Национальный исследовательский университет – Высшая школа экономики

Факультет бизнес-информатики, отделение программной инженерии

УТВЕРЖДЕНО

Заведующий кафедрой «Управление разработкой программного обеспечения»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Авдошин С.М./

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2012 г.

**КОмпонентная модель с декларативным описанием составных типов: парсеры**

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. инв. № |  | | Подп. и дата |  | | Инв. № подп. |  | | |  | | --- | | Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Гринкруг Е.М./  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2012 г. | | Исполнитель: студент группы 271ПИ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Дубов М.С. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2012 г. |   2012 |

Национальный исследовательский университет – Высшая школа экономики

Факультет бизнес-информатики, отделение программной инженерии

УТВЕРЖДЕНО

**Компонентная модель с декларативным описанием составных типов: парсеры**

Программа и методика испытаний

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. инв. № |  | | Подп. и дата |  | | Инв. № подп. |  | | Листов 7  2012 |

# Содержание

[Содержание 2](#_Toc325474388)

[1. Объект испытаний 3](#_Toc325474389)

[1.1. Назначение и область применения 3](#_Toc325474390)

[2. Цели испытаний 3](#_Toc325474391)

[3. Требования к программе 3](#_Toc325474392)

[4. Требования к программной документации 4](#_Toc325474393)

[5. Средства и порядок испытаний 4](#_Toc325474394)

[5.1. Требования к информационной и программной совместимости 4](#_Toc325474395)

[5.2. Требования к составу и параметрам технических средств 4](#_Toc325474396)

[6. Методы испытаний 4](#_Toc325474397)

[6.1. Испытание библиотеки средств синтаксического анализа 4](#_Toc325474398)

[6.2. Испытание библиотеки средств кодогенерации 5](#_Toc325474399)

[7. Приложение А. Код тестовых файлов 6](#_Toc325474400)

[7.1. Example.wrl 6](#_Toc325474401)

[7.2. Example2.wrl 6](#_Toc325474402)

[7.3. Example\_errors.wrl 6](#_Toc325474403)

[7.4. Example.x3d 7](#_Toc325474404)

[7.5. Example2.x3d 7](#_Toc325474405)

# Объект испытаний

Библиотека средств синтаксического анализа VRML и X3D-файлов и построения на их основе компонентных моделей состоит из двух компонент:

* Первая компонента – набор средств для синтаксического анализа (парсинга) декларативного описания компонентных моделей;
* Вторая компонента – набор средств для генерации декларативного описания компонентных моделей.

## Назначение и область применения

* + 1. *Функциональное назначение*

Программный комплекс предназначен для построения компонентных моделей на основе их описания на одном из поддерживаемых декларативных языков (VRML/X3D), а также для генерации декларативного описания уже существующих моделей.

* + 1. *Эксплуатационное назначение*

Библиотека предназначена для использования сторонними разработчиками при разработке ими других приложений.

# Цели испытаний

Целью проведения испытаний является проверка корректности работы компонент программного комплекса и правильности комплекта поставки.

# Требования к программе

В процессе испытаний необходимо проверить соответствие программного комплекса следующим требованиям:

Синтаксический анализ описаний моделей на языках VRML и X3D проходит без ошибок для корректных исходных файлов; результате синтаксического анализа VRML и X3D файлов, описывающих одну и ту же модель, строится один и тот же граф сцены.

В случае наличия ошибок в исходном файле парсер останавливает свою работу и выдает соответствующее сообщение;

Осуществляется диагностика лексических ошибок в коде описания модели («опечатки» в указании типов узлов и др.);

Осуществляется диагностика синтаксических ошибок в коде описания модели (отсутствие открывающих/закрывающих скобок и др.);

Осуществляется диагностика семантических ошибок в коде описания модели (несоответствия типов и др.);

Генерация VRML- и X3D-кода осуществляется без ошибок и без искажений/потерь информации;

Конвертирование файлов из VRML-формата в X3D-формат и обратно осуществляется без ошибок и без искажений/потерь информации.

# Требования к программной документации

В процессе разработки должны быть созданы следующие документы:

* Текст программы (ГОСТ 19.401-78)
* Пояснительная записка (ГОСТ 19.404-79)
* Программа и методика испытаний (ГОСТ 19.301-79)
* Руководство программиста (ГОСТ 19.504-79)

# Средства и порядок испытаний

## Требования к информационной и программной совместимости

Для работы библиотеки необходима реализация виртуальной машины Java версии не ниже 6 (например, Java Runtime Environment). При запуске юнит-тестов библиотеки требуется наличие средств Java Development Kit и библиотеки JUnit.

Для визуального тестирования работы средств кодогенерации требуется наличие VRML- и X3D-плееров (например, пакета Instant Reality).

## 5.2. Требования к составу и параметрам технических средств

|  |  |  |  |
| --- | --- | --- | --- |
| Необходимый процессор | Рекомендуемый процессор | Необходимое ОЗУ | Рекомендуемое ОЗУ |
| Pentium 2  266 MHz\* | Pentium 2  266 MHz или с более высоким быстродействием | 128 MB\* | 128 MB или больше |

\*Или минимум, требуемый операционной системой, какой бы она ни была.

# Методы испытаний

Испытания проводятся на основе юнит-тестов, код которых поставляется вместе с библиотекой посредством веб-сервиса для хостинга открытых проектов GitHub или на носителе типа CD-ROM. К коду юнит-тестов прилагается набор примеров исходных VRML- и X3D-файлов, используемых в ходе испытаний.

## 6.1. Испытание библиотеки средств синтаксического анализа

Для испытания средств синтаксического анализа используются юнит-тесты пакета *ru.hse.se.parsers.test*, среди которых:

* *VRMLParserTest.java* – класс, тестирующий работу парсера VRML. Осуществляет:
  + загрузку двух корректных тестовых VRML-файлов (*Example.wrl* и *Example2.wrl*);
  + вызов парсера VRML;
  + обход построенного графа сцены и вывод информации о его узлах на консоль;
  + загрузку VRML-файла, содержащего ошибки (*Example\_errors.wrl*), и вывод на консоль список ошибок, выявленных в ходе работы парсера.
* *X3DParserTest.java* – класс, тестирующий работу парсера X3D. Осуществляет:
  + загрузку двух корректных тестовых X3D-файлов (*Example.x3d* и *Example2.x3d*);
  + вызов парсера X3D;
  + обход построенного графа сцены и вывод информации о его узлах на консоль;
* *ValueTypeParseMethodsTest.java* – класс, тестирующий работу методов *parse(String str)* в классах, представляющих типы значений.

По результатам выполнения юнит-тестов осуществляется анализ выведенной на консоль информации с целью установления корректности построенных графов; в случае с ошибочным файлом осуществляется анализ обнаруженных парсером ошибок (среди них должны быть как лексические, так и синтаксические, а также одна семантическая ошибка и одно предупреждение).

## 6.2. Испытание библиотеки средств кодогенерации

Для испытания средств синтаксического анализа используются юнит-тесты пакета *ru.hse.se.codogenerators.test*, среди которых:

* *CodeGeneratorTest.java* – класс, тестирующий работу кодогенераторов VRML и X3D. Осуществляет:
  + Конвертацию двух корректных VRML-файлов (*Example.wrl* и *Example2.wrl*) в два X3D-файла (*out.x3d* и *out2.x3d*);
  + Конвертацию двух корректных X3D-файлов (*Example.x3d* и *Example2.x3d*) в два VRML-файла (*out.wrl* и *out2.wrl*).

По результатам генерации двух файлов осуществляется сравнение файлов *Example.wrl* и *out.wrl*, *Example2.wrl* и *out2.wrl*, *Example.x3d* и *out.x3d*, *Example2.x3d* и *out2.x3d*. Каждая из этих пар файлов описывает аналогичные модели. Их сравнение может осуществляться как посредством их чтения, так и их запуска в VRML-плеере. Положительные результаты испытаний свидетельствуют о корректной работе как парсеров, так и кодогенераторов.

# Приложение А. Код тестовых файлов

## 7.1. Example.wrl

|  |
| --- |
| #VRML 97  DEF shape1 Shape  {  appearance Appearance  {  material DEF mat Material  {  diffuseColor -2e-1 -0.2 +0.71  } # Material  } # Appearance  geometry Sphere  {  radius 1.2  } # Sphere  } # Shape  Shape {  geometry Box {}  }  Shape {  appearance DEF app\_1 Appearance {  material USE mat  }  geometry Text {  string [ "VRML" "XML" ]  length [ 3.5, 3.0 ]  }  }  USE shape1 |

## 7.2. Example2.wrl

|  |
| --- |
| #VRML 97  Group {  children [  Box {}  Sphere { radius 1.3 }  ]  } |

## 7.3. Example\_errors.wrl

|  |
| --- |
| #VRML 97  DEF shape1 Shape  {  appearance Appearance {  material DEF mat Material  {  diffuseColor -2e-1 +0.71 # 2 floats instead of 3  } # Material  } # Appearance  geomeetry Sphere # geomeetry <-> geometry  {  radius 1.2  } # Sphere  } # Shape  Shape # no '{'  appearance Appearance {  material DEF mat Material # Warning: 'mat' defined  {diffuesColor 1 0 1}} # diffuesColor  geometry Bx {} # Bx <-> Box  }  Shape {  appearance DF app\_1 Appearance { # DF <-> DEF  material USE mat  }  geometry Material { # Type mismatch  diffuseColor 0 0 0.2  # no '}'  }  USE shap1 # read as a field |

## 7.4. Example.x3d

|  |
| --- |
| <Scene>  <Shape DEF='shape1'>  <Appearance>  <Material DEF='mat' diffuseColor='-2e-1 -0.2 +0.71' />  </Appearance>  <Sphere radius='1.2'/>  </Shape>    <Shape>  <Box />  </Shape>    <Shape>  <Appearance DEF='app\_1'>  <Material USE='mat' />  </Appearance>  <Text string='"VRML" "XML"'>  <fieldValue name='length' value='3.5, 3.0' />  </Text>  </Shape>    <Shape USE='shape1' />  </Scene> |

## 7.5. Example2.x3d

|  |
| --- |
| <X3D>  <Group>  <fieldValue name='children'>  <Box />  <Sphere radius='1.3'/>  </fieldValue>  </Group>  </X3D> |