Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Инновационного непрерывного образования

Кафедра Информатики

К защите допустить:

Заведующий кафедрой информатики

\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Волорова

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему

система нормирования материалов в конструкторской документации

БГУИР ДП 1-40 04 01 00 103 ПЗ

Студент А.В. Авхимович

Руководитель А.В. Жвакина

Консультанты:

от кафедры информатики А.С. Летохо

по экономической части Е.Е. Марченкова

Нормоконтролер В.В. Шиманский

Рецензент

Минск 2019

Реферат

Дипломный проект представлен следующим образом. Чертежи и плакаты: 6 листов формата А1. Пояснительная записка: 42 страниц, 30 рисунков, 4 таблицы, 16 литературных источников, 1 приложение.

Ключевые слова: КЛИЕНТСКАЯ ЧАСТЬ, СЕРВЕРНАЯ ЧАСТЬ, БАЗА ДАННЫХ

Целью дипломного проекта является разработка Системы нормирования материалов в конструкторской документации для отдела технологического инжиниринга машиностроительного завода, которая позволяет удобно вести базу данных материалов, график посчитаных(пронармированных) узлов изделий, генерировать отчеты в формате данных Microsoft Excel, и другие возможности.

Для достижения цели дипломного проекта было разработано клиент-серверное приложение, а также подобраны и настроены другие программные средства, необходимые для работы системы.

При разработке и внедрении приложения использовался такой стек технологий как CSS, MySQL.

В разделе введения дано определение CRM, описаны преимущества, которые дает использованиие CRM, а также описаны потребности заказчиков, которые разработанное программное средство покрывает.

В первой главе описан результат сбора и анализа требований.

Во второй главе рассматривается процесс анализа рынка программных средств и выбора подходящей основы для разработки CRM под требования заказчиков.

В третьей главе представлен процесс проектирования системы программных средств, взаимодействия по API и модели хранения данных в CRM.

В четрвертой главе рассмотрен процесс разработки программного средства: дан обзор используемым технологиям, описана работа с базой данных, приведен программный код, а также визуальный интерфейс.

В пятой главе приведено технико-экономическое обоснование, произведен расчет затрат организации-подрядчика на выполнение заказываемых работ по проекту, рассчитана прибыль и рентабельность для организации-подрядчика.

Заключение содержит выводы по данному дипломному проекту и результаты его выполнения и внедрения для всех сторон.

Дипломный проект является завершенным, поставленная задача решена в полной мере.

Министерство образования Республики Беларусь

Учреждение образования

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Факультет | ИНО | | | Кафедра | | | Информатики | | | | | | | | | |
| Специальность | 1-40 04 01 | | | Специализация | | | | | 00 | | | | | | | |
| УТВЕРЖДАЮ | | | | | | | | | | | | | | | | |
|  | | | | | | | |  | | | | | Н.А. Волорова | | | |
| « | | | | | | | |  | | » |  | | | | 2019 г. | | |
| **ЗАДАНИЕ** | | | | | | | | | | | | | | | | |
| **по дипломному проекту студента** | | | | | | | | | | | | | | | | |
| Авхимовича Алексея Валерьевича | | | | | | | | | | | | | | | | |
| (фамилия, имя, отчество) | | | | | | | | | | | | | | | | |
| 1. Тема проекта: | | Система нормирования материалов в конструкторской документации | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| утверждена приказом по университету от | | | | « | 28 | » | | апреля | | | | 2019 г. | | № | | 692-с |
| 2. Срок сдачи студентом законченной работы | | | | |  | | | | | | | | | | | |
| 3. Исходные данные к проекту | | | Тип программного средства – CRM-система; | | | | | | | | | | | | | |
| Языки программирования – Java. | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| Назначение разработки: дать возможность пользователям программного средства | | | | | | | | | | | | | | | | |
| управлять взаимоотношениями с клиентами. | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| 4. Содержание пояснительной записки (перечень подлежащих разработке вопросов) | | | | | | | | | | | | | | | | |
| Введение | | | | | | | | | | | | | | | | |
| 1.АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ | | | | | | | | | | | | | | | | |
| 2. ПОСТАНОВКА ЗАДАЧИ | | | | | | | | | | | | | | | | |
| 3. ПРОЕКТИРОВАНИЕ ПС | | | | | | | | | | | | | | | | |
| 4. РАЗРАБОТКА ПС | | | | | | | | | | | | | | | | |
| 5. ОПИСАНИЕ ИСПОЛЬЗОВАНИЯ | | | | | | | | | | | | | | | | |
| 6. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА ПО РАЗРАБОТКЕ ПС | | | | | | | | | | | | | | | | |
| ЗАКЛЮЧЕНИЕ | | | | | | | | | | | | | | | | |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | | | | | | | | | | | | | | | | |
| ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММНОГО КОДА | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. Перечень графического материала (с точным указанием наименования) и обозначения | | | |
| вида и типа материала) | |  | |
| Схема структуры программы – формат А1, лист 1 | | | |
| Диаграмма классов – формат А1, лист 1 | | | |
| Схема базы данных – формат А1, лист 1 | | | |
| Миграция связанных записей. Схема программы – формат А1, лист 1 | | | |
| Диаграмма потоков данных системы. Плакат – формат А1, лист 1 | | | |
| Визуальный интерфейс программного средства. Плакат – формат А1, лист 1 | | | |
|  | | | |
|  | | | |
| 6. Содержание задания по технико-экономическому обоснованию | | | |
| Расчет затрат на разработку, прибыли и рентабельности проекта для | | | |
| компании-подрядчика. | | | |
| Задание выдал |  | | Е.Е. Марченкова |
|  | | | |

**КАЛЕНДАРНЫЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов дипломного проекта (работы) | Объем этапа в % | Срок выполнения этапа | Приме­чание |
| Анализ предметной области, разработка технического задания, выбор подходящей основы для реализации | 20 | 28.04 – 06.05 |  |
| Проектирование архитектуры приложения | 10 | 07.05 – 11.05 |  |
| Разработка модулей | 20 | 12.05 – 20.05 |  |
| Разработка API и миграция данных | 20 | 21.05 – 29.05 |  |
| Расчет экономической эффективности | 10 | 30.05 – 04.06 |  |
| Оформление графического материала и пояснительной записки | 20 | 05.06 – 14.06 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Дата выдачи задания | 28 апреля 2019 г. | | Руководитель | |  | А.В. Жвакина |
| Задание принял к исполнению | |  | | А.В. Авхимович | | |

Содержание

[ВВЕДЕНИЕ 6](#_Toc7568447)

[**1.** **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 7](#_Toc7568448)

[**1.1** **Обзор аналогов программного продукта** 7](#_Toc7568449)

[**1.2** **Формирование требований к проектируемому ПС** 8](#_Toc7568450)

[2. Постановка задачи 10](#_Toc7568451)

[2.1 Требования к программе 10](#_Toc7568452)

[2.2 Цели и задачи проектирования 10](#_Toc7568453)

[3. Проектирование ПС 12](#_Toc7568454)

[3.1 Используемые технологии и средства разработки 12](#_Toc7568455)

[**3.1.1** **Программная платформа Java** 12](#_Toc7568456)

[**3.1.2** **NetBeans** 13](#_Toc7568457)

[**3.1.3** **JavaFX** 14](#_Toc7568458)

[**3.1.4** **MySQL** 15](#_Toc7568459)

[**3.1.5** **Библиотека Apache POI** 15](#_Toc7568460)

[3.2 Архитектура приложения 15](#_Toc7568461)

[4. Разработка ПС 18](#_Toc7568462)

[4.1 Реализация клиентского-приложения 18](#_Toc7568463)

[5. Описание использования 20](#_Toc7568464)

[5.1 Система нормирования материалов в конструкторской документации 20](#_Toc7568465)

[6. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА ПО РАЗРАБОТКЕ ПРОГРАММНОГО СРЕДСТВА 28](#_Toc7568466)

[6.1 Общая характеристика программного средства 28](#_Toc7568467)

[6.2 Расчет выручки, полученной по договору 28](#_Toc7568468)

[6.3 Расчет затрат на разработку программного средства 29](#_Toc7568469)

[6.4 Расчет экономической эффективности проекта 34](#_Toc7568470)

[ЗАКЛЮЧЕНИЕ 35](#_Toc7568471)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc7568472)

[ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММНОГО КОДА 37](#_Toc7568473)

ВВЕДЕНИЕ

На сегодняшний день трудно представить формирование сводной ведомости материалов технологами вручную используя Microsoft Excel.

В связи с этим будет разработано программное обеспечение, Система нормирования материалов в конструкторской документации призванное облегчить, систематизировать и автоматизировать формирование сводной ведомости материалов.

Для реализации поставленной цели необходимо выполнить следую-щие задачи:

Изучить базу материалов предприятия;

Собрать информацию о расходе материалов предприятия, такие как припуски на обработку, резку и коэффиценты материала на отход;

Рассмотреть и обобщить собранную информацию.

1. **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 
   1. **Обзор аналогов программного продукта**

В настоящий момент на рынке представлено приложение с похожей функциональностью. Проведем его анализ.

1.Intermech АРМ материального нормирования.

Приложение, предназначенное для расчет материальных нормативов и формирование сводных ведомостей по материалам.

Расчет материальных нормативов

АРМ материального нормирования представляет собой узкоспециализированное автоматизированное рабочее место, посредством которого осуществляется:

Назначение вариантов заготовок, вспомогательных материалов на изделие/версию изделия с учетом входимости в различные сборочные узлы или заказы.

Автоматизация расчета заготовок и вспомогательных материалов при помощи встроенной Экспертной системы.

Выпуск извещений на изменение заготовок и вспомогательных материалов с сохранением истории изменений.

Получение сводных ведомостей норм расхода материалов на изделие, заказ, по цеху и другим условиям.

Единое информационное пространство

Единое информационное пространство между технологическими и конструкторскими службами предприятия осуществляется при помощи PDM-системы Search, которая входит в состав АРМ материального нормирования.

Электронный документооборот между различными подразделениями.

Ведение списка пользователей, которые могут работать с системой (вход в систему по паролю), обеспечение безопасности путем назначения пользователям прав доступа на выполнение тех или иных действий.

Получение выборок изделий и техпроцессов по разнообразным критериям.

Система IMBase обеспечивает ведение единой конструкторско-технологической базы данных. Широкий спектр возможностей по сопровождению информационной базы позволяет адаптировать систему к условиям конкретного предприятия. Настройка прав доступа к любому элементу системы обеспечивает требуемый уровень безопасности для конкретного пользователя.

База данных по технологическому назначению содержит следующую информацию:

Применяемые основные и вспомогательные материалы.

Виды заготовок и их применяемость по сортаменту.

Методика расчета материального нормирования.

Комплекты форм выходных документов.

* 1. **Формирование требований к проектируемому ПС**

Главной целью при разработке приложения будет являться реализация простого, понятного и удобного пользовательского интерфейса, с использованием технологии JavaFX от компании Oracle.

Основной упор будет делаться на качественную поддержку отслеживания посчитанных узлов изделия, формированию понятных и доходчивых сводных ведомостей материалов. Помимо этого, должны быть реализованы следующие возможности:

* Возможность редактирования справочник Формул;
* Возможность редактирования справочник Материалов;
* Возможность редактирования справочник Обозначений;
* Возможность редактирования справочник Вспомогательных материалов;
* График учёта посчитанных узлов изделия;
* Представление узла изделия в виде спецификации
* Древовидное представление изделия
* Возможность поиска входимости узла;
* Возможность формирования отчета сводной ведомости материалов в формате данных Excel
* Возможность формирования данных в формате Excel для проверки корректности введенных данных с конструкторской документации
* Возможность формирования данных в формате Excel для раскроя материалов
* Возможность представление узла в виде спецификации
* Возможность задания параметров для расчет материала
* Возможность задания параметров для расчет вспомогательного материала

Анализ предметной области показал, что аналоговое программное

средство имеет ряд недостатков таких как, неудобный пользовательский интерфейс, нет возможности вести учет(график) пронармированных узлов изделия, отсутствует возможность замены продублированных элементов, нет возможности сохранять сводную ведомость материалов в формате данных Microsoft Excel, а также отсутствует кроссплатформенность. В ходе разработки ПС будут устранены данные недостатки.

# Постановка задачи

## Требования к программе

Необходимо реализовать клиент-серверное приложение с использованием выбранной технологии (в данном случае был выбран фреймворк JavaFX), которая позволяет обеспечить запуск необходимых задач и выполнять соответствующую работу клиент-серверного приложения по взаимодействию с пользователем через клиентскую часть приложения.

Система нормирования материалов в конструкторской документации должна:

1. Поддерживать одновременно работу 5-10 пользователей(размер отдела технологов);
2. Поддерживать операционные системы: Windows xp, Windows 7, Windows 8, Windows 10, Linux.

## Цели и задачи проектирования

**Задачи:**

1. Изучить предметную область, ознакомиться с возможностями фреймворка JavaFX при разработке приложений
2. Изучить фреймворки, используемые совместно с изучаемой технологией JavaFX
3. Создать проект Система нормирования материалов в конструкторской документации с наиболее общим функционалом.

Основной функциональностью системы нормирования материалов можно обозначить следующие возможности:

1. Должно иметь представление в виде:

* Спецификации;
* Дерева.

1. Добавление/удаление/редактирование элементов(обозначений) конструкторской документации.
2. Добавление/удаление/редактирование основных, вспомогательных материалов.
3. Добавление/удаление/редактирование формул.
4. Добавление/удаление/редактирование производственных заказов.
5. Поиск вхождение элемента(обозначения) наследника к элементу (обозначению) родителя.
6. График учета пронармированных узлов изделий.
7. Формирование отчёта пронумерованного узла изделия в формате данных excel.
8. Формирование данных узла изделия в формате данных excel для последующего раскроя материалов.
9. Формирование данных узла изделия в формате данных excel для проверки корректности вводимых данных.
10. Замена продублированных элементов.

В ходе постановки задач были определены требования к ПС такие как, поддержка одновременной работы 5-10 человек и возможность запуска клиентской части приложения на разных операционных системах: Windows XP/7/8/10, Linux, macOS**.**

# Проектирование ПС

## Используемые технологии и средства разработки

Выбор технологий является очень важным этапом разработки приложения. Выделим основные факторы, повлиявшие на выбор технологий:

* разрабатываемое ПО рассчитано на использование с платформой семейства Windows, но следует иметь в виду возможность смены платформы на такие операционные системы как Linux, macOS);
* в качестве технологий следует максимально ориентироваться на технологии с открытым исходным кодом;
* имеющийся опыт.

Основываясь на этих факторах, целесообразно выбрать для разработки платформу Java. Разрабатываемому приложению потребуется выполнять массу разнообразных задач, для реализации которых крайне желательно использование сторонних библиотек. Платформа Java обеспечена поддержкой множества фреймворков, подходящих для реализации практически любой задачи. Кроме того, сам язык Java является элегантным, мультипарадигменным языком, позволяющим с легкостью писать красивый, поддерживаемый, расширяемый код.

### **Программная платформа Java**

Программная платформа Java - ряд продуктов и спецификаций компании Sun Microsystems, ныне являющейся дочерней компанией корпорации Oracle, которые предоставляют систему для разработки программного обеспечения и развертывания его на любой платформе. Java используется в совершенно разнообразных системах, начиная от настольных компьютеров, заканчивая встраиваемыми устройствами. Язык Java отличают следующие особенности:

- Простота и мощь. Одним из ключевых факторов философии Java является простота обучения и написания кода.

- Безопасность. Программы на Java запускаются изолированно и не получают напрямую доступ к системным ресурсам.

- Объектная ориентированность. Объектная модель представлена в множестве языков. Java придерживается строгих принципов ООП, максимально стараясь не нарушать его парадигмы, но в то же время легко расширяется.

- Надежность. Java является языком со строгой типизацией, что является в современном мире как плюсом, так и минусом. Однако, благодаря строгой типизации Java способствует обнаружению ошибок на ранних этапах разработки программ. Кроме того, обеспечивается динамическое управление памятью, что является большим плюсом, так как в платформах, не обладающих такой возможностью, одной из главных из сложных проблем являются труднообнаружимые и зачастую критические ошибки, связанные с утечками памяти.

- Мультиплатформенность. Разработчики Java позволили не на словах, а на деле, написав программу один раз, запускать ее на любой платформе, где существует виртуальная машина Java.

Виртуальная машина Java - основная часть Java Runtime Environment. Виртуальная машина исполняет байт-код Java, предварительно созданный компилятором из исходного кода. JVM может также использоваться для запуска программ на других языках, не только на Java. Обычно виртуальные машины Java содержат интерпретатор байт-кода и JIT-компилятор. Схема работы системы и набор байт-кодов виртуальной машины Java таковы, что позволяют достичь высокой производительности на этапе выполнения программы:

- код анализируется на соблюдение правил безопасности до запуска кода на выполнение

- работа с базовыми типами максимально эффективна

- методы в классах могут связываться как динамически, так и статически

- автоматический сборщик мусора работает в фоновом режиме, не замедляя основной поток исполнения и обеспечивает возврат свободной памяти в систему

- стандарт Java позволяет при необходимости писать критические к производительности участки кода на языках, являющимися нативными для целевой платформы

- язык Java обладает поддержкой многопоточности на уровне языка - часть примитивов синхронизации встроена в систему реального времени.

Важным фактором является и то, что Java включает в себя набор стандартных библиотек, которые можно разбить на следующие пакеты:

- java.lang - базовый набор типов

- java.io - потоки и файлы произвольного доступа

- java.util - коллекции и некоторые другие утилиты

### **NetBeans**

NetBeans IDE — свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада[3] и ряда других.

Проект NetBeans IDE поддерживается и спонсируется компанией Oracle, однако разработка NetBeans ведётся независимым сообществом разработчиков-энтузиастов (NetBeans Community) и компанией NetBeans Org.

Последние версии NetBeans IDE поддерживают рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету и множество предопределённых шаблонов кода.

Для разработки программ в среде NetBeans и для успешной инсталляции и работы самой среды NetBeans должен быть предварительно установлен Sun JDK или J2EE SDK подходящей версии. Среда разработки NetBeans по-умолчанию поддерживала разработку для платформ J2SE и J2EE. Начиная с версии 6.0 NetBeans поддерживает разработку для мобильных платформ J2ME, C++ (только g++) и PHP без установки дополнительных компонентов.

### **JavaFX**

JavaFX представляет инструментарий для создания кроссплатформенных графических приложений на платформе Java.

JavaFX позволяет создавать приложения с богатой насыщенной графикой благодаря использованию аппаратного ускорения графики и возможностей GPU.

С помощью JavaFX можно создавать программы для различных операционных систем: Windows, MacOS, Linux и для самых различных устройств: десктопы, смартфоны, планшеты, встроенные устройства, ТВ. Приложение на JavaFX будет работать везде, где установлена исполняемая среда Java (JRE).

JavaFX предоставляет большие возможности по сравнению с рядом других подобных платформ, в частности, по сравнению со Swing. Это и большой набор элементов управления, и возможности по работе с мультимедиа, двухмерной и трехмерной графикой, декларативный способ описания интерфейса с помощью языка разметки FXML, возможность стилизации интерфейса с помощью CSS, интеграция со Swing и многое другое.

История JavaFX фактически началась в первой половине 2000-х годов, когда разработчик по имени Крис Оливер (Chris Oliver), будучи работником компании SeeBeyond, разработал для создания графических интерфейсов новый язык F3 (Froms Follows Functions). Впоследствии в 2005 году SeeBeyond была приобретена компанией Sun Microsystems (которая на тот момент развивала язык Java до покупки компанией Oracle). F3 был переименован в JavaFX, а Крис Оливер продолжил работу над новой платформой уже в рамках компании Sun. И в мае 2007 года Sun Microsystems публично анонсировала новую платформу для создания графических приложений. А 4 декабря 2008 года вышел JavaFX 1.0 SDK.

После приобретения Sun Microsystems компанией Oracle в 2010 году была анонсирована, а в 2011 году вышла в релиз версия JavaFX 2.0. В первой версии JavaFX фактически представлял скиптовый язык. Во второй версии был полностью изменен подход. Скриптовый язык был убран, а платформа была полностью переписана фактически с нуля. Теперь создавать приложения можно было с помощью любого языка, который поддерживала JVM. Были добавлены новые API, интеграция со Swing и много других вещей.

Следующими важными вехами в развитии платформы стали версии JavaFX 8 и особенно JavaFX 9, которая вышла в сентябре 2017 года вместе с Java 9 и привнесла в платформу модульность.

В сентябре 2018 года вышла последняя версия фреймворка - JavaFX 11, которая была отделена от основной функциональности Java SE и стала использоваться как отдельный модуль.

На данный момент JavaFX представляет предпочтительный способ для создания графических приложений с помощью языка Java, который пришел на смену AWT и Swing. Также стоит отметить, что для работы с JavaFX вместо Java теоретически можно использовать любой язык программирования, который поддерживается JVM.

### **MySQL**

MySQL – реляционная система управления базами данных со свободным кодом. Распространяется как свободно, так и под коммерческой лицензией. MySQL является решением для малых и средних приложений. СУБД портирована на большое количество платформ, имеет API для множества языков. MySQL была выбрана в качестве реляционной базы данных для проекта из-за того, что она максимально удовлетворяет потребностям: на БД не ожидается большой нагрузки, развернуть ее можно на большинстве популярных платформ. Для связи с БД используем библиотеку JOOQ. Это легковесная современная библиотека, написанная в функциональном стиле и идеально интегрируется с возможностями Java 8, в частности StreamAPI.

### **Библиотека** **Apache POI**

Apache POI - предоставляет чистые библиотеки Java для чтения и записи файлов в форматах Microsoft Office, таких как Word, PowerPoint и Excel.

## Архитектура приложения

Проект представляет собой клиент-серверное приложение, использующее Фреймворк JavaFX. Отдельного описания заслуживает используемый при разработке шаблон MVC.

**Model-view-controller** (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, пользовательский интерфейс и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные. Данная схема проектирования часто используется для построения архитектурного каркаса, когда переходят от теории к реализации в конкретной предметной области.



Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

* Модель (англ. Model). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
* Представление, вид (англ. View). Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами.
* Контроллер (англ. Controller). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

В ходе проектирования ПС были выбраны:

* язык программирования java обоснованный необходимостью кроссплатформенности ПС;
* среда разработки приложений NetBeans обоснованная своей доступностью;
* технология JavaFX позволяющая создавать приложения с богатой насыщенной графикой благодаря использованию аппаратного ускорения графики и возможностей GPU;
* чистая библиотека Java - Apache POI для формирования сводных ведосомтей материалов в формате данных Microsoft Excel;
* база данных MySQL обоснованная своей доступностью.

А также был выбран шаблон MVC основной целью применения концепции которого состоит в отделении бизнес логики(модели) от её визуализации (представления, вида).

# Разработка ПС

## Реализация клиентского-приложения

Перед вами дерево файлов/папок проекта:



Основные сущности реализованы в папке Models.

Сущность CommonMatreials описывает вспомогательные материаоы. Описана в файле CommonMatreials.java.

Сущность CommonMaterialsValues описывает параметры вспомогательных материалов. Описана в файле CommonMaterialsValues.java

Сущность Elements описывает элементы соответствующие обозначениям в конструкторской документации. Описана в Elements.java

Сущность Formulas описывает формулы материалов. Описана в Formulas.java

Сущность Materials описывает материалы. Описана в Materials.java

Сущность MaterialsValues описывает параметры материалов. Описана в MaterialsValues.java

Сущность Node описывает узел иерархической структуры данных. Описана в Node.java

Сущность Orders описывает заказы. Описана в Orders.java

Сущность Schedule предназначена для проставления даты в графике учёта нормированных узлов. Описана в Schedule.java

Сущность Specification описывает конструкторскую спецификацию. Описана в Specification.java

Сущность SpecificationCommonMaterials параметры вспомогательного материала. Описана в SpecificationCommonMaterials.java

Контролеры описаны в папке *Controllers*.

Контроллер *MainViewController* предоставляет взаимодействие с основными функциями приложения.

Контроллер *ParametersController* предоставляет диалоговое окно для введения параметров вспомогательного материала.

Контроллер *SplashScreenController* предоставляет загрузочный экран.

Представления находятся в папке *Views*. Для простоты отметим, что по соглашениям и для корректной работы, представления для каждого контроллера находится в соответствующей папке с тем же именем, что и имя контроллера.

В ходе разработки ПС было разработано согласно шаблону MVC: классы модели 11 шт., классы контроллеры и представления 3 шт. соответственно.

# Описание использования

## Система нормирования материалов в конструкторской документации

Главная окно программы выглядит следующим образом. С левой стороны располагаться следующие вкладки:

* Спецификация в неё вводятся все спецификации конструкторской документации.
* Дерево. Предназначена для просмотра/редактирования введённого узла в виде древовидного представления
* График. Предназначена для введения учёта пронумерованных узлов изделия.
* Заказы. Предназначена для добавления/удаления производственных заказов изделий.
* Обозначение. Предназначена для добавления/удаления/редактирования элементов(обозначений).
* Вспомогательные. Предназначена для добавления/удаления/редактирования вспомогательных материалов
* Формулы. Предназначена для добавления/удаления/редактирования формул материалов
* Материалы. Предназначена для добавления/удаления/редактирования материалов.
* Справой стороны располагаться 3 вкладки:
* Обозначение. Предназначена для редактирования элемента (обозначения) таких параметров как: Позиция, Обозначение, Кол(количество), ЕИ(единица измерения), Наименование, Примечание соответствующая конструкторской документации.
* Материал. Предназначена для присвоения материала деталям.
* Вспомогательные. Предназначена для присвоения вспомогательных материалов.



Для начало работы, следует нажать на кнопку D:\git\practise\img\home.bmp , которая перекинет в кореньевой каталог



Далее вводим обозначение по чертежу конструкторской документации ЛУИФ.К501-28.01.200. Нажимаем Enter или кнопку Добавить. В результате чего обозначение добавляется в конец списка.



Выбираем введенное обозначение и нажимаем кнопку D:\git\practise\img\right.bmp.



Далее вводим обозначения входящие в конструкторскую спецификацию



После добавления двух обозначений, проставляем их наименование и количество.





Теперь выбрав обозначение, переходим на вкладку материал, где выбираем материал соответствующий материалу на чертеже, вводим его параметры (длину, ширину) и проставляем участок на который будет далее получен материал. Проделываем аналогичное со вторым обозначением.



После завершения нормирования узла нажимаем кнопку D:\git\practise\img\left.bmp. Попадаем по иерархии на уровень выше. Выбрав пронумерованный узел ЛУИФ.К501-28.01.200 нажимаем кнопку файл->Сохранить НРМ.



Далее создается отчет в файле ЛУИФ.К501-28.01.200.xlsx.



В ходе описания использования было наглядно показан цикл нормирования одной спецификации. Начиная от ввода данных с конструкторской документации и выводом сводной ведомости материалов в формате данных MS Excel.

# ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА ПО РАЗРАБОТКЕ ПРОГРАММНОГО СРЕДСТВА

## Общая характеристика программного средства

Целью дипломного проекта является разработка Системы нормирования материалов в конструкторской документации для компания из Беларуси.

Основными пользователями продукта являются технологи им система помогает считать(нормировать) узлы конструкторских изделий.

Также продуктом пользуется главный технолог компании. Система нормирования материалов в конструкторской документации позволяет ему контролировать работу технологов по посчитаным(пронормированным) узлам конструкторского изделия.

В технико-экономическом обосновании рассмотрены расчеты по разработке проекта для компании-подрядчика, поскольку именно в этой компании и работает автор дипломного проекта.

## Расчет выручки, полученной по договору

При разработке программного средства используется Agile-подход, не требующий от заказчика подробного предоставления задания. Это позволяет клиенту менять задание по ходу разработки, анализируя предварительно результаты, полученные на предыдущей фазе. Такой подход значительно уменьшает риск того, что заказчик на выходе получит то, что ему совсем не нужно. Но для обеспечения подобной гибкости, к сожалению, плохо подходит договор с фиксированным объемом работ, фиксированной ценой и, соответственно, фиксированными сроками. Вместо этого используется подход «Time and materials», при котором заказчик оплачивает фактически потраченное специалистами время, которое предварительно выкупает на несколько месяцев вперед. Именно такой подход используется и в разработке CRM, описываемой в дипломном проекте.

При таком подходе мы оперируем несколькими понятиями:

* часовая ставка категории специалиста для заказчика;
* количество отрабатываемых часов категории специалиста в день (может быть больше 8, если работает несколько специалистов);
* длительность проекта.

Длительность проекта составляет 90 дней.

Выручку организации-подрядчика по каждой категории специалистов можно посчитать по формуле:

, (5.1)

где *Е*с – выручка организации-подрядчика по каждой категории специалистов, руб;

*R*ч – часовая ставка специалиста данной категории для заказчика, руб / час;

*T*д– количество отрабатываемых часов в день в данной категории специалистов;

*T*п– длительность проекта, час.

Итоговая выручка считается как сумма выручек по каждой категории спецалистов.

Таблица 5.1 – Расчет выручки органищации-подрядчка

|  |  |  |  |
| --- | --- | --- | --- |
| Категория специалистов | Часовая ставка для заказчика, руб. | Рабочих часов в день | Выручка по категории, руб |
| Программист | 60,89 | 16 | 87681,6 |
| Менеджер проекта | 74,64 | 4 | 26870,4 |
| Тестировщик | 62,85 | 8 | 45252 |
| Итоговая выручка | | | 159804 |

## Расчет затрат на разработку программного средства

Начнем расчет с накладных расходов организации-подрядчика, понесенных во время разработки проекта, разделенных на команду, задействованных на проекте, специалистов. Расходы всей организации, разделенные на общее количество специалистов в ней, посчитаны экономистами организации и уже известны, они составляют 90 рублей на 1 специалиста в день. Этот показатель складывается из ряда расходов, разделенных на количество людей:

* аренда офисов;
* оплата труда сотрудников, не оказывающих продаваемых клиентам услуг: системные администраторы, менеджеры по продажам, маркетологи, бухгалтеры, экономисты, отдел кадров, специалисты по подбору кадров, управляющая команда;
* оплата техники и мебели: холодильники, кофеварки, электрочайники, столы, стулья, фильтры для воды;
* оплата соцпакета: аренда бассейна, площадок для командных видов спорта;
* оплата маркетинговых активностей;
* оплата различного рода мероприятий: конференции, хакатоны, копоративные вечеринки;
* другие расходы: авиабилеты, командировки и т.д.

Количество отрабатываемых часов в день всех специалистов составляет 28 часов.

Итоговую сумму по этой статье расходов посчитаем по формуле:

, (5.2)

где *Sо* – накладные расходы организации с пересчетом на команду, руб;

*Sd*1 – накладные расходы с пересчетом на одного специалиста в день, руб;

*Qh* – количество часов специалистов, отработанных на проекте;

*T*п – длительность проекта, час.

Подставив в формулу значения, получим:

 руб. (5.3)

Теперь посчитаем комиссионные менеджера по продажам, которые выплачиваются ему за приведенный проект. Их объем известен и составляет 3% от стоимости проекта.

Несложно посчитать, что расходы по этой статье составляют:

 руб. (5.4)

Расчитаем расходы на компьютерную технику и различного рода сопутствующие устройства, которые понесла организация в ходе выполнения проекта командой специалистов. Для чего возьмем данные по всей организации за больший период и пропорционально посчитаем за меньший период длительности проекта.

Размер расходов по всей организации на компьютерную технику в пересчете на одного специалиста уже посчитан ретроспективно экономистами организации и составляет 400 рублей в год. Именно столько закладывается и в этом году на эти нужды. Сюда включены:

* покупка новых компьютеров и устройств;
* обновление старых компьютеров и устройств;
* ремонт и замена сломанных элементов.

Посчитаем расходы по этой статье:

 руб. (5.5)

Посчитаем теперь выплаты в Фонд Социальной Защиты Населения как для организации так и для сотрудника.

Компания-подрядчик является членом Парка Высоких Технологий, что позволяет ей пользоваться налоговой льготой в части уплаты взносов в ФСЗН, а именно все взносы начисляются со средней по стране заработной платы.

Средняя заработная плата в Республике Беларусь за январь составила 859 рублей.

Организация выплачивает в фонд 34%, что составляет 292,06 рубля на человека в месяц.

Подсчитаем заодно и выплаты сотрудников в ФСЗН, т.к. это понадобится для будущих расчетов. Каждый сотрудник выплачивает в фонд 1% от средней по стране зарплаты, что составляет 8,59 руб.

Расчитаем выплаты организации в фонд за все время проекта:

 руб. (5.6)

Посчитаем расходы на заработную плату, для чего возьмем данные по зарплатам соответствующих специалистов из открытых источников, собирающих статистические данные.

На портале dev.by [15] приведены медианные зарплаты специалистов, задействованных на проекте. Однако зарплаты приведены в долларах США и «чистыми», т.е. указана выплаченная заработная плата, оставшаяся после выплаты всех налогов и сборов.

Компания-подрядчик является резидентом Парка Высоких Технологий, и сотрудники пользуются налоговыми льготами в уплате подоходного налога, он составляет 9%.

Кроме этого, каждый специалист выплачивает 1% от средней заработной платы по стране в пенсионный фонд, размер которой был расчитан выше.

Расчитаем начисленную заработную плату по формуле, предварительно сконвертировав доллары США в белорусские рубли по курсу НБРБ:

, (5.7)

где *S*н – начисленная заработная плата, руб;

*S*в – выплаченная заработная плата, руб;

*T*п – размер подоходного налога, %;

*T*ф – размер выплат в фонд соцзащиты насления, руб.

Таблица 5.2 – Заработные платы специалистов в месяц

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Специалист | Выплаченная заработная плата, руб | Начисленная заработная плата, руб | Подоходный, 9% | ФСЗН |
| Менеджер проекта | 5254 | 5783,07 | 520,48 | 8,59 |
| PHP-программист | 3928 | 4325,92 | 389,33 | 8,59 |
| Тестировщик | 1768 | 1952,3 | 175,71 | 8,59 |

Посчитаем расходы по этой статье, учитывая разное количество специалистов, задействованных на проекте и общий срок проекта в 3 месяца:

 руб. (5.8)

Посчитаем расходы организации на обязательное страхование сотрудников.

По обязательному страхованию от несчастных случаев на производстве и профессиональных заболеваний в нашей стране установлен страховой тариф 0,6 процента от начисленной заработной платы.

Поскольку процент одинаковый для всех, то можно не считать отдельно по каждому специалисту, взяв процент от общей суммы начисленных зарплат, посчитанных ранее.

 руб. (5.9)

Посчитаем также и расходы на оплату непроизводительного времени. К сожалению, между проектами неизбежны вынужденные простои, в ходе которых заработная плата сотрудникам выплачивается, но доход они не приносят.

Посчитано, что такое время в среднем составляет 16 рабочих дней в год, которые не включевы в отпуск, т.е. исключительно рабочих дней. Есть также 19 рабочих дней отпуска в году. В 2018 году 253 рабочих дня.

Посчитаем количество непроизводительных дней, которые выпадут на время проекта и которые, следовательно, нужно будет оплатить из бюджета проекта.

 дн. (5.10)

Теперь посчитаем стоимость одного дня работы команды для компании по формуле:

, (5.11)

где *So* – накладные расходы организации с пересчетом на команду, руб;

*S*т – расходы на технику, руб;

*Sf* – расходы на выплаты в ФСЗН, руб;

*S*н – расходы на заработную плату сотрудников, руб;

*Si* – расходы на страхование сотрудников, руб;

*T*п – длительность проекта, дни.

 руб. (5.12)

Умножим на количество непроизводителных дней и получим затраты на эту статью расходов:

 руб. (5.13)

Далее расчитаем расходы на оплату отпуска сотрудников.

Каждому сотруднику компании положено 19 рабочих дней отпуска в год. В 2018 году 253 рабочих дня.

Посчитаем количество дней отпуска, которые нужно будет оплатить из бюджета проекта, учитывая его длительность.

 дн. (5.14)

Расчитаем затраты на эту статью расходов, используя расчитанное ранее значение затрат на 1 рабочий день команды:

 руб. (5.15)

Посчитаем также и налоги, которые организация-подрядчик оплачивает с данного проекта, учитывая, что компания является членом Парка высоких технологий.

Парк высоких технологий – единственная организация в республике, наделенная правом предоставления налоговых льгот на систематической основе. Резиденты Парка высоких технологий освобождаются от всех корпоративных налогов, включая налог на добавленную стоимость и налог на прибыль [16].

Другими словами, налог на прибыль отсутствует, а НДС (в Республике Беларусь) не возникает, поскольку местом реализации услуг считается страна резидентства заказчика (США).

Таким образом, расходы по этой статье равны нулю.

Проссумируем все расходы из описанных выше статей расходов (таблица 5.3).

Таблица 5.3 – Расходы организации

|  |  |
| --- | --- |
| Тип расхода | Размер, руб |
| Оплата труда | 40487 |
| Выплаты в ФСЗН | 3000,62 |
| Выплаты по обязательному страхованию | 242,92 |
| Оплата отпуска специалистов | 6309,08 |
| Оплата непроизводительного времени между проектами | 5319,27 |
| Комиссионные менеджера по продажам, 2% | 4794,12 |
| Расходы на технику | 345,21 |
| Накладные расходы в пересчете на команду | 28350 |
| Налоги | 0 |
| Итого расходы | 88848,22 |

Таким образом, затраты составляют 88848 рублей 22 копейки.

## Расчет экономической эффективности проекта

Экономическим эффектом у организации-подрядчика является прибыль, которая считается как разница между выручкой и расходами. Посчитаем ее:

 руб. (5.16)

Затраты по проекту уже посчитаны ранее:

 руб. (5.17)

Рентабельность считается по формуле:

. (5.18)

Посчитаем ее:

 (5.19)

Такая рентабельность является достаточно большой, что оправдано, поскольку компания является одним из лидеров в Республике Беларусь в своей отрасли с хорошо поставленными процессами продаж и привлечения персонала.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта было разработано клиент-серверное приложение, представляющее собой систему нормирования материалов, призванное облегчить, систематизировать и автоматизировать нормирование материалов изделий.

В ходе работы были изучены многие интересные технологии, при-званные облегчить и разнообразить разработку клиент-серверного при-ложения на платформе java. Были подробно изучены Фреймворк JavaFX и база данных MySQL. Так же был изучен и применён шаблон MVC.

На основе анализа, проектирования и расчетов создана Система нормирования материалов для заказчика, которая затем была внедрена в его предприятие. Мое решение позволило предприятию эффективно работать с поступающими заказами.

Разработанная Система нормирования материалов в конструкторской документации уже активно пользуется отдел технологического инжиниринга, что, в дополнение к хорошим отзывам от заказчика, является отличным доказательством востребованности разработки, а также удачности выбранных мной решений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Java 8. Полное руководство, Герберт Шилдт, 2015
2. MySQL — Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/MySQL>. – Дата доступа: 01.04.2019.
3. JavaFX | Введение [Электронный ресурс]. – Режим доступа: <https://metanit.com/java/javafx/1.1.php>. – Дата доступа: 01.04.2019.
4. NetBeans — Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/NetBeans>. – Дата доступа: 01.04.2019.

ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММНОГО КОДА

Исходный код некоторых контроллеров:

*ParametersController.java*

*package controllers;*

*import com.jfoenix.controls.JFXTextField;*

*import java.net.URL;*

*import java.util.ResourceBundle;*

*import javafx.fxml.FXML;*

*import javafx.fxml.Initializable;*

*import javafx.scene.input.KeyCode;*

*import controllers.MainController.\*;*

*import static controllers.Variables.\*;*

*import models.SpecificationCommonMaterials;*

*import java.sql.SQLException;*

*import java.util.ArrayList;*

*import javafx.application.Platform;*

*import javafx.collections.FXCollections;*

*import javafx.collections.ObservableList;*

*import javafx.scene.Node;*

*import javafx.stage.Stage;*

*public class ParametersController implements Initializable*

*{*

*@FXML*

*private JFXTextField jfxtextfieldLength;*

*@FXML*

*private JFXTextField jfxtextfieldWidth;*

*@FXML*

*private JFXTextField jfxtextfieldNomberOfLayers;*

*double a;*

*double b;*

*double c;*

*double result1;*

*double result2;*

*/\*\**

*\* Initializes the controller class.*

*\*/*

*@Override*

*public void initialize(URL url, ResourceBundle rb)*

*{*

*jfxtextfieldLength.setOnKeyPressed((event) ->*

*{*

*if (event.getCode() == KeyCode.ENTER)*

*{*

*jfxtextfieldWidth.requestFocus();*

*}*

*});*

*jfxtextfieldWidth.setOnKeyPressed((event) ->*

*{*

*if (event.getCode() == KeyCode.ENTER)*

*{*

*jfxtextfieldNomberOfLayers.requestFocus();*

*}*

*});*

*jfxtextfieldNomberOfLayers.setOnKeyPressed((event) ->*

*{*

*if (event.getCode() == KeyCode.ENTER)*

*{*

*try*

*{*

*a = Double.parseDouble(jfxtextfieldLength.getText());*

*b = Double.parseDouble(jfxtextfieldWidth.getText());*

*c = Double.parseDouble(jfxtextfieldNomberOfLayers.getText());*

*result1 = (a \* b \* 0.5 \* c) / 1000000;*

*result2 = (a \* b \* 0.1) / 1000000;*

*if (type == "Bar")*

*{*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа,вспб,вспв, НРМ)\n"*

*+ "VALUES(?,414,4," + a + "," + b + "," + c + "," + result1 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,333,4," + result1 + "," + result1 / 5 + ");", idElement);*

*}*

*else*

*{*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа,вспб,вспв, НРМ)\n"*

*+ "VALUES(?,414,4," + a + "," + b + "," + c + "," + result1 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,333,4," + result1 + "," + result1 / 5 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,78,4, " + a + "," + (a / 1000) \* 2 + ");", idElement);*

*if ((a \* b) / 1000000 > 6.25)*

*{*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,63,4," + 1 + "," + 1 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,415,4," + 0.5 + "," + 0.5 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,358,4," + 2 + "," + (0.3 \* 2) + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа,вспб,вспв, НРМ)\n"*

*+ "VALUES(?,1511421945,4," + a + "," + b + "," + 1 + "," + result2 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа,вспб,вспв, НРМ)\n"*

*+ "VALUES(?,1511421945,8," + a + "," + b + "," + 1 + "," + result2 + ");", idElement);*

*}*

*else*

*{*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,63,4," + 0.5 + "," + 0.5 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,415,4," + 0.25 + "," + 0.25 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа, НРМ)\n"*

*+ "VALUES(?,358,4," + 1 + "," + (0.3 \* 1) + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа,вспб,вспв, НРМ)\n"*

*+ "VALUES(?,1511421945,4," + a + "," + b + "," + 1 + "," + result2 + ");", idElement);*

*insertCommonElementToDb("INSERT INTO ЭлементыВспомогательные ( IDЭлемента,IDВспомогательного,IDЦехаВ, вспа,вспб,вспв, НРМ)\n"*

*+ "VALUES(?,1511421945,8," + a + "," + b + "," + 1 + "," + result2 + ");", idElement);*

*}*

*}*

*}*

*catch (java.lang.NumberFormatException exception)*

*{*

*Stage stage = (Stage) ((Node) (event.getSource())).getScene().getWindow();*

*stage.close();*

*}*

*Stage stage = (Stage) ((Node) (event.getSource())).getScene().getWindow();*

*stage.close();*

*}*

*});*

*}*

*private void insertCommonElementToDb(String sql, int id)*

*{*

*try*

*{*

*if (connection == null)*

*{*

*System.out.println("Нет соединения с БД!");*

*System.exit(0);*

*}*

*preparedStatement = connection.prepareStatement(sql);*

*preparedStatement.setInt(1, id);*

*preparedStatement.executeUpdate();*

*}*

*catch (SQLException ex)*

*{*

*ex.printStackTrace();*

*}*

*finally*

*{*

*if (preparedStatement != null)*

*{*

*try*

*{*

*preparedStatement.close();*

*}*

*catch (SQLException ex)*

*{*

*}*

*}*

*}*

*}*

*}*

Пример файла представления

SplashScreenView.fxml

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.image.\*?>

<?import java.lang.\*?>

<?import java.util.\*?>

<?import javafx.scene.\*?>

<?import javafx.scene.control.\*?>

<?import javafx.scene.layout.\*?>

<StackPane fx:id="rootPane" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" xmlns="http://javafx.com/javafx/8" xmlns:fx="http://javafx.com/fxml/1" fx:controller="controllers.SplashScreenController">

<children>

<ImageView pickOnBounds="true" preserveRatio="true" StackPane.alignment="CENTER">

<image>

<Image url="@../resources/background.png" />

</image>

</ImageView>

</children>

</StackPane>

Исходный код некоторых моделей:

Elements.java

package models;

import javafx.beans.property.SimpleIntegerProperty;

import javafx.beans.property.SimpleStringProperty;

/\*\*

\*

\* @author Unlimited

\*/

public class Elements

{

private final SimpleIntegerProperty Элементы\_IDЭлементов;

private final SimpleStringProperty Элементы\_КТДЭ;

private final SimpleStringProperty Элементы\_Обозначение;

private final SimpleStringProperty Элементы\_Наименование;

private final SimpleStringProperty Элементы\_примечание;

public Elements(int Элементы\_IDЭлементов, String Элементы\_КТДЭ, String Элементы\_Обозначение, String Элементы\_Наименование, String Элементы\_примечание)

{

this.Элементы\_IDЭлементов = new SimpleIntegerProperty(Элементы\_IDЭлементов);

this.Элементы\_КТДЭ = new SimpleStringProperty(Элементы\_КТДЭ);

this.Элементы\_Обозначение = new SimpleStringProperty(Элементы\_Обозначение);

this.Элементы\_Наименование = new SimpleStringProperty(Элементы\_Наименование);

this.Элементы\_примечание = new SimpleStringProperty(Элементы\_примечание);

}

public int getЭлементы\_IDЭлементов()

{

return Элементы\_IDЭлементов.get();

}

public void setЭлементы\_IDЭлементов(int value)

{

Элементы\_IDЭлементов.set(value);

}

public SimpleIntegerProperty Элементы\_IDЭлементовProperty()

{

return Элементы\_IDЭлементов;

}

public String getЭлементы\_КТДЭ()

{

return Элементы\_КТДЭ.get();

}

public void setЭлементы\_КТДЭ(String value)

{

Элементы\_КТДЭ.set(value);

}

public SimpleStringProperty Элементы\_КТДЭProperty()

{

return Элементы\_КТДЭ;

}

public String getЭлементы\_Обозначение()

{

return Элементы\_Обозначение.get();

}

public void setЭлементы\_Обозначение(String value)

{

Элементы\_Обозначение.set(value);

}

public SimpleStringProperty Элементы\_ОбозначениеProperty()

{

return Элементы\_Обозначение;

}

public String getЭлементы\_Наименование()

{

return Элементы\_Наименование.get();

}

public void setЭлементы\_Наименование(String value)

{

Элементы\_Наименование.set(value);

}

public SimpleStringProperty Элементы\_НаименованиеProperty()

{

return Элементы\_Наименование;

}

public String getЭлементы\_примечание()

{

return Элементы\_примечание.get();

}

public void setЭлементы\_примечание(String value)

{

Элементы\_примечание.set(value);

}

public SimpleStringProperty Элементы\_примечаниеProperty()

{

return Элементы\_примечание;

}

}

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Обозначение* | | | | | *Наименование* | | | | *Дополнительные сведения* | |
|  | | | | | **Текстовые документы** | | | |  | |
|  | | | | |  | | | |  | |
| БГУИР ДП 1–40 04 01 00 103 ПЗ | | | | | **Пояснительная записка** | | | | 42 с. | |
|  | | | | | **Отзыв руководителя** | | | |  | |
|  | | | | | **Рецензия** | | | |  | |
|  | | | | | **Справка о внедрении** | | | |  | |
|  | | | | |  | | | |  | |
|  | | | | | **Графические документы** | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.400401.103 ПД1 | | | | | **CRM система для свадебного** | | | | Формат А1 | |
|  | | | | | **салона. Схема структуры** | | | |  | |
|  | | | | | **программы.** | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.400401.103 ПД2 | | | | | **CRM система для свадебного** | | | | Формат А1 | |
|  | | | | | **салона. Диаграмма классов.** | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.400401.103 ПД3 | | | | | **CRM система для свадебного** | | | | Формат А1 | |
|  | | | | | **салона. Схема базы данных.** | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.400401.103 ПД4 | | | | | **Система** | | | | Формат А1 | |
|  | | | | | **салона. Схема программы.** | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.400401.103 ПЛ1 | | | | | **CRM система для свадебного** | | | | Формат А1 | |
|  | | | | | **салона. Диаграмма потоков** | | | |  | |
|  | | | | | **данных системы. Плакат.** | | | |  | |
|  | | | | |  | | | |  | |
| ГУИР.400401.103 ПЛ2 | | | | | **CRM система для свадебного** | | | | Формат А1 | |
|  | | | | | **салона. Визуальный интерфейс** | | | |  | |
|  | | | | | **Программного средства.** | | | |  | |
|  | | | | | **Плакат.** | | | |  | |
|  | | | | |  | | | |  | |
|  |  |  |  |  | ***БГУИР ДП 1-40 04 01 00 103 Д1*** | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| *Изм.* | *Л.* | *№ докум.* | *Подп.* | *Дата* | ***Система нормирования материалов в конструкторсокй документации.***  ***Ведомость дипломного  проекта.*** | *Лит* | | | *Лист* | *Листов* |
| ***Разраб.*** | | ***Авхимович А.В.*** |  |  |  | *Т* |  | *42* | *42* |
| ***Пров.*** | | ***Жвакина А.В.*** |  |  | ***ИиТП***  ***гр. 493551*** | | | | |
| ***Т.контр.*** | | ***Летохо А.С.*** |  |  |
| ***Н.контр.*** | | ***Шиманский В.В.*** |  |  |
| ***Утв.*** | | ***Волорова Н.А.*** |  |  |