|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»  Факультет бизнес-информатики, отделение программной инженерии  Кафедра управления разработкой программного обеспечения | | | | | |
| СОГЛАСОВАНО  Старший преподаватель каф. УРПО отделения Программной инженерии Национального исследовательского университета «Высшая школа экономики»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.В. Дворянский  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 г. | |  | УТВЕРЖДАЮ  Заведующий отделением Программной инженерии Национального исследовательского университета «Высшая школа экономики»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин  «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 г. | | |
| **«СИСТЕМА КОДОГЕНЕРАЦИИ ИЗ МОДЕЛИ ВЛОЖЕННЫХ СЕТЕЙ ПЕТРИ В СИСТЕМЫ КОМПОНЕНТ НА БАЗЕ ENTERPRISE JAVABEANS»**  **«CODE GENERATION SYSTEM FROM NESTED PETRI NETS TO SYSTEMS OF ENTERPRISE JAVABEANS COMPONENTS»**  Программа и методика испытаний | | | | | |
| Подп. и дата |  | **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729. 501620-01 51 01-1-ЛУ**  **Листов 10** | | | | | |
| Инв. № дубл. |  |  | | | | | |
| Взам. инв. № |  |  |  | | | Исполнитель  студент группы 372 ПИ  \_\_\_\_\_\_\_\_\_\_\_\_/Николенко А.П. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2013 г. | |
| Подп. и дата |  |  | | | | |  |
| Инв. № подл. |  | **2013** | | | | | |

|  |  |  |
| --- | --- | --- |
|  |  | УТВЕРЖДЕН  RU.17701729. 501620-01 51 01-1-ЛУ  «Система кодогенерации из модели вложенных сетей Петри в системы компонент на базе Enterprise JavaBeans»  «Code Generation System from Nested Petri Nets to Systems of Enterprise JavaBeans Components»  Программа и методика испытаний  RU.17701729. 501620-01 51 01-1    Листов 10 |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл. |  |

2013

ОГЛАВЛЕНИЕ

[1 Объект испытаний 2](#_Toc367948641)

[1.1 Наименование программы 2](#_Toc367948642)

[1.2 Область применения программы 2](#_Toc367948643)

[2 Цель испытаний 2](#_Toc367948644)

[3 Требования к программе 2](#_Toc367948645)

[3.1. Требования к функциональным характеристикам 2](#_Toc367948649)

[4 Требования к программной документации 2](#_Toc367948650)

[5 Средства и порядок испытаний 3](#_Toc367948651)

[5.1. Технические средства 3](#_Toc367948654)

[5.2. Программные средства 3](#_Toc367948655)

[5.3. Порядок проведения испытаний 3](#_Toc367948656)

[6 Методы испытаний 4](#_Toc367948657)

[6.1. Установка программы 4](#_Toc367948664)

[6.2. Запуск программы 4](#_Toc367948665)

[6.3. Задание путей к необходимым компонентам 4](#_Toc367948666)

[6.4. Задание входного файла 5](#_Toc367948667)

[6.5. Обработка входного файла 5](#_Toc367948668)

[6.6 Проверка выходных файлов 6](#_Toc367948669)

[6.7 Проверка работоспособности 8](#_Toc367948670)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 9](#_Toc367948671)

# Объект испытаний

## Наименование программы

«Система кодогенерации из модели вложенных сетей Петри в системы компонент на базе Enterprise JavaBeans».

## Область применения программы

Программа предназначена для создания систем EJB компонент без необходимости прибегать к программированию.

# Цель испытаний

Целью испытаний является проверка правильности выполнения программой функций, перечисленных в разделе «Требования к программе».

# Требования к программе



## Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1. Загрузка и парсинг файла с описанием вложенной сети Петри.
2. Генерация EJB компонент на основе описания.
3. Компиляция EJB компонент.
4. Упаковка скомпилированных компонент в JAR файл для тестирования на локальном сервере.
5. Развертывание системы компонент на локальном сервере.

# Требования к программной документации

В процессе разработки должны быть созданы следующие документы:

* Пояснительная записка (ГОСТ 19.404-79);
* руководство оператора (ГОСТ 19.505-79);
* программа и методика испытаний (ГОСТ 19.301-79);
* текст программы (ГОСТ 19.401-78).

# Средства и порядок испытаний



## Технические средства

|  |  |  |
| --- | --- | --- |
| *Необходимый процессор* | *Необходимое ОЗУ* | *Прочие требования* |
| Intel Pentium/Celeron, AMD K6/Athlon/Duron с частотой 1.5 ГГц | 512 МБ | * Графический интерфейс (монитор и видеоадаптер Super VGA с разрешением 1024 X 768 или более высоким); * Клавиатура и мышь; * Видеокарта VGA, минимум 64 МБ; * CD – ROM. |
| *Рекомендуемый процессор* | *Рекомендуемое ОЗУ* |
| С частотой 2 ГГц и выше.  \* большая производительность при частоте больше 1.5 ГГц | 1024 МБ или больше  \* при наличии 512 МБ ОЗУ возможно снижение производительности и функциональности |

## Программные средства

Для работы программы необходимы:

* Установленный Ruby 1.9.3.
* Установленный Ruby gem Nokogiri 1.6.0
* Установленный Java Development Kit 7u40
* Установленный GlassFish Server 4.0 with Java EE 7

## Порядок проведения испытаний

Испытания должны выполняться в следующем порядке:

* 1. Выполнить стандартную процедуру начальной загрузки операционной системы;
  2. Установить программу
  3. Провести необходимые испытания, описанные в разделе «Методы испытаний»;
  4. Выйти из программы.
  5. Удалить ненужные сгенерированные файлы.

# Методы испытаний

Испытания представляют собой процесс установления соответствия программы заданным требованиям и программной документации.



## Установка программы

Скопируйте файлы EJBGenerator.rb и EJBGenerator.jar с носителя (CD-ROM) в необходимую директорию на жестком диске.

## Запуск программы

Откройте файл EJBGenerator.jar в каталоге, в который вы скопировали файл на предыдущем шаге.

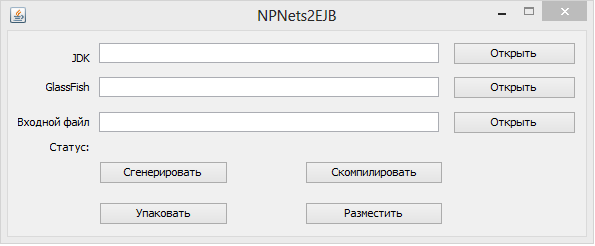


Рисунок Главное окно

## Задание путей к необходимым компонентам

В первом и втором полях приложения необходимо задать пути к установленным на системе JDK и GlassFish серверу. Это можно сделать вручную, либо с помощью диалога выбора папки.

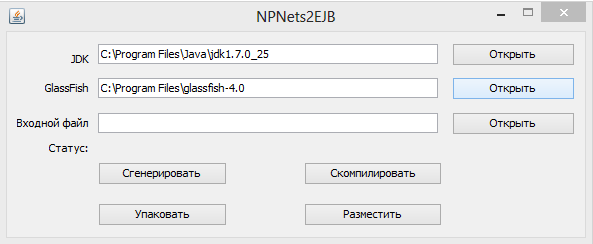


Рисунок 2. Пути к JDK и GlassFish заданы.

## Задание входного файла

В последнем поле задайте путь к тестовому XML файлу с описание структуры сети Петри.

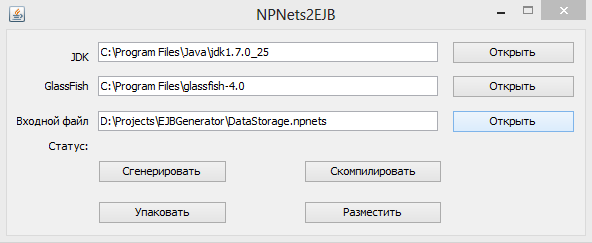


Рисунок 3. Входной файл задан

## Обработка входного файла

С помощью расположенных внизу четырех кнопок произведите генерацию EJB компонент, их компиляцию, упаковку и размещение на локальном сервере.

Статус операций будет показан в графе статус.

Результаты работы будут сохранены в папке output в той же директории что и входной файл.

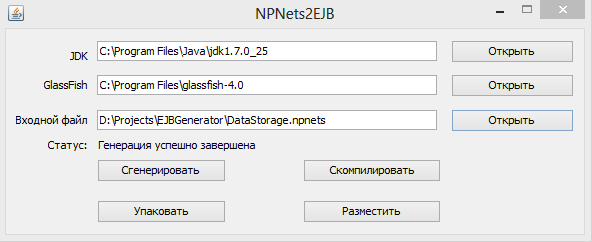


Рисунок 4. Окно программы после генерации EJB компонент

## 6.6 Проверка выходных файлов

В результате работы программы в директории входного файла должна быть создана папка

оutput, содержащая:

* Папки java/<имя сети>/ со сгенерированными java файлами

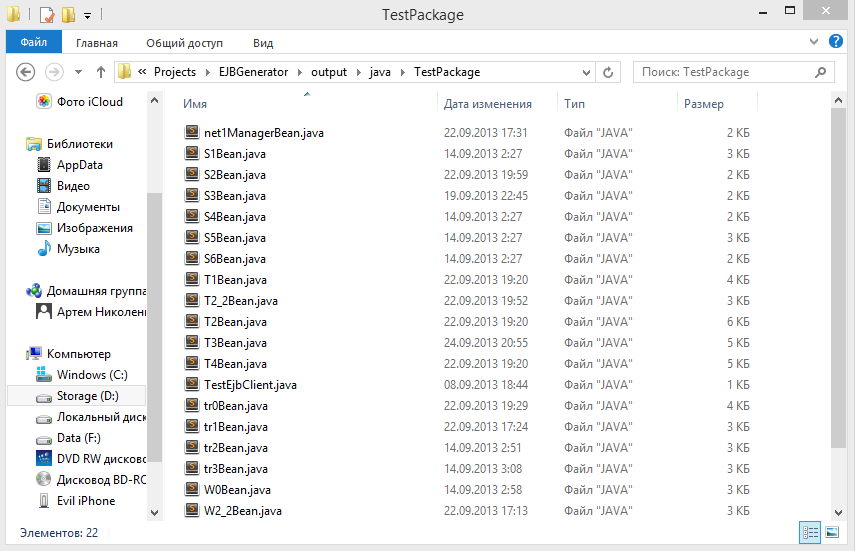


Рисунок 5. Сгенерированные java файлы

* Папки classes/<имя сети>/ со скомпилированными java файлами

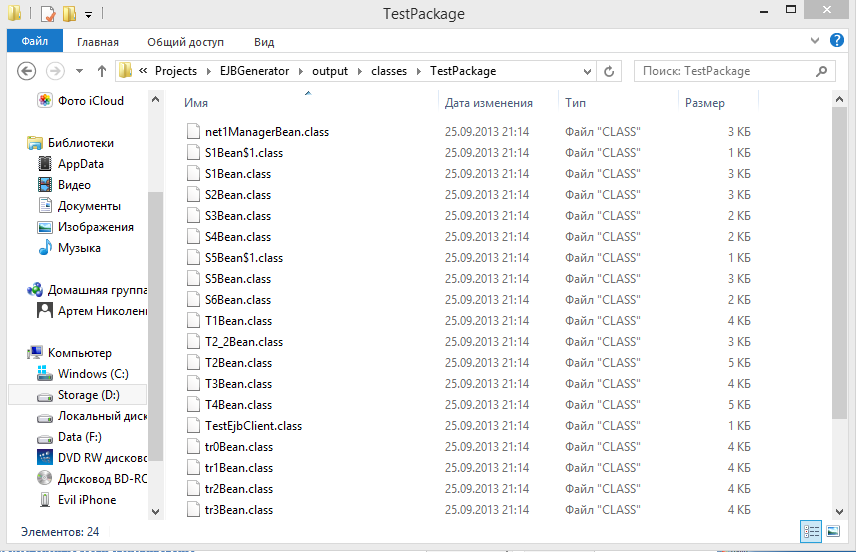


Рисунок 6. Скомпилированные java файлы

* Упакованный jar в корне директории output

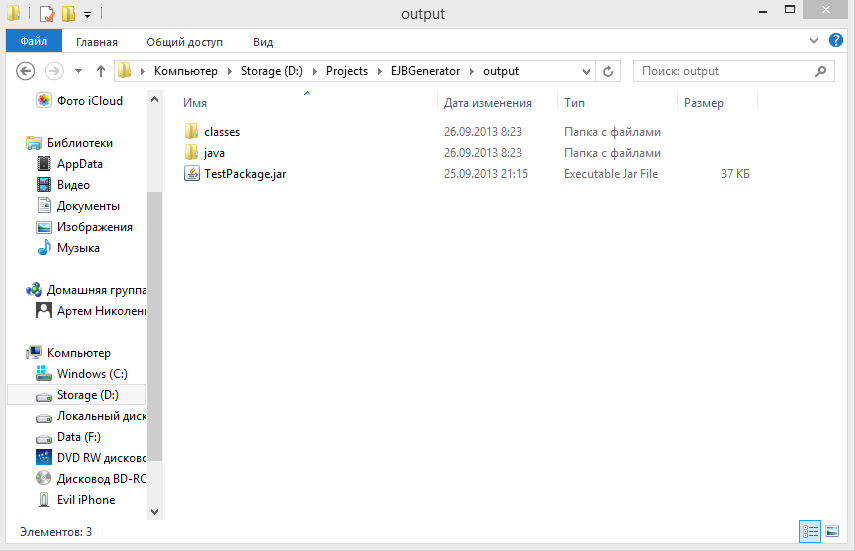


Рисунок 7. Jar файл

* Также jar файл должен быть скопирован в директорию autodeploy локального сервера Glassfish

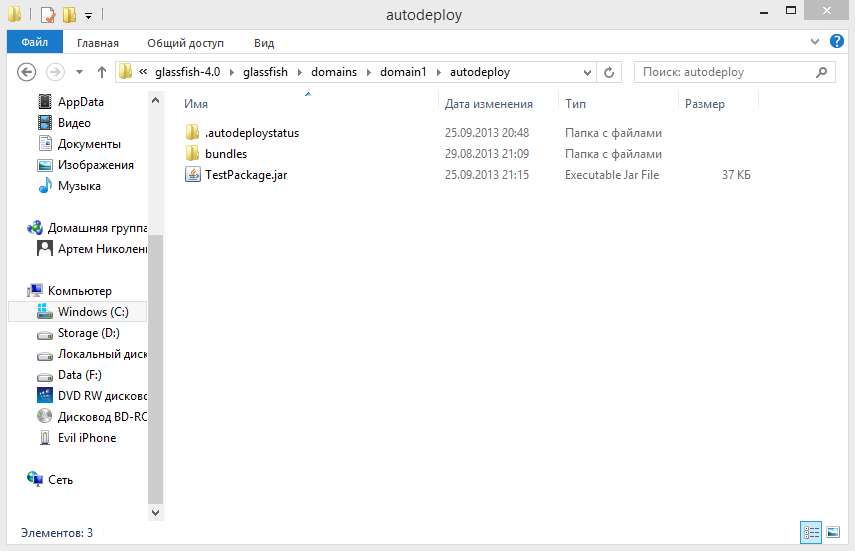


Рисунок 8. Jar файл в директории autodeploy

## 6.7 Проверка работоспособности

После этого можно запустить сервер командой asadmin start-domain –verbose

через командную строку из папки bin находящейся в корневой директории вашего сервера.

Приложение будет развернуто автоматически. Если в вашем приложении имеется консольный вывод, то вы сможете наблюдать его в консоли.

В качестве тестового примера используется модель склада инструментов из книги И. А. Ломазовой “Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой”

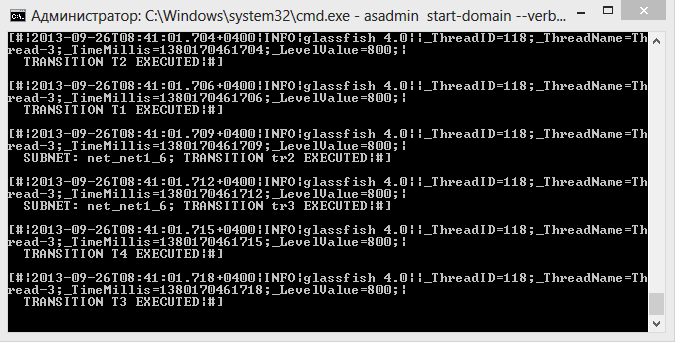


Рисунок 9. Консольный вывод приложения

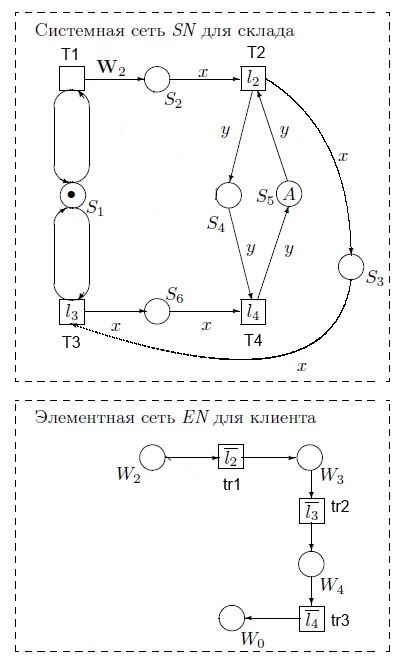


Рисунок 10. Модель сети

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |