Котов Виталий Евгеньевич

**Система учета студентов из числа иностранных граждан**

Дипломная *работа* по специальности

1-31-03-4 «Информатика»

Научный руководитель

Доцент кафедры информатика

Платонов Александр Сергеевич

Дипломная работа допущена к защите

Заведующий кафедрой информатики

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. Н. Батан

\_\_ \_\_\_\_\_\_\_\_\_\_\_ 2015

# **РЕФЕРАТ**

Котов Виталий Евгеньевич. Тема дипломной работы: система учета студентов из числа иностранных граждан.

Java, REST-сервисы, MySql, AngularJS, Intellij Idea, TomCat, Maven.

Разработанное приложение позволяет вести учет студентов из числа иностранных граждан и составление статистики.

# **ОГЛАВЛЕНИЕ**

[РЕФЕРАТ 2](#_Toc417298907)

[ОГЛАВЛЕНИЕ 3](#_Toc417298908)

[ВВЕДЕНИЕ 4](#_Toc417298909)

[ОБЗОР ЛИТЕРАТУРНЫХ ИССТОЧНИКОВ ПО ТЕМЕ 6](#_Toc417298910)

[ГЛАВА 1 ТЕОРИТИЧЕСКАЯ ЧАСТЬ 7](#_Toc417298911)

[1.1 Сведения о MySQL 7](#_Toc417298912)

[1.2 Сведения о REST 7](#_Toc417298913)

[1.3 Сведения о Java 9](#_Toc417298914)

[ГЛАВА 2 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ 10](#_Toc417298915)

[2.1 Анализ предметной области и объекта исследования 10](#_Toc417298916)

[2.2 Постановка задачи 10](#_Toc417298917)

[2.3 Входные и выходные документы 11](#_Toc417298918)

[2.4 Анализ бизнес-процессов 11](#_Toc417298919)

[ГЛАВА 3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 13](#_Toc417298920)

[3.1 Построение инфологической модели 13](#_Toc417298921)

[3.2 Требования к разрабатываемой 14](#_Toc417298922)

[3.3 Логическая модель данных 14](#_Toc417298923)

[3.4 Физическое проектирование 14](#_Toc417298924)

[3.5 Модель интерфейса 14](#_Toc417298925)

[ГЛАВА 4 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 15](#_Toc417298926)

[4.1 Обоснование выбора ПО для разработки 15](#_Toc417298927)

[4.2 Структура ИС и функционирование каждого блока 16](#_Toc417298928)

[4.3 Структура базы данных 16](#_Toc417298929)

[4.4 Результаты работы ИС 17](#_Toc417298930)

[ЗАКЛЮЧЕНИЕ 18](#_Toc417298931)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc417298932)

[ПРИЛОЖЕНИЯ 20](#_Toc417298933)

# ВВЕДЕНИЕ

Основные идеи современной информационной технологии базируются на концепции, согласно которой данные должны быть организованы в базы данных с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей. Эти базы данных создаются и функционируют под управлением специальных программных комплексов, называемых системами управления базами данных (СУБД).

Одним из ключевых направлений в области автоматизация бизнес-процессов с использованием информационных технологий является разработка баз данных, позволяющих решить проблему хранения и систематизации информации согласно индивидуальным требованиям компании.

Увеличение объема и структурной сложности хранимых данных, расширение круга пользователей информационных систем привели к широкому распространению наиболее удобных и сравнительно простых для понимания реляционных (табличных) СУБД. Для обеспечения одновременного доступа к данным множества пользователей, нередко расположенных достаточно далеко друг от друга и от места хранения баз данных, созданы сетевые мультипользовательские версии БД основанных на реляционной структуре. В них тем или иным путем решаются специфические проблемы параллельных процессов, целостности (правильности) и безопасности данных, а также санкционирования доступа.

Для проектируемой системы предметной областью является отдел университета, который занимается учетом студентов из числа иностранных граждан. Необходимо спроектировать базу данных, информация которой будет использоваться для учета студентов и составления статистики.

**Предметом исследования** является организация процесса учета студентов из числа иностранных граждан, а также расчет статистики.

**Объектом исследования** является реальный экономический объект: отдел, который отвечает за учет студентов.

**Цель исследования:** разработать автоматизированную систему для оптимизации работы отдела. Разрабатываемый прототип будет предоставлять возможность вести автоматизированный учет студентов и расчет статистических данных.

# ОБЗОР ЛИТЕРАТУРНЫХ ИССТОЧНИКОВ ПО ТЕМЕ

1. ТЕОРИТИЧЕСКАЯ ЧАСТЬ

## Сведения о MySQL

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

MySQL портирована на большое количество платформ: Linux, Mac OS, Solaris, SunOS, Windows. MySQL имеет API для языков Delphi, C, C++, Java, PHP, Python, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC. MyODBC работает на всех системах Microsoft Windows и на большинстве платформ Unix.

В реализуемом приложении используется MySQL Server версии 5.6.

## Сведения о REST

REST (сокр. от англ. Representational State Transfer — «передача состояния представления») — метод взаимодействия компонентов распределённого приложения в сети Интернет, при котором вызов удаленной процедуры представляет собой обычный HTTP-запрос (обычно GET или POST), а необходимые данные передаются в качестве параметров запроса. Этот способ является альтернативой более сложным методам, таким как SOAP и CORBA.

В широком смысле REST означает концепцию построения распределённого приложения, при которой компоненты взаимодействуют наподобие взаимодействия клиентов и серверов во Всемирной паутине.

Хотя данная концепция лежит в самой основе Всемирной паутины, термин REST был введён Роем Филдингом, одним из создателей протокола HTTP, лишь в 2000 году. В своей диссертации в Калифорнийском университете в Ирвайне он подвёл теоретическую основу под метод взаимодействия клиентов и серверов во Всемирной паутине, абстрагировав его и назвав «передачей репрезентативного состояния». Филдинг описал концепцию построения распределённого приложения, при которой каждый запрос (REST-запрос) клиента к серверу содержит в себе исчерпывающую информацию о желаемом ответе сервера (желаемом репрезентативном состоянии), и сервер не обязан сохранять информацию о состоянии клиента («клиентской сессии»).

В качестве необходимых условий для построения распределенных REST-приложений Филдинг перечислил следующие:

* Клиент-серверная архитектура.
* Сервер не обязан сохранять информацию о состоянии клиента.
* В каждом запросе клиента должно явно содержаться указание о возможности кэширования ответа и получения ответа из существующего кэша.
* Клиент может взаимодействовать не напрямую с сервером, а с произвольным количеством промежуточных узлов. При этом клиент может не знать о существовании промежуточных узлов, за исключением случаев передачи конфиденциальной информации.
* Унифицированный программный интерфейс сервера. Филдинг приводил URI в качестве примера формата запросов к серверу, а в качестве примера ответа сервера форматы HTML, XML и JSON, различаемые с использованием идентификаторов MIME.

Филдинг указывал, что приложения, не соответствующие приведённым условиям, не могут называться REST-приложениями. Если же все условия соблюдены, то, по его мнению, приложение получит следующие преимущества:

* надёжность (за счет отсутствия необходимости сохранять информацию о состоянии клиента, которая может быть утеряна);
* производительность (за счет использования кэша);
* масштабируемость;
* прозрачность системы взаимодействия, особенно необходимая для приложений обслуживания сети;
* простота интерфейсов;
* портативность компонентов;
* легкость внесения изменений;
* способность эволюционировать, приспосабливаясь к новым требованиям (на примере Всемирной паутины).

## Сведения о Java

Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине вне зависимости от компьютерной архитектуры.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

В реализуемом приложении используется Java 8, релиз версии состоялся 19 марта 2014 года.

1. ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

## Анализ предметной области и объекта исследования

Для проектируемой базы данных предметной областью является процесс учета студентов. В учебном отделе собрана информация о студентах, которые обучаются на различных факультетах, а также в определенных группах. При этом, необходим учет их регистрации, места проживания, формы обучения, а также личной информации. Упрощение процесса учета студентов, подсчет их общего количества, а также составление статистики представляет интерес для данного исследования.

База данных студентов создается для экономии времени которое тратится на составление документации (отчетов) об учащихся студентах. При работе с такой базой, добавлять, изменять информацию в отчете гораздо проще и быстрее.

## Постановка задачи

Ставиться задача разработать программный комплекса, который позволит упростить работу отдела университета, отвечающего за учет студентов. Описываемый прототип позволит хранить информацию о факультетах, специальностях, группах, личную информацию, а также сведения регистрации и места проживания. Также функционал прототипа будет включать возможность различных видов подсчетов (общего количества студентов по специальностям, по странам, по общежитиям и т.д.).

## Входные и выходные документы

После проведения необходимых исследований и анализа предметной области, было выявлено, что необходимыми для разработки базы данных являются следующие входные документы:

1. Информация о студентах
2. Информация о факультетах
3. Информация о специальностях
4. Информация о группах
5. Информация о курсах
6. Информация о странах
7. Информация о общежитиях

Выходными документами будут списки студентов по специальностям, по странам, по общежитиям, списки с информацией о регистрации. В которых будет представлена информация о количестве студентов по курсам, а также их общее количество.

## Анализ бизнес-процессов

Трикотажная фабрика производит продукцию различного вида (пальто, костюмы и т.д.). Каждый продукт имеет несколько моделей (костюм в полоску, костюм клетчатый и т.д.). При составлении отчетности о поступлениях на склад фиксируются сведения именно о модели товара, которая прибыла на склад, а не о виде продукции. Учитывая это, в отчетную ведомость входят сведения о модели, а не о самом продукте.

Для составления отчетной ведомости потребуется информация о поступлениях моделей произведенных товаров на склад. Данные о поступлениях на склад должны содержать сведения о поступившей модели, дату поступления и количество поступивших моделей.

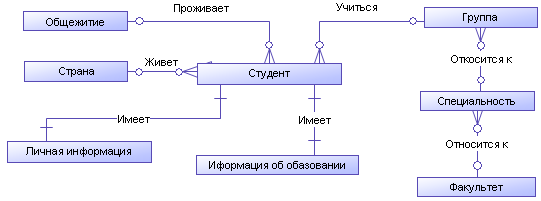
Также необходимо хранить информацию непосредственно о самих моделях. А именно название модели, цену модели, а также вид товара к которому относится модель.

У каждого потока имеется свой приоритет, который отчасти определяет, насколько часто поток получает доступ к ЦП. Низкоприоритетные потоки получают доступ к ЦП реже, чем высокоприоритетные. Таким образом, в течение заданного промежутка времени низкоприоритетному потоку будет доступно меньше времени ЦП, чем высокоприоритетному. Как и следовало ожидать, время ЦП, получаемое потоком, оказывает определяющее влияние на характер его выполнения и взаимодействия с другими потоками, исполняемыми в настоящий момент в системе.

1. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## Построение инфологической модели

Информация, предоставляемая в базе данных, в первую очередь должна отображать реальные объекты прикладной области и связи между ними.

Результатом инфологического проектирования является концептуальная модель, которая представляет структуру данных не зависимую от любой физической реализации. Инфологическую модель исследуемой области можно представить следующим образом:

* Студент – сущность, в которой храниться информация о студенте.
* Группа – сущность, содержащая информацию о группе в которой учатся студенты.
* Специальность – сущность, которая содержит данные о специальности которой обучаются группы студентов.
* Факультет – сущность.

Связи между сущностями следующие:

* В группе учатся студенты, причем студент связан только с одной группой, а группа хотя бы с одним студентом. Получаем, что сущности связаны между собой связью один ко многим (1:М);
* Специальность связана с несколькими группами, а группа с одной специальностью. Получаем, что сущности связаны между собой связью один ко многим (1:М). Аналогичным образам связаны сущности факультет и специальность, общежитие и студент, а также страна и студент;
* Студент имеет личную информацию и информацию об образовании. Эти сущности связаны между собой связью один к одному(1:1);

Выявим все атрибуты, описывающие сущности ER-модели.

……………

…………

.

## Требования к разрабатываемой

Учитывая особенности предметной области, разрабатываемый прототип должен поддерживать следующие возможности:

* Добавление, изменение и удаление студентов
* Перевод студента на следующий курс
* Отчисление студента

При этом функционал прототипа должен включать возможность просмотра списка всех студентов. А также в просмотр отчетов по курсам, специальностям, по странам и т. д.

## Логическая модель данных

Была выбрана, как чаще всего и поступают, реляционная модель данных в связи с наглядностью табличного представления данных и удобства работы с ними. Для каждой сущности ER-модели создадим таблицу. Имя сущности – имя таблицы.

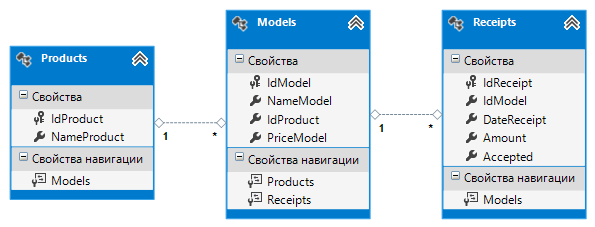
|  |
| --- |
| **Товары** |
| Код товара |
| Наименование товара |

|  |
| --- |
| **Модели** |
| Код модели |
| Наименование модели |
| Код товара |
| Цена модели |

|  |
| --- |
| **Поступления** |
| Код поступления |
| Код модели |
| Дата поступления |
| Количество |
| Кто принял товар |

А так же определим связи между таблицами посредством механизма первичных и внешних ключей.

Таблица Товары связана с таблицей Модели с помощью внешнего ключа Код товара из таблицы Модели. Таблица Модели связана с таблицей Поступления посредством внешнего ключа Код модели из таблицы Поступления. На рисунке 2 изображена схема связей между таблицами.



## Физическое проектирование

## Модель интерфейса

1. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

## Обоснование выбора ПО для разработки

Для создания программного комплекса была выбрана Intellij Idea, которая содержит все необходимое для начала работы, в том числе встроенный сервер для запуска и отладки MVC приложений, релиз SQL сервера для разработки приложений на основе баз данных, инструменты для тестирования и, конечно, компилятор и отладчик (дебаггер).

Одно из дополнений к Visual Studio 2012 и SQL Server 2012 – версия LocalDB. Это без административная реализация функций ядра SQL Server, предназначенных только для разработчиков. Используя эту версию можно пропустить процесс настройки базы данных, пока создаем проект, а затем развернуть приложение на полном экземпляре SQL Server.

В качестве доступа к данным использовалась технология ADO.NET Entity Framework. Entity Framework (EF) — это объектно-реляционный модуль сопоставления, позволяющий разработчикам .NET работать с реляционными данными с помощью объектов, специализированных для доменов. Это устраняет необходимость в написания большей части кода для доступа к данным, который обычно требуется разработчикам.

Создадим класс Forest в котором будет реализован лес.

struct ForestPoint

{

public bool heardAyy; //истина если в точке леса

//слышна «Ауу»!

public int personID;

}

static class Forest

{

private static int \_size = 50;//размер леса

public static ForestPoint[,] forest =

new ForestPoint[\_size, \_size];

public static int Size

{

get { return \_size; }

}

public static void Print()

{

for (int i = 0; i < \_size; i++)

{

for (int j = 0; j < \_size; j++)

{

//выводим в консоль букву в

//соответствии c personID

}

}

}

public static void PrintChanges()

{

Thread.Sleep(30);

Console.Clear();

Forest.Print();

}

}

где метод Print() выводит в консоль состояние леса.

## Структура ИС и функционирование каждого блока

размера леса.

## Структура базы данных

Классы Father, Mother, GrandFather, GrandMother наследуют класс LifeSaver. В классах Mother и GrandMother перегружается метод MovementAlgorithm(), реализующий алгоритм движения спасателя.

## Результаты работы ИС

В классе Girl реализован объект девочка. Класс имеет следующие поля: \_x, \_y – первоначальные координаты девочки; \_region – размер области где слышно девочку; movement – ссылка на объект класса Thread.

Класс Girl имеет следующий конструктор:

public Girl()

{

\_random = new Random();

\_region = 4;

\_x = \_random.Next(1, Forest.Size - 2);

\_y = \_random.Next(1, Forest.Size - 2);

lock (Forest.forest)

{

SetRegion(true);

Forest.PrintChanges();

}

movement = new Thread(MovementGirls);

movement.Start();

}

Реализации этого класса имеет следующие методы:

private void MovementGirls()

этот метод осуществляется движение девочки по лесу.

private void MoveGirl\_and\_Region(Move moveTo)

перемещает девочку и ее голос по лесу.

# ЗАКЛЮЧЕНИЕ

В данной дипломной работе была рассмотрена разработка и реализация автоматизированной системы учета студентов. В процессе разработки освоена технология доступа к данным Hibernate и MySQL Server, язык программирования Java и интегрированная среда разработки Intellij Idea.

В ходе дипломной работы были изучены принципы проектирования приложений с Rest архитектурой, а также разработка и реализация систем управления базой данных.

Был разработан компактный пользовательский интерфейс и использования JavaScript фреймворка AngularJS, который предоставляет возможность добавления, изменения и удаления данных. Функционирование приложения можно расширить возможностью выгрузки данных в Excel.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Рихтер Дж., CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C#, 2012. – 928 с.
2. Троелсен, Э., Язык программирования C# 5.0 и платформа .NET 4.5 6-е издание, 2012. – 1311 с.
3. Шилдт, Г., C# 4.0 Полное руководство, 2011. – 1056 с.
4. Microsoft Developer Network [Электронный ресурс] Режим доступа: http://msdn.microsoft.com/ru-ru/library/system.threading.aspx

Дата доступа: 10.05.2014.

# ПРИЛОЖЕНИЯ