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](#_Toc416992133)

[ОГЛАВЛЕНИЕ 3](#_Toc416992134)

[ВВЕДЕНИЕ 4](#_Toc416992135)

[ОБЗОР ЛИТЕРАТУРНЫХ ИССТОЧНИКОВ ПО ТЕМЕ 6](#_Toc416992136)

[ГЛАВА 1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ 7](#_Toc416992137)

[1.1 Анализ предметной области и объекта исследования 7](#_Toc416992138)

[1.2 Постановка задачи 7](#_Toc416992139)

[1.3 Входные и выходные документы 8](#_Toc416992140)

[1.4 Анализ бизнес-процессов 8](#_Toc416992141)

[ГЛАВА 2 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 10](#_Toc416992142)

[2.1 Построение инфологической модели 10](#_Toc416992143)

[2.2 Требования к разрабатываемой 11](#_Toc416992144)

[2.3 Логическая модель данных 11](#_Toc416992145)

[2.4 Физическое проектирование 11](#_Toc416992146)

[2.5 Модель интерфейса 11](#_Toc416992147)

[ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 12](#_Toc416992148)

[3.1 Обоснование выбора ПО для разработки 12](#_Toc416992149)

[3.2 Структура ИС и функционирование каждого блока 13](#_Toc416992150)

[3.3 Структура базы данных 13](#_Toc416992151)

[3.4 Результаты работы ИС 14](#_Toc416992152)

[ЗАКЛЮЧЕНИЕ 15](#_Toc416992153)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc416992154)

[ПРИЛОЖЕНИЯ 17](#_Toc416992155)

# 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. 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 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.

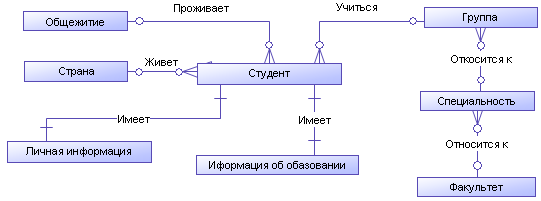
Также необходимо хранить информацию непосредственно о самих моделях. А именно название модели, цену модели, а также вид товара к которому относится модель.

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 студенты.
* …

Связи между сущностями следующие:

* Студент учится в группе, ;
* модель связана с несколькими поступлениями на склад, а поступление, связано с одной моделью;

## Требования к разрабатываемой

При разработке многопоточных программ следует быть особенно внимательным, чтобы избежать взаимоблокировки и состояний гонок. Взаимоблокировка, как подразумевает само название, — это ситуация, в которой один поток ожидает определенных действий от другого потока, а другой поток, в свою очередь, ожидает чего-то от первого потока. В итоге оба потока приостанавливаются, ожидая друг друга, и ни один из них не выполняется.

## Логическая модель данных

## 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