

Реализация графической части интерпретатора языка PostScript

Артур Гудиев

444 группа

29 мая 2014 г.

Научный руководитель: Д.Ю. Булычев

PostScript – динамический интерпретируемый язык

- Создавался с целью представления цифровой графики в машино-независимой форме
- Используется в современных принтерах для управления процессом печати

Важной особенностью интерпретатора является его переносимость

- Программы на Java переводятся в специальный байт-код Java
- Виртуальная машина Java широко распространена

Введение

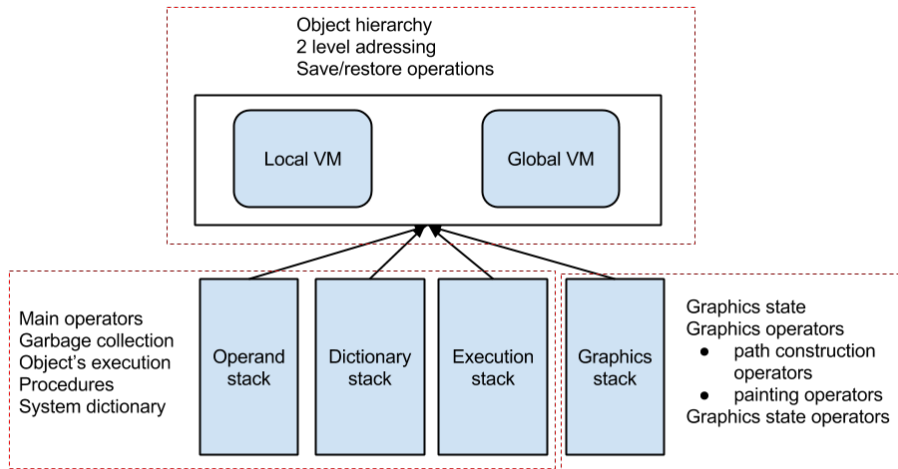
Графические возможности языка PostScript

- Графические объекты, состоящие из векторов, дуг и кубических кривых
- Графические операторы прорисовки траекторий, заливки и ограничения области рисования
- Координатная система, которая поддерживает все комбинации линейных преобразований
- Операторы сохранения и восстановления графического состояния `gsave` и `grestore`

Интерпретация программ PostScript на JVM

- Реализация графической части интерпретатора языка PostScript

Общая архитектура



Первый этап реализации

- Собственная реализация класса траектории, описывающего графические объекты
 - ▶ Отдельные классы для векторов, дуг и кубических кривых
- Собственная реализация алгоритмов прорисовки, заливки области рисования
 - ▶ Попытка реализации заливки по правилу *Non-zero winding number rule*
- Нереализованные графические параметры

Второй этап реализации

- Класс траектории реализован с помощью стандартного класса `GeneralPath`
- Есть встроенные алгоритмы заливки и прорисовки
- Есть стандартные возможности добавления графических элементов

Особенности реализации

Несоответствие между координатными системами Java и PostScript

Реализация графической среды исполнения

Разработанные классы:

- Класс `GState`, характеризующий графическое состояние
- Классы графических операторов:
 - ▶ операторы конструирования траекторий (`LineToOp`, `ArcOp`, `CurveOp`, `NewPathOp`...)
 - ▶ операторы раскраски (`StrokeOp`, `FillOp`, `EoFillOp`, `RectFillOp`)
 - ▶ операторы графического состояния (`GSaveOp`, `GRestoreOp`, `SetGrayOp`...)
 - ▶ операторы линейной трансформации координатной системы (`ScaleOp`, `RotateOp`, `TranslateOp`...)
- Класс `PSDrawer`, отображающий графическое состояние на экране

Использованные технологии

- библиотека `java.awt`
- библиотека `java.swing`

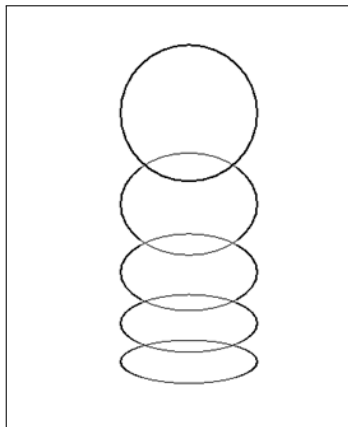
Пример программы на языке PostScript

```
/doACircle  
{ 0 0 54 0 360 arc stroke } def
```

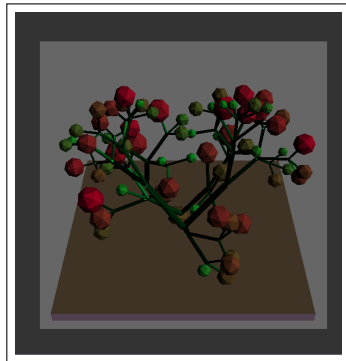
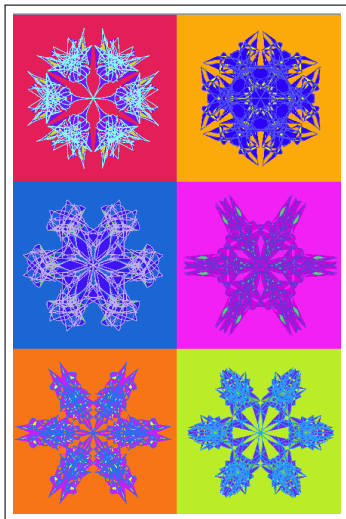
```
/doAnEllipse  
{ 1 .75 scale doACircle } def
```

```
300 500 translate doACircle
```

```
4 {0 -72 translate  
doAnEllipse } repeat  
showpage
```



Примеры



- Разработана графическая среда исполнения
 - ▶ Реализованы основные графические операторы
 - ▶ Разработаны операторы сохранения и восстановления графического состояния
- Есть возможность вывода на экран результата работы интерпретатора – получившейся картинке