

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
Петрозаводский государственный университет
Математический факультет
Кафедра информатики и математического обеспечения

Отчет по дисциплине «Верификация ПО»

РАЗРАБОТКА СЕРВИСА ДЛЯ ОПТИМИЗАЦИИ
ПЛАНИРОВАНИЯ ПРОИЗВОДСТВА ФАНЕРЫ

Выполнил:

студент 6 курса группы 22608 М. А. Торозеров

подпись

Лектор:

к.т.н., доцент К.А. Кулаков

Итоговая оценка:

подпись

Петрозаводск

2014

Содержание

1	Выбор и согласование тематики проекта	5
2	Разработка плана реализации проекта. Анализ привлекательности плана.	5
3	Описание вариантов использования программного обеспечения.	6
4	Разработка проекта архитектуры.	6
4.1	Подсистема «Бизнес-логика»	7
4.2	Подсистема «Работа с БД»	7
4.3	Подсистема «Пользовательский интерфейс»	8
4.4	Универсальный решатель ОСОП «Фанера»	8
4.5	Схема взаимодействия методов подсистем	8
5	Разработка модели потоков данных.	9
6	План тестирования	10
6.1	Блочное тестирование	11
6.1.1	Проверка метода <code>SelectById<T>(int id)</code> с идентификатором, который есть в таблице с наименованием <code>T</code>	11
6.1.2	Проверка метода <code>SelectById<T>(int id)</code> с идентификатором, которого нет в таблице с наименованием <code>T</code>	11
6.1.3	Проверка метода <code>SelectById<T>(int id)</code> при несуществующей таблице с наименованием <code>T</code>	12
6.1.4	Проверка метода <code>SelectById<T>(int id)</code> с идентификатором, который есть в таблице с наименованием <code>T</code> , но нет доступа	12
6.1.5	Проверка метода <code>SaveOrUpdate<T>(T object)</code> с идентификатором, которого нет в таблице с наименованием <code>T</code>	12
6.1.6	Проверка метода <code>SaveOrUpdate<T>(T object)</code> с идентификатором, который есть в таблице с наименованием <code>T</code>	12
6.1.7	Проверка метода <code>SaveOrUpdate<T>(T object)</code> при несуществующей таблице с наименованием <code>T</code>	13
6.1.8	Проверка метода <code>SaveOrUpdate<T>(T object)</code> с идентификатором, который есть в таблице с наименованием <code>T</code> , но нет доступа	13

6.1.9	Проверка метода Delete<T>(T object) с идентификатором, который есть в таблице с наименованием T	13
6.1.10	Проверка метода Delete<T>(T object) с идентификатором, которого нет в таблице с наименованием T	14
6.1.11	Проверка метода Delete<T>(T object) при несуществующей таблице с наименованием T	14
6.1.12	Проверка метода Delete<T>(T object) с идентификатором, который есть в таблице с наименованием T, но нет доступа	14
6.1.13	Проверка метода TM MappingToModel<T, TM>(T object)	14
6.1.14	Проверка метода MappingToDb<T, TM>(T object, TM modelObject)	15
6.1.15	Проверка метода SetConnection(EndpointAddress endpoint)	15
6.1.16	Проверка метода Solve (Plan plan) с установленным соединением	15
6.1.17	Проверка метода Solve (Plan plan) без установленного соединения	15
6.1.18	Проверка метода GetResult(PlanResult res) через 5 минут после вы- зова метода Solve	16
6.1.19	Проверка метода GetResult(PlanResult res) без вызова метода Solve	16
6.2	Интеграционное тестирование	16
6.2.1	Проверка взаимодействия методов SelectById<T>(int id) и метода TM MappingToModel<T, TM>(T object)	16
6.2.2	Проверка взаимодействия методов SaveOrUpdate<T>(T object) (с несущ- ествующим идентификатором) и метода TM MappingToModel<T, TM>(T object)	17
6.2.3	Проверка взаимодействия методов SaveOrUpdate<T>(T object) (с су- ществующим идентификатором) и метода TM MappingToModel<T, TM>(T object)	17
6.2.4	Проверка взаимодействия методов MappingToDb<T, TM>(T object, TM modelObject) и SaveOrUpdate<T>(T object) (с несуществующим идентификатором)	17
6.2.5	Проверка взаимодействия методов MappingToDb<T, TM>(T object, TM modelObject) и SaveOrUpdate<T>(T object) (с существующим идентификатором)	18
6.2.6	Взаимодействие методов Solve(Plan plan), GetResult(PlanResult res) и SaveOrUpdate<T>(T object)	18
6.3	Подход к тестированию	18

6.4	Критерий прохождения тестов	19
6.5	Критерий приостановления работы	19
6.6	Критерий возобновления работы	19
6.7	Требуемая документация	19
6.8	Необходимое оборудование	19
6.9	Участники тестирования	19
7	Отчет о проведении тестирования	20
7.1	Пример исходного кода	23
8	Отчет об ошибке	26
9	Методы покрытия	26
10	Текущие результаты	26
11	Глоссарий терминов	31
	Библиографический список использованной литературы	32

1 Выбор и согласование тематики проекта

Целью разработки является оптимизация производства листов фанеры из имеющихся запасов шпонов, которые различаются своим качеством, толщиной и пр.

Лист фанеры, в зависимости от вида и сорта, в данной модели рассматривается как комплект листов шпона, который включает наружные и внутренние слои, причем обычно предусмотрено несколько способов комплектации фанеры одной марки. Продукция может различаться по толщине, маркам, сортам и качеству, быть обрезной или шлифованной посредством дополнительной обработки краев или поверхностей.

Содержание рассматриваемой задачи оптимизации производственного плана такого. Исходя из имеющегося портфеля заказов на производство фанеры, наличия сырья (древесных кражей и готового шпона различных видов и сортов), технологической обстановки (графика работы оборудования), необходимо рассчитать оптимальный по одному из критериев (объему выработки или ожидаемому доходу предприятия) производственный план, который содержит следующие рекомендации:

1. Перечень заказов, рекомендованных к выполнению в течение рассматриваемого периода планирования и объемы выработки соответствующей продукции, исходя из текущей производственной и коммерческой обстановки (производительности оборудования, наличия шпонов различного вида, границ возможных объемов выработки, коммерческих характеристик и срочности выполнения заказов);
2. Рекомендуемых способов комплектации продукции при наличии выбора; назначения единиц оборудования (пресса) для формирования заготовок листов фанеры каждого вида;
3. Рекомендуемого выбора различного сырья и выработки шпонов, необходимых для выполнения заказов с учетом вероятностей появления листов шпона различного качества в зависимости от исходного материала и технологии производства.

Для решения рассматриваемых задач используются различные показатели и параметры, представленные в последующих разделах данного документа.

2 Разработка плана реализации проекта. Анализ привлекательности плана.

План реализации проекта состоит из следующих пунктов:

1. Разработка архитектуры проекта
2. Разработка сервиса
3. Составление плана тестирования
4. Отчет о тестировании

3 Описание вариантов использования программного обеспечения.

ОСОП «Фанера» предназначен для оптимизации планирования производства фанеры.

4 Разработка проекта архитектуры.

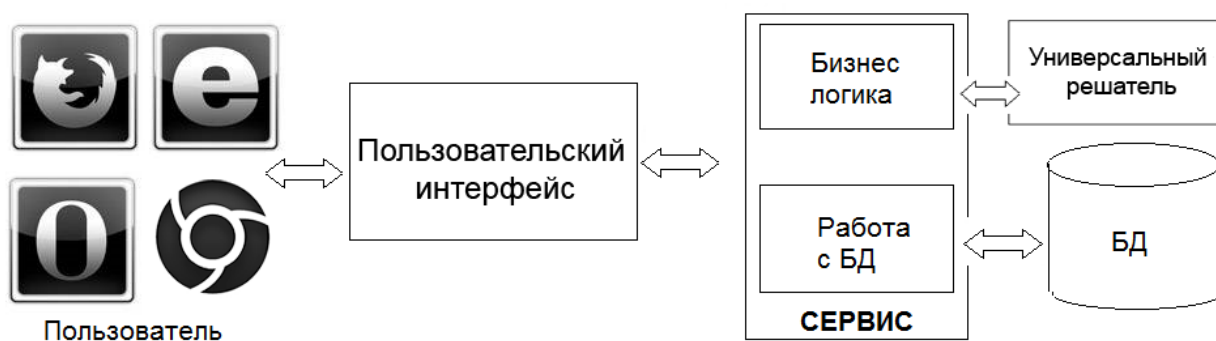


Рис. 1: Архитектура

- Подсистема «Бизнес-логика»;
- Подсистема «Работа с БД»;
- Подсистема «Пользовательский интерфейс»;
- Универсальный решатель ОСОП «Фанера».

Пользователь - это клиент, который будет взаимодействовать с нашим сервисом (необходим браузер).

Нет необходимости тестировать данное взаимодействие.

БД - база данных, которая содержит в себе данные об объектах, которые перечислены в подсистеме «Бизнес-логика».

Нет необходимости тестировать БД.

4.1 Подсистема «Бизнес-логика»

Подсистема содержит следующие *бизнес объекты*: Veneer, Plywood, VeneerThickness, Wood, WoodPeeling, Press, PressPlywood, Plan, PlanWood, PlanPlywood, PlanPress, PlanVeneerBalance, PlanWoodBalance, PlanPlywoodResult, PlanPressResult, PlanVeneerRemains, PlanWoodRemains.

Также система содержит следующие *объекты модели*: VeneerModel, PlywoodModel, VeneerThicknessModel, WoodModel, WoodPeelingModel, PressModel, PressPlywoodModel, PlanModel, PlanWoodModel, PlanPlywoodModel, PlanPressModel, PlanVeneerBalanceModel, PlanWoodBalanceModel, PlanPlywoodResultModel, PlanPressResultModel, PlanVeneerRemainsModel, PlanWoodRemainsModel.

Все свойства бизнес-объектов можно найти в соответствующих им классах, которые находятся в проекте "Business а для объектов модели в проекте "Web".

Объекты модели дублируют информацию соответствующего бизнес-объекта (имя бизнес-объекта плюс постфикс Model даёт наименование соответствующей модели) и необходимы для ограничения важной информации, которую мы будем передавать по сети.

Подсистема представляет собой библиотеку базовых классов и predefined типов для создания бизнес-логики приложения. Подсистема исполняет роль каркаса для создания бизнес-объектов, ускоряющего процесс разработки пользовательской системы. Подсистема включает в себя следующие методы:

1. TM MappingToModel<T, TM>(T object) where T, TM : class
2. MappingToDb<T, TM>(T object, TM modelObject) where T, TM : class

, которые будут тестироваться.

4.2 Подсистема «Работа с БД»

Подсистема предназначена для упрощения написания и выполнения запросов к СУБД MSSQL S2008. Данная подсистема взаимодействует с подсистемой «Бизнес-логика» на определённых соглашениях, которые не будут рассмотрены в текущем документе. Подсистема включает в себя следующие методы:

1. SelectById<T>(int id) where T : class - метод получение объекта по его идентификатору из базы, где T - наименование класса бизнес-объекта

2. `SaveOrUpdate<T>(T object)` where `T : class` - метод создания объекта или обновления его данных в базе, в зависимости от существования такого объекта в базе с текущим идентификатором, где `object` - бизнес-объект, `T` - наименование класса бизнес-объекта
3. `Delete<T>(T object)` where `T : class` - метод удаления объекта из базы, где `object` - бизнес-объект, `T` - наименование класса бизнес-объекта

, которые будут тестироваться. Данные методы реализованы в классе `Repository` и могут быть вызваны только оттуда. Каждый бизнес-объект класса `T` содержит такие поля, как: `Id`, `Description`, `IdCompany`. `IdCompany` - идентификатор компании, которой принадлежит текущий объект класса `T` и его данные.

4.3 Подсистема «Пользовательский интерфейс»

Данная подсистема предназначена для отображения экранных форм справочников, журналов и отчетов системы. Информация для отображения, как правило, передается через подсистему БЛ ОСОП «Фанера». Бизнес-объекты, входящие в состав подсистемы, наследуются от базовых бизнес-объектов, входящих в состав подсистемы «Пользовательский интерфейс». Данная подсистема не будет тестироваться

4.4 Универсальный решатель ОСОП «Фанера»

Универсальный решатель предназначен для поиска оптимального решения различных модификаций задачи планирования выпуска продукции.

1. `SetConnection(EndpointAddress endpoint)` - метод, который устанавливает соединение с сервисом по указанному адресу конечной точки с типом безопасности - `BasicHttpBinding`.
2. `Solve (Plan plan)` - метод, который принимает объект `Plan`, который содержит данные следующих объектов: `Veneer`, `Press`, `PressPlywood`, `Wood`, `WoodPeeling`;
3. `GetResult (Result res)` - метод, который необходимо вызвать после метода `Solve`, иначе ответ будет пуст. `PlanResult` содержит данные следующих объектов: `PlanResultPress`, `PlanResultPlywood`, `PlanResultPlywoodComplectation`, `PlanResultVeneerRemains`, `PlanResult`

4.5 Схема взаимодействия методов подсистем

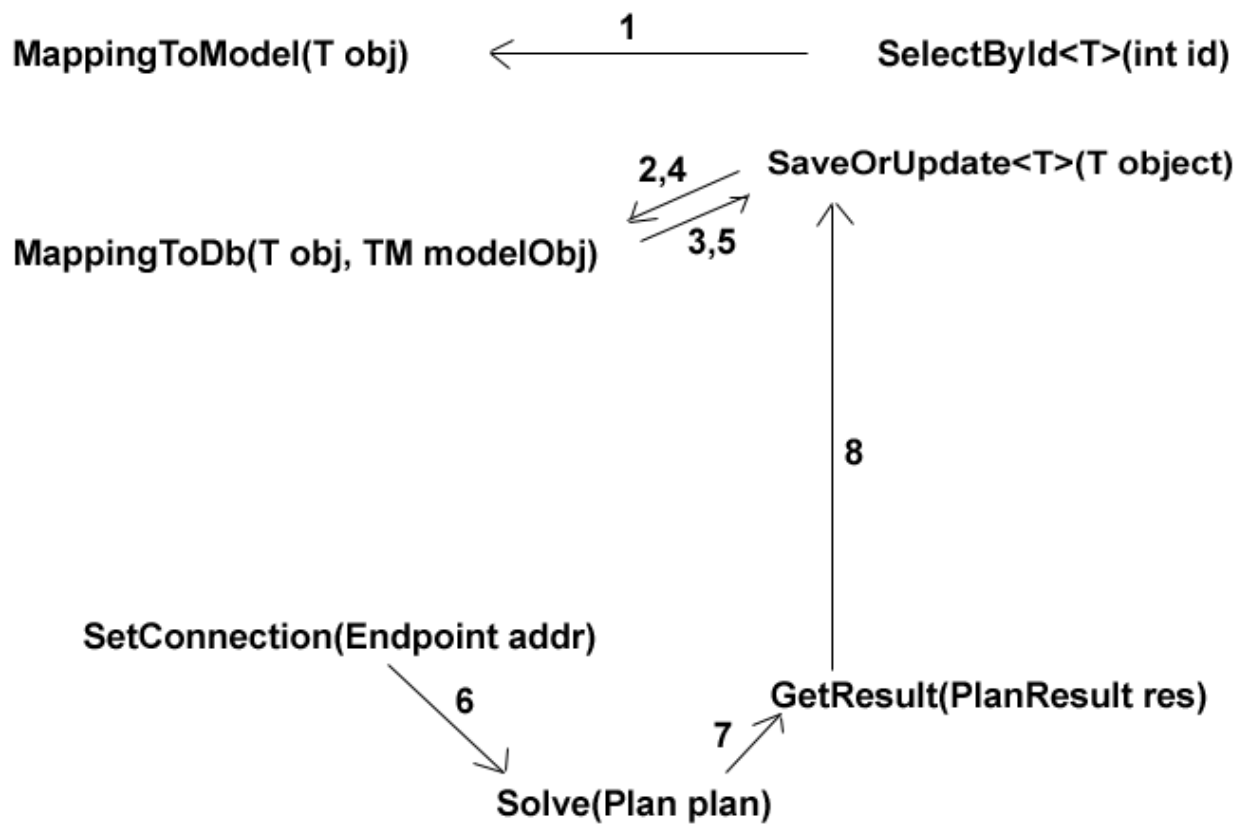


Рис. 2: Схема взаимодействия методов подсистем

5 Разработка модели потоков данных.



Рис. 3: Поток данных

6 План тестирования

Документ №2014-15-12

В проекте существует два типа объектов.

Первый - бизнес объект, второй - объект модели.

Все бизнес объекты описаны в подсистеме «Бизнес-логика». *Объект модели* - объект, данные которого дублируются из данных бизнес объекта. Находятся в проекте 'Web'. Преобразование объекта модели из объекта это полное дублирование всех данных объекта.

Отличие объекта модели от бизнес объекта в том, что он содержит лишь необходимую информацию для пользователя, тем самым повышая производительность сервиса и безопасность пользовательских данных.

Каждый объект сохранён в БД таблицу (наименование таблицы соответствует наименованию класса, экземпляром которого является данный объект).

Каждый объект имеет *идентификатор* - уникальное числовое значение.

У двух объектов из одной таблицы идентификаторы различны, однако могут совпадать для объектов из разных таблиц.

Каждый объект принадлежит к одной из компаний (компания - пользователь). Данные одной компании не доступны другой компании.

Для того, чтобы определить существующий идентификатор T объекта (где T - наименование класса бизнес-объекта), необходимо взять любую запись из таблицы с именем T (наименование таблицы совпадает с наименованием класса объекта), и взять значение поля Id.

Если нет записи с заданным Id класса T, то данный Id считается несуществующим для данного класса T.

При выполнении тестов для подсистемы «Работа с БД» необходимо использовать класс Repository, в котором реализованы методы SelectById, SaveOrUpdate, Delete. Для того, чтобы определить, принадлежит ли текущий объект класса T текущей компании, необходимо сравнить идентификатор T.IdCompany с Repository.IdCompany;

6.1 Блочное тестирование

6.1.1 Проверка метода `SelectById<T>(int id)` с идентификатором, который есть в таблице с наименованием T

Тест 1

Тип: Общий

Входные данные: Существующий идентификатор объекта, наименование класса объекта

Ожидаемый результат: Объект получен из базы

6.1.2 Проверка метода `SelectById<T>(int id)` с идентификатором, которого нет в таблице с наименованием T

Тест 2

Тип: Общий

Входные данные: Несуществующий идентификатор объекта, наименование класса объекта

Ожидаемый результат: Объект получен из базы

6.1.3 Проверка метода `SelectById<T>(int id)` при несуществующей таблице с наименованием `T`

Тест 3

Тип: Общий

Входные данные: Таблица с наименованием `T` отсутствует, любой идентификатор, наименование класса объекта

Ожидаемый результат: Пустой объект

6.1.4 Проверка метода `SelectById<T>(int id)` с идентификатором, который есть в таблице с наименованием `T`, но нет доступа

Тест 4

Тип: Общий

Входные данные: Таблица с наименованием `T`, существующий идентификатор, идентификатор другой компании, наименование класса объекта

Ожидаемый результат: Пустой объект

6.1.5 Проверка метода `SaveOrUpdate<T>(T object)` с идентификатором, которого нет в таблице с наименованием `T`

Тест 5

Тип: Общий

Входные данные: Объект с несуществующим в базе `Id`, наименование класса объекта

Ожидаемый результат: Объект сохранился в базе

6.1.6 Проверка метода `SaveOrUpdate<T>(T object)` с идентификатором, который есть в таблице с наименованием `T`

Тест 6

Тип: Общий

Входные данные: Объект с существующим в базе Id, наименование класса объекта

Ожидаемый результат: Данные объекта успешно обновлены в базе

6.1.7 Проверка метода SaveOrUpdate<T>(T object) при несуществующей таблице с наименованием T

Тест 7

Тип: Общий

Входные данные: Таблица с наименованием T отсутствует, любой идентификатор, наименование класса объекта

Ожидаемый результат: Пустой объект

6.1.8 Проверка метода SaveOrUpdate<T>(T object) с идентификатором, который есть в таблице с наименованием T, но нет доступа

Тест 8

Тип: Общий

Входные данные: Таблица с наименованием T, существующий идентификатор, идентификатор другой компании, наименование класса объекта

Ожидаемый результат: В базе не произошло никаких изменений

6.1.9 Проверка метода Delete<T>(T object) с идентификатором, который есть в таблице с наименованием T

Тест 9

Тип: Общий

Входные данные: Объект с существующим в базе Id, наименование класса объекта

Ожидаемый результат: Объект удалился из базы

6.1.10 Проверка метода Delete<T>(T object) с идентификатором, которого нет в таблице с наименованием T

Тест 10

Тип: Общий

Входные данные: Объект с несуществующим в базе Id, наименование класса объекта

Ожидаемый результат: В базе не произошло никаких изменений

6.1.11 Проверка метода Delete<T>(T object) при несуществующей таблице с наименованием T

Тест 11

Тип: Общий

Входные данные: Таблица с наименованием T отсутствует, любой идентификатор, наименование класса объекта

Ожидаемый результат: В базе не произошло никаких изменений

6.1.12 Проверка метода Delete<T>(T object) с идентификатором, который есть в таблице с наименованием T, но нет доступа

Тест 12

Тип: Общий

Входные данные: Таблица с наименованием T, существующий идентификатор, идентификатор другой компании, наименование класса объекта

Ожидаемый результат: В базе не произошло никаких изменений

6.1.13 Проверка метода TM MappingToModel<T, TM>(T object)

Тест 13

Тип: Общий

Входные данные: Объект модели класса TM и бизнес объект класса T, наименование класса T, наименование класса TM (класс TM должен состоять из наименования T и постфикса Model)

Ожидаемый результат: Объект модели класса ТМ с данными бизнес объекта класса Т

6.1.14 Проверка метода MappingToDb<Т, ТМ>(Т object, ТМ modelObject)

Тест 14

Тип: Общий

Входные данные: Объект модели класса ТМ и бизнес объект класса Т, наименование класса Т, наименование класса ТМ (класс ТМ должен состоять из наименования Т и постфикса Model)

Ожидаемый результат: Бизнес объект класса Т с данными объекта модели класса ТМ

6.1.15 Проверка метода SetConnection(EndpointAddress endpoint)

Тест 15

Тип: Общий

Входные данные: Адрес конечной точки, используя класс EndpointAddress

Ожидаемый результат: Соединение установлено

6.1.16 Проверка метода Solve (Plan plan) с установленным соединением

Тест 16

Тип: Общий

Входные данные: Соединение, установленное методом SetConnection, Данные класса Plan

Ожидаемый результат: Данные успешно отправлены, ответ от сервиса пришел положительный

6.1.17 Проверка метода Solve (Plan plan) без установленного соединения

Тест 17

Тип: Общий

Входные данные: Отсутствие соединения, Данные класса Plan

Ожидаемый результат: Ответ от сервиса отрицательный, сбой не произошёл

6.1.18 Проверка метода `GetResult(PlanResult res)` через 5 минут после вызова метода `Solve`

Тест 18

Тип: Общий

Входные данные: Соединение, установленное методом `SetConnection`, сервис с загруженными данными метода `Solve`

Ожидаемый результат: Данные успешно получены

6.1.19 Проверка метода `GetResult(PlanResult res)` без вызова метода `Solve`

Тест 19

Тип: Общий

Входные данные: Соединение, установленное методом `SetConnection`, сервис без данных

Ожидаемый результат: Ответ от сервиса положительный, результат пуст.

6.2 Интеграционное тестирование

6.2.1 Проверка взаимодействия методов `SelectById<T>(int id)` и метода `TM MappingToModel<T, TM>(T object)`

Тест 20

Тип: Интеграционный

Описание: Получить бизнес-объект из таблицы `T` по существующему идентификатору и преобразовать его в объект модели класса `TM`

Входные данные: Существующий идентификатор объекта класса `T`, наименование класса объекта, наименование класса объекта модели

Ожидаемый результат: Объект получен из базы и преобразован в объект модели

6.2.2 Проверка взаимодействия методов SaveOrUpdate<T>(T object) (с несуществующим идентификатором) и метода TM MappingToModel<T, TM>(T object)

Тест 21

Тип: Интеграционный

Описание: Сохранить бизнес-объект в таблицу T с несуществующим идентификатором и преобразовать его в объект модели класса TM

Входные данные: Несуществующий идентификатор объекта класса T, наименование класса объекта, наименование класса объекта модели

Ожидаемый результат: Объект создан в базе и преобразован в объект модели

6.2.3 Проверка взаимодействия методов SaveOrUpdate<T>(T object) (с существующим идентификатором) и метода TM MappingToModel<T, TM>(T object)

Тест 22

Тип: Интеграционный

Описание: Сохранить бизнес-объект в таблицу T по существующему идентификатору и преобразовать его в объект модели класса TM

Входные данные: Существующий идентификатор объекта класса T, наименование класса объекта, наименование класса объекта модели

Ожидаемый результат: Данные об объекте обновлены в базе и он преобразован в объект модели

6.2.4 Проверка взаимодействия методов MappingToDb<T, TM>(T object, TM modelObject) и SaveOrUpdate<T>(T object) (с несуществующим идентификатором)

Тест 23

Тип: Интеграционный

Описание: Преобразовать объект модели класса TM с несуществующим id в таблице T в бизнес объект класса T и сохранить в таблицу T

Входные данные: Несуществующий идентификатор в таблице T у объекта класса TM,

наименование класса объекта, наименование класса объекта модели

Ожидаемый результат: Объект модели преобразован в бизнес объект и создана запись в таблице T

6.2.5 Проверка взаимодействия методов `MappingToDb<T, TM>(T object, TM modelObject)` и `SaveOrUpdate<T>(T object)` (с существующим идентификатором)

Тест 24

Тип: Интеграционный

Описание: Преобразовать объект модели класса TM с существующим id в таблице T в бизнес объект класса T и сохранить в таблицу T

Входные данные: Несуществующий идентификатор в таблице T у объекта класса TM, наименование класса объекта, наименование класса объекта модели

Ожидаемый результат: Объект модели преобразован в бизнес объект и обновлены данных у записи в таблице T с текущим id

6.2.6 Взаимодействие методов `Solve(Plan plan)`, `GetResult(PlanResult res)` и `SaveOrUpdate<T>(T object)`

Тест 25

Тип: Интеграционный

Описание: Используя методы решателя, не подключаясь удалённо, получить данные из сервиса методом `GetResult(PlanResult)`, заведомо запустив метод `Solve()`, и сохранить их

Входные данные: Данные о плане (Plan), метод `Solve`, метод `GetResult`

Ожидаемый результат: Каждый бизнес-объект полученный из `PlanResult` сохранен в базу.

6.3 Подход к тестированию

Тестирование будет проводиться в автоматическом режиме, с помощью VS2010 и его инструментариев. Тестировщику необходимо запускать проект для тестирования и ожидать результат. В случае ошибки, тестировщик сам составляет отчет об ошибке, однако

таблица с результатами ему будет предоставлена.

6.4 Критерий прохождения тестов

Тест считается пройденным, если конечный результат соответствует ожидаемому результату.

Ожидаемый результат должен присутствовать в описании каждого теста в данном документе плане тестирования.

6.5 Критерий приостановления работы

Необходимо приостановить тестирование, если количество непройденных тестов превысит 10 процентов от их общего количества.

6.6 Критерий возобновления работы

После получения заявки на повторное тестирование, необходимо начать тестирование с самого начала. Тестирование с места, когда была приостановлена работа, считается недопустимым.

6.7 Требуемая документация

Необходим отчет о проведении тестирования, также необходимо указывать даты проведения тестирования.

Необходим подробный отчет об ошибке после проведения тестирования.

Необходимо указать какие средства тестирования были использованы.

6.8 Необходимое оборудование

Требования к системе: ОС Windows/ОС Linux/Mac OS X, доступ в интернет.

Требования к ПК: устройство с установленным .Net v4.0+, Visual Studio 2010.

6.9 Участники тестирования

Для тестирования необходим лишь один тестировщик, обладающий навыками работы с такими инструментами, как:

1. Visual Studio 2010
2. Unit testing in visual studio 2010

Приветствуется навык пользования Excel.

7 Отчет о проведении тестирования

Тестирование проводится с помощью инструментов VS 2010. При тестировании методов, которые должны возвращать информацию об успехе операции не будет указываться ничего, иначе - минус и после обязательно отчет об ошибке.

При тестировании операций CRUD(сохранение/удаление/редактирование/загрузку) используется по одному объекту из каждого класса (Veneer, Press and etc.).

Тестировщик: Торозёров М.А.

В таблице указан день проведения теста, а не дата, ибо все тесты проводятся в ноябре 2014 года.

Также идентификатор сокращён до id, существующий до сущ., несуществующий до несущ., объект модели до ОМ., наименование класса объекта до НК.

Самая первая колонка в таблице должна идентифицировать номер теста.

№	День	Вх. данные	Вых. данные	Кол-во тестов	Рез-т
1	20	Сущ. id. объекта, НК	Объект	21	
2	20	Несущ. id. объекта, НК	Объект	21	
3	20	Несущ. таблица T, любой id. объекта, НК	Пустой объект	21	
4	20	Сущ. таблица T, сущ. id. объекта другой компании, НК	Пустой объект	21	
5	20	Несущ. id, НК		21	
6	20	Сущ. id, НК	Объект обновлен	21	
7	20	Несущ. таблица T, любой id. объекта, НК	Пустой объект	21	
8	20	Сущ. таблица T, сущ. id. объекта, id другой компании, НК		21	
9	20	Сущ. id. объекта, НК	Объект удален	21	
10	20	Несущ. id. объекта, НК		21	
11	20	Несущ. таблица T, любой id. объекта, НК		21	

Таблица 1: Отчет о тестировании

№	День	Вх. данные	Вых. данные	Кол-во тестов	Рез-т
12	20	Сущ. таблица Т, сущ. id. объекта, id другой компании, НК		21	
13	20	Объект ТМ, объект Т, НК, ОМ	ОМ с данными НК	21	
14	20	Объект ТМ, объект Т, НК, ОМ	НК с данными ОМ	21	-
15	20	Адрес конечной точки	Соединение установлено	21	
16	20	Соединение, данные класса Plan	Данные отправлены	21	
17	20	Нет соединения, данные класса Plan	Ответ отрицательный	21	
18	20	Соединение, сервис с данными	Данные получены	21	
19	20	Соединение, сервис без данных	Пустой результат	21	

Таблица 2: Отчет о тестировании

№	День	Вх. данные	Вых. данные	Кол-во тестов	Рез-т
20	22	Сущ. id. объекта, НК, ОМ	Объект получен и преобразован	21	
21	22	Несущ. id. объекта, НК, ОМ	Объект создан и преобразован	21	
22	22	Сущ. id. объекта, НК, ОМ	Данные объекта обновлены и он преобразован	21	
23	22	Несущ. id. объекта, НК, ОМ	ОМ преобразован и создан в таблице	21	
24	22	Несущ. id. объекта, НК, ОМ	ОМ преобразован и обновлен в таблице	21	
25	22	Данные класса Plan, Solve, PlanResult	Объекты сохранены	1	

Таблица 3: Отчет о тестировании

7.1 Пример исходного кода

```
#region Test14

public void Test14<T, TM>(T obj, TM modelObj)
    where T : Entity
    where TM : EntityModel
{
    MappingToDb(obj, modelObj);

    #region TestMainProperty

    Assert.AreEqual(obj.Id, modelObj.Id);
    Assert.AreEqual(obj.Description, modelObj.Description);

    #endregion

    TestOtherProperties;
}


#region TestVeneer14

[TestMethod]
public void TestVeneer14()
{
    var veneer = new Veneer("");
    var veneerModel = new VeneerModel("");
    Test14(veneer, veneerModel);
}

#endregion

TestVeneerThickness14
```

Рис. 4: Исходный код теста 14

 **Result Summary**

Test run name: Team4D.Stels@TEAM4DSTELS 2014-12-11 10:40:57

Run result:  [24/25 tests passed, 1 failed, 0 skipped](#)

Test settings: Local

Submitted by: Team4DStels\Team4D.Stels

Started on: 11.12.2014 10:41:00

Completed on: 11.12.2014 10:50:02

Рис. 5: Результат тестирования в VS2010

8 Отчет об ошибке

№ теста: 14 Приложение было протестировано под ОС Windows 7 с использованием Visual studio 2010,.Net v4.0, Stimulsoft v2013.2.1602.0.

Основной номер версии 4, дополнительный номер версии - 3, номер сборки - 92689.

Тест успешно пройден на всех предыдущих версиях.

Алгоритм воспроизведения ошибки.

1. Создать экземпляр(бизнес-объект) класса Veneer, используя беспараметрический конструктор.
2. Создать экземпляр(объект модели) класса VeneerModel с любыми наименованием.
3. Вызвать метод MappingToDb(бизнес-объект, объект-модели);
4. Если ошибка выявлена, конец алгоритма, иначе необходимо установить ПО, описанное выше и повторить алгоритм.
5. Если ошибка не выявлена, составить отчет.

9 Методы покрытия

Расчет тестового покрытия относительно исполняемого кода программного обеспечения проводился по формуле:

$Tcov = (Ltc/Lcode) * 100$ процентов, где:

Tcov - тестовое покрытие

Ltc - кол-во строк кода, покрытых тестами

Lcode - общее кол-во строк кода.

Ltc = 1460 физических строк кода;(исключая комментарии и пробелы)

Lcode = 1747 физических строк кода; (исключая комментарии и пробелы)

Tcov = 83,57 процентов физических строк кода покрыто тестами.

10 Текущие результаты

На данный момент получены результаты:

1. Разработан сервис;

Редактирование плана

Справка

Наименование плана **Дата**

План

Показатели

Доход (у.е.)

Объем (м3)

Исходные данные

Результат

Настройки плана

Остатки шпона

+ Добавить				Действия	
Шпон	Толщина шпона (мм)	Кол-во листов		<input type="button" value="Изменить"/>	<input type="button" value="Удалить"/>
1	средина	1,5	14		
<input type="button" value="←"/> <input checked="" type="button" value="1"/> <input type="button" value="→"/> Страница <input type="text" value="1"/> из <input type="text" value="1"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="10"/> элементов на странице		Обработаны записи 1 - 1 из 1 <input type="button" value="↻"/>			

Остатки древесины

+ Добавить				Действия	
Древесина	Объем (м3)			<input type="button" value="Изменить"/>	<input type="button" value="Удалить"/>
Берёза 1 с	1 255,0				
Берёза 2 с	1 255,0			<input type="button" value="Изменить"/>	<input type="button" value="Удалить"/>
Берёза 3 с	1 255,0			<input type="button" value="Изменить"/>	<input type="button" value="Удалить"/>
<input type="button" value="←"/> <input checked="" type="button" value="1"/> <input type="button" value="→"/> Страница <input type="text" value="1"/> из <input type="text" value="1"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="10"/> элементов на странице		Обработаны записи 1 - 3 из 3 <input type="button" value="↻"/>			

Заказы по фанере

+ Добавить				Действия		
Клиент	Фанера	Цена фанеры за лист (у.е.)	Дата отгрузки	Кол-во листов фанеры по заказу	<input type="button" value="Изменить"/>	<input type="button" value="Удалить"/>
ООО "Онега холдинг"	ВВ/СР (12)	1	19.09.2013	1400		
<input type="button" value="←"/> <input checked="" type="button" value="1"/> <input type="button" value="→"/> Страница <input type="text" value="1"/> из <input type="text" value="1"/> <input type="button" value="↶"/> <input type="button" value="↷"/> <input type="button" value="10"/> элементов на странице		Обработаны записи 1 - 1 из 1 <input type="button" value="↻"/>				

Рис. 6: ОСОП «Фанера»

2. Сервис реализует все необходимые функции;

Примеры:

Сервис для оптимального планирования производства фанеры

Фанера

Главная | Справочники | План | Отчеты

Древесина

Добавить

Наименование древесины: Береза 1 с

Береза 1 с

Береза 2 с

Береза 3 с

Страница 1 из 1 | 10 элементов на странице

Изменить

Наименование древесины: Береза 1 с

Цена древесины (за м3): 11

Статистическое распределение шпона на 1 м3 древесины

Добавить

Шпон	Толщина шпона (мм)	Кол-во листов шпона	Действия
8	1,2	7,00	Изменить Оценить
Выборите шпон	1,2	10	Изменить Удалить
В	1,2	15	Изменить Удалить
ВВ	1,2	20	Изменить Удалить
СР	1,2	20	Изменить Удалить
С	1,2	30	Изменить Удалить
1 середина	1,2	20	Изменить Удалить
2 середина	1,2	20	Изменить Удалить
3 середина	1,5	20	Изменить Удалить
СР	1,5	30	Изменить Удалить

Обновить | Оценить

Обработаны записи: 1 - 3 из 3

Справка

Дорого пожаловать, user! [Выход]

Рис. 7: Справочник «Древесина и технология лущения»

Объем выработки и расходования шпона
План 1, 10.05.2013

№	Шпон		Поступление и расходование шпона (мст)			
	Марка	Толщ. (мм)	Нач. ост.	План выр.	План расх.	Кон. ост.
1	Б	1,5	4	990	0	83
2	В	1,5	0	990	0	990
3	ВВ	1,2	0	110	0	110
4	ВВ	1,5	0	650	0	650
5	СР	1,2	0	165	0	165
6	СР	1,5	25	890	0	1 005
7	С	1,2	0	30 390	0	30 390
8	С	1,5	0	330	0	330
9	1 серия	1,2	0	330	329	1
10	1 серия	1,5	0	0	0	0
11	2 серия	1,2	0	220	0	220
12	2 серия	1,5	0	6 930	0	6 930
13	3 серия	1,2	0	0	0	0
14	3 серия	1,5	0	0	0	0
Всего :			19	40 992	987	40 024

Рис. 8: Отчет «Баланс шпона»

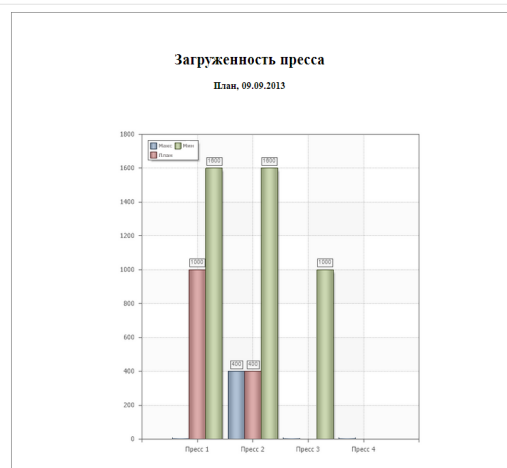


Рис. 9: Отчет «Загруженность прессы»

3. Сервис удовлетворяет всем требованиям;
4. Сервис положительно прошел тестирование.

11 Глоссарий терминов

- Локаль - это набор параметров, которые пользователь ожидает видеть в пользовательском интерфейсе.
- Конвертация данных — преобразование данных из одного формата в другой. Обычно с сохранением основного логически-структурного содержания информации.

- Веб-служба, веб-сервис (англ. web service) - идентифицируемая веб-адресом программная система со стандартизированными интерфейсами.
- URI (англ. Uniform Resource Identifier) - унифицированный (единообразный) идентификатор ресурса.
- Интерфейс (англ. interface - сопряжение, поверхность раздела, перегородка) - совокупность возможностей, способов и методов взаимодействия двух систем, устройств или программ для обмена информацией между ними, определенная их характеристиками, характеристиками соединения, сигналов обмена и т. п.
- Id - идентификатор
- ПК - персональный компьютер
- CRUD - (create/read/update/delete) создать/прочитать/обновить/удалить, основные операции работы с объектами.

Список литературы

1. Veselov A.A., Galyk L.G., Doronin Y.G. and others. *Reference book of plywood production*
Edited by PhD Kachalin N.V. М/ : Forest industry, 1984 pp. 432. (in Russian)
2. Voronin A.V., Kuznetsov V.A., Shabaev A.I., Arkhipov I.V. *Software system for sawmill operation planning*
SPb.: Proceedings of 12th conference of Open Innovations Association FRUCT, 2012. Pp. 165-171.
3. Gladkov L.A., Kureichik V.V., Kureichik V.M. . *Genetic algorithms*
Text book. - 2-nd edition - М: Physics and Mathematics Literature, 2006. - 320 p. (in Russian)
4. Kositsyn D.P., Kuznetsov V.A., Shabaev A.I. .*Optimal planning software platform development with cloud computing technology*
SPb.: Proceedings of 12th conference of Open Innovations Association FRUCT, 2012. Pp. 129-135
5. Dantzig G.B. *Linear programming and extensions*
Princeton University Press, 1963