронной форме процесса написания символов на доске при чтении лекций по дисциплинам из области точных наук. Материалы IV Всесоюзной научно-практической конференции МарГУ, 2007, с. 120-124.
2. Попов А.А. Программирование на языке Java приложений для сопровождения и иллюстрации лекций // В сборнике: Информатизация инженерного образования ИНФОРИНО -2014. Труды международной научно-практической конференции. 2014. С. 121-122.
3. Попов А.А. Комплекс программ для моделирования процесса написания символов при чтении лекций. Фонд алгоритмов и программ СО РАН, 2013. Свидетельство о регистрации программы № PR13036.