МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информационные технологии»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

ПО ДИСЦИПЛИНЕ

«СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ»

Ростов-на-Дону

ДГТУ

2020

УДК 372.8:004

Составители: М. В. Привалов, Е.В. Рашидова

Методические указания к выполнению лабораторных работ по дисциплине «Современные технологии программирования» / Сост.

М.В. Привалов, Е.В. Рашидова ; Донской гос. техн. ун-т. - Ростов-на-Дону: ДГТУ, 2020. – 16 с.

Рассматриваются современные подходы и технологии разработки программного обеспечения для Web-ориентированных и распределённых информационных систем.

Предназначены для обучающихся направления 09.04.02 «Информационные технологии» всех форм обучения.

УДК 372.8:004

Печатается по решению редакционно-издательского совета

Донского государственного технического университета

Ответственный за выпуск зав. кафедрой «Информационные технологии», д-р техн. наук, профессор Б.В. Соболь

В печать 13.07. 2020 г.

Формат 60×84/16. Объем 1,0 усл.п.л. Тираж 30 экз. Заказ № 138.

Издательский центр ДГТУ

Адрес университета и полиграфического предприятия:

344000, г. Ростов-на-Дону, пл. Гагарина,1

©Донской государственный технический университет, 2020

Лабораторная работа №1  
«ОПРЕДЕЛЕНИЕ ГРАНИЦ ПРОДУКТА И РАЗРАБОТКА ТРЕБОВАНИЙ»

Цель работы: научиться определять функциональные границы продуктов, определять свойства, необходимые для решения проблем заказчика, формулировать требования. Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

В соответствии с индивидуальным заданием:

1. Определите функциональные границы продукта и представьте их в виде документа Vision & Scope.
2. Выделите свойства продукта и разработайте соответствующий документ, используя лекционный пример Product Features в качестве шаблона.
3. Сформулируйте основные функциональные и нефункциональные требования к программному обеспечению (без детализации сценариев Use Case).

Индивидуальное задание

Индивидуальное задание согласовывается с преподавателем. Рекомендуется в качестве индивидуального задания использовать тему дипломного проекта или одно из перечисленных ниже заданий.

1. Спроектировать компьютерную подсистему учета Интернет-услуг
2. Спроектировать подсистему учета ресурса воздушных судов
3. Спроектировать компьютерную подсистему начисления и учета страховых выплат клиентам в условиях страховой компании
4. Спроектировать подсистему учета и анализа экономической деятельности в условиях агентства недвижимости
5. Спроектировать подсистему формирования и обработки договоров добровольного страхования наземного автотранспорта
6. Спроектировать подсистему обработки торговых агентов
7. Спроектировать подсистему планирования и учета чартеров
8. Спроектировать подсистему формирования и учета счетов-заказов в для туристических агентств
9. Спроектировать подсистему бронирования билетов в кинотеатр
10. Спроектировать подсистему расчета себестоимости для рыбного хозяйства
11. Спроектировать подсистему учета потребительских кредитов
12. Спроектировать подсистему реализации и движения горюче-смазочных материалов в условиях сети АЗС
13. Спроектировать подсистему учета продаж товаров
14. Спроектировать подсистему учета сбора и реализации зерновых культур
15. Спроектировать подсистему учета убытков автогражданской ответственности
16. Спроектировать подсистему учета выплат за услуги газоснабжения для населения
17. Спроектировать подсистему расчета арендной платы в условиях коммунального предприятия
18. Спроектировать подсистему учета движения грузового подвижного железнодорожного транспорта в условиях металлургического завода
19. Спроектировать подсистему учета сырья и материалов
20. Спроектировать подсистему приема электронных коммунальных платежей в условиях банка
21. Спроектировать подсистему учета и планирования ремонтных работ
22. Спроектировать подсистему учета технического состояния компьютерного оборудования

Лабораторная работа №2

«ОБРАБОТКА ИЕРАРХИЧЕСКИХ ДАННЫХ. ЯЗЫК ЗАПРОСОВ ДЛЯ ФОРМИРОВАНИЯ

ВЫБОРОК ПО ИЕРАРХИЯМ»

Цель работы: научиться представлять информацию в виде XML-документов, формировать их в приложениях C# и выполнять выборку интересующих данных с помощью языка XPath Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

1. Сформировать с помощью текстового редактора исходный XML-документ, содержащий не менее 10 записей, соответствующих индивидуальному заданию
2. Считать документ в программе и выполнить обработку в соответствии с заданием. При обработке обязательно использование выражений XPath для разбора информации и/или отбора узлов, соответствующих заданию.
3. Сформировать на диске результирующий XML-документ.
4. В заданиях с чётными вариантами при формировании узлов запрещается использование метода InnerXml. Вместо этого требуется работать с объектной моделью XML-документа

(присоединять элементы с помощью метода класса XmlElement AppendChild, и т.п.)

Индивидуальные задания

* 1. В расписании движения самолетов из аэропорта г. Донецка указаны следующие сведения: номер рейса (4 цифры), аэропорт назначения, расстояние в км, стоимость билета (взрослый билет, детский билет), время в часах и мин. (отправление, прибытие в аэропорт назначения). Сформировать документ со сведениями о трех рейсах, имеющих наибольшую продолжительность полета при расстоянии не большем заданного пользователем.
  2. В каталоге студии звукозаписи имеются следующие данные: название группы, название альбома, год выпуска альбома, название студии, записавшей альбом. Необходимо сформировать XML-каталог групп, выпустивших альбомы в заданном году и на заданной студии.
  3. В журнале успеваемости академгруппы по программированию имеются следующие данные: фамилия студента, оценки по пяти лабораторным работам, количество пропусков занятий. Определить трех студентов, имеющих наибольшее количество пропусков (студентов, сдавших все работы не включать). Записи в XML-документе расположить в алфавитном порядке.
  4. В магазине имеются следующие данные о товарах: название, единица измерения, цена, норма отпуска в одни руки. Составить документ, включающий элемент, содержащий список товаров, норма отпуска которых не более двух единиц, а также элемент, содержащий список товаров, стоимость которых превышает указанную пользователем. Списки упорядочить по названию товаров в алфавитном порядке.
  5. На заводе радиоэлектроники выпускают звуковоспроизводящую технику и имеются следующие данные: название прибора, назначение (магнитофон, магнитола, проигрыватель), год создания, стоимость, гарантийный срок эксплуатации. Составить XML-документ, включающий список магнитофонов, разработанных в заданном году, а также список проигрывателей, гарантийный срок эксплуатации которых более 3-х лет.
  6. В каталоге программного обеспечения имеются следующие данные: имя файла, расширение, размер файла, дата создания. Составить каталог текстовых файлов (расширение TXT, DOC), а также таблицу файлов, размером более 64 Кбайт. Каталоги отсортировать по имени файла в алфавитном порядке.
  7. В больнице ведется учет больных по следующим данным: фамилия больного, номер палаты, дата поступления, дата выписки (может отсутствовать), диагноз (название болезни). Необходимо выдать список больных, лежавших в больнице на заданную дату. Список сортировать по номеру палаты.
  8. На станции технического обслуживания автомобилей (СТО) ведется учет автомобилей, прошедших капитальный ремонт, по следующим данным: марка машины, серийный номер, пробег (в км) после предыдущего ремонте, год выпуска автомобиля. Необходимо составить документ, содержащий список машин, имеющих пробег более 100 000 км, а также список автомобилей, выпущенных после заданного года и прошедших ремонт. Списки сортировать по году выпуска машины.
  9. В библиотеке имеются следующие данные о книгах: название, фамилия автора, год издания, издательство, количество экземпляров книг в библиотеке. Необходимо сформировать XML-документ, содержащий список книг, изданных в заданном году и список книг, имеющихся в библиотеке в одном экземпляре. Список упорядочить по названию книг в алфавитном порядке.

11. В военкомате ведется учет юношей допризывного и призывного возраста. Имеются следующие данные: фамилия, год рождения, номер личного дела, годность к службе («годен» или «не годен»). Необходимо вывести список юношей, призываемых на службу в заданном году (по достижении 18 лет). Список упорядочить по году рождения.

* 1. В аптеке ведется учет лекарственных средств. Имеются следующие данные: название лекарства, цена одной упаковки, количество упаковок в аптеке, год выпуска, срок хранения (в годах). Необходимо вывести документ, содержащий список лекарств, не годных к употреблению на заданный год, и и список лекарств, стоимость которых выше заданной пользователем. Упорядочить по названию лекарства в алфавитном порядке.
  2. В заводском цеху ведется журнал расхода материалов по следующим данным: название материала, ГОСТ, расход в сутки, количество имеющихся в цеху. Необходимо вывести документ, содержащий список материалов, которые закончатся через заданное количество дней, а также список 5-ти наименее расходуемых материалов. Списки упорядочить по названию материала в алфавитном порядке.
  3. За материально ответственным лицом числятся материальные ценности, записанные в журнале: название предмета, количество, дата приобретения (год), срок службы (в годах). Необходимо вывести документ, содержащий список предметов, подлежащих списанию в заданный год, а также список предметов, срок службы которых превышает указанный пользователем. Списки упорядочить по дате приобретения.
  4. В заводском цеху ведется учет электроэнергии, расходуемой машинами и приборами. Имеются следующие данные: название машины или прибора, инвентарный номер, потребляемая мощность, количество таких приборов в цеху. Вывести список десяти наиболее энергоемких приборов, в котором атрибутом корневого узла указать суммарную мощность, потребляемую цехом при всех включенных приборах. Список упорядочить по инвентарному номеру.
  5. На АТС ведется учет междугородних разговоров абонентов по следующим данным: фамилия абонента, домашний адрес, номер телефона, сумма междугородних телефонных разговоров за месяц (в рублях). Необходимо вывести список абонентов, тратящих на междугородние звонки больше указанной суммы. Списки упорядочить по номеру телефона.

Лабораторная работа №3 «ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ WEB-СЕРВИСОВ»

Цель работы: научиться определять функциональные границы Web-сервисов, реализовывать, развёртывать и использовать их.

Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

В соответствии с индивидуальным заданием

* 1. Определить функциональные границы веб-сервиса.
  2. Создать WSDL-описание сервиса и задокументировать его.
  3. Реализовать сервис на платформе и выполнить его развертывание в контейнере IIS.
  4. Реализовать клиентское приложение и с помощью вызовов созданного сервиса и получения результатов продемонстрировать работу методов.

Индивидуальное задание

* 1. В соответствии с индивидуальным заданием по предыдущей лабораторной работе выполнить описание бизнес-процесса в виде одной Activity-диаграммы или их набора.
  2. Определите функциональные границы двух-трех сервисов, которые могли бы быть использованы для представления первых шагов бизнес-процесса.
  3. Выделите функциональность для организации первого из сервисов таким образом, чтобы можно было создать операции: однонаправленные и запрос-ответ.
  4. Реализуйте первую службу (первый шаг процесса).
  5. Создайте клиентское приложение, с помощью которого продемонстрируйте все виды запросов, выполняя сериализацию запроса и ответа в файл. Основные требования к клиентскому приложению:
     + - приложение должно использовать разработанную веб-службу или сервис
       - приложение должно быть оснащено графическим пользовательским интерфейсом, который должен обеспечивать демонстрацию работы всех функций, а также ввод пользователем всех основных параметров и отображение результатов

Лабораторная работа №4 «ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ WEB-СЕРВИСОВ С ИСПОЛЬЗОВАНИЕМ WINDOWS COMMUNICATION FOUNDATION (WCF), NODE.JS, JAX-WS»

Цель: научиться создавать сервисы на основе WCF, Node.js, JAX-WS. Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

В соответствии с индивидуальным заданием:

* 1. Определить функциональные границы веб-сервиса.
  2. Определить контракт данных и функциональный контракт службы.
  3. Создать сервис с контрактом для двухстороннего обмена информацией сервера и клиента.
  4. Реализовать сервис с использованием WCF и протестировать его.

Индивидуальное задание

* 1. В соответствии с согласованным заданием (темой дипломного проекта или индивидуальным заданием из лабораторной работы №2) выполнить описание выполнения функций в виде Activity-диаграмм.
  2. Выбрать одну из функций для демонстрации вызова метода с получением уведомления с использованием механизмов обратной связи.
  3. Реализовать вышеуказанный сервис с использованием WCF, JAX-WS или Node.js развернуть его, как управляемое клиентское приложение с интерфейсом.
  4. Создать клиентское приложение, с помощью которого продемонстрировать работу с сервисом и обратную связь (только для WCF) с асинхронной передачей данных (для любых технологий).

Лабораторная работа №5

«ОБЪЕДИНЕНИЕ ВЕБ-СЛУЖБ И ПОЛЬЗОВАТЕЛЬСКИХ АЛГОРИТМОВ В ПОТОК

АВТОМАТИЗАЦИИ БИЗНЕС-ПРОЦЕССА»

Цель работы: научиться строить поток бизнес-процесса с использованием технологий Windows Workflow Foundation. Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

В соответствии с индивидуальным заданием

* 1. Определить поток исполнения, автоматизирующий часть бизнес-процесса, используя созданные в работах 2 и 3 веб-службы. Поток исполнения должен отвечать следующим требованиям:
     + Должны присутствовать шаги процесса, где вызываются созданные службы.
     + Должен присутствовать минимум один шаг интерактивного пользовательского ввода информации.
     + Обязательно применение закладок и демонстрация асинхронной работы допустимых действий.
     + Должен присутствовать хотя бы один узел ветвления.
  2. Продемонстрировать получение результата

Обеспечить возможность изменения потока выполнения без пересборки приложения.

Лабораторная работа №6 «ОСНОВЫ ООП В JAVA. ВВОД/ВЫВОД, СЕРИАЛИЗАЦИЯ, НАСЛЕДОВАНИЕ»

Цель работы: изучить структуру программы и объектную модель Java. Научиться корректно использовать ввод/вывод и сериализацию.

Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

Разработать программу, которая получает на вход CSV файл с информацией об объектах согласно варианту. Программа считывает эти данные и помещает их в массив. Затем массив сериализуется в бинарный файл, после чего десериализуется из этого файла в новый.

При выполнении лабораторной работы требуется воспользоваться интерфейсами и абстрактными классами. Интерфейс должен содержать типовые операции, необходимые для обработки информации по алгоритму, заданному в задании. Второй интерфейс является базовым для сущностей индивидуального задания. При наличии общих операций рекомендуется создать дополнительный уровень иерархии с применением абстрактного класса.

Требования к результирующей программе.

− Входной файл задается в качестве параметра при запуске программы.

− Необходимо проанализировать вариант и использовать наследование, отражающее иерархию классов.

− Каждый класс должен содержать 3 - 5 свойств.

− Программа должна корректно обрабатывать ошибочные данные во входном файле!

− Программа должна вести отдельный текстовый файл журнала с информацией о создаваемых объектах, ошибках во входном файле, результатах сериализации и десериализации.

Варианты заданий

1. студент, преподаватель;
2. телевизор, монитор;
3. авторучка, карандаш;
4. катер, водный мотоцикл;
5. диван, кровать;
6. самолет, корабль;
7. штатный сотрудник, совместитель;
8. эллипс, многоугольник;
9. арифметическая и геометрическая прогрессия;
10. книга, журнал; 11) зачет, экзамен;
11. город, село;
12. млекопитающее, птица;
13. детская коляска, автокресло;
14. цифровая и обычная фоторамка;
15. кондиционер, обогреватель;
16. квадрат, прямоугольник;
17. стол, стул;
18. ноутбук, планшет; 20) HDD, SDD;

21) клавиатура, мышь;

Лабораторная работа №7

«ОСНОВЫ РАБОТЫ С JAVA GRAPHICS2D API, МНОГОПОТОЧНОСТЬ»

Цель работы: получить навыки по созданию многопоточных приложений с графическим интерфейсом пользователя. Изучить возможности пакетов java.awt.geom, java.awt, классов java.awt.Graphics и java.awt.Graphics2D. Освоить основные графические примитивы и закрепить работу с потоками.

Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

Разработать приложение, отображающее не менее двух «пассивных» движущихся объектов. Каждый объект должен управляется своим потоком, что должно быть визуально заметно (разная скорость движения). Количество этих объектов может быть произвольным.

Кроме этих объектов должен быть один «активный» объект, управление которым осуществляется с помощью клавиатуры, или мыши (реализовать оба варианта). При столкновении объектов они должны менять направление движения. Вид фигуры определяется вариантом. Варианты заданий:

1. Три концентрические окружности различных цветов и радиусов, движутся в различных направлениях и меняют свои цвета. Внутренняя окружность меняет заливку.
2. Два квадрата, расположенные один внутри другого, движутся по периметру экрана и меняют цвет линий при повороте. Внутренний квадрат меняет заливку.
3. Треугольник, вписанный в квадрат, движется по диагонали экрана и изменяет свои размеры и цвет при достижении края экрана.
4. Два пересекающихся эллипса, изменяют цвета заливки, положение одного относительно другого и общее местоположение.
5. Окружность, вписанная в квадрат, движется из стороны в сторону, изменяя свои размеры и цвета при достижении края экрана.
6. Пять, касающиеся друг друга окружностей, движутся по периметру экрана и меняют свои цвета при повороте, причем у каждой окружности изменяется свой цвет заливки.
7. Окружность, вписанная в квадрат, движется по окружности с центром в середине панели и изменяет свои цвет заливки и размеры при достижении верхней точки траектории.
8. Круг, вписанный в эллипс, движется по окружности с центром в середине панели и изменяет свой цвет заливки и размеры при достижении правой крайней точки траектории.
9. Окружности движутся по разным прямым траекториям и меняют свои цвета при достижении края экрана.
10. Ромб, вписанный в прямоугольник, попеременно изменяет свой цвет, местоположение и размеры.

11.Окружность, вписанная в ромб, изменяют свои цвета и размеры. Центр фигуры постоянно находится в центре панели.

12.Набор прямоугольников разных цветов, отбрасывающих тень, движутся по экрану вправо, а при столкновении с «активным» объектом – меняют направление на противоположное.

13.Набор пересекающихся квадратов разных цветов и размеров движутся по форме вниз и на каждом следующем шаге изменяют цвета и заливку. При столкновении с «активным» объектом – меняют направление на противоположное.

1. Набор непересекающихся эллипсов разных цветов и размеров движутся по экрану вверх и на каждом следующем шаге изменяют цвета и заливку. При столкновении с «активным» объектом – меняют направление на противоположное.
2. Набор кругов разных цветов и размеров движутся по экрану вверх и на каждом следующем шаге изменяют цвета и заливку. При столкновении с «активным» объектом – меняют направление на противоположное.
3. Набор прямоугольников и треугольников различных цветов изменяют свои цвета и положение друг относительно друга.
4. В наборе кругов и прямоугольников различных цветов, круги движутся по своим траекториям, а прямоугольники стоят на месте и на каждом следующем шаге все изменяют цвета и заливку.
5. В наборе кругов и окружностей различных размеров и цветов, круги движутся и на каждом следующем шаге меняют цвета, а окружности остаются постоянными.
6. В наборе кругов и треугольников различных цветов круги движутся влево, а треугольники вправо и на каждом следующем шаге меняют цвета. При столкновении с «активным» объектом круги меняют направление на противоположное.
7. Окружности различных размеров и цветов движутся по экрану, постепенно увеличивая радиус, и изменяют цвет на каждом новом шаге.
8. Произвольная решетка из прямых линий разных цветов движется по панели по периметру, изменяя расцветку на каждом повороте.
9. Компактный набор окружностей одного цвета хаотически движется по экрану, изменяя размеры и общий цвет при достижении края экрана.
10. Компактный набор квадратов одного цвета движется по экрану по синусоидальной траектории, изменяя общий цвет при достижении края экрана.
11. Компактный набор кругов одного цвета и квадратов другого цвета произвольно движутся по экрану, изменяя размеры только окружностей и общие цвета.
12. Компактный набор концентрических окружностей различных цветов движется по экрану, изменяя постепенно размеры и изменяет цвет фона при достижении края экрана.
13. Компактный набор вписанных друг в друга квадратов различных цветов движется по экрану, изменяя размеры и общий цвет.
14. Компактный набор квадратов различных цветов движется по экрану, принимая то собственный цвет, то цвет поля, сохраняя собственным цвет периметра.
15. Набор вписанных друг в друга квадратов, имеющих различное положение диагоналей и различные цвета, движется по экрану, изменяя размеры и положения относительно друг друга.

Лабораторная работа №8

«ИНТЕРФЕЙС РАБОТЫ С БАЗАМИ ДАННЫХ В JAVA»

Цель работы: получить навыки по работе с реляционными базами данных в Java с применением стандартного API JDBC. Ознакомиться со способами управления зависимостями и сборки Javaпроектов.

Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

Разработать программу, которая получает на вход CSV файл с информацией об объектах согласно варианту (из работы №6). Программа считывает эти данные и помещает их в БД.

Реализовать 2 запроса к БД.

К программе имеются следующие требования:

− Проект должен быть собран с применением Maven.

− Входной файл задается в качестве параметра при запуске программы.

− Каждый объект должен содержать 3 - 5 свойств.

− Необходимо использовать наследование.

− Реализовать запросы на добавление нескольких записей с использованием JDBC.

− Реализовать запрос на вывод всех объектов из БД и связанных с ними сущностей.

− Реализовать подготавливаемый запрос с условием, параметры которого заполняются в зависимости от пользовательского ввода.

− Реализовать изменение одного объекта и удаление другого внутри транзакции с уровнем изоляции Dirty Read.

Для выполнения работы необходимо:

1. Создать базу данных, в соответствии с описанием предметной области в задании.
2. Настроить проект, подключив к нему соответствующий драйвер, поставляемый в виде JAR-библиотеки (рекомендуется для этого и предыдущего случаев использовать СУБД MySQL или PostgreSQL).
3. Реализовать разбор командной строки и чтение исходного файла.
4. Реализовать подключение к БД и обработку ошибок.
5. Для выполнения требований реализовать создание всех требуемых SQL Statement, их вызов и обработку.
6. Обеспечить корректное освобождение всех задействованных ресурсов.
7. Продемонстрировать запуск проекта как с применением Maven, так и без.

Варианты заданий

* 1. студент, преподаватель;
  2. телевизор, монитор;
  3. авторучка, карандаш;
  4. катер, водный мотоцикл;
  5. диван, кровать;
  6. самолет, корабль;
  7. штатный сотрудник, совместитель;
  8. эллипс, многоугольник;
  9. арифметическая и геометрическая прогрессия;
  10. книга, журнал; 11) зачет, экзамен;
  11. город, село;
  12. млекопитающее, птица;
  13. детская коляска, автокресло;
  14. цифровая и обычная фоторамка;
  15. кондиционер, обогреватель;
  16. квадрат, прямоугольник;
  17. стол, стул;
  18. ноутбук, планшет; 20) HDD, SDD;
  19. клавиатура, мышь;
  20. беговая дорожка, степпер;
  21. мобильный телефон, смартфон.

Лабораторная работа №9

«JAVA PERSISTENCE API, ORM, HIBERNATE»

"

Цель работы: получить навыки по работе с реляционными базами данных в Java. Научиться использовать отображение объектов на реляционную модель с использованием современных ORM-каркасов.

Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

По аналогии с предыдущей работой создайте программу, которая получает на вход CSV файл с информацией об объектах согласно варианту. Программа считывает эти данные и помещает их в БД. При этом вся работа с БД должна вестись посредством объектнореляционного отображения.

Требования к программе.

− Конфигурация зависимостей должна быть выполнена посредством Maven.

− Входной файл задается в качестве параметра при запуске программы.

− Каждый объект должен содержать 3 - 5 свойств.

− Отражение таблиц на объектную модель должно быть задано с применением аннотаций.

− Реализуйте DAO для доступа к соответствующим сущностям.

− Реализуйте и продемонстрируйте работу следующих запросов:

− запроса на вставку новых данных (внесите в БД данные из CSV файла);

− запроса на вывод всех объектов из БД;

− запроса на изменение одного и нескольких объектов в БД;

− запроса на выборку данных по условию (требуется задать условие, задействующее все таблицы, а сам запрос – в виде HQL). Варианты заданий

1. студент, преподаватель, группа;
2. телевизор, монитор, категория товаров;
3. авторучка, карандаш, отдел канцтоваров;
4. катер, водный мотоцикл, причал;
5. диван, кровать, комната;
6. самолет, корабль, (аэро)порт;
7. штатный сотрудник, совместитель, кафедра;
8. эллипс, многоугольник, презентация;
9. арифметическая и геометрическая прогрессия;
10. книга, журнал, полка;
11. зачет, экзамен, предмет;
12. город, село, регион;
13. млекопитающее, птица, зоопарк;
14. детская коляска, автокресло, гараж;
15. цифровая и обычная фоторамка, фото; 16) кондиционер, обогреватель;
16. квадрат, прямоугольник, рисунок;
17. стол, стул, офис;
18. ноутбук, планшет, установленная операционная система;
19. HDD, SDD, сервер;
20. клавиатура, мышь, компьютер;
21. беговая дорожка, степпер, зал;
22. мобильный телефон, смартфон, приложение.

Лабораторная работа №10

«КОНФИГУРИРОВАНИЕ WEB-ПРИЛОЖЕНИЙ НА JAVA»

Цель работы: получить навыки по созданию и конфигурированию простого Web-приложения, использующего технологию JSP.

Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

1. Установите и настройте Maven.

После скачивания с <https://maven.apache.org/>и установки Maven, чтобы начать работу со сборщиком приложения, необходимо провести настройку среды. Для этого нужно установить переменную окружения M2\_HOME. В свойствах окружения необходимо добавить переменную «M2\_HOME», значение которой – это путь к распакованному Maven.

Установите переменную окружения PATH. Для этого к путям PATH требуется добавить к списку директорий строку «%M2\_HOME%\bin». Проверьте установку переменной окружения

JAVA\_HOME, которая должна указывать путь к установленному JDK (по аналогии с предыдущими пунктами).

Проверьте правильность сделанных настроек, запустив в командной строке

*mvn --version* 2. Создайте Web-приложение.

Воспользуйтесь командной строкой и сгенерируйте приложение с помощью Maven. Для этого используйте команду: *mvn archetype:generate -DgroupId={project-packaging}*

*-DartifactId={project-name} –DarchetypeGroupId=org.apache.maven.archetypes*

*-DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false*

В качестве groupId и artifactId задайте пакет и имя приложения, которые придумайте самостоятельно.

Ознакомьтесь с содержимым сгенерированного проекта: изучите структуру каталогов и содержимое сгенерированных файлов. Создайте каталог src/main/java, после чего выполните тестовую сборку проекта: *mvn clean install*

Если всё сделано правильно, Maven скачает необходимые зависимости, выполнит очистку проекта и его сборку, а сам процесс будет завершён выводом строки «[INFO] BUILD SUCCESS», а также времени, затраченного на построение.

3. Изучите результаты построения проекта

Просмотрите содержимое директории target/ в проекте. Изучите созданные директории и файлы. Найдите Web-архив, подготовленный к развёртыванию и изучите его содержимое (файл с расширением .war).

Изучите содержимое локального репозитория Maven и найдите артефакты, созданные в результате построения проекта. Архив, предназначенный для развёртывания должен располагаться по следующему пути:

<домашняя директория>/.m2/repository/ <groupIdPath>/<artifactId>/<version>/ <artifactId><version>.war. 4. Подготовьте проект и откройте его в IDE

Проект Maven, сгенерированный данным образом, годится для разработки, автоматизированного построения и тестирования, но любую Java-разработку удобно делать в популярных IDE. В этом случае требуется подготовить проект к использованию в IDE, импортировать его и начинать разработку. В случае IDEA от JetBrains специальная подготовка не требуется и проект можно импортировать, используя дерево каталогов Maven: среда разработки «из коробки» способна корректно импортировать Maven-проекты. При использовании Eclipse требуется выполнить *mvn eclipse:clean eclipse:eclipse*

Затем импортировать проект в рабочее пространство IDE как существующий сторонний проект.

5. Разработка и наполнение проекта

Наш проект готов к тому, чтобы менять его содержимое, тестировать, запускать и развёртывать внутри контейнера сервлетов. В качестве последнего подойдут следующие популярные и вполне легковесные контейнеры: Tomcat и Jetty. Для получения возможности запустить наше приложение с помощью Maven в контейнере необходимо включить в секцию

<build> нашего проекта плагин, который позволит это сделать. Для Tomcat:

<plugin>

<groupId>org.apache.tomcat.maven</groupId>

<artifactId>tomcat7-maven-plugin</artifactId>

<version>2.2</version>

<configuration>

<port>9090</port>

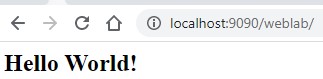
</configuration>

</plugin> После этого достаточно дать команду: *mvn tomcat7:run*

Ваше приложение будет запущено в контейнере Tomcat и будет доступно по ссылке

<http://localhost:9090/имя>

В случае успеха в браузере вы должны увидеть примерно следующее:



Аналогичным образом можно использовать для просмотра страниц сервер Jetty.

Контрольные вопросы.

1. Чем в Maven зависимости отличаются от плагинов?
2. Каковы основные этапы жизненного цикла приложения при использовании сборки Maven? 3. В чем отличие этапа clean от запуска плагина eclipse:clean?
3. Какие бывают репозитории Maven (maven repository)?
4. Для чего нужен плагин archetype:generate? Как его использовать?
5. Разобрать в настройке дополнительных параметров плагина maven Tomcat и плагина Jetty. Как применяются дополнительные параметры и на что они влияют?

Лабораторная работа №11

«СОЗДАНИЕ WEB-ПРИЛОЖЕНИЯ НА JAVA, ИСПОЛЬЗУЮЩЕГО ТЕХНОЛОГИЮ

JAVA SERVER PAGES"

Цель работы: получить навыки по разработке Web-приложения, использующего технологию Java EE – Java Server Pages, изучить жизненный цикл и способы применения сервлетов. Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

1. Создайте пакет для размещения исходного кода Java
2. Создайте и реализуйте сервлеты в соответствии с вариантом задания. Студенты, чей номер в списке нечётный, выбирают вариант 1, остальные – вариант 2.

Вариант 1. Создайте сервлет, который:

* + принимает 5 входящих параметров: имя, фамилию, отчество студента, название дисциплины и полученный балл;
  + создает новый файл по фамилии, имени и отчеству студента в любом заданном каталоге на жестком диске и записывает все переданные параметры в каждую строку в файле.

В качестве результата необходимо выдать html-страницу с указанием созданного файла и его текущим содержимым. Если в файле с именем в качестве ФИО уже существует запись о добавляемом предмете и оценка отличается от неудовлетворительной, требуется отобразить html-страницу с ошибкой.

Вариант 2. Создать сервлет, который:

* + в качестве параметра принимает имя файла с изображением (считается, что все изображения находятся в одном каталоге, имя которого заранее известно);
  + необходимо считать картинку с диска в память, после чего выдать html-страницу с ней и её именем в качестве ответа на запрос (сервлет должен выдать само изображение, а не обеспечивать прямой доступ к файлу по URL).

Если картинка с переданными именем файла не была найдена, требуется отобразить htmlстраницу с ошибкой.

Необходимо создать jsp-страницу, содержащую html-форму для отправки данных и параметров сервлетам в соответствии с заданным вариантом задания.

Контрольные вопросы.

1. Что такое сервлет?
2. Что необходимо для запуска сервлетов?
3. Перечислите этапы жизненного цикла сервлетов.
4. Что требуется для того, чтобы сервлету можно было передать параметры формы?

Лабораторная работа №12.

«СОЗДАНИЕ ПРОСТОГО WEB-ПРИЛОЖЕНИЯ, ИСПОЛЬЗУЮЩЕГО ПАТТЕРН MVC НА

ОСНОВЕ SPRING FRAMEWORK»

Цель работы: получить навыки по разработке Web-приложения с применением Spring Framework, ознакомиться с принципами работы Spring и шаблонизатора Thymeleaf. Форма отчета: демонстрация выполненной работы преподавателю.

Методические указания и задания:

1. Создайте с помощью онлайн-конфигуратора приложение Spring Boot, использующее шаблонизатор Thymeleaf и сборщик Maven.
2. Импортируйте приложение в вашу IDE и настройте JPA для соединения с базой данных (требуется любая SQL РСУБД, рекомендуется MySQL или PostgreSQL).
3. Согласно варианту задания работы №9 создайте сущности, используя аннотации JPA.
4. Создайте шаблоны страниц с использованием Thymeleaf, которые будут давать возможность выполнять следующие действия: - добавлять сущность;

* добавлять связанную сущность;
* выводить список сущностей;
* выводить список связанных сущностей (доступно по щелчку на главной сущности в списке);

1. Создайте Java-классы, представляющие формы из п.4
2. Создайте контроллер, обеспечивающий обработку форм и отображение списков. Рекомендуется создать два контроллера: один для главной сущности, второй – для связанной.

Контрольные вопросы.

1. Для чего используется Spring Framework? 2. Для чего нужны шаблонизаторы?

1. Как шаблонизатор обеспечивает интерактивную работу приложения?
2. Какие существуют альтернативные варианты работы с шаблонизатором?

ЛИТЕРАТУРА

1. Платёнкин А.В. Проектирование информационных систем. Проектный практикум: учеб. пособие / А.В. Платёнкин. – Тамбов: Изд. ФГБОУ ВПО «ТГТУ», 2015.
2. Столбовский Д.Н. Основы разработки Web-приложений на ASP.NET: учеб. пособие / Д.Н. Столбовский. – М.: Интернет- Университет Информационных Технологий (ИНТУИТ), 2016.
3. Гуськова О.И. Объектно-ориентированное программирование в Java: учеб. пособие / О.И. Гуськова. – М.: Московский педагогический государственный университет, 2018.

4.Руководство по Maven // Способ дост. URL: http://www.apache-maven.ru/