Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра компьютерных систем и программных технологий

Отчет

По дисциплине «Базы данных»

«Разработка консольного приложения»

Работу выполнил студент группы № 43501/4 Смирнов Л.А. \_\_\_\_\_\_\_

Работу принял преподаватель Мяснов А.В. \_\_\_\_\_\_\_

Санкт-Петербург

2016

**1. Цель работы**

Разработать консольное приложение, реализующее следующие возможности:

* импорт и экспорт в xml нескольких таблиц
* проверка с помощью схемы

**2. Программа работы**

* Создать консольное приложение, которое сможет общаться с базой данных путем добавления и выборки данных при помощи sql-запросов
* Данные для импорта выбираются из файла .json и связаны с данными о призывнике
* Экспортируемые данные помещаются в файл .xml

**3. Выполнение задания**

Для доступа к базе данных посредством языка Java используется библиотека под названием JDBC (Java DataBase Connectivity – соединение с базами данных на Java). С помощью его возможно получение доступа к БД, а также выполнение SQL-запросов.

Приложение имеет консольный интерфейс. Все действия выполняются посредством ввода команд в консоли. В соответствии с заданием, приложение имеет 3 команды:

* Load <имя\_файла.json> - в ответ на запрос приложение загружает с заданного файла информацию о призывнике. Примерный вид файла для загрузки представлен ниже

|  |
| --- |
| {  "personalFile":[  {  "idFile":"1000",  "idRecruit":"1000",  "registrDate":"10-12-2016",  }  ],  "recruit" : [  {  "idRecruit": "1000",  "name": "Sokolov Vasya Vasilievich",  "birthDate": "01-01-1990",  "registrPlace": "Spb",  "idMother": "1000",  "idFather": "1000",  "idAcademic": "3"  }  ],  "fatherInfo" : [  {  "idFather": "1000",  "name": "Sokolov Vasya Alexeevich",  "birthDate": "03-01-1967",  "registrPlace": "Spb"  }  ],  "motherInfo" : [  {  "idMother": "1000",  "name": "Sokolova Maria Pavlovna",  "birthDate": "03-01-1970",  "registrPlace": "Spb"  }  ]  } |

Скрипт в формате json построен следующим образом: в любом порядке следования указывается название таблицы и все его поля с соответствующими значениями. Если какие-либо поля не заполнены, то программа выдаст ошибку и причину ее появления.

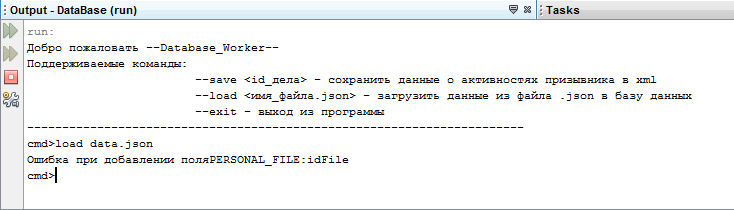
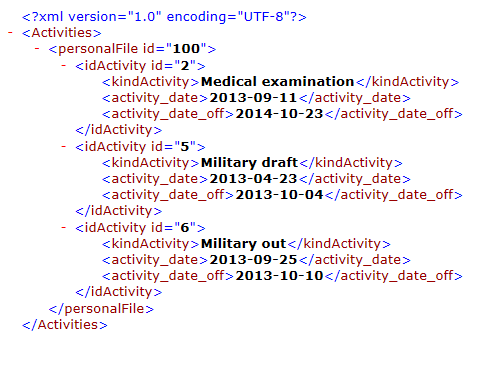


Рис.1

* Save <id\_дела> - команда позволяет вывести в xml-файл данные об «Активностях» призывника по ID личного дела. Формат вывода представлен ниже



Указывается ID-личного дела, ID-активности, вид активности, и даты начала и конца активностей.

* Exit – прекращение выполнения программы

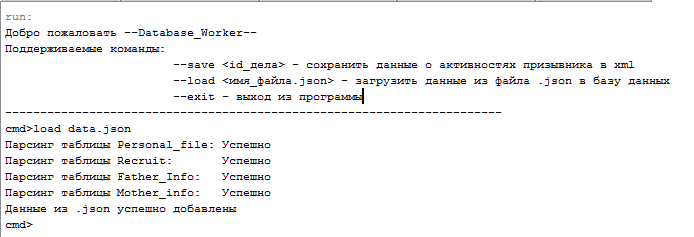
**4. Тестирование приложения**

Добавим в базу данных данные из файла data.json

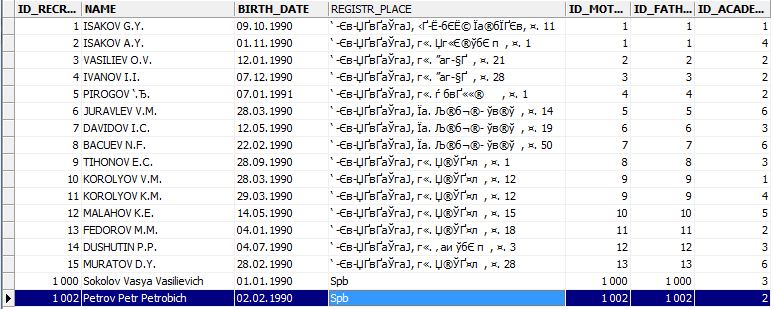
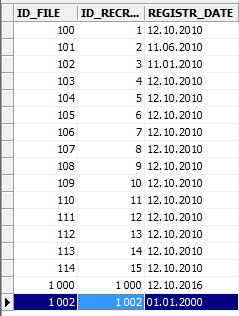
Листинг файла data.json:

|  |
| --- |
| {  "personalFile":[  {  "idFile":"1002",  "idRecruit":"1002",  "registrDate":"01-01-2000"  }  ],  "recruit" : [  {  "idRecruit": "1002",  "name": "Petrov Petr Petrobich",  "birthDate": "02-02-1990",  "registrPlace": "Spb",  "idMother": "1002",  "idFather": "1002",  "idAcademic": "2"  }  ],  "fatherInfo" : [  {  "idFather": "1002",  "name": "Petrov Petr Alexeevich",  "birthDate": "01-04-1969",  "registrPlace": "Spb"  }  ],  "motherInfo" : [  {  "idMother": "1002",  "name": "Petrova Elena Petrova",  "birthDate": "03-01-1974",  "registrPlace": "Spb"  }  ]  } |

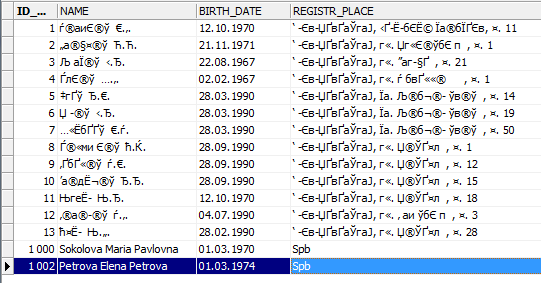
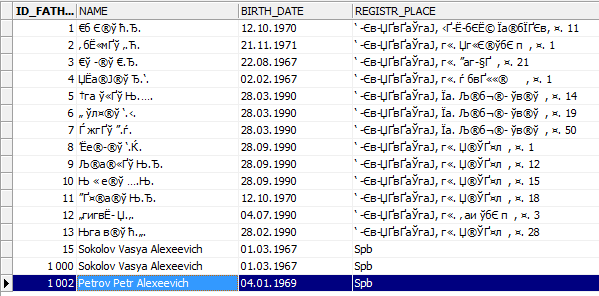
Выполним команду:



Теперь проверим с помощью программы IBExpert, добавились ли данные из json файла:



PERSONAL\_FILE RECRUIT

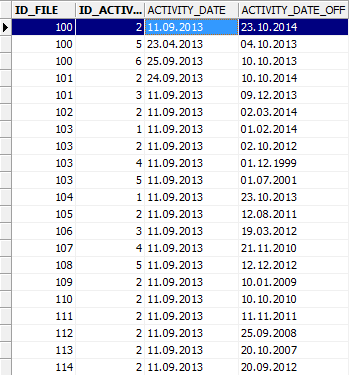


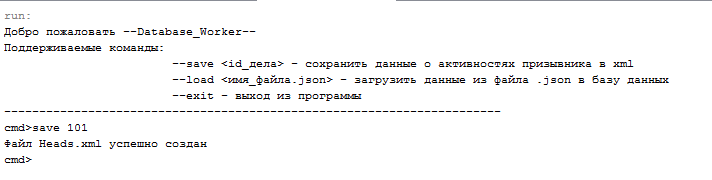
MOTHER\_INFO FATHER\_INFO

Видим, что все данные были успешно добавлены в каждую из таблиц.

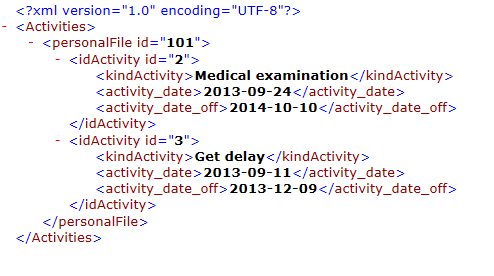
Также проверим, как сохраняются данные из таблицы Activities.

Рассмотрим таблицу Activities и попробуем сохранить данные о призывнике с номером дела 101





Файл Heads.xml:



**Вывод:**

В результате выполнения работы было написано и оттестировано приложение работающее с базой данных. Доступ к базе данных осуществлялся с помощью библиотеки JDBC.

Преимущества JDBС:

* Лёгкость разработки: разработчик может не знать специфики базы данных, с которой работает;
* Код практически не меняется, если компания переходит на другую базу данных (количество изменений зависит исключительно от различий между диалектами SQL);
* Не нужно устанавливать громоздкую клиентскую программу;
* К любой базе можно подсоединиться через легко описываемый [URL](https://ru.wikipedia.org/wiki/URL).

Все действия с любой базой данных выполняются через SQL запросы, поэтому нам достаточно использовать данную библиотеку для манипуляцией данными в БД.