

## **ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ**

ПО – программное обеспечение

АСУ – автоматизированная система управление

УМК – учебно-методический комплекс

ЭУМК – электронный учебно-методический комплекс

RAD – Rapid Application Development (модель быстрой разработки приложений)

4GL – Fourth Generation Language (язык программирования четвертого поколения)

CASE-средства – Computer Aided Software Engineering (компьютерная поддержка проектирования ПО)

SADT – Structured Analysis And Design Technique (методология структурного анализа и проектирования)

DFD – Data Flow Diagrams (методология структурного анализа потоков данных)

JVM – Java Virtual Machine (виртуальная машина Java)

HTML – HyperText Markup Language (язык гипертекстовой разметки)

CSS – Cascading Style Sheets (каскадные таблицы стилей)

JPA – Java Persistence API

API – application programming interface (программный интерфейс приложения)

ORM – Object-Relational Mapping (объектно-реляционное отображение)

DAO – Data Access Object (объект доступа к данным)

CRUD-операции – create, read, update, delete (создать, прочесть, обновить, удалить)

## ВВЕДЕНИЕ

В современном мире, по причине быстрого развития информационных технологий, происходит их повсеместное внедрение во все сферы человеческой жизни. Также это происходит и в образовательном процессе. Тенденция развития информационных технологий и повсеместного внедрения Интернета приводит к тому, что образование перейдет полностью в Интернет на форму удаленного обучения. В настоящее время можно найти множество различных ресурсов для удаленного получения образования. Множество сертификатов мирового уровня можно пройдя тестирование находясь в любой точке земного шара.

В связи с этой тенденцией в данном дипломном проекте разрабатывается платформа для обучения программированию на Java. Целью проекта является повышение доступности и качества образовательной деятельности учебного заведения.

Перечень средств разработки:

- Средства проектирования приложения: IBM Rational Rose Enterprise Edition, Microsoft Visio 2010;
- Язык программирования: Java8;
- Фреймворки: Spring Framework, Spring Boot
- Средства создания пользовательского интерфейса: HTML, CSS;
- Средства получения исполняемого кода: IntelliJ IDEA 2017.3.5 x64;
- Средства отладчики исполняемого кода: IntelliJ IDEA 2017.3.5 x64;
- Средства тестирования программ: IntelliJ IDEA 2017.3.5 x64;
- Средства управления базами данных: MySQL;
- Средства запуска готового проекта: JVM.

# **1 ЦЕЛЬ И НАЗНАЧЕНИЕ ПРОЕКТА. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1 Цели разработки**

Платформа предназначена для повышения доступности и качества образовательной деятельности в рамках учебного заведения, а также призвана облегчить работу преподавательской деятельности.

## **1.2 Основание для разработки**

Разработка системы осуществляется на основании темы к дипломному проекту в соответствии с документацией деканата факультета «Электросвязи» Белорусской государственной академии связи и регламентируется Уставом Белорусской государственной академии связи.

## **1.3 Назначение**

Назначение разрабатываемого программного продукта заключается в передаче студентам теоретических знаний концепции объектно-ориентированного программирования, практических навыков по разработке объектно-ориентированных программ на одном из популярных языков программирования – Java.

## **1.4 Описание предметной области**

Опишем понятия, которые входят в предметную область выбранной темы.

**Программное обеспечение (ПО)** представляет собой программу, либо совокупность программ, предназначенных для управления компьютером.

Программное обеспечение в зависимости от назначения разделяется на: системное ПО, прикладное ПО и инструментальное ПО.

Системное программное обеспечения включает в себя программы, обеспечивающие функционирование компьютера и компьютерных сетей. Данное ПО позволяет пользователю осуществлять управление компьютером и компьютерной сетью, а также обеспечивает работоспособность других программ. В качестве примеров

системного ПО можно привести операционные системы, архиваторы, файловые менеджеры, утилиты и другие.

Программы, предназначенные для решения задач определенной предметной области, относятся к прикладному программному обеспечению.

Прикладное ПО принято разделять на:

- программы общего назначения (необходимы для работы любого пользователя);
- программы специального назначения (требуются для использования профессионалами в различных сфера деятельности);
- компьютерные игры.

Примерами прикладных программ являются текстовые редакторы, графические редакторы и другие.

Инструментальное программное обеспечение предназначения для проектирования, разработки и сопровождения программ.

Различают такие виды инструментального ПО как: компиляторы, интерпретаторы, отладчики, SDK, интегрированные среды разработки и другие.

**Автоматизированные системы управления (АСУ)** – это комплекс программных и аппаратных средств, предназначенный для управления различными процессами в рамках технологического процесса, предприятия, производства. АСУ применяются в различных отраслях промышленности, энергетике, транспорте и т. п.

Наиболее важной задачей автоматизированных систем управления является повышение эффективности управления объектом на основе роста производительности труда и совершенствования методов планирования процесса управления.

Различают автоматизированные системы управления объектами (технологическими процессами – АСУТП, предприятием – АСУП, отраслью – ОАСУ) и функциональные автоматизированные системы, например, проектирование плановых расчётов, материально-технического снабжения и т.д.

**Java** является универсальным языком программирования, который основан на концепции объектно-ориентированного программирования. Он предназначен для того, чтобы позволить разработчикам приложений «писать один раз, выполнять где угодно» (WORA), что означает, что скомпилированный код Java может работать на всех платформах, поддерживающих Java, без необходимости перекомпиляции.

**Учебная программа** (Учебный план) учреждения образования – созданный в рамках системы обучения документ, определяющий содержание и количество знаний, умений и навыков, предназначенных к обязательному усвоению по той или иной учебной дисциплине, распределение их по темам, разделам и периодам обучения.

**Учебно-методический комплекс (УМК)** дисциплины – стандартное название для совокупности учебно-методической документации, средств обучения и контроля, разрабатываемых в учреждениях образования для каждой дисциплины. УМК должен включать полную информацию, достаточную для прохождения дисциплины. УМК предназначены для обеспечения открытости образовательного процесса и должны быть доступны любому желающему.

**Электронный учебно-методический комплекс (ЭУМК)** дисциплины – продукт, имеющий электронный формат, который используется в учебном процессе учебного заведения в рамках дисциплины.

**Контроль знаний** – оценка знаний учащихся по истечении конкретного периода учебного года (четверти, полугодия), позволяющая определить качество и объем усвоенных знаний.

**Журнал успеваемости** – документ, позволяющий участникам учебного процесса вести учет о текущих и итоговых оценках.

В качестве предметной области выбрано учреждение образования «Белорусская государственная академия связи», которое занимается обучением будущих специалистов в области телекоммуникаций, а точнее кафедра программного обеспечения сетей телекоммуникаций (ПОСТ) [1].

Кафедра является профилирующей и осуществляет подготовку специалистов на уровне высшего образования по направлению специальности 1-45 01 02-03 «Инфокоммуникационные системы (сопровождение программного обеспечения)», (квалификация специалиста – инженер по инфокоммуникационным системам) [1].

Кафедра осуществляет подготовку специалистов на уровне среднего специального образования по специальности 2-45 01 33 «Сети телекоммуникаций», специализации 2-45 01 33 02 «Программное обеспечение сетей телекоммуникаций» (квалификация специалиста – техник по телекоммуникациям) [1].

На уровне высшего образования преподается дисциплина «Конструирование программ и языки программирования», цель которой является приобретение студентами теоретических знаний концепции объектно-ориентированного программирования, практических навыков по разработке объектно-ориентированных программ, а также умений объектно-ориентированных программ [2].

В рамках дисциплины предусматривается изучение языка Java SE, аспекты применения библиотек классов языка Java, включая файлы, коллекции, сетевые и многопоточные приложения, а также взаимодействие с XML [2].

## **2 ВЫБОР ЖИЗНЕННОГО ЦИКЛА ПРОЕКТА, АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, ВЫЯВЛЕНИЕ ТРЕБОВАНИЙ К СИСТЕМЕ**

### **2.1 Выбор модели жизненного цикла проекта**

Существуют различные схемы классификации проектов, связанных с разработкой программных средств и систем.

В ГОСТ Р ИСО/МЭК ТО 12182–2002 – Информационная технология. Классификация программных средств [3] – приведена схема классификации ПС по 16 видам, каждый из которых подразделяется на классы. Данная классификация имеет общий характер и в целом не может использоваться для обоснования выбора модели ЖЦ ПС и систем [4].

Институтом качества программного обеспечения SQI (Software Quality Institute, США) специально для выбора модели ЖЦ предложена схема классификации проектов по разработке ПС и систем [5]. Основу данной классификации составляют четыре категории критериев. По каждому критерию проекты подразделяются на два альтернативных класса [4].

Критерии классификации проектов, предложенные Институтом SQI, объединены в следующие категории.

#### *1. Характеристики требований к проекту*

Критерии данной категории классифицируют проекты в зависимости от свойств требований пользователя (заказчика) к разрабатываемой системе или программному средству. Например, известны ли требования к началу проекта, сложны ли они, будут ли они изменяться.

#### *2. Характеристики команды разработчиков*

Чтобы иметь возможность пользоваться критериями данной категории классификации проектов, состав команды разработчиков необходимо сформировать до выбора модели ЖЦ. Характеристики команды разработчиков играют важную роль при выборе модели ЖЦ, поскольку разработчики несут основную ответственность за успешную реализацию проекта. В первую очередь следует учитывать квалификацию разработчиков, их знакомство с предметной областью, инструментальными средствами разработки и т.п.

### 3. Характеристики пользователей (заказчиков)

Чтобы иметь возможность пользоваться критериями данной категории классификации проектов, до выбора модели ЖЦ необходимо определить возможную степень участия пользователей (заказчиков) в процессе разработки и их взаимосвязь с командой разработчиков на протяжении проекта. Это важно, поскольку отдельные модели ЖЦ требуют усиленного участия пользователей в процессе разработки. Выбор таких моделей при отсутствии реальной возможности пользователей участвовать в проекте приведет к существенному снижению качества результатов разработки.

### 4. Характеристики типов проектов и рисков

В некоторых моделях в достаточно высокой степени предусмотрено управление рисками. В других моделях управление рисками вообще не предусматривается. Поэтому при выборе модели ЖЦ следует учитывать реальные риски проекта, критичность и сложность продуктов разработки.

Для выбора подходящей к условиям конкретного проекта модели ЖЦ ПС и систем Институтом качества программного обеспечения SQI рекомендуется использовать специальную процедуру [5]. Данная процедура базируется на применении четырех таблиц вопросов. Примеры вопросов приведены в таблицах А.1-А.4 приложения А.

Для моего проекта наиболее важными вопросами из таблиц А.1-А.4 являются: 1, 2, 3, 6, 7 – вопросы из таблицы А.1; все вопросы из таблицы А.2; 3, 4 – вопросы из таблицы А.3; 3, 5, 6, 7, 8, 10 – вопросы из таблицы А.4.

Таблица 2.1 – Итоги выбора жизненного цикла

Количество отмеченных вопросов	Каскадная	Вобразная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
Важные	14	13	14	8	7	5
Не важные	2	2	5	1	6	4
Итого	16	15	19	9	13	9

Исходя из итогов таблицы 2.1 лидирует модель жизненного цикла RAD по количеству важных вопросов и суммарному количеству вопросов. Так как к процессу разработки нет ограничений, выбираем **модель жизненного цикла RAD**.

## 2.2 Моделирование и анализ предметной области

В качестве модели жизненного цикла я выбрал модель RAD. Как правило, RAD-модели используются в составе другой модели для ускорения цикла разработки прототипа

(версии) системы или программного средства. Но так как проект не является сложным, буду применять RAD-модель как независимую модель разработки.

Основу RAD-модели составляет использование мощных инструментальных средств разработки. Такими средствами являются языки четвертого поколения 4GL (Fourth Generation Language – язык программирования четвертого поколения) и CASE-средства (Computer Aided Software Engineering – компьютерная поддержка проектирования ПО), благодаря наличию в них сред визуальной разработки и кодогенераторов.

Использование инструментальных средств позволяет задействовать пользователя, а следовательно, дать оценку продукту на всех этапах его разработки.

Характерной чертой RAD-модели является короткое время перехода от анализа требований до создания полной системы или программного средства. Разработка прототипа, как правило, ограничивается четко определенным периодом времени.

Существует 3 основные RAD-модели:

- *Базовая RAD-модель* применяется совместно с другой моделью разработки;
- *RAD-модель, основанная на моделировании предметной области* – используются различные виды моделирования предметной области, например, функциональное моделирование, моделирование данных, моделирование процесса (поведения). С этой целью широко используются CASE-средства;
- *RAD-модель параллельной разработки приложений* – данный вариант модели поддерживает параллельную разработку программных компонентов, реализующих базовые функции программного средства различными группами разработчиков, с дальнейшей интеграцией разработанных компонентов в единую систему или программное средство.

Исходя из описания, базовая RAD-модель и RAD-модель параллельной разработки приложений не подходят, так как базовая модель учитывает, что она будет использоваться с другой моделью разработки, чего я не предусматриваю. RAD-модель параллельной разработки приложений не подходит, так как в ней должны участвовать группа разработчиков, что в данном случае невозможно. Следовательно, я выбрал **RAD-модель, основанную на моделировании предметной области.**

В данной модели выделяется пять этапов (рисунок 2.1).



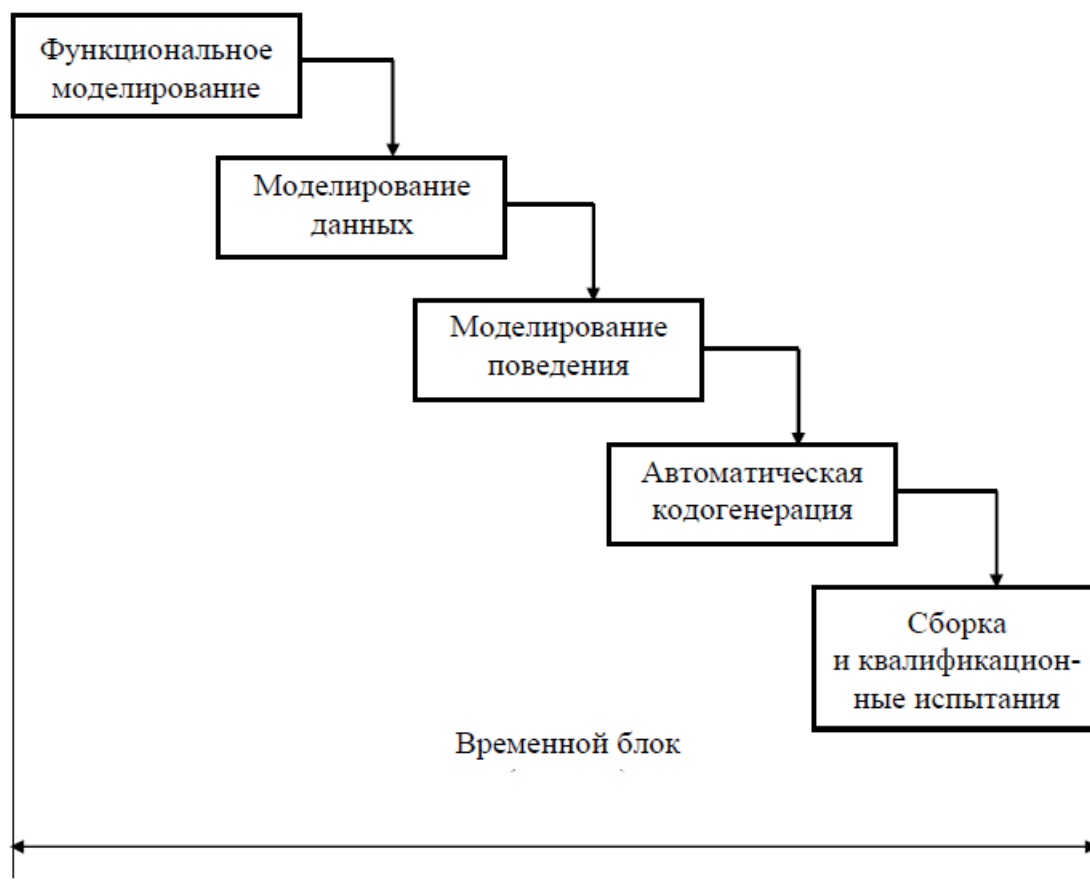


Рисунок 2.1 – Вариант модели быстрой разработки приложений, основанной на моделировании предметной области

*На этапе функционального моделирования* определяются и анализируются функции и информационные потоки предметной области, в которой используются методологии функционального моделирования IDEF0 и DFD.

*На этапе моделирования данных на базе информационных потоков*, определенных на предыдущем этапе, разрабатывается информационная модель предметной области в которой используется методология информационного моделирования IDEF1X.

*На этапе моделирования поведения* выполняется динамическое (поведенческое) моделирование предметной области. Одной из известных методологий динамического моделирования является методология IDEF3.

*На этапе автоматической кодогенерации* на основе информационной, функциональной и поведенческой моделей выполняется генерация текстов программных компонентов. При этом используются языки программирования четвертого поколения (Fourth Generation Language – 4GL) и CASE-средства. Широко применяются повторно используемые программные компоненты.

На этапе сборки и квалификационных испытаний выполняется сборка и испытания результирующей системы, подсистемы или программного средства.

### 2.2.1 Этап функционального моделирования.

Основой для методологии функционального моделирования IDEF0 является методология структурного анализа и проектирования SADT. Основным назначением методологии SADT является моделирование предметной области с целью определения требований к разрабатываемой системе или программному средству и с целью их проектирования.

В моделировании SADT определены два направления: *функциональные модели* (IDEF0) выделяют события в системе, *модели данных* (DFD) выделяют объекты (данные) системы, связывающие функции между собой и с их окружением.

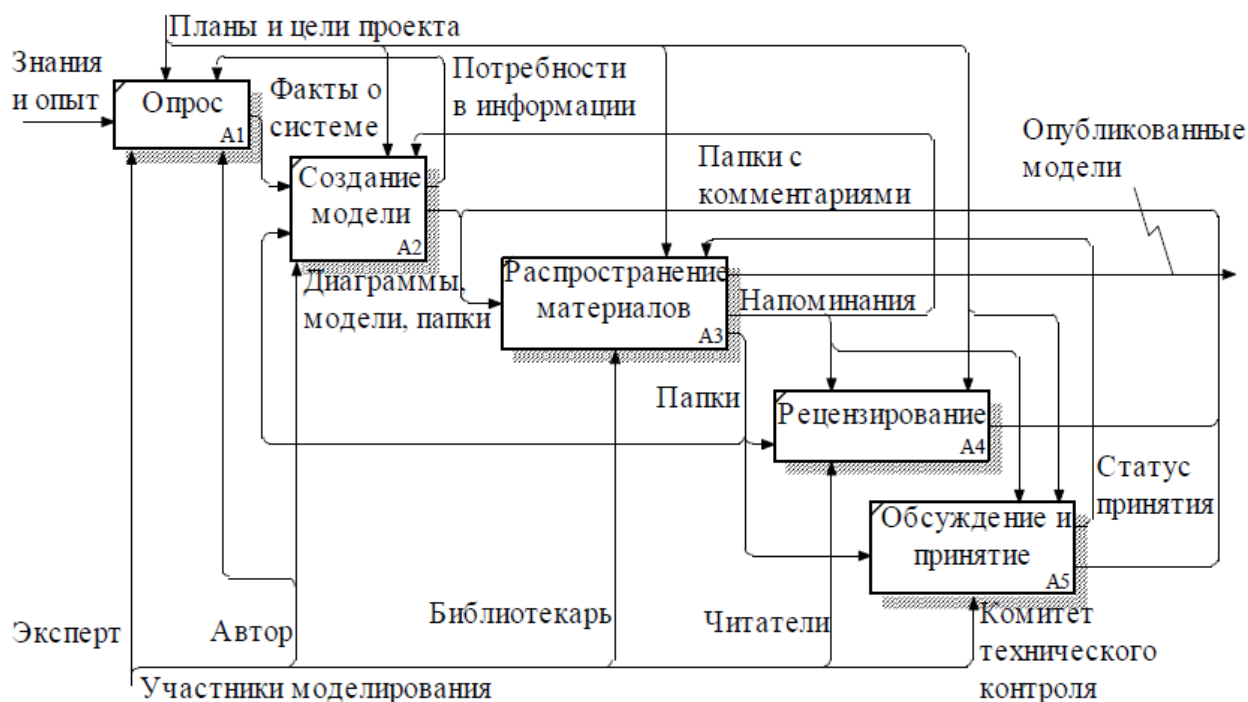


Рисунок 2.2 – Процесс моделирования в IDEF0

На первом этапе разработки IDEF0-модели формулируются вопросы к ней, формируется цель модели, определяются претенденты на точку зрения, выбирается точка зрения.

Перечень вопросов:

- Какие этапы обучения необходимо пройти студенту?
- Какие сотрудники учувствуют в процессе обучения и аттестации студентов?

– Какие виды работ должен осуществлять преподаватель во время обучения студентов и их аттестации?

– Какая информация должна быть входной для обучения студентов?

– Как влияют результаты отдельных этапов на итоги обучения студента?

– Что необходимо для завершения обучения студентом?

*Цель модели:* определить основные этапы прохождения обучения студентами, влияние этапов друг на друга и на результаты завершения обучения с целью обучения студентов концепциям объектно-ориентированного программирования на Java.

*Претенденты на точку зрения:* преподаватель, студент.

С учетом цели модели предпочтение следует отдать точке зрения преподавателя, так как она наиболее полно охватывает все этапы обучения и только с этой точки зрения можно показать взаимосвязи между отдельными этапами и обязанности участников обучения.

**Вторым этапом** моделирования является создание модели. На данном этапе полученные на предыдущем этапе факты о системе автор представляет в виде одной или нескольких IDEF0-диаграмм. Процесс создания модели осуществляется с помощью метода декомпозиции ограниченного субъекта. При его использовании автор модели вначале анализирует объекты (информацию, данные, механизмы и т.п.), входящие в систему, а затем использует полученные знания для анализа функций системы (см. п. 5.2.5). На основе этого анализа создается диаграмма, в которой объединяются сходные объекты и функции.

IDEF0 диаграммы представлены в приложении Б.

Третий, четвертый и пятый этапы моделирования можно пропустить, так как эти этапы не являются информативными и не несут важных сведений. Целью моделирования было получение полного представления о процессах обучения в учебном заведении, что было достигнуто.

Методология структурного анализа потоков данных DFD (Data Flow Diagrams) основана на методах, ориентированных на потоки данных. Существуют различные графические нотации данной методологии. В данном проекте используется методология DFD в нотации Гейна–Сарсона.

DFD-методология выделяет функции (действия, события, работы) системы. Функции соединяются между собой с помощью потоков данных (объектов).

Аналогично IDEF0-методологии DFD-модель должна иметь единственную цель, точку зрения, субъект и точно определенные границы. Однако если в IDEF0 дуги имеют

различные типы и определяют отношения между блоками, то в DFD дуги отражают реальное перемещение объектов от одной функции к другой.

*Цель модели:* определить основные этапы прохождения обучения студентами, влияние этапов друг на друга и на результаты завершения обучения с целью обучения студентов концепциям объектно-ориентированного программирования на Java.

*Претенденты на точку зрения:* преподаватель, студент.

С учетом цели модели предпочтение следует отдать точке зрения преподавателя, так как она наиболее полно охватывает все этапы обучения и только с этой точки зрения можно показать взаимосвязи между отдельными этапами и обязанности участников обучения.

IDEF0 диаграммы представлены на рисунках 2.3-2.6.



Рисунок 2.3 – Концептуальная схема

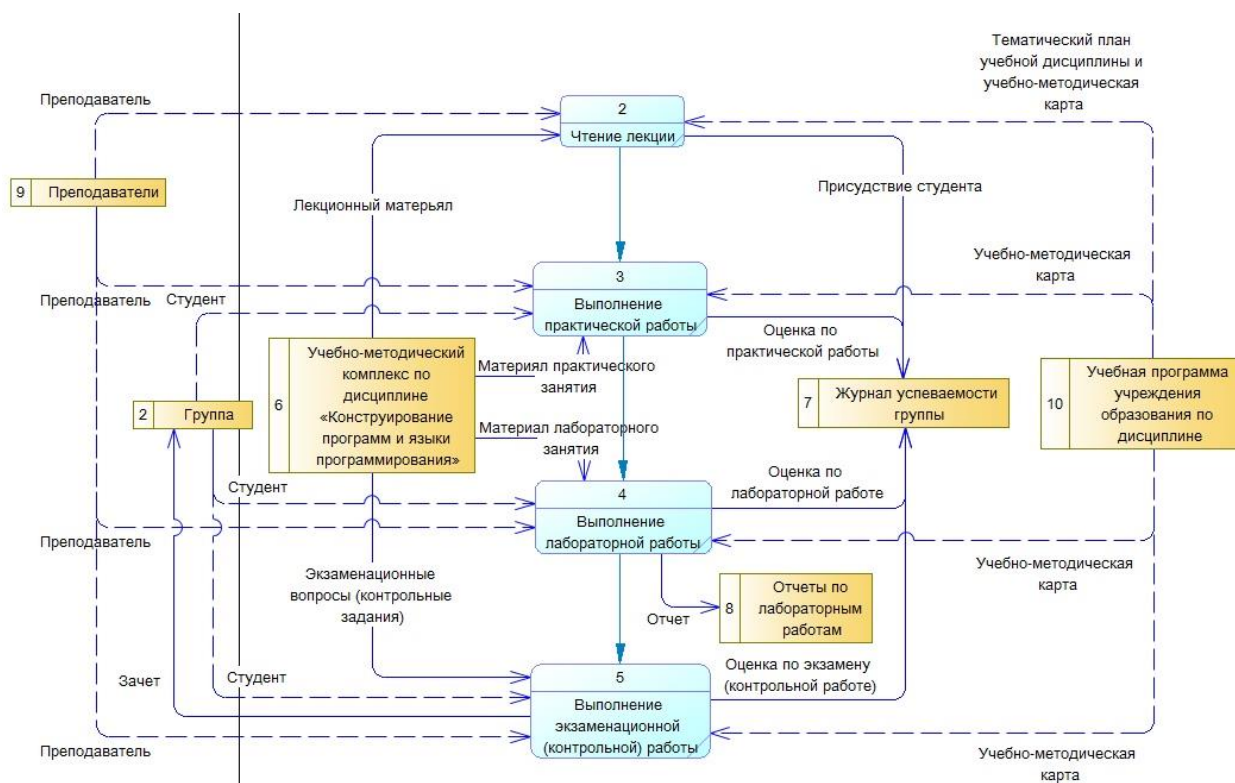


Рисунок 2.4 – Декомпозиция концептуальной схемы

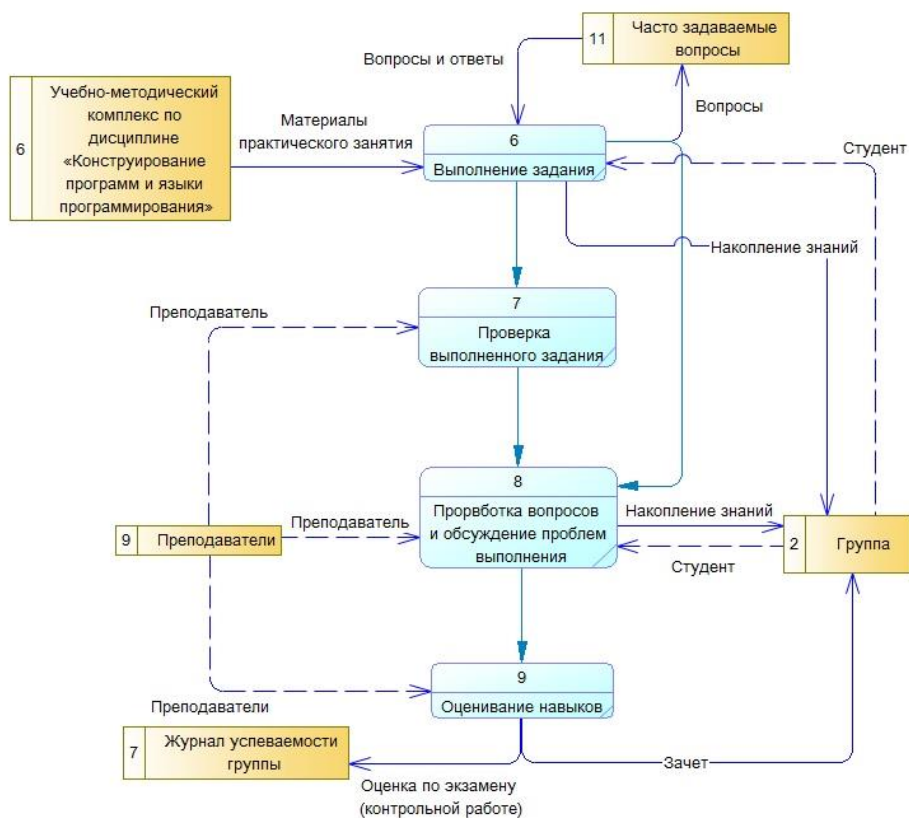


Рисунок 2.5 – Декомпозиция процесса выполнения практического задания

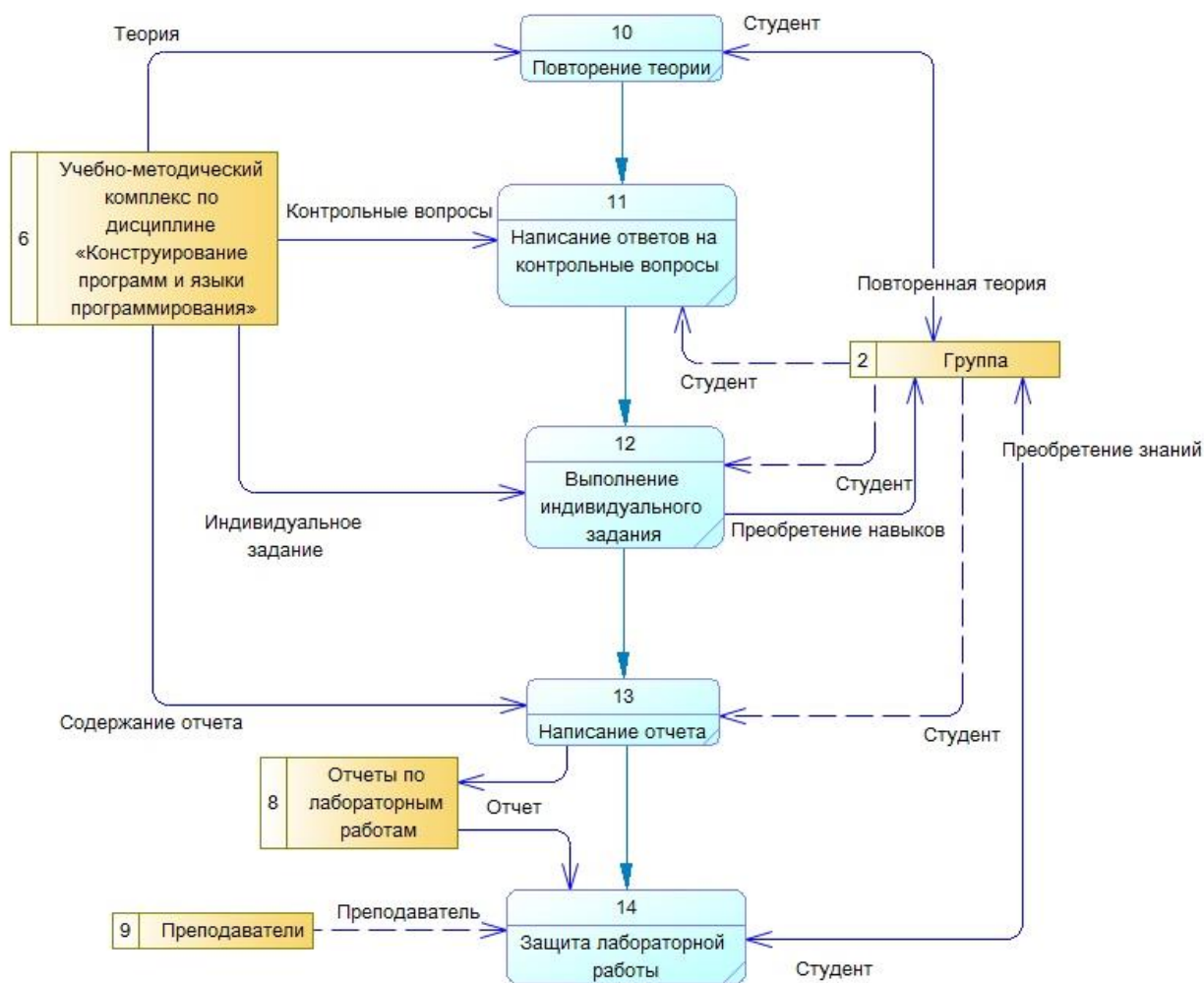


Рисунок 2.6 – Декомпозиция выполнения лабораторной работы

### 2.2.2 Этап информационного моделирования

Методология информационного моделирования IDEF1X известна также под названиями методологии семантического моделирования данных [6] и методологии концептуального моделирования [7].

Под информационной моделью (моделью данных) подразумевается графическое и текстовое представление результатов анализа предметной области, которое идентифицирует данные, используемые в организации [7].

*Целью* этапа является выявление сущностей, составляющих предметную область, и отношений между ними. *Результатом* является информационная модель предметной области, содержащая сущности, их атрибуты и отражающая отношения между сущностями.

Результаты этапа информационного моделирования будут использоваться для проектирования базы данных разрабатываемой системы.

Общепринятым стандартом представления информационных моделей (моделей данных) в настоящее время является стандарт IDEF1X [7], разработанный на основе диаграмм «Сущность–Связь» Чена.

Информационная модель предметной области, разработанная на основе моделирования потоков данных DFD, представлена на рисунке 2.7.

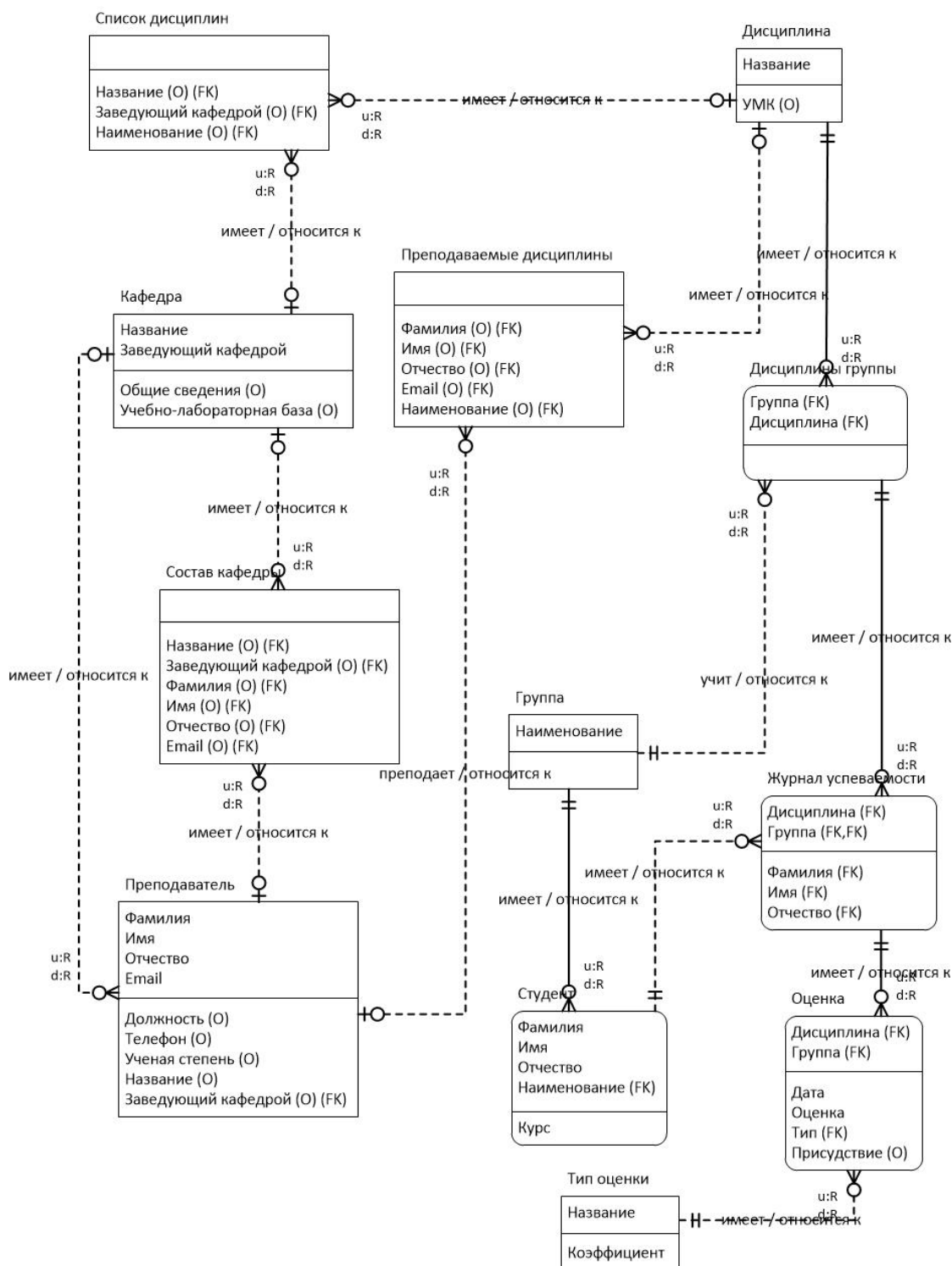


Рисунок 2.7 – Информационная модель предметной области IDEF1X



### 2.2.3 Этап моделирования поведения.

Динамическое (поведенческое) моделирование предметной области я считаю можно пропустить, так как поведение предметной области функционально полно отражено на IDEF0 диаграммах в приложении Б.

## 2.3 Требования к системе

В этом подразделе классификация требований описана по К. Вигерсу [8]. Классификация разделяет требования к разрабатываемой системе на 3 уровня:

- *Бизнес-требования* определяют высокоуровневые цели организации или клиента;
- *Пользовательские требования* описывают цели/задачи пользователей системы, которые должны выполняться пользователями при помощи создаваемого ПО;
- *Функциональные требования* определяют функциональность ПО, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требования.

Классификация требований по К. Вигерсу представлена на рисунке 2.8.



Рисунок 2.8 – Классификация требований по К. Вигерсу



### **2.3.1 Бизнес-требования.**

Главной целью учебного заведения является передача и контроль знаний. В следствии чего выделяются следующие бизнес-требования:

- Система должна предоставлять теоретический материал в полном объеме в соответствии с УМК предоставляемым учебным заведением;
- Система должна учитывать учебный план учебного заведения и быть адаптивной к возникающим изменениям;
- В течении обучения система должна предоставлять помощь в контроле знаний участников обучения.

### **2.3.2 Пользовательские требования.**

Пользователями системы являются Преподаватели и Студенты.

Требования к системе для пользователя Студент:

- Система должна легко и удобно предоставлять доступ к изучаемому материалу;
- Система должна обеспечивать среду для выполнения тестовых и практических заданий;
- Система должна предоставлять возможность сдачи отчета о выполненной лабораторной работе и практических работ;

Требования к системе для пользователя Преподаватель:

- Система должна предоставлять инструменты для создания, модернизации, расширения и управлять базой данных материала обучения, тестовых и практических заданий, дополнительных материалов;
- Система должна хранить отчетность и выполненную работу студентов, и предоставлять доступ к ней;
- Система должна предоставлять инструменты для создания и управления учетными записями студентов и преподавателей, а также студенческими группами;
- Система должна позволять вести учет успеваемости студентов в течении их обучения.

### **2.3.3 Функциональные требования**

Из пользовательских требований вытекают следующие функциональные требования:

- В системе должен присутствовать конструктор курса обучения, с помощью которого преподаватель сможет создавать, модернизировать, расширять и управлять курсами и планом обучения;

- Система должна предоставлять графический интерфейс для получения доступа к материалам курса для обучения студентов;

- В системе должна присутствовать форма для отправки отчетности о выполнении лабораторных работ, практических заданий и курсовых проектов на проверку преподавателем с обратной связью;

- В системе должен присутствовать онлайн компилятор для выполнения практических заданий студентами;

- В системе должен присутствовать журнал ведения учета успеваемости студентов;

- После выполнения студентами тестовых заданий система должна проверить ответы студентов на правильность выполнения теста и выставить оценку;

Из этой классификации можно выделить еще системные требования:

- Web-приложение запускаемое любым контейнером приложений;

- Кроссплатформенное приложение.

## 3 ОПИСАНИЕ СРЕДСТВ РАЗРАБОТКИ

### 3.1 Язык программирования и фреймворк

**Java** – сильно типизированный объектно-ориентированный кроссплатформенный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Исходный код, написанный на Java, транслируемый компилятором в байт-код, выполняется виртуальной машиной Java (JVM). JVM – программа, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее. Он предназначен для того, чтобы позволить разработчикам приложений «писать один раз, запускать где угодно» (WORA) [9], что означает, что скомпилированный код Java может работать на всех платформах, поддерживающих Java, без необходимости перекомпиляции [10].

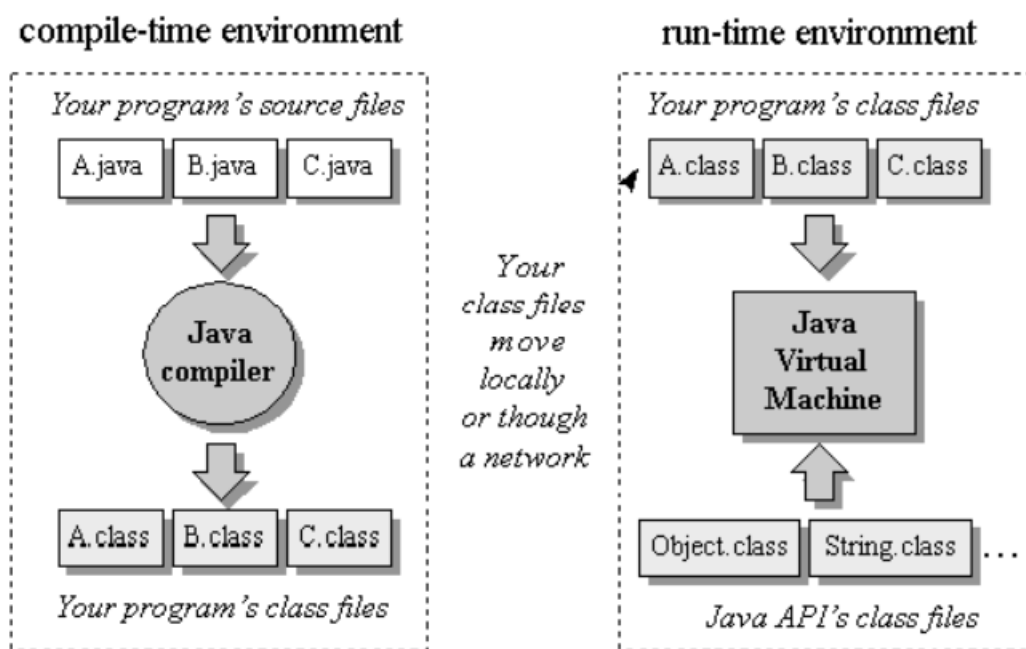


Рисунок 3.1 – Среда программирования Java

Также известно, что Java ориентирована на Internet. С помощью Java web-страницу можно наполнить не только обычным текстом, но и динамическими элементами.

Java-платформа обладает следующими преимуществами:

- Кроссплатформенность;
- Объектная ориентированность, создана эффективная объектная модель;

- Встроенная и прозрачная модель безопасности;
- Ориентация на Internet-задачи, сетевые распределенные приложения;
- Динамичность, легкость развития и добавления новых возможностей;

**Spring Framework** предоставляет комплексную модель программирования и конфигурации для современных корпоративных приложений на основе Java – на любой платформе развертывания [11].

Ключевым элементом Spring является инфраструктурная поддержка на уровне приложений: Spring фокусируется на «подключении» корпоративных приложений, чтобы группы могли сосредоточиться на бизнес-логике на уровне приложений без ненужных связей с конкретными средами развертывания.

Характеристика Spring Framework:

- Основные технологии: внедрение зависимостей, события, ресурсы, i18n, проверка, привязка данных, преобразование типов, SpEL, AOP.
- Тестирование: макет объектов, каркасные TestContext, Spring MVC Test, WebTestClient.
- Доступ к данным: транзакции, поддержка DAO, JDBC, ORM, Marshalling XML.
- Spring MVC и Spring WebFlux веб-фреймворки.
- Интеграция: удаленное взаимодействие, JMS, JCA, JMX, электронная почта, задачи, планирование, кэш.

**Spring Boot** позволяет легко создавать автономные производственные приложения на основе Spring, которые можно «просто запустить» [12].

Мы принимаем взвешенное мнение о платформе Spring и сторонних библиотеках, чтобы вы могли начать работу с минимальными усилиями. Большинству приложений Spring Boot требуется очень небольшая конфигурация Spring.

Характеристика Spring Boot:

- Создание автономных приложений Spring
- Внедрить Tomcat, Jetty или Undertow напрямую (не нужно развертывать файлы WAR)
- Предоставьте самоуверенные зависимости для начинающих, чтобы упростить конфигурацию сборки
- Автоматическая настройка Spring и сторонних библиотек, когда это возможно
- Обеспечить готовые к работе функции, такие как метрики, проверки работоспособности и внешняя конфигурация
- Абсолютно нет генерации кода и не требуется настройка XML

### 3.2 Средства получения, отладки исполняемого кода и средства тестирования программ

IntelliJ IDEA – интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains [13]. Обладает широким набором интегрированных инструментов для рефакторинга, которые позволяют программистам быстро реорганизовывать исходные тексты программ, и идеально подходит для создания коммерческих, мобильных и веб-приложений. IntelliJ IDEA включает поддержку всех последних технологий и фреймворков, работает с автоматизированными инструментами сборки и управления проектом, включая Maven, Gradle, Ant и другими. Также IntelliJ IDEA обладает встроенным отладчиком исполняемого кода и средствами тестирования программ.

### 3.3 Средства управления базами данных

MySQL – свободная реляционная система управления базами данных. MySQL является решением для малых и средних приложений [14]. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk, Компонентный Паскаль и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

### 3.4 Средства запуска готового проекта

В основе технологии Java лежит **виртуальная машина Java** (JVM) – абстрактный компьютер, на котором работают все программы Java [15] – основная часть исполняющей

системы Java, так называемой Java Runtime Environment (JRE). Виртуальная машина Java, Java API и файл классов Java работают вместе с языком для запуска программ Java.

Виртуальные машины Java обычно содержат Интерпретатор байт-кода, однако, для повышения производительности во многих машинах также применяется JIT-компиляция часто исполняемых фрагментов байт-кода в машинный код.

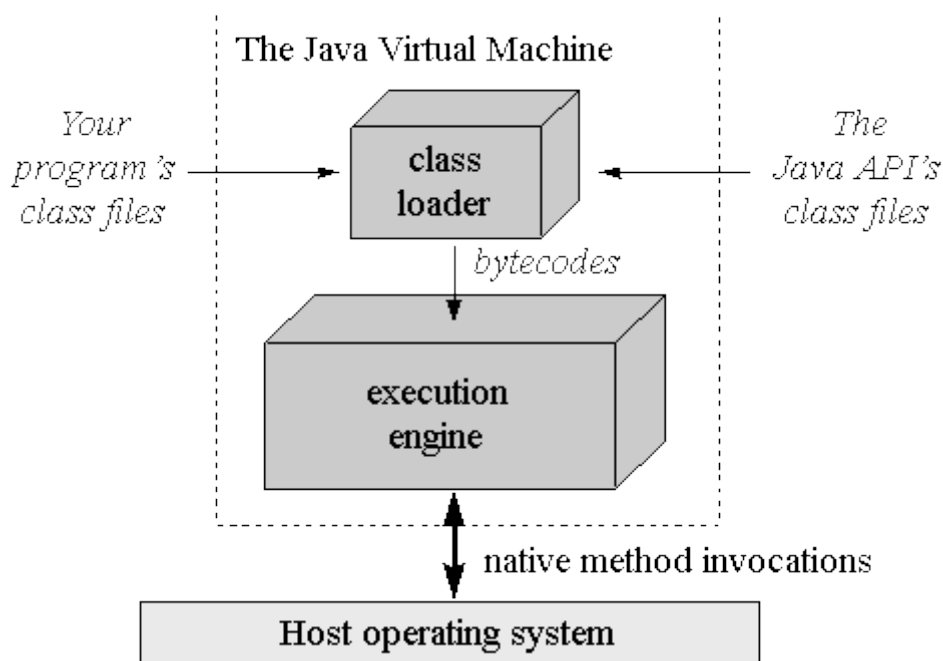


Рисунок 3.2 – Виртуальная машина Java, реализованная в программном обеспечении поверх операционной системы хоста

**Apache Tomcat** – контейнер сервлетов с открытым исходным кодом, разрабатываемый Apache Software Foundation. Реализует спецификацию сервлетов, спецификацию JavaServer Pages (JSP) и JavaServer Faces (JSF) [16]. Написан на языке Java.

Tomcat позволяет запускать веб-приложения, содержит ряд программ для самоконфигурирования.

Tomcat используется в качестве самостоятельного веб-сервера, в качестве сервера контента в сочетании с веб-сервером Apache HTTP Server, а также в качестве контейнера сервлетов в серверах приложений JBoss и GlassFish.

### 3.5 Средства проектирования приложения

Microsoft Visio – векторный графический редактор, редактор диаграмм и блок-схем для Windows. Основным назначением Visio является визуализация данных или, проще

говоря, изготовление практически любых схем, иллюстраций и диаграмм, необходимых для оформления деловой документации.

Используя эту программу, вы можете отобразить ход выполнения проекта, подготовить план здания, составить информационную модель предметной области IDEF1X, составить функциональную модель IDEF0 и смоделировать информационные потоки DFD предметной области. Содержание иллюстраций зависит только от вашего воображения. При этом вы сможете гибко настраивать внешний вид элементов, что позволит не только использовать функциональную окраску частей иллюстрации, но и получить опрятный и привлекательный документ.

CASE-средство IBM Rational Rose со времени своего появления претерпело серьезную эволюцию, и в настоящее время представляет собой современный интегрированный инструмент для проектирования архитектуры, анализа, моделирования и разработки программных систем. Именно в IBM Rational Rose язык UML стал базовой технологией визуализации и разработки программных систем, что определило популярность и стратегическую перспективность этого инструментария.

Наиболее характерные функциональные особенности этой программы заключаются в следующем:

- интеграция с MS Visual Studio 6, которая включает поддержку на уровне прямой и обратной генерации кодов и диаграмм Visual Basic и Visual C++ с использованием ATL (Microsoft Active Template Library), Web-Классов, DHTML и протоколов доступа к различным базам данных;

- непосредственная работа (инжиниринг и реинжиниринг) с исполняемыми модулями и библиотеками форматов EXE, DLL, TLB, OCX.

- поддержка технологий MTS (Microsoft Transaction Server) и ADO (ActiveX Data Objects) на уровне шаблонов и исходного кода, а также элементов технологии Microsoft - COM+ (DCOM);

- полная поддержка компонентов CORBA и J2EE, включая реализацию технологии компонентной разработки приложений CBD (Component-Based Development), языка определения интерфейса IDL (Interface Definition Language) и языка определения данных DDL (Data Definition Language);

- полная поддержка среды разработки Java-приложений, включая прямую и обратную генерацию классов Java формата JAR, а также работу с файлами формата CAB и ZIP.

### 3.6 Средства создания пользовательского интерфейса

HTML – стандартизированный язык гипертекстовой разметки документов для создания веб-страниц и веб-приложений [17]. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

CSS (каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием HTML [18]. CSS предназначен для разделения представления и содержимого, включая макеты, цвета и шрифты.

Bootstrap – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения [19].

Основные инструменты Bootstrap:

- Сетки – заранее заданные размеры колонок, которые можно сразу же использовать, например, ширина колонки 140 px относится к классу `.span2` (`.col-md-2` в третьей версии фреймворка), который можно использовать в CSS-описании документа;
- Шаблоны – фиксированный или резиновый шаблон документа;
- Типографика – описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.;
- Медиа – предоставляет некоторое управление изображениями и видео;
- Таблицы – средства оформления таблиц, вплоть до добавления функциональности сортировки;
- Формы – классы для оформления форм и некоторых событий, происходящих с ними;
- Навигация – классы оформления для панелей, вкладок, перехода по страницам, меню и панели инструментов;
- Алерты – оформление диалоговых окон, подсказок и всплывающих окон.



## 4 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

### 4.1 Разработка диаграммы вариантов использования

В рамках эксплуатации программного продукта преподаватель столкнется с необходимостью добавлять, создавать, изменять, удалять, сохранять, просматривать аккаунты студентов, а также списки групп. Еще преподавателю необходим инструмент для создания и модернизации курса обучения.

Студентам нужно предоставить возможность просмотра материала, выполнение тестов, выполнение практических, лабораторных и контрольных работ, а также возможность отправки отчетности о выполненной работе.

Для более наглядного представления о возможностях системы, я разработал диаграмму вариантов использования, которая представлена на рисунке 4.1.

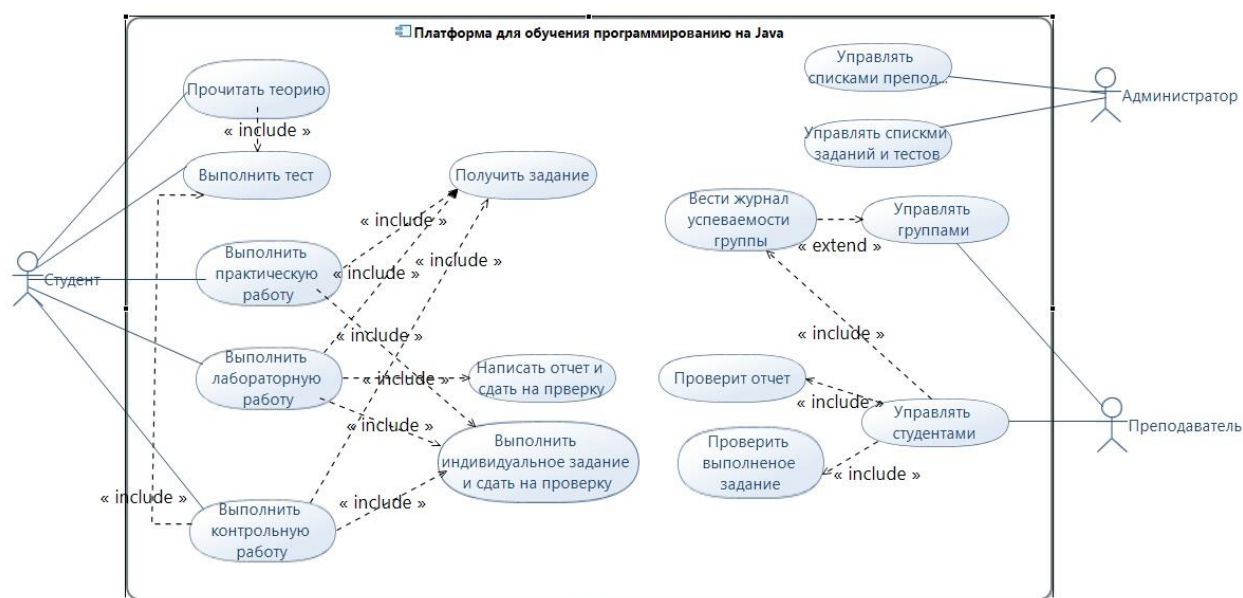


Рисунок 4.1 – Диаграмму вариантов использования

### 4.2 Проектирование и разработка модели хранения данных

При анализе предметной области, я выявил следующие сущности, которые учувствуют в бизнес-модели предметной области:

– Участники: преподаватели, студенты;

- Материалы: теоретический материал, практический материал, тесты;
- Документы: отчеты по лабораторным работам, результаты практической работы, курсовые проекты, контрольные работы, журнал успеваемости;

На основе сущностей предметной области я разработал список сущностей, которые буду учувствовать в разрабатываемой системе для хранения данных:

- Пользователи: обычные пользователи, преподаватели, администраторы;
- Теоретический и практический материалы: курсы, темы, разделы, вопросы, ответы, тесты, задания;
- Документы: отчеты по лабораторным работам, результаты практической работы, оценки;

На основе данных сущностей я разработал диаграмму сущностей представленную на рисунке 4.2.

Для создания модели данных я использовал встроенный инструмент Spring Boot, а именно Spring JPA – спецификация API Java EE, предоставляет возможность сохранять в удобном виде Java-объекты в базе данных [20].

Реализует JPA библиотека Hibernat – самая популярная реализация спецификации JPA, предназначенная для решения задач объектно-реляционного отображения (ORM) [21].

ORM (Object-Relational Mapping, объектно-реляционное отображение) – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных» [22].

Схема преобразования POJO объектов (простой Java-объект) в таблицы базы данных показана на рисунке 4.3.

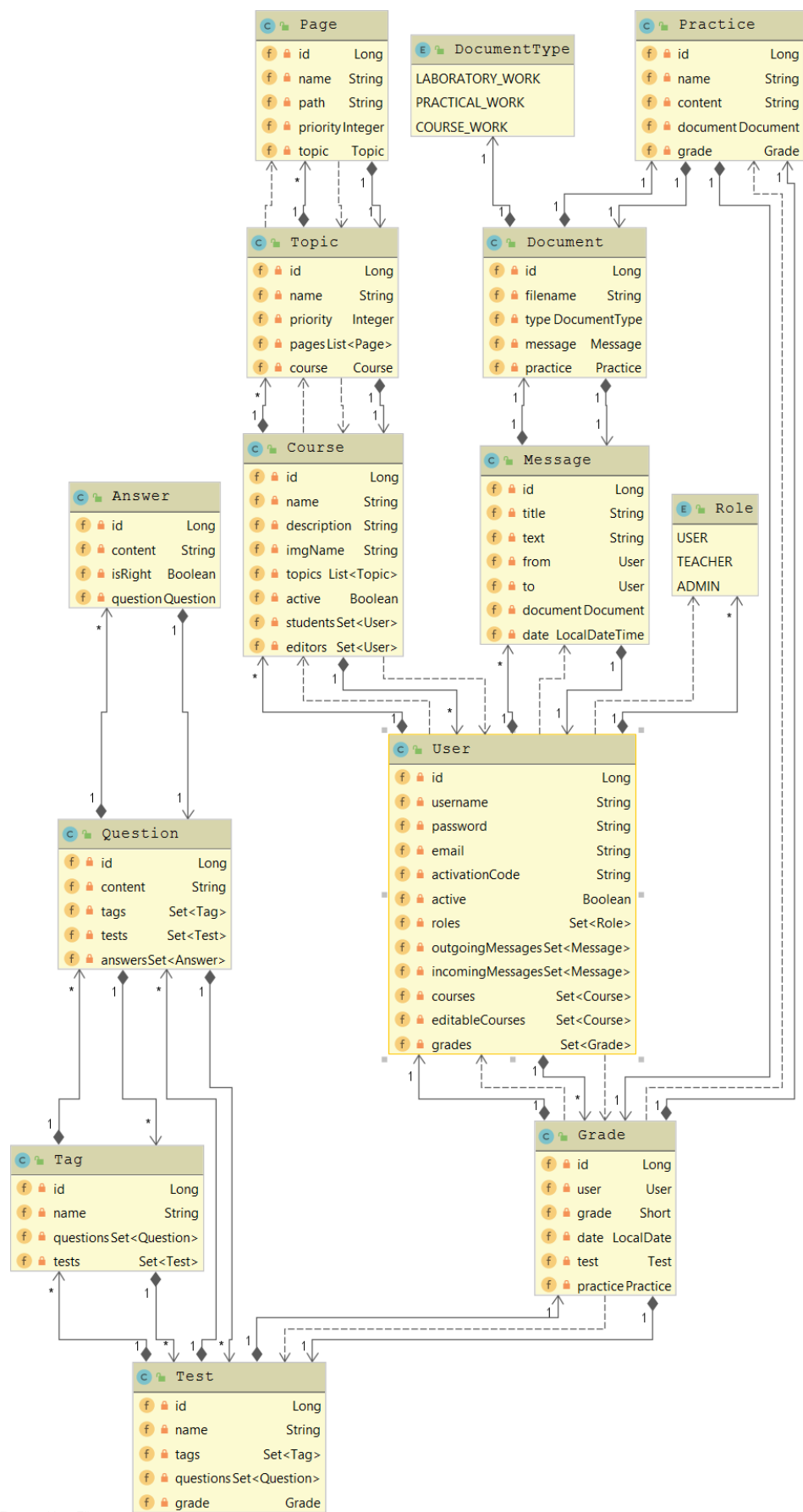
Результат преобразования сущностей в таблицы базы данных представлен на рисунке 4.4.

В итоге я получил готовую базу данных, но чтобы избежать проблем в развертывании и проблем безопасности данных на основе инструмента миграции баз данных Flyway я разработал 2 файла sql файла.

Flyway – это инструмент миграции базы данных с открытым исходным кодом.

Он основан на 7 основных командах: Migrate, Clean, Info, Validate, Undo, Baseline и Repair.

Миграции могут быть написаны на SQL ( поддерживается специфичный для базы данных синтаксис, такой как PL/SQL, T-SQL и т. Д.) или Java (для сложных преобразований данных или работы с большими объектами).



Powered by yFiles

Рисунок 4.2 – Диаграмма сущностей

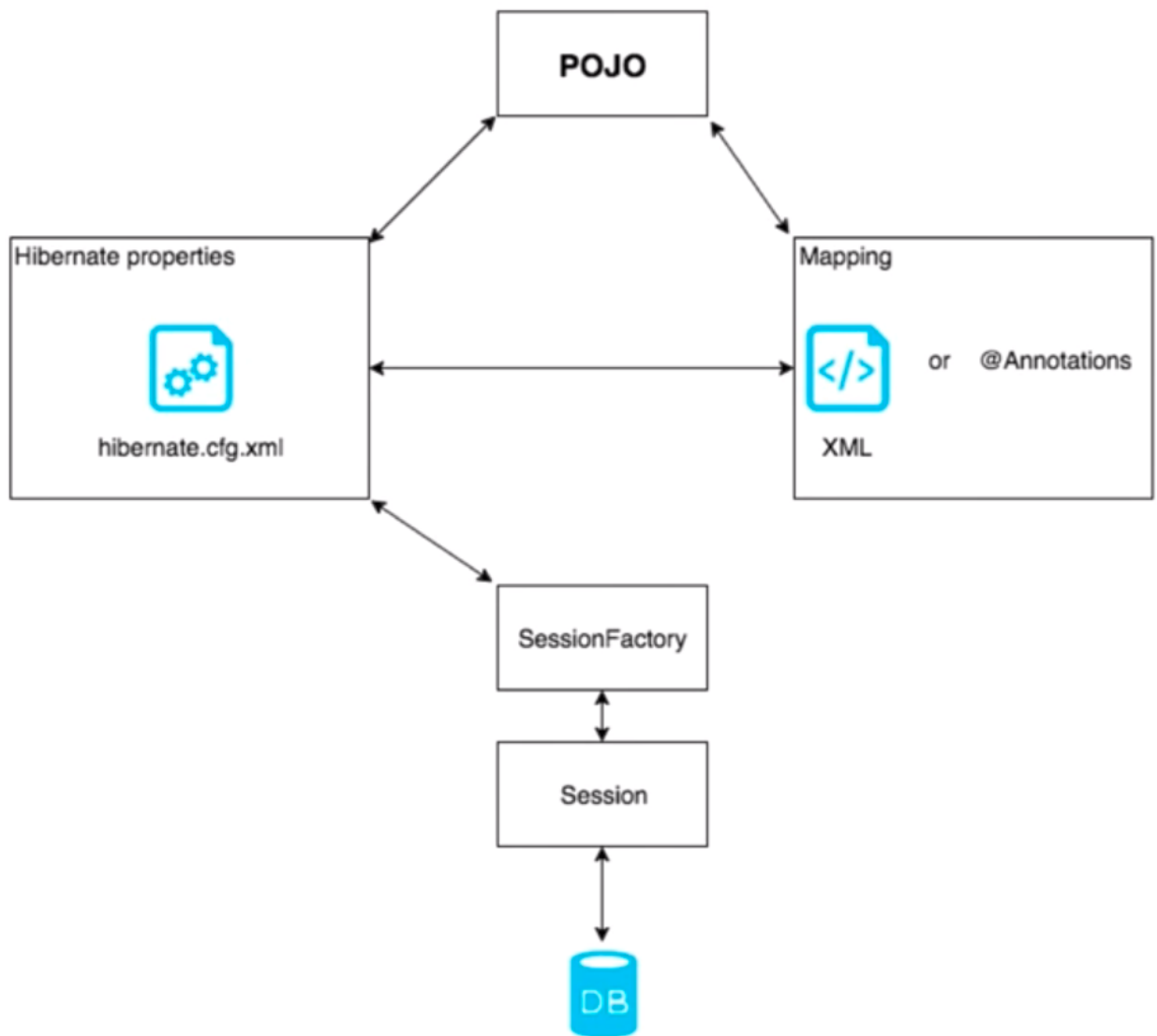


Рисунок 4.3 – Схема преобразования Java-объектов в таблицы базы данных

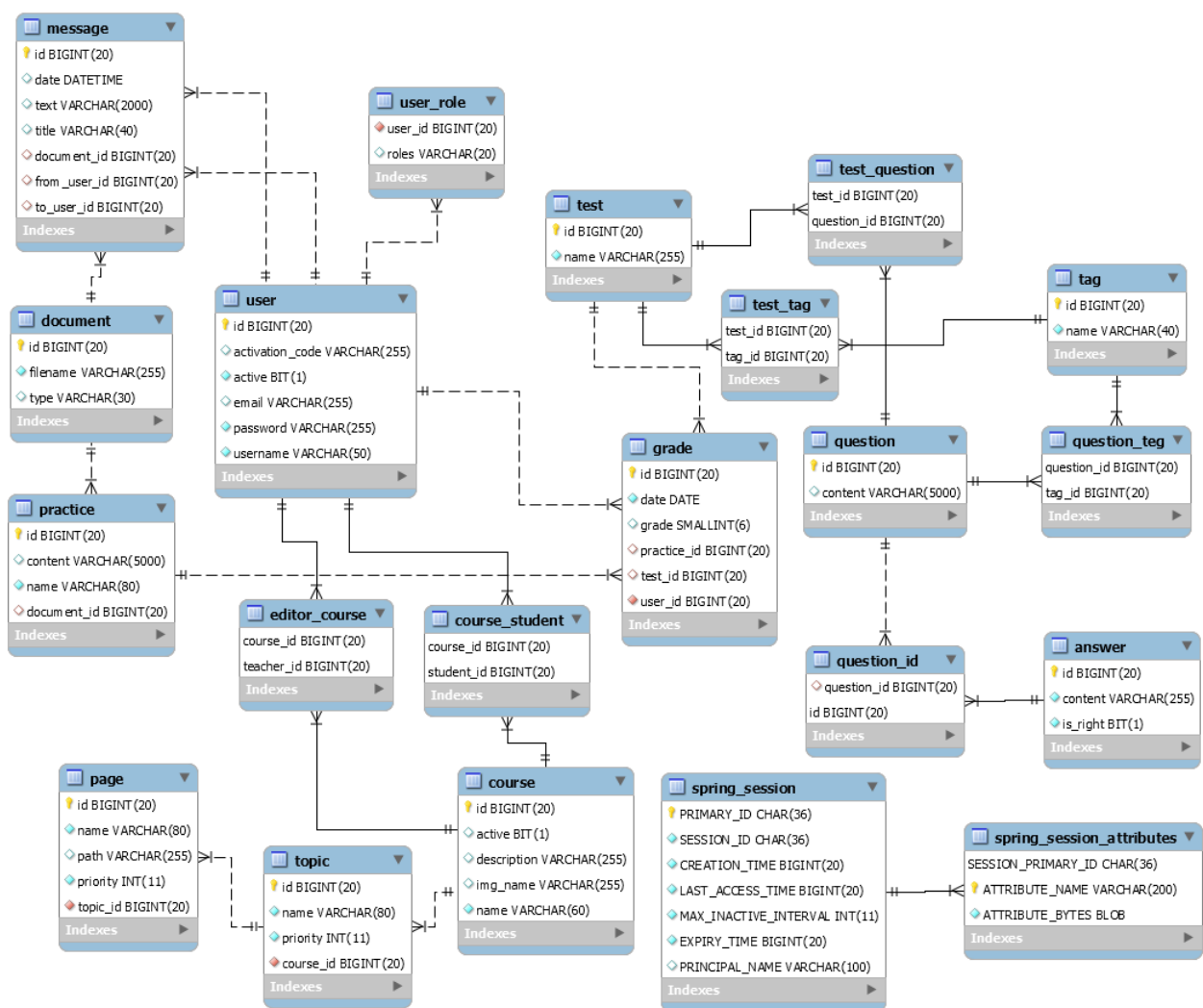


Рисунок 4.4 – Модель базы данных

SQL файлы используются для создания структуры базы данных и начальной конфигурацией для начала работы разработанной системы. Файлы находятся в директории db.migration:

– V1\_\_Create\_tables.sql – текстовый файл, содержащий SQL скрипт, который создает таблицы в базе данных и создает связи между ними.

– V2\_\_Init\_db.sql – текстовый файл, содержащий SQL скрипт, который инициализирует таблицы стартовыми данными для начальной работы платформы. Скрипт создает трех пользователей с разными правами доступа к функционалу платформы. Это необходимо, так как при регистрации аккаунта система автоматически дает самые минимальные пользовательские права, но только аккаунт с наивысшими правами доступа способен раздавать более высокие права. Это говорит о том, что в системе должен быть заранее подготовлен аккаунт с наивысшими правами доступа.

## 4.3 Разработка структуры проекта

### 4.3.1 Spring MVC.

Так как система строится «клиент-серверной» архитектуре, я воспользуюсь реализацией Spring Web MVC – это оригинальный веб-фреймворк, основанный на Servlet API, который был включен в Spring Framework с самого начала [23].

Структура Spring Web MVC представлена на рисунке 4.5.

DispatcherServlet – это главный контроллер (Front controller), который обрабатывает все входящие и все исходящие запросы. Все поступившие запросы он передает на обработку HandlerMapping контроллеру, который должен решить какому, из все созданных разработчиком контроллеров должен делегировать этот запрос для выполнения запрашиваемой пользователем операции.

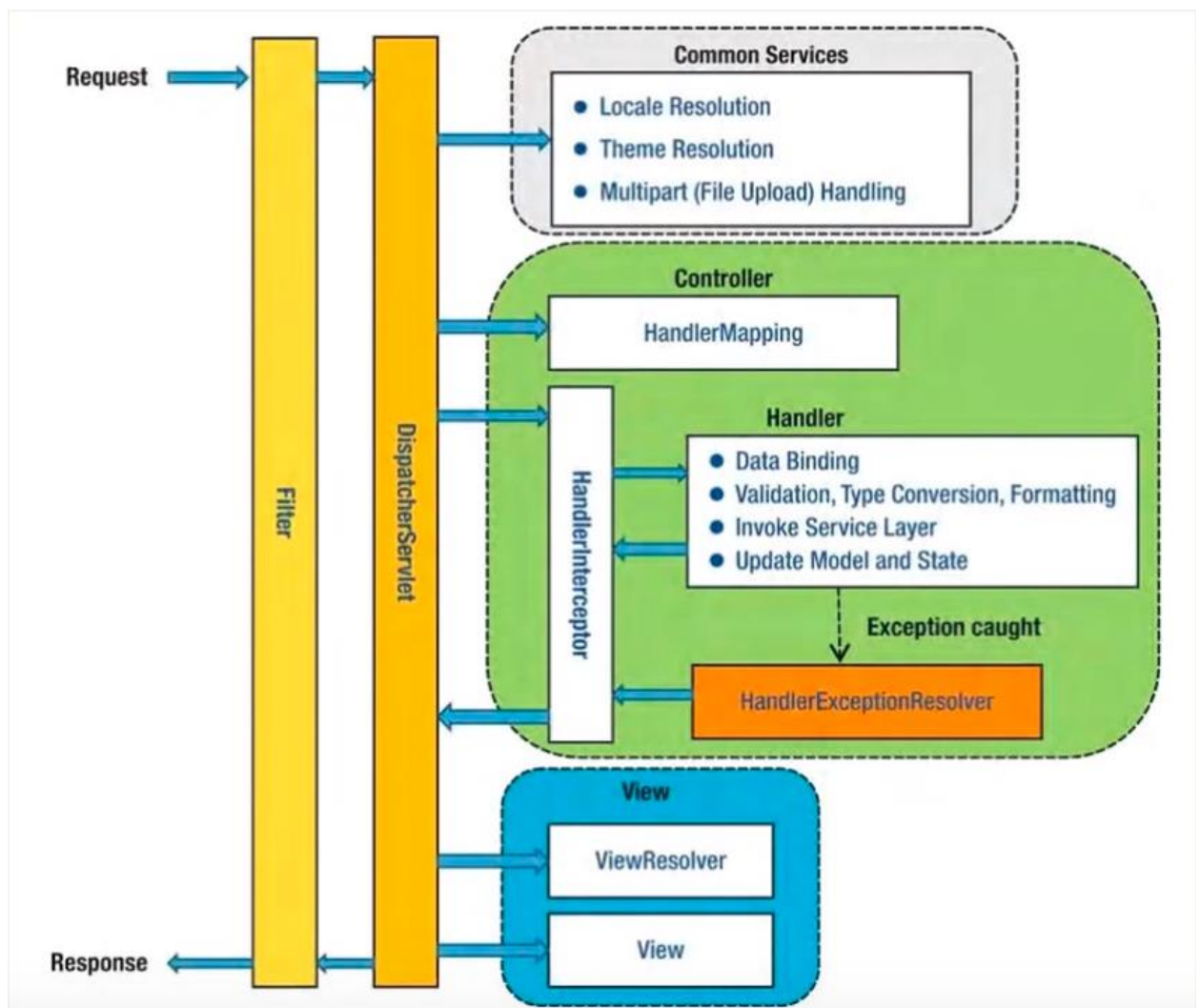


Рисунок 4.5 – Структура Spring Web MVC

После HandlerMapping дает команду главному контроллеру (DispatcherServlet), чтобы тот передал запрос нужному контроллеру. Контроллер выполняет операцию, создает модель (Model), где храниться данные для пользователя и имя представления (View Name). Контроллер передает главному контроллеру данные (ModelAndView), достает View Name, передает контроллеру представлений (ViewResolver), который находит нужное представление (View) – шаблон, на основе которого будет строиться ответ пользователю, для построения ответа пользователю, отдает нужное представление главному контроллеру. Контроллер на основе представления (View) и данных (Model) строит ответ пользователю и отправляет его.

#### 4.3.2 Структура разрабатываемого проекта

Структура проекта представлена на рисунке 4.6.

Исходя из требований к разрабатываемой системе и выбранной архитектуры приложения, я спроектировал следующую структуру проекта:

- Контроллеры (controller), которые выполняют обработку запросов пользователей и выполняют требуемые операций. Диаграмма классов-контроллеров проекта представлена на рисунке 4.7;

- Сущности (entity), которые занимаются хранением данных в удобной форме. Диаграмма классов-сущностей представлена на рисунке 4.2;

- Сервисы (service), которым делегируется выполнение сложных операция запрашиваемые пользователем. Диаграмма классов-сервисов представлена на рисунке 4.8;

- DAO (Data Access Object) слой, который выполняет CRUD-операции для получения доступа к хранимым данным в базе данных. Диаграмма классов-dao представлена на рисунке 4.9;

- Представления (templates), которые предназначены для построения ответа пользователю;

- Настройки окружения проекта (config, application.properties) – настраивает проект для работы приложения. Диаграмма классов-конфигурации представлена на рисунке 4.10.

Диаграмма пакетов представлена на рисунке 4.11.

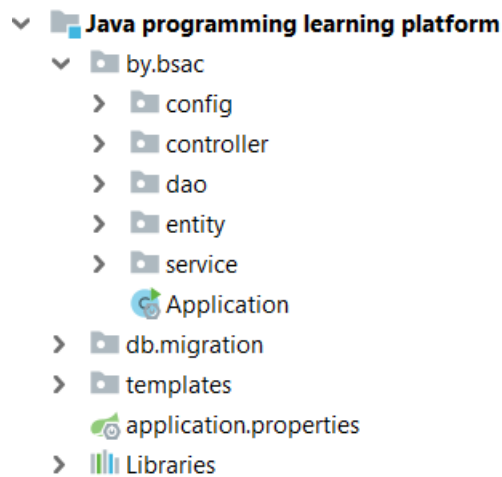


Рисунок 3.6 – Структура разрабатываемого проекта

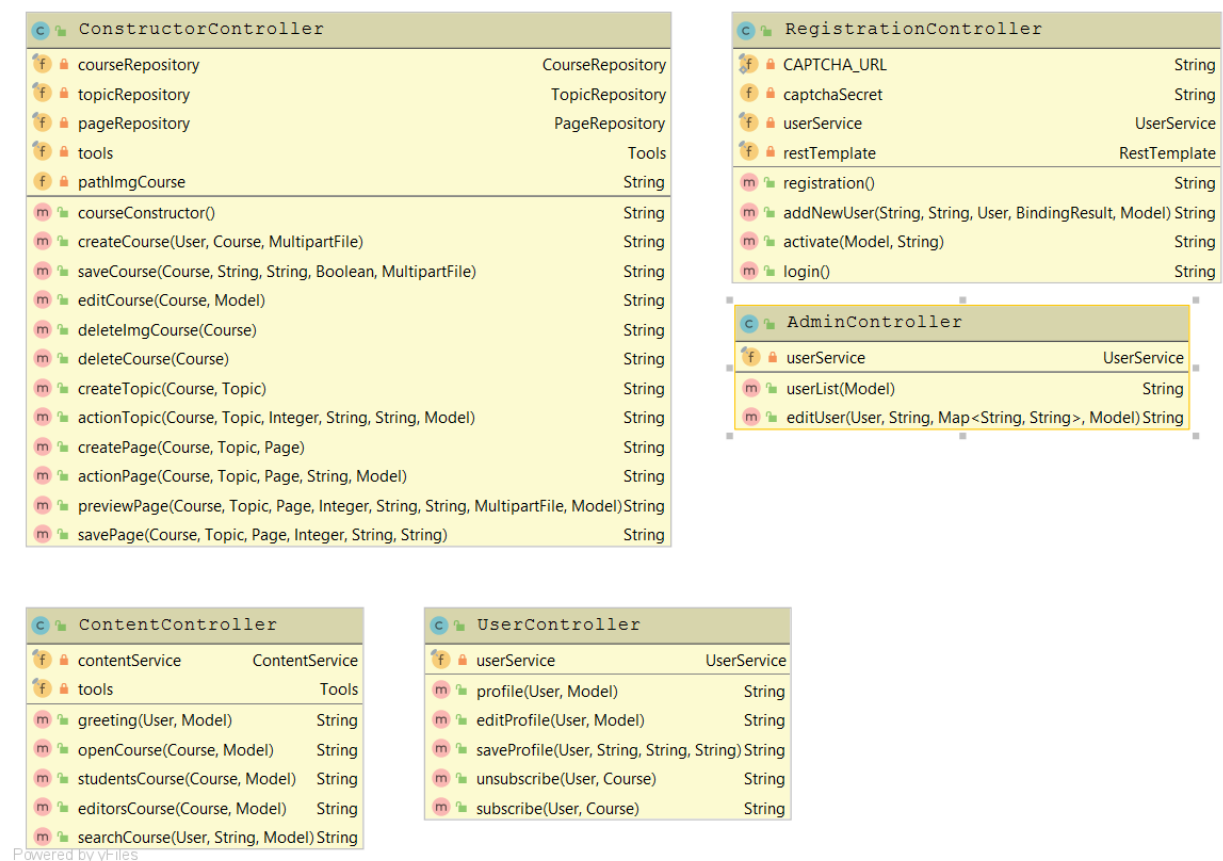


Рисунок 4.7 – Диаграмма классов-контроллеров



UserService		
f	userRepository	UserRepository
f	mailSend	MailSend
f	passwordEncoder	PasswordEncoder
m	loadUserByUsername(String)	UserDetails
m	addUser(User)	boolean
m	sendActivationCode(User)	void
m	activateUser(String)	boolean
m	findAll()	Iterable<User>
m	save(User, Map<String, String>)	void
m	updateUser(User, String, String, String)	void
m	deleteUser(User)	void
m	updateUser(User)	void

Tools		
f	pathContentPage	String
f	pathImgCourse	String
m	getPasswordEncoder()	PasswordEncoder
m	readContentPage(String)	String
m	readContentPage(MultipartFile)	String
m	writeContentPage(String)	String
m	convertStringToStringByte(String, String)	String
m	convertStringByteToString(String, String)	String
m	deleteContentPage(String)	void
m	saveImg(MultipartFile)	String
m	getErrorMap(BindingResult)	Map<String, String>

ContentService		
f	courseRepository	CourseRepository
m	getCourses()	Iterable<Course>
m	getCourses(Boolean)	Iterable<Course>
m	getCourses(User)	Iterable<Course>
m	searchCoursesByName(Boolean, String)	Iterable<Course>
m	searchCoursesByName(User, String)	Iterable<Course>

MailSend		
f	username	String
f	mailSender	JavaMailSender
m	send(String, String, String)	void

Powered by yFiles

Рисунок 4.8 – Диаграмма классов-сервисов

CourseRepository		
m	findCoursesByActive(Boolean)	Iterable<Course>
m	findCoursesByEditorsContains(User)	Iterable<Course>
m	findCoursesByActiveAndNameLike(Boolean, String)	Iterable<Course>
m	findCoursesByEditorsContainsAndNameLike(User, String)	Iterable<Course>

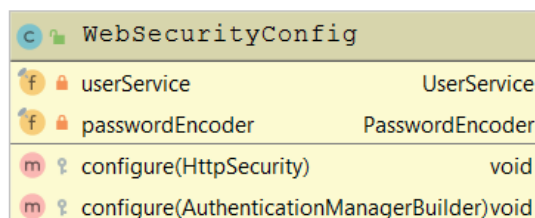
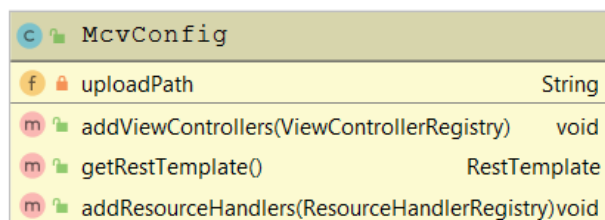
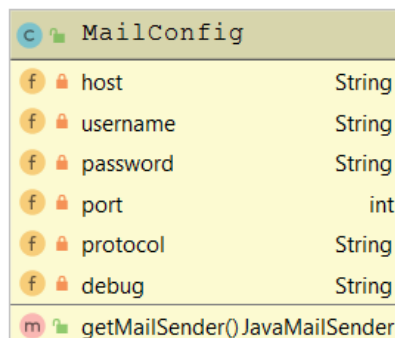
UserRepository		
m	findByUsername(String)	User
m	findByActivationCode(String)	User

TopicRepository		
-----------------	--	--

PageRepository		
----------------	--	--

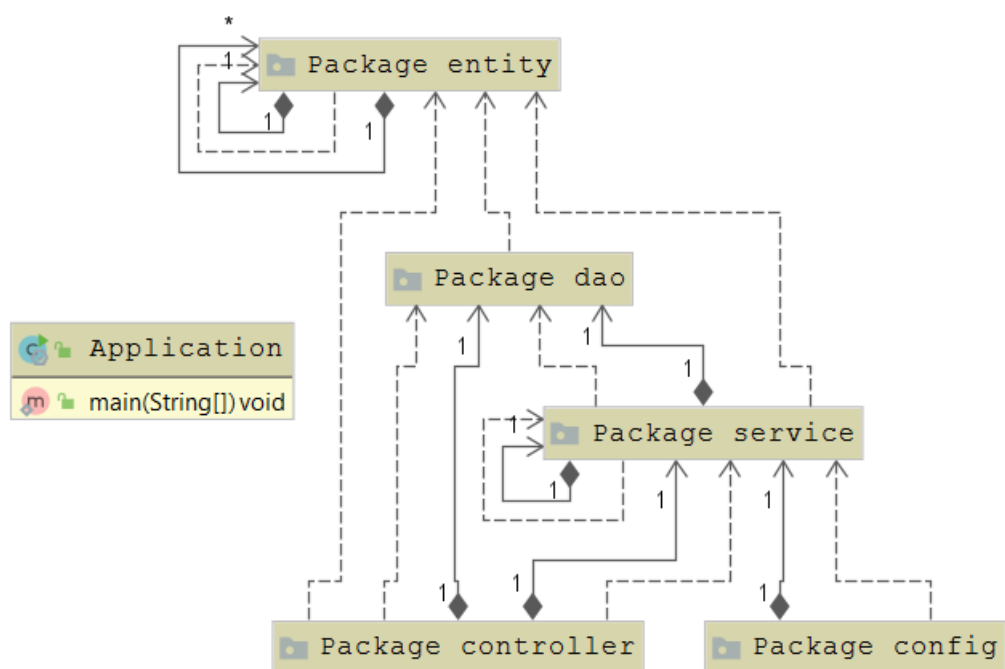
Powered by yFiles

Рисунок 4.9 – Диаграмма классов-dao



Powered by yFiles

Рисунок 4.10 – Диаграмма классов-конфигурации



Powered by yFiles

Рисунок 4.11 – Диаграмма пакетов

### 4.3.3 Spring IoC и DI.

В основе работы Spring лежит 2 термина: инверсия управления (Inversion of Control) и внедрение зависимостей (Dependency Injection).

Инверсия управления (IoC) – это принцип в разработке программного обеспечения, при котором управление объектами или частями программы передается в контейнер или среду [24]. IoC позволяет инфраструктуре управлять потоком программы и выполнять вызовы нашего пользовательского кода. Чтобы включить это, фреймворки используют абстракции со встроенным дополнительным поведением. Если мы хотим добавить наше собственное поведение, нам нужно расширить классы фреймворка или подключить наши собственные классы. Инверсия управления может быть достигнута с помощью различных механизмов, таких как: шаблон проектирования стратегии, шаблон локатора служб, шаблон фабрики и *внедрение зависимостей (DI)*.

Внедрение зависимостей (DI) – это шаблон для реализации IoC, где инвертируемый элемент управления – это установка зависимостей объекта [24]. Акт соединения объектов с другими объектами или «инъекция» объектов в другие объекты выполняется транслятором, а не самими объектами.

Платформа Spring предоставляет несколько реализаций интерфейса ApplicationContext (контейнером IoC) – ClassPathXmlApplicationContext и FileSystemXmlApplicationContext для автономных приложений и *WebApplicationContext* для *веб-приложений*, который я использовал в разработке данной системы.

Аннотация @SpringBootApplication, определенная в классе Application.java, эквивалентна использованию таких аннотаций, как @Configuration, @EnableAutoConfiguration и @ComponentScan с их атрибутами по умолчанию [25]. Зависимости объекта application представлены на рисунке 4.12.

@EnableAutoConfiguration – включает автоконфигурацию приложения, которая настраивает приложение на основе добавленных зависимостей jar. Для добавления нужных зависимостей я использовал сборщик проектов Maven с помощью конфигурационного файла – pom.xml.

@ComponentScan – вкачает автоматическое сканирование пакетов в приложении для поиска и дальнейшего внедрения зависимостей bean-объектов отмеченные аннотацией @Component.

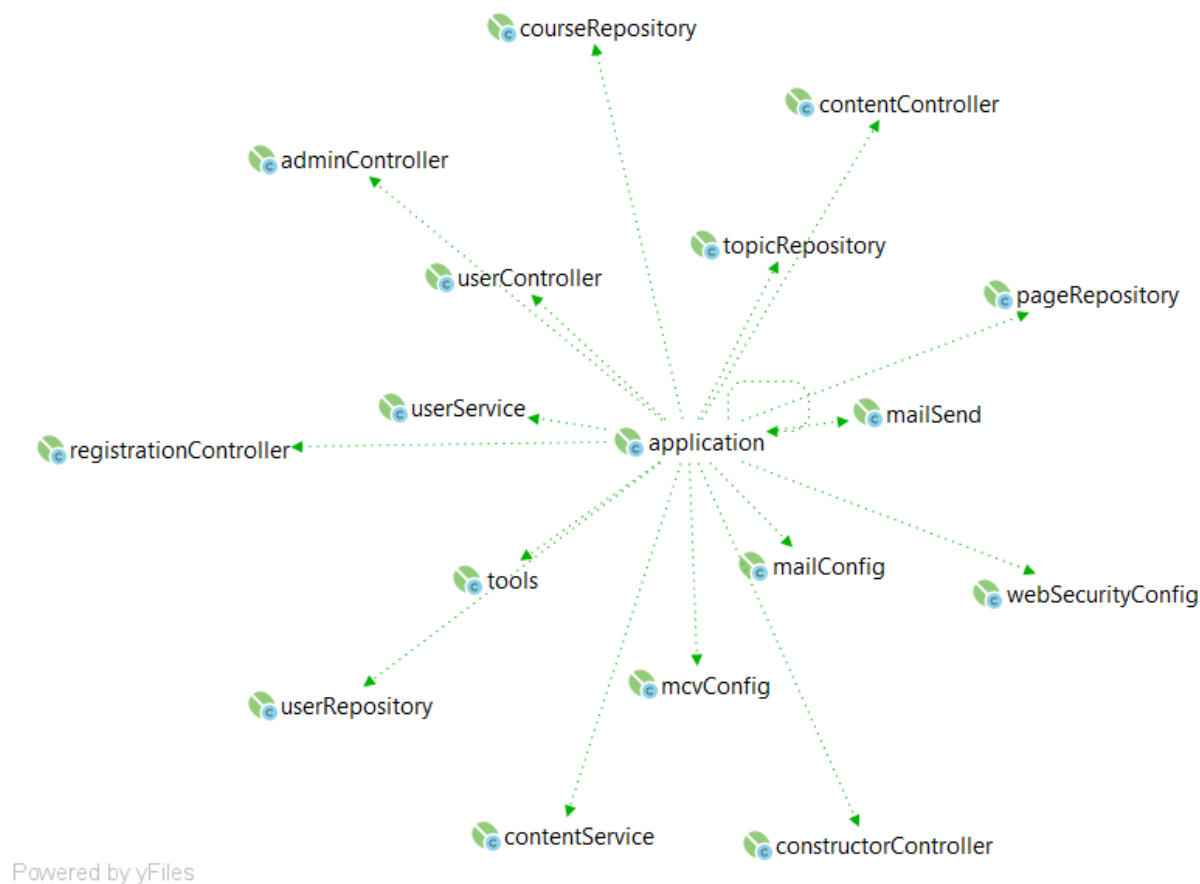


Рисунок 4.12 – Spring зависимости объекта application

@Configuration – позволяет регистрировать дополнительные компоненты в контексте или импортировать дополнительные классы конфигурации, что позволяет в любой момент определять свою собственную конфигурацию для замены определенных частей автоконфигурации. В структуре проекта классы, которые заменяют конфигурацию по умолчанию, являются:

– MvcConfig, реализующий интерфейс WebMvcConfigurer – позволяет конфигурировать настройки MVC архитектуры. Зависимости объекта mvcConfig представлены на рисунке 4.13.

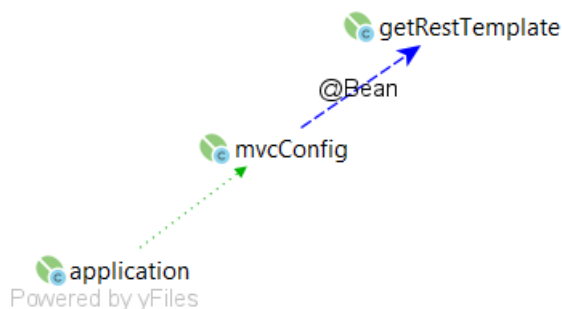


Рисунок 4.13 – Spring зависимости объекта mvcConfig



@EnableGlobalMethodSecurity), а также позволяет шифровать пароли пользователей при регистрации в системе.

– MailConfig – настраивает почту, с которой будут приходить письма разного рода уведомлений пользователям. Зависимости объекта mailConfig представлены на рисунке 4.15.



Рисунок 4.15 – Spring зависимости объекта mailConfig

## 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

### 5.1 Начальная конфигурация платформы

Перед развертывание платформы необходимо указать несколько параметров в файле *application.properties*:

– В поле *spring.datasource.url* указать имя базы данных, где будут храниться все данные платформы:

```
spring.datasource.url=jdbc:mysql://localhost:3306/{Имя_базы}?serverTimezone=UTC
```

– В полях *spring.datasource.username* и *spring.datasource.password* указать имя и пароль учетной записи с полномочиями редактирования базы данных, указанной выше:

```
spring.datasource.username={Наименование_учетной_записи}
```

```
spring.datasource.password={Пароль_учетной_записи}
```

– В поле *upload.path* указать директорию, где будут храниться загружаемые изображения:

```
upload.path={Путь_к_директории}
```

– В поле *upload.path.page.content* указать директорию, где будут храниться содержимое курсов:

```
upload.path.page.content={Путь_к_директории}
```

– В полях *spring.mail.host*, *spring.mail.username*, *spring.mail.password*, *spring.mail.port*, *spring.mail.protocol* указать хост, наименование почты, пароль, порт и протокол соответственно для отправления различного рода уведомлений от платформ:

```
spring.mail.host={Имя_хоста}
```

```
spring.mail.username={Наименование_почты}
```

```
spring.mail.password={Пароль_почты}
```

```
spring.mail.port={Порт}
```

```
spring.mail.protocol={Протокол}
```

После необходимо создать базу данных на сервере MySQL и учетную запись пользователя. Скрипт создания базы данных и учетной записи пользователя предоставлен ниже:

```
create database {наименование_базы_данных};
```

```
create user '{наименование_учетной_записи}'@'%' identified by '{пароль}';
```

```
grant all on {наименование_базы_данных}.* to '{наименование_учетной_записи}'@'%';
```

После развертывания платформа содержит в себе 3 аккаунта с разными правами доступа для тестирования функционала (все данные аккаунтов можно изменить, внося изменения в файл V2\_\_Init\_db.sql перед развёртыванием):

- Права доступа «Администратор» (Role = {USER, TEACHER, ADMIN}). Наименование аккаунта – admin, пароль – admin, email – admin@admin.com, id – 0, активность аккаунта – true;

- Права доступа «Учитель» (Role = {USER, TEACHER}). Наименование аккаунта – teacher, пароль – teacher, email – teacher@teacher.com, id – 1, активность аккаунта – true;

- Права доступа «Пользователь» (Role = {USER}). Наименование аккаунта – user, пароль – user, email – user@user.com, id – 2, активность аккаунта – true.

## 5.2 Регистрация и вход в профиль пользователя

### 5.2.1 Регистрация на платформе разделено на 2 этапа:

- Ввод данных в форму регистрации на странице регистрации. Форма регистрации представлен на рисунке 5.1.

The registration form consists of the following elements:

- Имя:** Input field with placeholder text "Имя пользователя".
- Email:** Input field with placeholder text "some@some.com".
- Пароль:** Input field with placeholder text "Пароль".
- Повторите пароль:** Input field with placeholder text "Повторите пароль".
- reCAPTCHA:** A widget containing a checkbox labeled "Я не робот", the reCAPTCHA logo, and the text "reCAPTCHA" and "Конфиденциальность - Условия использования".
- Зарегистрироваться:** A blue button with white text.

Рисунок 5.1 – Форма регистрации



При вводе неunikального имени будет отображено предупреждение об этом и система попросит ввести имя данные заново. Пример ошибки подставлен на рисунке 5.2.

User exists!

Имя: user

Email: babanin.dima@gmail.com

Пароль: Пароль

Повторите пароль: Повторите пароль

☐ Я не робот

geCAPTCHA

Конфиденциальность - Условия использования

Зарегистрироваться

Рисунок 5.2 – Пример ошибки регистрации

При корректно вводе данных и нажатии на кнопку регистрации система перебросит вас на страницу входа в профиль.

– На введенный вами почтовый ящик система отправить письмо со ссылкой активации профиля для подтверждения почты. Пример сообщения изображен на рисунке 5.3.

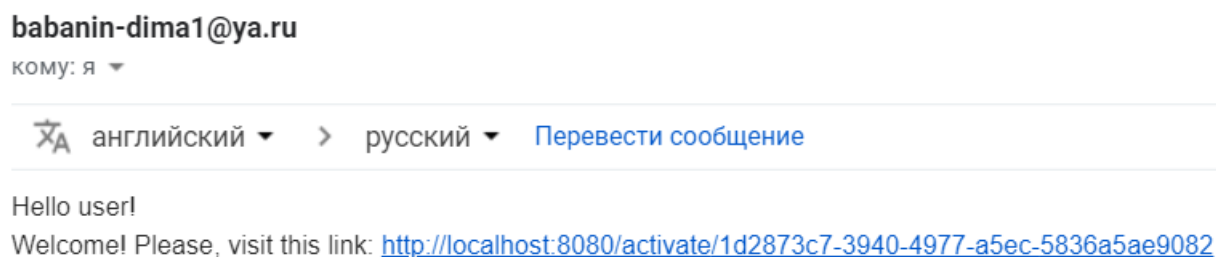
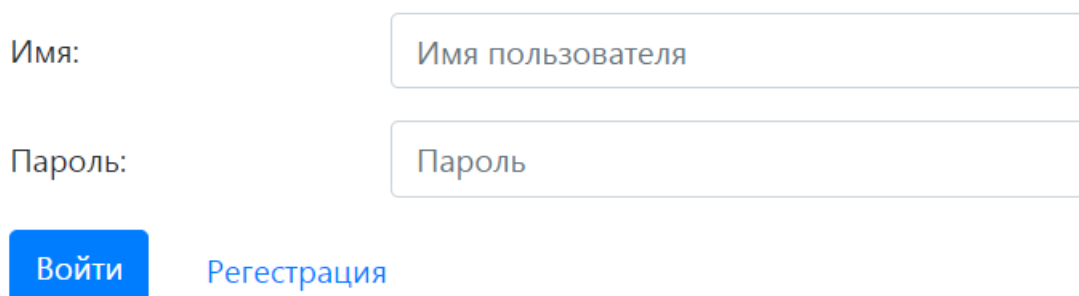


Рисунок 5.3 – Пример письма с ссылкой активации

### 5.2.2 Вход в профиль пользователя.

После прохождения регистрации для входа в профиль необходимо перейти на страницу входа и ввести соответствующие данные. Форма входа в профиль пользователя представлена на рисунке 5.4.



The image shows a login form with two input fields. The first field is labeled 'Имя:' and contains the placeholder text 'Имя пользователя'. The second field is labeled 'Пароль:' and contains the placeholder text 'Пароль'. Below the password field, there is a blue button labeled 'Войти' and a link labeled 'Регистрация'.

Рисунок 5.4 – Форма входа в профиль пользователя

### 5.3 Функционал администратора

Войдя в профиль с правами «Администратор», вы попадаете на главную страницу платформы. Главная страница представлена на рисунке 5.5.

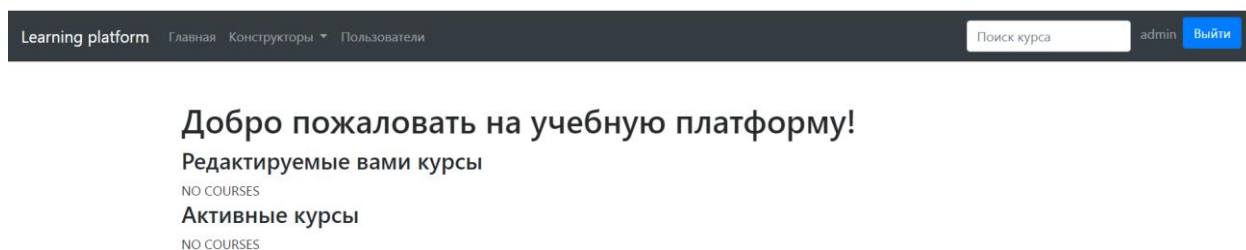


Рисунок 5.5 – Главная страница

Администратор способен просматривать и предоставлять повышенные права доступа другим пользователям системы, перейдя на страницу «Пользователи» в главном меню платформы. Страница «Пользователи» изображена на рисунке 5.6.

## Зарегистрированные пользователи

#	Наименование	Email	Роли	Подписки	Модерация	Действия
0	<a href="#">admin</a>	admin@admin.com	USER, TEACHER, ADMIN	0	0	Редактировать Удалить
1	<a href="#">teacher</a>	teacher@teacher.com	USER, TEACHER	0	0	Редактировать Удалить
2	<a href="#">user</a>	user@user.com	USER	0	0	Редактировать Удалить

Рисунок 5.6 – Страница «Пользователи»

На странице «Пользователи», нажав на кнопку «Редактировать», администратор способен давать или убирать дополнительные права доступа для пользователей. Пример редактирования пользователей представлен на рисунке 5.7.

## Зарегистрированные пользователи

#	Наименование	Email	Роли	Подписки	Модерация	Действия
0	<a href="#">admin</a>	admin@admin.com	USER, TEACHER, ADMIN	0	0	Редактировать Удалить
1	<a href="#">teacher</a>	teacher@teacher.com	USER, TEACHER	0	0	Редактировать Удалить
2	<a href="#">user</a>	user@user.com	<input checked="" type="checkbox"/> USER <input type="checkbox"/> TEACHER <input type="checkbox"/> ADMIN	0	0	Сохранить Удалить

Рисунок 5.7 – Пример редактирования пользователей

Администратор способен удалить профиль пользователя нажав на кнопку «Удалить».

Также администратор имеет функционал, как преподавателя, так и обычного пользователя, которые описаны дальше.

## 5.4 Функционал преподавателя

Главным функционалом платформы является «Конструктор курсов», благодаря которому можно разместить теоретический материал в удобном виде и доступного на любом устройстве с подключением к интернету. Благодаря библиотеки Bootstrap, контент курса будет отображаться корректно на устройствах разного размера экранов.

Конструирование курса разделено на 4 этапа:

– На первом этапе необходимо дать название курсу и можно добавить картинку и описание, которое будет, в краткой форме, отражаться смысл и назначение курса. У курса есть отметка «Активировать курс», которая означает, если отметка активна, то этот курс будет виден обычным пользователям на главной странице платформы и смогут иметь доступ к его содержанию. Форма для создания курса представлена на рисунке 5.8.

## Создание курса

Название курса

Картинка

Выберите файл

Файл не выбран

Описание

Создать курс

☐ Активировать курс

Рисунок 5.8 – Форма создания курса

– На втором этапе, после нажатия на кнопку «Создать курс», вы попадаете на страницу «Редактирования курса». Где можно создать темы курса и содержимое тем (рисунок 5.9). Для создания темы введите название темы и нажмите на кнопку «Добавит тему». Тема создана с введенным вами названием (рисунок 5.10).

## Редактирование курса

Другое

#	Название темы	Количество страниц	Действия
NO TOPICS			
1	<div>Название темы</div>		<div>Добавить тему</div>

Рисунок 5.9 – Страница редактирования курса

## Редактирование курса

Другое

#	Название темы	Количество страниц	Действия
1	Переменные в Java	0	<div>Редактировать</div> <div>Удалить</div>
2	<div>Название темы</div>		<div>Добавить тему</div>

Рисунок 5.10 – Пример добавления темы

– На третьем этапе создаться содержимое тем. Для добавления содержимого темы необходимо нажать на кнопку в столбце таблицы «Количество страниц» и в раскрывшейся форме заполнить название страницы и нажать кнопку «Добавить страницу» (рисунок 5.11).

## Редактирование курса

Другое

#	Название темы	Количество страниц	Действия
1	Переменные в Java	0	<div>Редактировать</div> <div>Удалить</div>

#

Название страницы

Действия

NO PAGES

1

Что это такое?

Добавить страницу

2

Название темы

Добавить тему

Рисунок 5.11 – Добавления содержимого темы

После нажатия на кнопку «Добавить страницу», конструктор перенаправляет вас на страницу с формой заполнения содержимого страницы темы, где присутствует редактор содержимого страницы, подсказки к нему и полезные ссылки (рисунок 5.12). Также предлагается загрузить содержимое страницы из файла формата txt или html. Если загружен файл, то содержимое редактора полностью игнорируется.

## Редактирование страницы

Номер страницы

1

Заголовок страницы

Что это такое?

Содержимое страницы

Получить подсказку

Разрешено использовать HTML теги и элементы **Bootstrap** **Подсветка синтаксиса** на странице

Или можно загрузить содержимое из файла (\*.txt, \*.html)

Выберите файл

Файл не выбран

Если загружен файл, то содержимое страницы будет браться только из файла

Предпросмотр

Рисунок 5.12 – Редактор содержимого страницы

– На четвертом, заключительном этапе, после заполнения содержимого страницы или загрузки содержимого из файла нажимаете на кнопку «Предпросмотр», которая вас перенаправляет на страницу предпросмотра содержимого страницы перед его сохранением, на которой можно сохранить внесенные изменения. На рисунках 5.13 – 5.15 можно увидит, какое содержимое можно добавлять на страницу.

## Что это такое?



Рисунок 5.13 – Пример содержимого страницы

You can use the mark tag to highlight text.

~~This line of text is meant to be treated as deleted text.~~

~~This line of text is meant to be treated as no longer accurate.~~

This line of text is meant to be treated as an addition to the document.

This line of text will render as underlined

This line of text is meant to be treated as fine print.

**This line rendered as bold text.**

*This line rendered as italicized text.*

Lorem ipsum dolor sit amet  
Consectetur adipiscing elit  
Integer molestie lorem at massa  
Facilisis in pretium nisl aliquet  
Nulla volutpat aliquam velit

- Phasellus iaculis neque
- Purus sodales ultricies
- Vestibulum laoreet porttitor sem
- Ac tristique libero volutpat at

Faucibus porta lacus fringilla vel  
Aenean sit amet erat nunc  
Eget porttitor lorem

Рисунок 5.14 – Пример содержимого страницы



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat a ante.  
— Someone famous in Source Title

Сохранить

Рисунок 5.15 – Пример содержимого страницы

После нажатия на кнопку «Сохранить», система перенаправляет вас в редактор курса для продолжения работы над курсом.

Для редактирования названия и описания курса необходимо нажать на кнопку «Другое», которая позволяет отобразить форму для редактирования (рисунок 5.16).

## Редактирование курса

Другое

Название курса

Java

Картинка

Выберите файл

Файл не выбран

Описание

Курс посвящен обучению одному из самых популярных объектно-ориентированных языков программирования - Java. В этом курсе вы увидите не только сухую теорию, но и практический и видео материалы для закрепления знаний.

Сохранить изменения

☐ Активировать курс

#	Название темы	Количество страниц	Действия
1	Переменные в Java	1	<div>Редактировать</div> <div>Удалить</div>
2	Thread в Java	0	<div>Редактировать</div> <div>Удалить</div>

#	Название страницы	Действия
NO PAGES		
1	<div>Название страницы</div>	<div>Добавить страницу</div>

Рисунок 5.16 – Форма редактирования курса

Если вы нажали кнопку «Добавить страницу», но не добавили содержимого страницы, то в таблицы редактора курса она будет отмечена жёлтым цветом, которое сигнализирует о том, что на странице отсутствует содержимое (рисунок 5.17).

2	Thread в Java	1	<div>Редактировать</div> <div>Удалить</div>
---	---------------	---	---

#	Название страницы	Действия
1	Что такое потоки и с чем их едят?	<div>Редактировать</div> <div>Удалить</div>
2	<div>Что такое потоки и с чем их едят?</div>	<div>Добавить страницу</div>

Рисунок 5.17 – Предупреждение об отсутствии содержимого страницы



На главной странице преподавателю доступен раздел, где отображаются все редактируемые им курсы (рисунок 5.18).

## Редактируемые вами курсы

### Java

Подписаться 0

Курс посвящен обучению одному из самых популярных объектно-ориентированных языков программирования - Java. В этом курсе вы узнаете не только сухую теорию, но и практический и видео материалы для закрепления знаний.

— Модераторы *teacher*

Редактировать

Удалить

Рисунок 5.18 – Раздел редактируемых курсов на главной странице платформы

В профиле преподавателя также имеется такой раздел в виде таблицы (рисунок 5.19).

### Модератор курсов

Название курса	Студенты	Модераторы	Действия	
Java	0	1	Редактировать	Удалить

Рисунок 5.19 – Раздел редактируемых курсов в профиле преподавателя

Также преподаватель имеет весь функционал пользователя.

## 5.5 Функционал пользователя

На главной странице платформы пользователю предоставляется возможность видеть все активные курсы (рисунок 5.20).

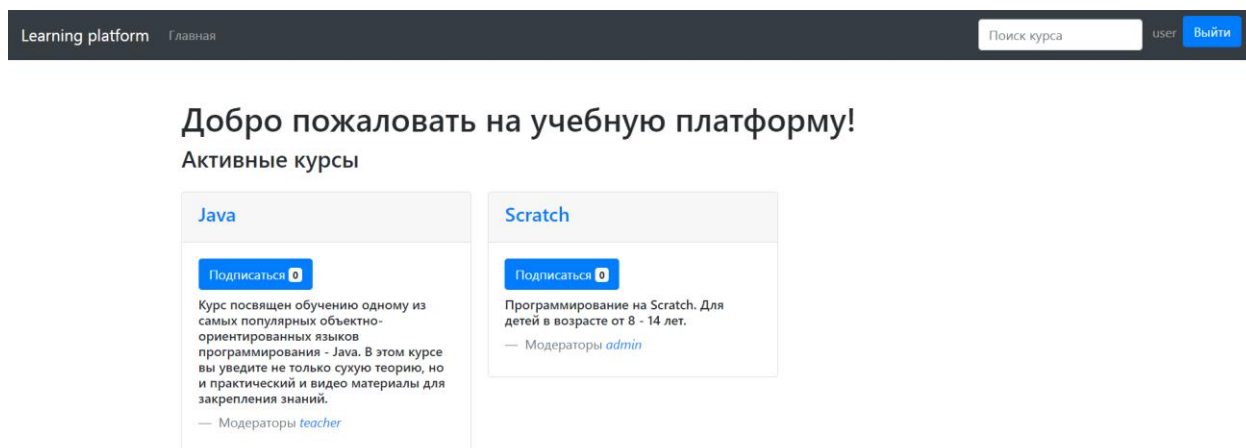


Рисунок 5.20 – Главная страница платформы

Выбрав курс он может перейти к обучению, нажав на название курса (рисунок 5.21). На странице курса она может переходить по темам и содержимому страниц благодаря меню в левой части экрана. В правой части экрана отображается содержимое страниц.

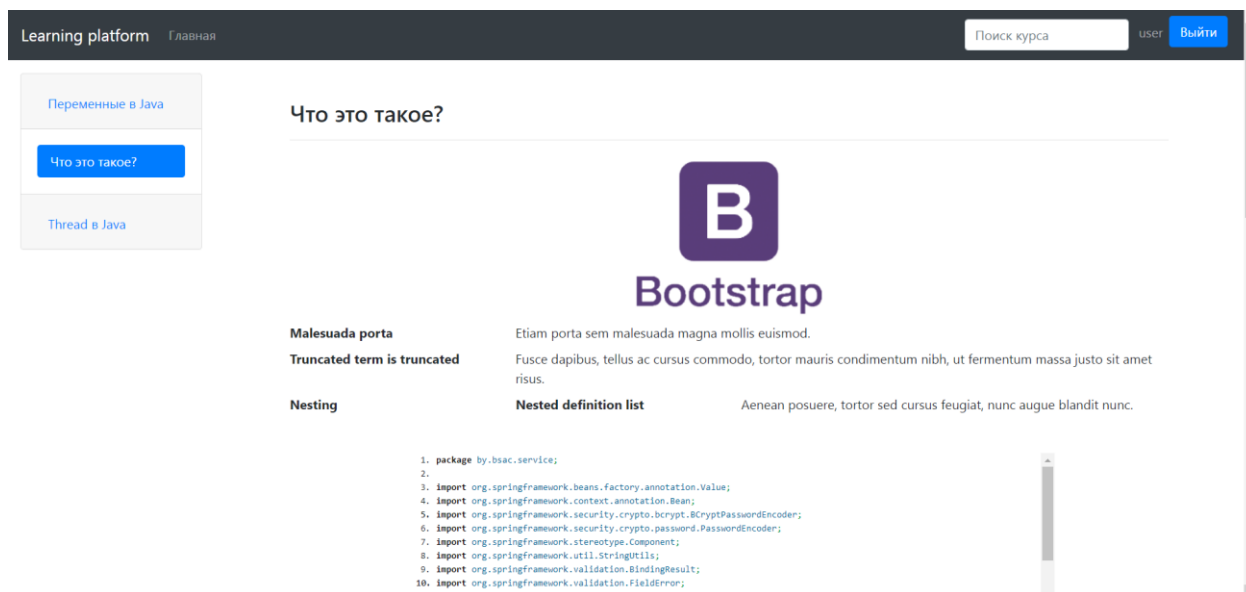


Рисунок 5.21 – Страница курса

Также на главной странице курса пользователь может подписаться (отписаться) на выбранный курс и видеть, сколько пользователей уже подписано (рисунок 5.22).

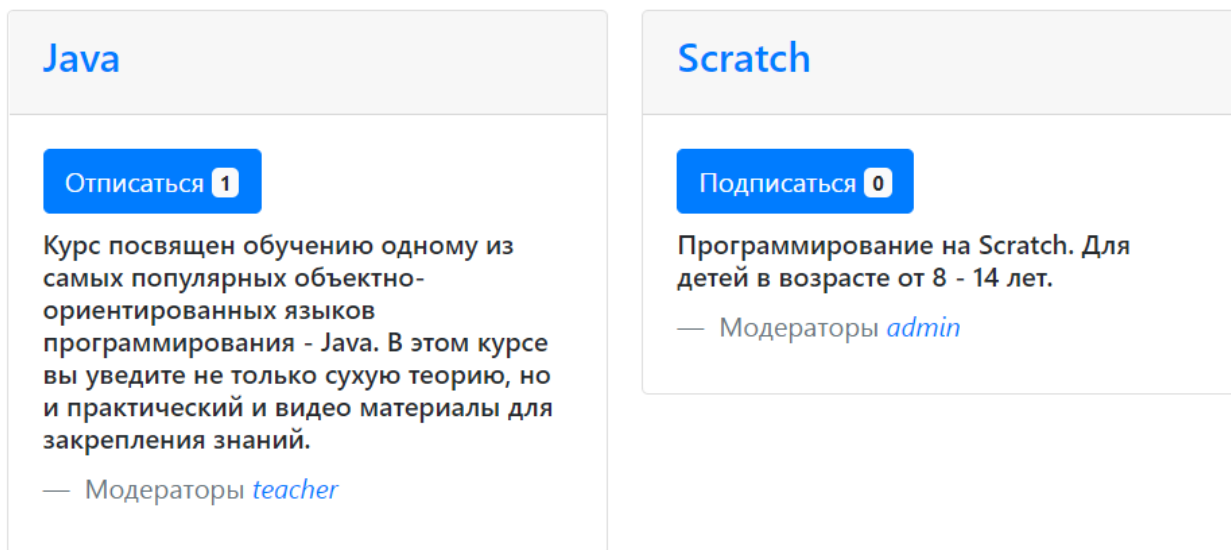


Рисунок 5.22 – Демонстрация подписки на курс

В профиле пользователя можно увидеть, на какие курсы вы уже подписаны, просмотреть свои данные и отредактировать их (рисунок 5.23).

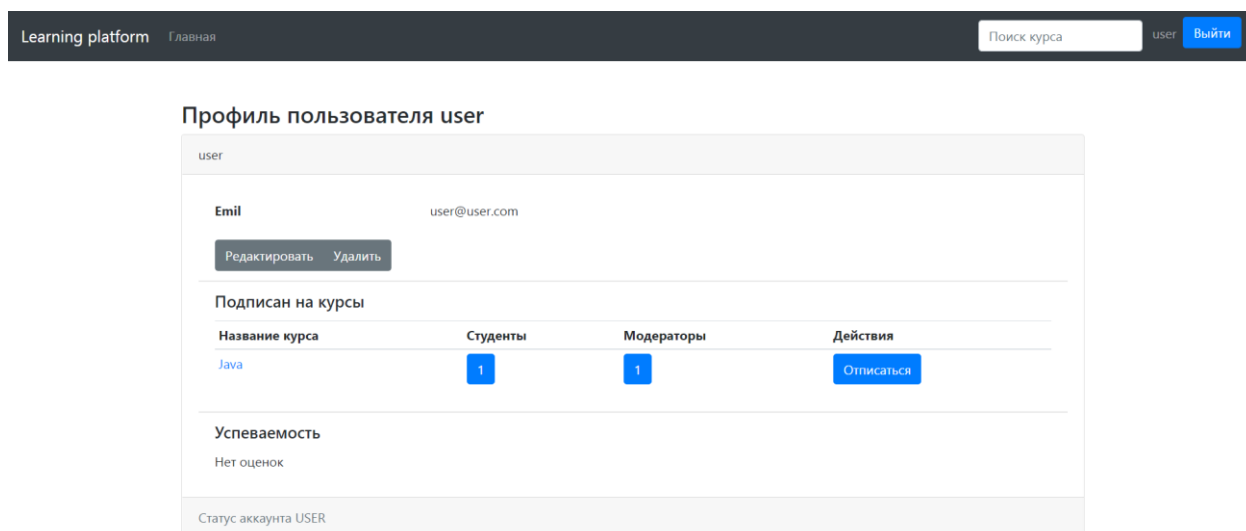


Рисунок 5.23 – Профиль пользователя

При нажатии на кнопку «Редактировать», вас перенаправят на страницу редактирования профиля (рисунок 5.24)

## Редактирование профиля

Наименование  
профиля:

user

Пароль:

Password

Email:

user@user.com

Сохранить

Рисунок 5.24 – Форма редактирования данных пользователя

В профиле пользователя в таблице подписок присутствуют кнопки, нажав на которые можно получить список студентов и модераторов курсов, на которые вы подписаны (рисунок 5.25).

### Зарегистрированные пользователи

#	Наименование	Email	Роли	Подписки	Модерация
2	user	user@user.com	USER	1	0

Рисунок 5.25 – Список студентов курса

Также можно просматривать профили других пользователей.

## **6 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ**

### **6.1 Краткая характеристика разрабатываемой системы**

Дипломный проект представляет собой проектирование и разработку платформы для обучения программированию на Java.

Платформа предназначена для повышения доступности и качества образовательной деятельности в рамках учебного заведения, а также призвана облегчить работу преподавательской деятельности.

Назначение разрабатываемого программного продукта заключается в передаче студентам теоретических знаний концепции объектно-ориентированного программирования, практических навыков по разработке объектно-ориентированных программ на одном из популярных языков программирования – Java.

### **6.2 Расчет себестоимости и цены программного продукта**

Себестоимость программного продукта представляет собой выраженные в денежной форме затраты организации- разработчика на создание программы и отражается в смете затрат, которая включает следующие статьи расходов:

- материалы и комплектующие изделия;
- затраты на оплату труда:
  - а) основная заработная плата научно-технического персонала;
  - б) дополнительная заработная плата научно-технического персонала;
- отчисления в Фонд социальной защиты населения;
- отчисления в Белгосстрах на страхование от несчастных случаев на производстве и профессиональных заболеваний;
- командировочные расходы;
- услуги сторонних организаций;
- прочие прямые расходы;
- накладные расходы.

Полная себестоимость разработки проекта рассчитывается по формуле 6.1:

$$C_{\pi} = MЗ + ЗОТ + ФСЗН + БГС + ПР + НР + КР + УСО, \quad (6.1)$$

где МЗ – материалы и комплектующие изделия;

ЗОТ – затраты на оплату труда;

ФСЗН – отчисления в Фонд социальной защиты населения;

БГС – отчисления в Белгосстрах на страхование от несчастных случаев на производстве и профессиональных заболеваний;

ПР – прочие прямые расходы;

НР – накладные расходы.

КР – командировочные расходы;

УСО – услуги сторонних организаций;

### 6.2.1 Расчет затрат по статье «Затраты на оплату труда».

В статью «Затраты на оплату труда» включаются основная и дополнительная заработная плата всех работников, непосредственно занятых выполнением разработки.

Произведем расчет основной заработной платы по формуле 6.2:

$$З_o = \sum_{i=1}^n T_{di} \cdot \Phi_p \cdot K_{пр}, \quad (6.2)$$

где  $T_{di}$  – дневная тарифная ставка  $i$ -го исполнителя (денежные единицы);

$\Phi_p$  – плановый фонд рабочего времени (т.е. время, в течение которого работники  $i$ -ой категории принимали участие в разработке), дней;

$K_{пр}$  – коэффициент премирования за выполнение плановых показателей (1,2 – 1,4).

Дневная тарифная ставка  $i$ -го исполнителя рассчитывается следующим образом:

$$T_{di} = T_{mi} / \Phi_p, \quad (6.3)$$

где  $T_{mi}$  – месячная тарифная ставка  $i$ -го исполнителя, руб.;

$\Phi_p$  – среднемесячный фонд рабочего времени, дни.

В соответствии с Постановлением Минтруда и соцзащиты Республики Беларусь от 22.10.2018 № 77 «Об установлении расчетной нормы рабочего времени на 2019 год» Расчетная норма рабочего времени в днях на 2019 г. при пятидневной рабочей неделе составляет 252 дня. Тогда среднемесячный фонд рабочего времени составляет:

$$\Phi_p = 252 / 12 = 21 \text{ дней}$$

Месячная тарифная ставка каждого исполнителя ( $T_m$ ) определяется путем умножения действующей месячной тарифной ставки первого разряда ( $T_{m1p}$ ) (на

предприятию установлен повышающий коэффициент – 2 к тарифной ставке первого разряда, действующей в Республике Беларусь для оплаты труда работников бюджетных организаций и иных организаций, работники которых приравнены по оплате труда к бюджетникам. С 01.05.2019 эта величина составляет 36,4 руб.) на тарифный коэффициент ( $T_k$ ), соответствующий установленному тарифному разряду. В соответствии с Единой тарифной сеткой работников Республики Беларусь исполнитель данного проекта по уровню квалификации, теоретических и практических знаний, выполняемых обязанностей тарифицируется по 10 разряду, которому соответствует тарифный коэффициент ( $T_k$ ) – 2,48

$$T_m = T_{m1p} \cdot T_k \quad (6.4)$$

Произведем расчет необходимых показателей по указанным формулам.

Месячная тарифная ставка 1-ого исполнителя:

$$T_{m1} = T_{m1p} \cdot T_k = 72,80 \cdot 2,48 = 180,54 \text{ (руб.)}$$

Дневная тарифная ставка 1-ого исполнителя:

$$T_{d1} = T_{m1} / \Phi_p = 180,54 / 21 = 8,60 \text{ (руб.)}$$

Оплата за отработанное время одного работника:

$$З_{п1} = T_{d1} \cdot \Phi_p = 8,60 \cdot 15 = 129,00 \text{ (руб.)}$$

Основная заработная плата одного работника с учетом премии:

$$З_{п_{полн1}} = З_{п1} \cdot K_{пр} = 129,00 \cdot 1,2 = 154,80 \text{ (руб.)}$$

В связи с участием только одного работника, общая основная заработная плата с учетом премии будет равна  $З_{п_{полн1}}$ , а именно:

$$З_o = 154,80 \cdot 1 = 154,80 \text{ (руб.)}$$

Результаты расчетов представим в виде таблицы 6.1.

Таблица 6.1 – Расчет основной заработной платы научно-производственного персонала

Категория работников	Кол-во, чел	Тарифный коэф-т	Месячная тарифная ставка соотв. разряда, руб.	Дневная тарифная ставка, руб.	Плановый фонд рабочего времени, дней	Оплата за отработанное время одного работника, руб.	Основная заработная плата одного работника с учетом премии, руб.	Основная заработная плата с учетом премии, руб.
Инженер программист	1	2,48	180,54	8,60	15	129,00	154,80	154,80
Итого								154,80

Дополнительная заработная плата научно-технического персонала включает выплаты, предусмотренные действующим законодательством за непроработанное время (оплата очередных и дополнительных отпусков, выплата за выслугу лет и др.) определяется по формуле:

Дополнительная заработная плата рассчитывается по формуле 6.5:

$$З_д = З_о \cdot Н_д / 100\%, \quad (6.5)$$

где  $Н_д$  – норматив дополнительной заработной платы, %.

Рассчитаем дополнительную заработную плату:

$$З_д = З_о \cdot Н_д / 100\% = 129,00 \cdot 10\% / 100\% = 12,90 \text{ (руб.)}$$

В целом затраты на оплату труда составят:

$$З_{ОТ} = З_о + З_д = 154,80 + 12,90 = 167,70 \text{ (руб.)}$$

**6.2.2** Расчет затрат по статье «Отчисления в фонд социальной защиты населения на социальные нужды».

Размер затрат определяется в процентах от суммы основной и дополнительной заработной платы всех категорий работников, причастных к выполнению данной разработки:

$$ОСН = З_{ОТ} \cdot S_{осн} / 100\% \quad (6.6)$$



$S_{\text{осн}}$  – ставка отчислений на социальные нужды, % (на сегодняшний день по РБ установлена ставка 34%)

Рассчитаем ОСН:

$$\text{ОСН} = \text{ЗОТ} \cdot S_{\text{осн}} / 100\% = 167,70 \cdot 34\% / 100\% = 57,02 \text{ (руб.)}$$

**6.2.3** Расчет затрат по статье «Отчисления в Белгосстрах на страхование от несчастных случаев на производстве и профессиональных заболеваний».

Отчисления в Белгосстрах на страхование от несчастных случаев на производстве и профессиональных заболеваний рассчитывается по следующей формуле:

$$\text{БГС} = \text{ЗОТ} \cdot \text{Н}_{\text{БГС}} / 100\%, \quad (5.7)$$

$\text{Н}_{\text{БГС}}$  – норматив отчислений в Белгосстрах установленный для организации, % (для организаций связи норматив равен 0,6%).

Рассчитаем БГС:

$$\text{БГС} = \text{ЗОТ} \cdot \text{Н}_{\text{БГС}} / 100\% = 167,70 \cdot 0,6\% / 100\% = 1,01 \text{ (руб.)}$$

Все остальные затраты включаются в себестоимость единицы продукции косвенным путем в процентах от основной заработной платы.

**6.2.4** Расчет затрат по статье «Прочие расходы».

Данная статья затрат включает в себя затраты на приобретение, перевод специальной научно-технической информации, использование технических средств связи и т.д.

$$\text{ПР} = \text{З}_0 \cdot \text{Н}_{\text{пр}} / 100\%, \quad (6.8)$$

где  $\text{Н}_{\text{пр}}$  – норматив прочих расходов для расчета себестоимости и отпускной цены НИОКР, %

Рассчитаем ПР:

$$\text{ПР} = Z_o \cdot H_{\text{пр}} / 100\% = 154,80 \cdot 20\% / 100\% = 30,96 \text{ (руб.)}$$

#### **6.2.5 Расчет затрат по статье «Накладные расходы».**

Данная статья затрат в равной степени относится ко всем выполняемым НИОКР и включает в себя затраты на содержание и текущий ремонт зданий, сооружений, оборудования, инвентаря, расходы по охране труда и т.д.

$$НР = Z_o \cdot H_{\text{нр}} / 100\%, \quad (6.9)$$

где  $H_{\text{нр}}$  – норматив накладных расходов для расчета себестоимости и отпускной цены НИОКР, %

Рассчитаем НР:

$$НР = Z_o \cdot H_{\text{нр}} / 100\% = 154,80 \cdot 100\% / 100\% = 154,80 \text{ (руб.)}$$

#### **6.2.6 Расчет затрат по статье «Командировочные расходы».**

Командировочные расходы рассчитываются по смете, или по формуле:

$$КР = Z_o \cdot H_{\text{кр}} / 100\%, \quad (6.10)$$

где  $H_{\text{кр}}$  – норматив командировочных расходов, %

Рассчитаем КР:

$$КР = Z_o \cdot H_{\text{кр}} / 100\% = 154,80 \cdot 10\% / 100\% = 15,48 \text{ (руб.)}$$

#### **6.2.7 Расчет затрат по статье «Услуги сторонних организаций».**

Услуги сторонних организаций (УСО) рассчитываются исходя их сметы затрат.

В нашем случае УСО будет равно 0.

#### **6.2.8 Расчет затрат по статье «Материалы и комплектующие изделия».**

По статье «Материалы и комплектующие изделия» отражаются расходы на внешние запоминающие устройства, бумагу, картриджи и другие материалы,

необходимые для разработки данного проекта. Сумма затрат на расходные материалы определяются по нормативу в процентах к фонду заработной платы по формуле:

$$КР = З_о \cdot Н_{кр} / 100\%, \quad (6.11)$$

где  $H_{МЗ}$  – норма расхода материалов от основной заработной платы, %

$$М = З_о \cdot Н_{кр} / 100\% = 154,80 \cdot 5\% / 100\% = 7,74 \text{ (руб.)}$$

#### **6.2.9 Итоговый расчет полной себестоимости разработки проекта.**

Таким образом, рассчитаем полную себестоимость разработки проекта:

$$\begin{aligned} C_{\pi} = МЗ + ЗОТ + ОСН + БГС + ПР + НР + КР = 7,74 + 167,70 + 57,02 + 1,01 + \\ + 30,96 + 154,80 + 15,48 = 434,71 \text{ (руб.)} \end{aligned}$$

### **6.3 Стоимость коммерческой реализации проекта**

Стоимость коммерческой реализации проекта рассчитывается по формуле 6.12:

$$Ц_{отп} = C_{\pi} + П + НДС, \quad (6.12)$$

где  $П$  – плановая прибыль, руб.

$НДС$  – налог на добавленную стоимость, руб.

**6.3.1** Прибыль необходимо определить в соответствии с запланированной нормой рентабельности по формуле:

$$П = C_{\pi} \cdot P_{\pi} / 100 = 434,71 \cdot 20\% / 100\% = 86,94 \text{ (руб.)}$$

**6.3.2** Налог на добавленную стоимость, включаемый в цену, определяется по формуле:

$$НДС = (C_{\pi} + П) \cdot S_{ндс} / 100 = (434,71 + 86,94) \cdot 20\% / 100\% = 104,33 \text{ (руб.)}$$

Результаты расчетов представим в виде таблицы 6.2.

Таблица 6.2 – Расчет себестоимости и отпускной цены технической разработки

Наименование статьи затрат	Условное обозначение	Значение, руб.
Материалы и комплектующие изделия	МЗ	7,74
Затраты на оплату труда	ЗОТ	167,70
Отчисления в фонд социальной защиты населения	ФСЗН	57,02
Отчисления в Белгосстрах	БГС	1,01
Командировочные расходы	КР	15,48
Услуги сторонних организаций	УСО	0
Прочие прямые расходы	ПР	30,96
Накладные расходы	НР	154,80
<b>Полная себестоимость</b>	<b>С<sub>п</sub></b>	<b>434,71</b>
Плановая прибыль	П	81,76
Налог на добавленную стоимость	НДС	98,11
<b>Отпускная цена</b>	<b>Ц<sub>отп</sub></b>	<b>614,58</b>

В ходе расчетов было получено, что полная себестоимость разработки системы составляет 434,71 рубля, конечная отпускная цена реализации потребителю – 614,58 рубля.

Данная сумма затрат оправдана, т. к. разработка позволяет повысить доступность и качество образовательной деятельности, а также позволяет облегчить работу преподавательской деятельности за счет автоматизации некоторых процессов.

## **7. ОХРАНА ТРУДА**

### **7.1 Организация системы управления охраной труда на предприятии**

Охрана труда является неотъемлемой частью профессиональной и производственной деятельности, направленной на сохранение трудоспособности работника и других приравненных к нему лиц, представляет собой систему правовых, социально-экономических, организационно-технических, санитарно-гигиенических, лечебно-профилактических, реабилитационных и других мер.

Основной целью системы охраны труда на каждом производственном предприятии является создание благоприятных условий для обеспечения безопасности, здоровья и эффективности работников при выполнении ими своих профессиональных обязанностей.

Охрана труда обуславливает выполнение двух основных задач: создание машин и инструментов, не представляющих опасности для человека, и разработка специальных защитных мер по обеспечению безопасности человека в рабочем процессе и обучение работников безопасным методам работы и применению защитных мер; создание безопасных условий труда.

Система управления охраной труда (СУОТ) представляет собой единый комплекс взаимосвязанных и взаимодействующих элементов, определяющих политику и цели в области охраны труда и промышленной безопасности на производстве у работодателя и порядок достижения этих целей. В соответствии со статьей 17 Закона Республики Беларусь "Об охране труда" каждый наниматель обязан разработать и внедрить на своем предприятии систему управления охраной труда, независимо от вида деятельности.

Основной целью СУОТ, действующей в секторе по разработке и сопровождения ПО технологических систем информационно-расчетного центра РУП «Белтелеком», является создание безопасных условий труда для снижения риска нанесения вреда здоровью и улучшения существующих условий труда.

Организацией функционирования системы управления охраны труда на предприятии занимаются следующие должностные лица:

- Генеральный директор: обеспечивает ресурсами для осуществления деятельности по обучению работников, повышению их компетентности и осведомленности в области ОТ, принимает решения о проведении аттестации работников.

- Ведущий инженер по ОТ: определение потребности в обучении в области охраны труда и СУОТ, доведение до работников политики и целей в области ОТ, проведение вводного инструктажа по ОТ,

– Инженер по ОТ: проведение первичного на рабочем месте, повторного, внепланового и целевого инструктажей по ОТ, организацию и контроль проведения стажировки работников

– Начальник отдела капитального строительства и снабжения: благоустройство и санитарное состояние помещений.

– Начальники отделов и секторов: содержание в исправности оборудования в кабинетах.

В соответствии с Законом Республики Беларусь «Об охране труда» работодатель обязан осуществлять обучение, стажировку, инструктаж и проверку знаний работающих по вопросам охраны труда, обеспечивать безопасность при эксплуатации территории, производственных зданий (помещений), сооружений, оборудования, технологических процессов и применяемых в производстве материалов, химических веществ, а также эффективную эксплуатацию средств индивидуальной и коллективной защиты, осуществлять обязательное страхование работающих от несчастных случаев на производстве и профессиональных заболеваний в соответствии с законодательством.

Согласно статье 49 ТК РБ работодатель обязан не допускать к работе в соответствующий день при нарушении следующих пунктов: появившегося на работе в состоянии алкогольного, наркотического или токсического опьянения, не прошедшего инструктаж, стажировку и проверку знаний по вопросам охраны труда, не использующего средства индивидуальной защиты, непосредственно обеспечивающие безопасность труда, не прошедшего медицинский осмотр, освидетельствование на предмет нахождения в состоянии алкогольного, наркотического или токсического опьянения в случаях и порядке, предусмотренных законодательством.

В соответствии со статьей 42 ТК РБ однократного грубого нарушения требований по охране труда, повлекшего увечье или смерть других работников – расторжение трудового договора по инициативе нанимателя.

## **7.2 Идентификация и анализ вредных и опасных факторов**

Работа с программным обеспечением, которое разрабатывается в данной дипломной работе, связана с вычислительной техникой, поэтому перечислим основные опасные и вредные производственные факторы согласно ГОСТ 12.0.003-74 [3], которые могут влиять на работу пользователя с данным программным обеспечением: физические и психофизиологические опасные и вредные производственные факторы:

К физическим факторам, которые влияют на пользователя данного программного обеспечения, относятся:

- повышенный уровень шума на рабочем месте;
- повышенное значение напряжения в электрической цепи, замыкание которой может произойти через тело человека;
- повышенная или пониженная температура воздуха рабочей зоны;
- отсутствие или недостаток естественного света;
- недостаточная освещенность рабочей зоны.

К психофизиологическим факторам, относятся:

- физические перегрузки;
- нервно-психические перегрузки.

Программное обеспечение, которое разрабатывается в данной дипломной работе предназначено для обучения программированию, поэтому это ПО будет применяться в лабораториях и кабинетах, в которых установлены ЭВМ. При работе с ЭВМ можно столкнуться с различными опасными и вредными производственными факторами, такими как повышенные значения напряжения в электрической цепи, недостаточная освещенность рабочей зоны пользователя, повышенный уровень статического напряжения на контроллерах, повышенный уровень электромагнитных излучений и т.д.

Также в работе с компьютером и программным обеспечением, монотонность труда играет большую роль, негативно сказывается на производительности и способствует развитию невротических расстройств, определяемых термином «монотонность», и ведет к профессиональной заболеваемости. При работе с нашим программным обеспечением на пользователя оказывают влияние и другие психофизиологически опасные и вредные производственные факторы, особенно нарушения зрения, головные боли, мышечные боли в области спины, а также повышенное чувство усталости и нервное напряжение. Поэтому требования к рабочим местам и условиям труда должны соблюдаться в соответствии с ГОСТ 22269-76. Например, основными требованиями к рабочему месту являются: рациональная высота рабочей поверхности стола и кресла, наличие спинки и подлокотников, удобные углы плечевого и локтевого суставов, удобный наклон головы, наличие необходимого пространства и подставки для ног. А к рабочей среде: рациональное расположение документов на столе, видеотерминале, клавиатуре, правильный угол обзора экрана и расстояние между глазами и монитором работника.

Компьютерная, копировальная и множительная техника в случае неправильного теплового режима в лабораториях приводит к повышению температуры и снижению влажности воздуха на рабочих местах. Оборудование, установленное в кабинетах, также

является источником шума, а именно жесткий диск, охлаждающие вентиляторы, принтеры и т.д. Это приводит к снижению производительности и увеличению усталости и могут вызывать стресс.

Освещение рабочего места является важнейшим фактором в создании нормальных условий труда. На практике существует потребность как в естественном, так и в искусственном освещении. Если глаза людей не освещены должным образом, они быстро устают, вызывая физическую усталость.

Рабочее пространство пользователя должно быть организовано в соответствии со всеми эргономическими принципами и эргономическими требованиями, а именно: между отдельными рабочими местами должен быть проход не менее одного метра, помещение должно регулярно проветриваться и не менее одного раза в смену, экран должен быть очищен от пыли. Положение пользователя должно быть настроено таким образом, чтобы исключить отражение света и света на экране дисплея, устранить напряжение мышц и обеспечить возможность регулировки угла наклона для оптимального угла обзора.

### **7.3 Организационные, технологические и иные решения по устранению опасных и вредных факторов**

При организации рабочего места необходимо соблюдать следующие принципы и рекомендации ГОСТ 22269-76: конструкция одного рабочего стола должна предусматривать две отдельные поверхности: первая горизонтальная для размещения ПЭВМ или ВДТ с бесступенчатой регулировкой высоты в диапазоне 520-760 мм и вторая для клавиатуры с бесступенчатой регулировкой высоты и наклона от 0 до 15° с надежной фиксацией в оптимальное рабочее положение 12-15°, что позволяет поддерживать правильное положение головы. Ширина поверхности для ПЭВМ, ВДТ и клавиатуры должна быть не менее 750 мм, а глубина - не менее 550 мм. При использовании принтера в работе необходимо увеличить ширину области до 1200 мм. Высота края стола, обращенного к ПЭВМ и ВДТ, и высота пространства для ног должны соответствовать высоте оператора в обуви. Рабочая зона оборудована креслом, основные размеры которого должны соответствовать высоте пользователя. Поверхность кресла должна быть легко дезинфицируемой. Не допускается вместо стульев использование кубов, табуреток, скамеек без опоры для спины.

В помещении, в котором мы работаем с нашим программным обеспечением, температура, относительная влажность и скорость движения воздуха должны соответствовать действующим санитарным нормам микроклимата производственных



помещений СанПиН №11-13-94. Например, оптимальными стандартами микроклимата для помещений в которых есть компьютерное оборудование в теплое время года являются: температура воздуха 23-25 °С, относительная влажность 40-60%, скорость воздуха 0,1 м/с. Для увеличения влажности воздуха в помещении необходимо использовать увлажнители, наполняемые ежедневно дистиллированной или кипяченой питьевой водой. Помещение должно периодически проветриваться для улучшения качества воздуха.

Во время основной работы уровень шума в помещении не должен превышать 60 дБА. На рабочих станциях в помещениях с шумными вычислительными устройствами (большие серверные шкафы с оборудованием, принтеры и т.д.) уровень шума не должен превышать 75 дБА. Все устройства уровень шума которых превышает нормализованный уровень, должны быть установлены снаружи. Снизить уровень шума в помещениях с мониторами и ПК можно с помощью звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в диапазоне частот 63-8000 Гц для отделки помещений, подтвержденными специальными акустическими расчетами. Для дополнительного звукопоглощения можно использовать шторы из плотной ткани, гармонично сочетающиеся с цветовой составляющей стен и висеть в складке на расстоянии 15-20 см от забора. Ширина занавеса должна быть в два раза шире, чем ширина окна.

Рабочая зона оператора должна быть расположена таким образом, чтобы со стороны, в основном с левой стороны, падало естественное освещение. При проектировании рабочих мест следует учитывать расстояние между рабочими станциями с видеомониторами, которое должно быть не менее 2,0 м, и расстояние между боковыми поверхностями видеомониторов, которое должно быть не менее 1,2 м. Рабочее место должно быть расположено таким образом, чтобы естественное освещение исходило сбоку, желательно с левой стороны. Оконные проемы должны быть оснащены регулируемыми устройствами, такими как жалюзи, занавески, козырьки и т.д. Экран видеомонитора должен быть 600-700 мм, но не ближе 500 мм, включая буквенно-цифровые символы и символы.

Для улучшения освещения рабочего места пользователя помещение должно быть оснащено естественным и искусственным освещением. Опишите основные требования к нашему рабочему месту. Естественное освещение должно осуществляться через световые проемы, ориентированные преимущественно на север и северо-восток для обеспечения коэффициента естественного освещения (КЕО) не менее 1,2% в районах со стабильным снежным покровом и не менее 1,5% в остальной части территории. Площадь каждого

рабочего места должна составлять не менее  $6,0 \text{ м}^2$  и не менее  $20 \text{ м}^3$ . Диффузные отражающие материалы с коэффициентом отражения для потолка должны использоваться для внутренней отделки помещения - 0,7 - 0,8; для стен - 0,5 - 0,6; для полов - 0,3 - 0,5. Искусственное освещение помещения должно обеспечиваться общей системой равномерного освещения. Освещение поверхности стола в области рабочих документов должно составлять 300-500 люкс. Для освещения документов могут быть установлены локальные светильники.

При выборе дисплеев или ВДТ необходимо соблюдать следующие правила. Эргономические визуальные параметры ВДТ являются параметрами безопасности, и, если они не будут выбраны правильно, здоровье пользователя ухудшится. ТДС должен быть спроектирован таким образом, чтобы обеспечить вид спереди экрана путем поворота корпуса по горизонтали вокруг вертикальной оси в диапазоне  $\pm 30^\circ$  градусов и по вертикали относительно горизонтальной оси в диапазоне  $\pm 30^\circ$  с учетом монтажного положения. Конструкция ВДТ должна быть такой, чтобы корпус был окрашен в мягкие, мягкие тона с рассеиванием света. ВДТ и корпус компьютера, клавиатура и другие компьютерные блоки, и устройства должны иметь матовую поверхность одного цвета с коэффициентом отражения от 0,4 до 0,6 и не должны содержать глянцевых частей, способных создавать светоотражающую способность. Не рекомендуется размещать элементы управления, маркировку, вспомогательные надписи или маркировку на передней части корпуса ТДС. При необходимости элементы управления на передней панели должны быть закрыты крышкой или утоплены в шкаф.

Уровень электромагнитного излучения мониторов, признанных безопасными для здоровья, регулируется действующими санитарными нормами СанПиН №11-13-94, согласно которым частотный диапазон электрического поля не должен превышать 2,5 В/м, а магнитного поля - 25 нТл. Частота регенерации в режиме положительного контраста должна быть не менее 60 Гц, а в режиме обработки текста - не менее 72 Гц. Встряхните элементы изображения не более чем на 0,1 мм.

Рекомендации по режиму работы и отдыха.

Для обеспечения оптимального режима труда и отдыха и профилактики профессиональных заболеваний необходимо рациональное сочетание умственной и физической активности и организации динамичного труда.

Необходимо планировать оптимальную рабочую нагрузку на человека с учетом его индивидуальных способностей и специфики данного программного обеспечения.

В течение рабочего дня должны быть предусмотрены запланированные перерывы. Рассчитайте их общую продолжительность. Общий процент перерывов состоит из

процентов по каждому фактору. Поскольку физические усилия незначительны, предполагается, что этот фактор представляет собой процент перерывов, равный 1%. Среднее нервное напряжение составляет еще 2% от общего числа простоев в работе. Умеренные темпы работы, ограниченность рабочего места и монотонность в среднем обеспечивают еще 4% от общего количества перерывов в работе. Другие факторы, такие как температура, влажность, загрязнение воздуха, производственный шум, вибрации и освещение, находятся в допустимых пределах. Поэтому они не будут влиять на общий процент перебоев в поставках. Общий процент перебоев в поставках составляет 7%. В течение восьмичасового рабочего дня общая продолжительность перерыва должна составлять около 35 минут.

Частота перерывов зависит как от характера работы, так и от индивидуальных особенностей пользователя. Однако в конце первого полугодия и второй половины смены должны быть сделаны дополнительные перерывы. Продолжительность перерыва должна выбираться субъективным пользователем, но не должна превышать 10 минут.

Пользователь, работающий с компьютером, должен делать перерывы в физической культуре во время смены, основанные на субъективных чувствах. Упражнения по расслаблению следует выполнять в сочетании со статическими или нестатическими упражнениями на растяжение мышц. Также следует широко использовать самообучение работающих и разгруженных мышц.

Как упоминалось ранее, изоляция линейных объектов с помощью персональных компьютеров требует четко скоординированных действий, визуального напряжения в микропаузах, поэтому рекомендуется выполнять различные расслабляющие упражнения для глаз.

Для борьбы с монотонной работой желательно менять виды деятельности, совмещать занятия и использовать изменение темпа. Поэтому выбор оптимальных схем работы и отдыха требует комплексного социально-экономического подхода. Целью такого подхода является полная и всесторонняя оценка его оптимизации с точки зрения личных и общественных интересов, продуктивных интересов и физиологического потенциала личности.

#### **7.4 Ионизирующие излучения**

Согласно Закону Республики Беларусь от 11.07.2017 № 122-З «О радиационной безопасности населения» ионизирующими излучениями называются такие виды лучистой энергии, которые, попадая в определенные среды или проникая через них, производят в

них ионизацию. Такими свойствами обладают радиоактивные излучения, излучения высоких энергий, рентгеновские лучи и др.

Источник ионизирующего излучения (ИИИ) – радиоактивное вещество или устройство, испускающее или способное испускать ионизирующее излучение сверх уровней, установленных нормативными правовыми актами, в т.ч. ТНПА.

Для категорий облучения населения и профессионального облучения в ситуациях планируемого облучения устанавливаются следующие основные пределы доз облучения на территории Республики Беларусь в результате воздействия источников ионизирующего излучения:

- при облучении населения предел средней годовой эффективной дозы равен 0,001 зиверта, допустимо облучение в размере годовой эффективной дозы до 0,005 зиверта при условии, что средняя годовая эффективная доза, исчисленная за пять последовательных лет, включая год, в котором предел средней годовой эффективной дозы был превышен, не превысит 0,001 зиверта;

- при профессиональном облучении предел средней годовой эффективной дозы равен 0,02 зиверта, допустимо облучение в размере годовой эффективной дозы до 0,05 зиверта при условии, что средняя годовая эффективная доза, исчисленная за пять последовательных лет, включая год, в котором предел средней годовой эффективной дозы был превышен, не превысит 0,02 зиверта;

- при профессиональном облучении женщины, уведомившей о беременности, условия труда беременной женщины должны обеспечить минимально возможную эффективную дозу облучения ребенка, не превышающую 0,001 зиверта в течение, по меньшей мере, оставшейся части беременности;

- при профессиональном облучении учащихся и студентов в возрасте от 16 до 18 лет, которые проходят обучение в целях последующего получения работы, связанной с ионизирующим излучением, и для облучения учащихся и студентов в возрасте от 16 до 18 лет, которые пользуются источниками ионизирующего излучения в процессе обучения, предел средней годовой эффективной дозы равен 0,006 зиверта.

Различают источники ионизирующего излучения:

- непосредственно ионизирующее;
- косвенно ионизирующее.

Непосредственно ионизирующее излучение состоит из заряженных частиц, кинетическая энергия которых достаточная для ионизации при столкновении с атомами вещества ( $\alpha$  и  $\beta$  – излучение радионуклидов, протонное излучение ускорителей и т.п.).

Косвенно ионизирующее излучение состоит из незаряженных (нейтральных) частиц, взаимодействие которых со средой приводит к возникновению заряженных частиц, способных непосредственно вызывать ионизацию (нейтронное излучение,  $\gamma$ -излучение).

Существенную часть облучения население получает от естественных источников радиации, т.е. из космоса и от радиоактивных веществ, находящихся в земной коре. Например, радиоактивный газ радон постоянно выделяется на поверхность и проникает в производственные и жилые помещения.

Часть помещений расположенных в подвальных, цокольных или первых этажах зданий согласно СНиП 2.6.2.11-4-2005 "Гигиенические требования по ограничению облучения населения за счет природных источников ионизирующего излучения" могут подвергаться обследованию радиационной обстановки, в виду возможного облучение работников природными источниками ионизирующего излучения дозой выше 1 мЗв/год.

Если среднегодовое значение эквивалентной равновесной объемной активности (ЭРОА) дочерних продуктов изотопов радона и торона в воздухе зданий (части помещений), сдающихся в эксплуатацию после окончания строительства (реконструкции, капитального ремонта), превышает 100 Бк/м<sup>3</sup>, то предусматриваются мероприятия по ее снижению (уменьшение поступления дочерних продуктов изотопов радона и торона в воздух помещений за счет дополнительной изоляции почвы под зданием, создания разрежения в пространстве под зданием, повышения кратности воздухообмена помещений и подпольного пространства здания и др.).

При невозможности в результате экономически обоснованных защитных мероприятий уменьшить ЭРОА дочерних продуктов изотопов радона и торона в воздухе до значений ниже 100 Бк/м<sup>3</sup>, рассматривается вопрос о перепрофилировании здания или части помещений здания.

Если мощность дозы гамма-излучения в эксплуатируемых жилых и общественных зданиях превышает мощность дозы на открытой местности более чем на 0,2 мкЗв/ч, то предусматриваются мероприятия по ее снижению. При невозможности снизить ее до указанного уровня без нарушения целостности здания решается вопрос о перепрофилировании здания или части помещений здания.

Также существует естественный радиационный фон, устоявшееся значение которого для города Минска составляет 0.1 мкЗв/ч.

Намного меньшее количество радиоактивного облучения проходит через человеческий организм от всевозможных технических устройств или систем.

Единственным искусственным источником при работе с нашей программой является устаревший тип мониторов – ЭЛТ. Такие мониторы при работе вырабатывают малое рентгеновское излучение. Поэтому раньше существовал очень жесткий регламент работы с таковыми мониторами.

## **7.5 Выводы и предложения**

Секторе по разработке и сопровождения ПО технологических систем информационно-расчетного центра РУП «Белтелеком», в силу своих производственных процессов имеет опасные и вредные факторы, влияющие на работников, и не имеет таковых по отношению к природе. Производственные помещения относятся к классу помещений без повышенной опасности.

Для профилактики и предотвращения производственного травматизма инженером по технике безопасности необходимо производить квалифицированные вводные инструктажи для новых сотрудников и практикантов. На предприятии созданы все условия для безопасного труда: планоно проводятся инструктажи, аттестации, отработка чрезвычайных ситуаций, имеется необходимое материально снабжение.

Так как все рабочие места для работы с разрабатываемым программным обеспечением оснащены современным типом жидкокристаллических мониторов, то можно сделать вывод о том, что работать с программным обеспечением абсолютно безопасно с точки зрения ионизирующих излучений. Никаких индивидуальных или коллективных средств защиты от ионизирующего излучения не требуется. Рекомендуется разработать в кабинетах на первых этажах график проветривания помещений для уменьшения времени задержки воздуха в помещении с измененным балансом ионизации.

## **ЗАКЛЮЧЕНИЕ**

На данный момент получение образование через интернет не является государственной нормой, но тема является актуальной и перспективной, так как это даёт возможность за относительно не продолжительный промежуток времени приобрести высококвалифицированные знания с малым количеством затрат в удобном виде. Также это выгодно организациям, занимающиеся обучением, так как резко уменьшаются затраты, а многие моменты управления можно автоматизировать.

В рамках дипломного проекта разработана платформа для обучения программирования на Java, которую можно использовать для других дисциплин. Система предоставляет удобный конструктор, позволяющий разместить теоретическую базу дисциплины на платформе с удобным доступом к нему.

Данный продукт является легко сопровождаемым и масштабируемым, так как система разработана при помощи Spring фреймворка, что дает гибкость и большой спектр возможностей в эволюции проекта.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 УО Белорусская государственная академия связи [Электронный ресурс]. – Режим доступа: <http://bsac.by/> – Дата доступа: 16.05.2019.
- 2 Рябычина, О. П. Конструирование программ и языки программирования. Учебная программа учреждения высшего образования по учебной дисциплине для специальности: 1-45 01 02 Инфокоммуникационные системы (по направлениям) по направлению: 1-45 01 02-03 Инфокоммуникационные системы (сопровождение программного обеспечения) / О. П. Рябычина. – Минск : УО БГАС, 2015. – 13 с.
- 3 ГОСТ Р ИСО/МЭК ТО 12182 – 2002. Информационная технология. Классификация программных средств.
- 4 Бахтизин В. В. Технология разработки программного обеспечения / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2010. – 267 с.
- 5 Фатрелл, Р. Управление программными проектами: достижение оптимального качества при минимуме затрат / Р. Фатрелл, Д. Шафер, Л. Шафер. – М. : Вильямс, 2003.
- 6 Каменнова, М. С. Системный подход к проектированию сложных систем / М. С. Каменнова // Журнал д-ра Добба. – 1993. – №1.
- 7 IEEE Std. 1320.2–1998. Стандарт IEEE по синтаксису и семантике языка концептуального моделирования IDEFIX97 (IDEF Object). – Введ. 1998-06-25. – Нью-Йорк : IEEE, 1998.
- 8 Классификация требований к ПО и ее представление в стандартах и методологиях [Электронный ресурс]. – Режим доступа: <http://2006.secrus.org/upload/files/63.pdf> – Дата доступа: 19.05.2019.
- 9 Write once, run anywhere? [Электронный ресурс]. – Режим доступа: <https://www.computerweekly.com/feature/Write-once-run-anywhere> – Дата доступа: 20.05.2019.
- 10 The Java Language Environment [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/technetwork/java/intro-141325.html> – Дата доступа: 20.05.2019.
- 11 Spring Framework [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-framework> – Дата доступа: 20.05.2019.
- 12 Spring Boot [Электронный ресурс]. – Режим доступа: <https://spring.io/projects/spring-boot> – Дата доступа: 20.05.2019.
- 13 IntelliJ IDEA [Электронный ресурс]. – Режим доступа: <https://jetbrains.ru/products/idea/> – Дата доступа: 20.05.2019.



14 What is MySQL? [Электронный ресурс]. – Режим доступа: <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html> – Дата доступа: 20.05.2019.

15 Introduction to Java's Architecture [Электронный ресурс]. – Режим доступа: <https://www.artima.com/insidejvm/ed2/introarch.html> – Дата доступа: 20.05.2019.

16 Apache Tomcat [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Apache\\_Tomcat](https://ru.wikipedia.org/wiki/Apache_Tomcat) – Дата доступа: 20.05.2019.

17 HTML [Электронный ресурс]. – Режим доступа: <https://en.wikipedia.org/wiki/HTML> – Дата доступа: 20.05.2019.

18 CSS [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) – Дата доступа: 20.05.2019.

19 Bootstrap [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Bootstrap\\_\(front-end\\_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)) – Дата доступа: 20.05.2019.

20 Java Persistence API FAQ [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/technetwork/java/javaee/persistence-jsp-136066.html> – Дата доступа: 21.05.2019.

21 Hibernate (framework) [Электронный ресурс]. – Режим доступа: [https://en.wikipedia.org/wiki/Hibernate\\_\(framework\)](https://en.wikipedia.org/wiki/Hibernate_(framework)) – Дата доступа: 21.05.2019.

22 ORM [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/ORM> – Дата доступа: 21.05.2019.

23 Web on Servlet Stack [Электронный ресурс]. – Режим доступа: <https://docs.spring.io/spring/docs/current/spring-framework-reference/web.html> – Дата доступа: 21.05.2019.

24 Intro to Inversion of Control and Dependency Injection with Spring [Электронный ресурс]. – Режим доступа: <https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring> – Дата доступа: 21.05.2019.

25 Using the @SpringBootApplication Annotation [Электронный ресурс]. – Режим доступа: <https://docs.spring.io/spring-boot/docs/current/reference/html/using-boot-using-springbootapplication-annotation.html> – Дата доступа: 21.05.2019.

## ПРИЛОЖЕНИЕ А

Таблица А.1 – Выбор модели жизненного цикла на основе характеристик требований

№ критерия	Критерии категории требований	Каскадная	V-образная	RAD	Итерационная	Быстрого прототипирования	Эволюционная
1	Являются ли требования к проекту легко определяемыми и реализуемыми?	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет	Нет
2	Могут ли требования быть сформулированы в начале ЖЦ?	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет
3	Часто ли будут изменяться требования на протяжении ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да
4	Нужно ли демонстрировать требования с целью их определения?	Нет	Нет	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>
5	Требуется ли проверка концепции программного средства или системы?	Нет	Нет	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>
6	Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
7	Нужно ли реализовать основные требования на ранних этапах разработки?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>

Таблица А.2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

№ критерия	Критерии категории требований	Каскадная	V-образная	RAD	Итерационная	Быстрого прототипирования	Эволюционная
1	Являются ли проблемы предметной области проекта новыми для большинства разработчиков?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

№ критерия	Критерии категории требований	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
2	Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков?	<u>Да</u>	<u>Да</u>	Нет	Нет	Нет	<u>Да</u>
3	Изменяются ли роли участников проекта на протяжении ЖЦ?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
4	Является ли структура процесса разработки более значимой для разработчиков, чем гибкость?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	Нет	Нет
5	Важна ли легкость распределения человеческих ресурсов проекта?	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет
6	Приемлет ли команда разработчиков оценки, проверки, стадии разработки?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>

Таблица А.3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

№ критерия	Критерии категории требований	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	Будет ли присутствие пользователей ограничено в ЖЦ разработки?	Да	Да	<u>Нет</u>	Да	<u>Нет</u>	<u>Нет</u>
2	Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
3	Будут ли пользователи вовлечены во все фазы ЖЦ разработки?	Нет	Нет	<u>Да</u>	Нет	<u>Да</u>	Нет
4	Будет ли заказчик отслеживать ход выполнения проекта?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

Таблица А.4 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

№ критерия	Критерии категории требований	Каскадная	V-образная	RAD	Икрементная	Быстрого прототипирования	Эволюционная
1	Разрабатывается ли в проекте продукт нового для организации направления?	Нет	Нет	Нет	Да	Да	Да
2	Будет ли проект являться расширением существующей системы?	Да	Да	Да	Да	Нет	Нет
3	Будет ли проект крупно- или среднемасштабным?	Нет	Нет	Нет	Да	Да	Да
4	Ожидается ли длительная эксплуатация продукта?	Да	Да	Нет	Да	Нет	Да
5	Необходим ли высокий уровень надежности продукта проекта?	Нет	Да	Нет	Да	Нет	Да
6	Предполагается ли эволюция продукта проекта в течение ЖЦ?	Нет	Нет	Нет	Да	Да	Да
7	Велика ли вероятность изменения системы (продукта) на этапе сопровождения?	Нет	Нет	Нет	Да	Да	Да
8	Является ли график сжатым?	Нет	Нет	Да	Да	Да	Да
9	Предполагается ли повторное использование компонентов?	Нет	Нет	Да	Да	Да	Да
10	Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)?	Нет	Нет	Нет	Нет	Да	Да

