Теоретический материал для индивидуального задания №4

1) Печать

Для выполнения печати в Java просто нужно получить объект `Graphics`, который использует в качестве поверхности рисования принтер. Если у вас есть объект `Graphics`, вы можете печатать текст и выводить графику на принтер, как вы делаете это с экраном. Единственная хитрость относится к получению объекта `Graphics`, связанного с принтером.

1.1 Печать с помощью API Java

В примере 1 показан графический компонент, позволяющий пользователю рисовать мышью. Этот пример запоминает координаты выделенного фрагмента и при нажатии пользователем кнопки `Print` распечатывает его.

В примере использованы библиотеки `Swing`, `Java 2D` и интерфейс печати Java в пакете `java.awt.print`. Используется класс `java.awt.print.PrinterJob`. В этом примере наш класс реализует интерфейс `java.awt.print.Printable` и для непосредственного выполнения печати используется собственный метод компонента `paintComponent`().

1.2 Печать многостраничных текстовых документов

Печать многостраничных документов требует определенных ухищрений, поскольку мы должны решить, где помещать разрывы страниц. В примере 2 показано, как это может быть сделано. Данный класс `HardcopyWriter` является производным потоком `java.io.Writer`, который использует интерфейс печати Java для печати пересылаемых через него символов, вставляя при необходимости разрывы строк и страниц.

Класс `HardcopyWriter` включает две демонстрационные программы, реализованные в виде внутренних классов. Первая, `PrintFile`, читает указанный текстовый файл и распечатывает его, посылая его содержимое в поток `HardcopyWriter`. Вторая, `Demo`, распечатывает демонстрационную страницу, показывающую возможности этого класса по управлению шрифтами и табуляцией.

1.3 Печать Swing-документов

В примере 3 заложена возможность печати, основанная на классе `PrintableDocument`, который реализует интерфейсы `Printable` и `Pageable`. `Pageable` представляет многостраничный документ и определяет три метода: `getNumberOfPages()`, возвращающий количество страниц в документе; `getPageFormat()`, возвращающий размер и ориентацию каждой страницы документа; `getPrintable()`, который возвращает объект `Printable`, представляющий каждую страницу.

Кроме интерфейсов `Pageable` и `Printable` в примере показана внутренняя работа пакета `javax.swing.text`. В этом пакете объекты `Document` составлены из вложенных объектов `Element`, отображаемых на экран (или принтер) с помощью параллельной иерархии объектов `View`.

Реализация интерфейса `Printable` показана также в примере 4, который мы рассматривали ранее. Изучите более подробно внутренний класс `PrintableExample` .

Демонстрационные примеры (в папке Примеры)

Для запуска примера используйте команду go.cmd в каталоге примера (требуется java.exe, возможно нужно скорректировать переменную PATH).

Дополнительный материал в книге _Books\ corejava2_1.djvu

Литература

1. Хабибуллин И. Ш. Java 7. — СПб.: БХВ-Питербург, 2012
2. Г. Шилдт. Java . Полное руководство, 8-е издание, М.: ООО «И.Д. Вильямс»,
3. 2012
4. Кей С. Хорстман. Java2 Основы. Том 1. С.-Питербург. 2007 (_Books\ corejava2_1.djvu)
5. Кей С. Хорстман. Java2 Тонкости программирования. Том 2. С.-Питербург. 2007 (_Books\ corejava2_1.djvu)
6. http://docs.oracle.com/javase/7/docs/