

Санкт-Петербургский государственный университет

***КОНОВАЛОВ Владислав Алексеевич***

**Выпускная квалификационная работа**

***Разработка модулей выполнения заданий Java-версии программного  
комплекса distolymr***

Уровень образования: магистр  
Направление 03.04.01 «Прикладные математика и физика»  
Основная образовательная программа ВМ.5510.2020  
«Прикладные физика и математика»

Научный руководитель:  
к. ф.-м. н., доцент  
Монахов В. В.

Рецензент:  
к. ф.-м. н., ст. преподаватель  
Смирнов А. В.

Санкт-Петербург  
2022

# Оглавление

<b>Введение.....</b>	<b>3</b>
<b>1. Существующие системы управления обучением и основные технологии серверной разработки программного комплекса distolymp ...</b>	<b>5</b>
1.1. Существующие системы управления обучением и виртуальные лаборатории .....	5
1.2. Основные технологии, используемые при разработке серверной части программного комплекса distolymp .....	7
1.2.1. Слой клиента.....	8
1.2.2. Слой логики .....	10
1.2.3. Слой управления данными.....	11
Выводы по главе 1 .....	13
<b>2. Программный комплекс distolymp .....</b>	<b>15</b>
2.1. PHP-версия программного комплекса distolymp .....	15
2.2. База данных программного комплекса distolymp .....	16
2.3. Программный комплекс BARSIC .....	17
2.4. Структура Java-версии программного комплекса distolymp.....	17
Выводы по главе 2 .....	19
<b>3. Полученные результаты и их обсуждение.....</b>	<b>21</b>
3.1. Создание актуальной схемы базы данных .....	21
3.2. Реализация справочников .....	24
3.2.1. Справочник классов (параллелей) обучения участников олимпиады .....	24
3.2.2. Справочник городов .....	27
3.3. Реализация модуля решения задач .....	29
3.4. Реализация модуля выполнения моделей BARSIC .....	38
3.5. Модуль выполнения тестов .....	46
3.5.1. Создание редактора тестов.....	46
3.5.2. Реализация модуля выполнения тестов .....	50
3.6. Обсуждение результатов.....	58
Выводы по главе 3 .....	59
<b>Выводы .....</b>	<b>61</b>
<b>Литература .....</b>	<b>62</b>
<b>Приложение.....</b>	<b>65</b>

## Введение

С 2005 года СПбГУ совместно с Национальным исследовательским университетом Информационных Технологий, Механики и Оптики (Университетом ИТМО) проводит интернет-олимпиаду школьников по физике для учащихся 7–11 классов. Олимпиада позволяет проверять не только знания школьников в области физики, но и их практические умения. Диплом заключительного тура данной олимпиады даёт существенные льготы при поступлении в ведущие вузы страны, например, 100 баллов ЕГЭ по физике или поступление в вуз без вступительных испытаний [1].

В физике важно не только обладать теоретическими знаниями, но и уметь применять их на практике в физическом эксперименте, поэтому олимпиада имеет несколько типов заданий, которые проверяют оба этих фактора. При прохождении олимпиады школьники решают три типа заданий: задачи, которые являются полным аналогом обычных школьных задач по физике; тесты; виртуальные лабораторные работы, максимально приближённые к реальным лабораторным работам. Задачи и тесты осуществляют проверку теоретических знаний, а виртуальные лаборатории — практических умений и навыков. Также виртуальные лаборатории хороши тем, что они бесплатные, одинаковые для всех участников, не требуют обустройства лаборатории и абсолютно безопасны для учащихся.

Для проведения интернет-олимпиады школьников был разработан программный комплекс `distolymp` (разработчики: В. В. Монахов, Л. А. Евстигнеев). Помимо этого, данный программный комплекс также используется преподавателями физического факультета СПбГУ для проведения занятий. На данный момент серверная часть этого комплекса работает на программном обеспечении LAMP (Linux, Apache, MySQL, PHP). Виртуальные лаборатории выполняются с помощью программного комплекса BARSIC [2] (Business And Research Scientific Interactive Calculator)

(разработчики: В. В. Монахов, А. В. Кожедуб, Л. А. Евстигнеев) [3]. Так как связка данного программного обеспечения имеет существенные недостатки при разработке крупных проектов [4], В. В. Монаховым было принято решение разработать аналог серверной части комплекса distolymp на языке программирования Java.

Разработка Java-версии distolymp началась в 2017 году студентом Д. А. Тимашовым [5]. В 2020 году Д. С. Усова выбрала более современный набор технологий, используемых при веб-разработке на Java, и новую архитектуру приложения [6]. Эти технологии и архитектура использованы в данной работе.

Одними из ключевых модулей в программном комплексе distolymp являются модули выполнения заданий (задач, моделей и тестов). Поэтому их реализация в Java-версии приложения является первостепенной задачей. В связи с этим целью данной работы является разработка модулей задач, моделей и тестов на языке программирования Java. Для достижения поставленной цели решались следующие задачи:

- установка и анализ PHP-версии программного комплекса distolymp;
- анализ структуры существующей базы данных;
- изучение технологий, используемых при веб-разработке на Java;
- разработка справочников городов и классов (параллелей) обучения участников олимпиады;
- разработка модуля решения задач;
- разработка модуля выполнения моделей BARSIC;
- разработка редактора тестов;
- разработка модуля выполнения тестов.

# **1. Существующие системы управления обучением и основные технологии серверной разработки программного комплекса distolymp**

## **1.1. Существующие системы управления обучением и виртуальные лаборатории**

В последнее время системы управления обучением (Learning Management Systems) набирают всё большую популярность в обучении школьников и студентов. Особенно сильно на это повлияла пандемия COVID-19. Такие системы позволяют организовывать процесс обучения, осуществлять текущий контроль успеваемости и легко получать доступ к учебным материалам в любое время в практически любом месте.

Самой популярной LMS является Moodle [7]. К её преимуществам можно отнести бесплатное программное обеспечение, мобильное приложение, встроенный редактор для создания текущего контроля обучения, возможность выгрузки статистики и открытый исходный код. Из недостатков системы можно выделить относительно сложный интерфейс, что усложнит работу в программе неопытным пользователям [7].

Также к популярным бесплатным LMS можно отнести Sakai [8]. Основными его преимуществами являются: открытый исходный код, поддержка оптимизации и гибкой настройки процессов обучения и, конечно же, бесплатное распространение. Открытый исходный код позволяет легко совершенствовать продукт даже самим пользователям. Из минусов можно выделить сложный и не интуитивный пользовательский интерфейс и отсутствие хорошей и быстрой обратной связи с преподавателем [8].

Также к наиболее популярным решениям в области систем управления обучением можно отнести EdApp, Violet LMS, Gyrus, Loop, Tovuti LMS и так далее [9]. В целом все вышеописанные системы объединяет то, что они сконцентрированы именно на обучении. В то же время программный

комплекс distolymp концентрируется в основном на проверке полученных знаний и умении их практического применения.

Программный комплекс distolymp положительно выделяется на фоне других LMS тем, что в него интегрирован комплекс виртуальных лабораторий, созданных с помощью языка программирования BARSIC.

Перенос лабораторий в виртуальную компьютерную среду, в которой модели не являются независимыми приложениями, а интегрированы с серверной системой и управляются ей, сложная задача. Поэтому все такие проекты работают в качестве отдельных программ. Рассмотрим наиболее популярные из них.

«National Science Digital Library: ChemCollective Virtual Labs» представляют собой виртуальные лаборатории по химии, которые охватывают такие темы как: стехиометрия, термехимия, кислотно-основная химия, электрохимия [10].

«University of Colorado Boulder: Interactive Simulations for Science and Math» или другое название «PhET Interactive Simulations» [11]. Один из самых богатых репозиториев интерактивных материалов и виртуальных лабораторий. На данном ресурсе можно найти виртуальные лабораторные по физике, химии, математике, биологии, наукам о Земле.

Из российских систем выделяются виртуальные лабораторные работы Томского политехнического университета и «Профессиональная группа» [12, 13]. В обеих системах расположены сотни платных виртуальных работ на различные темы.

Основной недостаток рассмотренных виртуальных лабораторий и множества им подобных — невозможность использования в олимпиадах, так как в них имеются только обычные лабораторные работы в виртуальном варианте, а не олимпиадные задания. Помимо этого, подавляющее большинство таких моделей являются чисто демонстрационными — в них можно самостоятельно задавать параметры эксперимента и смотреть, как они на что-то влияют, но невозможно проверить способность ученика получить

правильный результат. В этих моделях отсутствует возможность автоматической генерации псевдослучайных допустимых значений параметров физической системы и проверка правильности полученных при измерениях результатов в пределах допустимой погрешности. Кроме того, данные модели не сопрягаются с серверной системой, которая позволяет формировать задания туров олимпиады, хранить информацию о полученных участниками туров результатах и подводить итоги туров олимпиады.

Программный комплекс `distolymp`, совмещённый с моделями виртуальных лабораторий `BARSIC`, имеет значительные преимущества перед перечисленными системами. Это бесплатное некоммерческое использование и возможность создания любой лабораторной работы для своих нужд. Но у него есть и недостатки. Например, двумерная анимация лабораторий.

Учитывая вышеописанные факторы, можно сказать, что программный комплекс `distolymp`, с внедрённым программным комплексом виртуальных лабораторий `BARSIC`, является уникальным и мощным инструментом в области проверки знаний и практических умений учащихся, а также их способности к выполнению нестандартных заданий в области физического эксперимента.

## **1.2 Основные технологии, используемые при разработке серверной части программного комплекса `distolymp`**

Для разработки программного комплекса `distolymp` использовалась интегрированная среда разработки (IDE) `IntelliJ IDEA` [14]. Была выбрана именно эта среда разработки, так как она считается одной из лучших. Главные её отличительные черты — это мощные вспомогательные функции по проверке и рефакторингу кода.

Версией, используемого комплекта разработки на `Java`, является бесплатный для распространения `Oracle JDK 8`.

Для автоматизации сборки приложения используется `Maven` [15] — фреймворк на основе описания структуры приложения. Информация об этой

структуре хранится в файлах на языке POM, являющемся подмножеством XML.

Данная платформа позволяет автоматизировать загрузку фреймворков, библиотек, а также их обновление. Помимо процессов управления зависимостями, Maven автоматически генерирует структуру каталогов приложения.

Так как разработка данного программного комплекса ведётся несколькими людьми, то для координации действий разработчиков, а также для безопасности разработанного кода используется система контроля версий Git [16]. Используется данная система совместно с сайтом для хостинга исходного кода GitHub [17].

Для упрощения разработки и отладки приложения в качестве основы программного комплекса distolymp на Java была выбрана трёхслойная архитектура. Трёхуровневая архитектура обычно состоит из уровня представления (уровня пользовательского интерфейса), уровня логики и уровня данных (уровня доступа к данным).

Преимущества многослойной архитектуры заключаются в том, что различные слои разрабатываются как отдельные модули. Один модуль можно полностью заменить, не трогая остальные.

### **1.2.1. Слой клиента**

Основными технологиями, используемыми при разработке слоя пользовательского интерфейса, являются: язык гипертекстовой разметки HTML, таблицы стилей CSS, язык программирования JavaScript, шаблонизатор Thymeleaf.

HTML — это язык разметки, который сообщает веб-браузеру, как структурировать посещаемую страницу. HTML-документ представляет собой текстовый файл, который содержит инструкции (тэги), которые веб-браузер интерпретирует и выводит на страницу различные элементы, такие как кнопки, поля для ввода и тому подобное. Является неотъемлемой частью слоя представления.



Стиль отображаемых веб-браузером элементов можно настраивать с помощью свойства «style». Но это неудобно, так как на веб-страницах часто встречается множество элементов в одинаковом стиле, а также это нарушает принцип разделения ответственности. Поэтому для того, чтобы задать единый стиль всех веб-страниц приложения, используют таблицы стилей CSS. CSS-документ представляет собой текстовый файл, который содержит пары «свойство: значение», которые записываются для группы HTML-элементов. Чтобы стили из CSS-файла применились к веб-странице, необходимо в HTML-файле подключить CSS-файл, используя тэг «link».

Существуют заранее созданные библиотеки стилей для упрощения дизайна веб-страниц. Bootstrap одна из таких библиотек [18]. Она содержит HTML и CSS шаблоны стилей для форм, кнопок, навигации и других компонентов интерфейса. Также важной деталью является то, что Bootstrap изначально очень хорошо адаптирован под просмотр страниц с мобильных устройств.

Используя язык гипертекстовой разметки HTML и таблицы стилей CSS, можно создать веб-страницы собственного дизайна, но эти страницы будут статичными, и любое взаимодействие с ними обязательно будет осуществляться с помощью сервера. Для решения данного вопроса используется скриптовый язык программирования JavaScript. JavaScript используется для динамического изменения HTML и CSS контента. Программный код пишется в HTML-документе, внутри тэга «script».

Также при разработке пользовательского слоя программного комплекса distolymr использовалась библиотека jQuery [19], которая позволяет отделить структуру HTML от описания поведения элементов и обновлять содержимое веб-страницы без необходимости её полной перезагрузки.

Для обеспечения внедрения информации с сервера на веб-страницы используется Java-библиотека (шаблонизатор) Thymeleaf [20], которая преобразует HTML-шаблоны веб-страниц.

## 1.2.2. Слой логики

Основой слоя логики служит Spring Framework [21]. Данный фреймворк состоит из множества модулей (Рис. 1.1), созданных для различных задач.

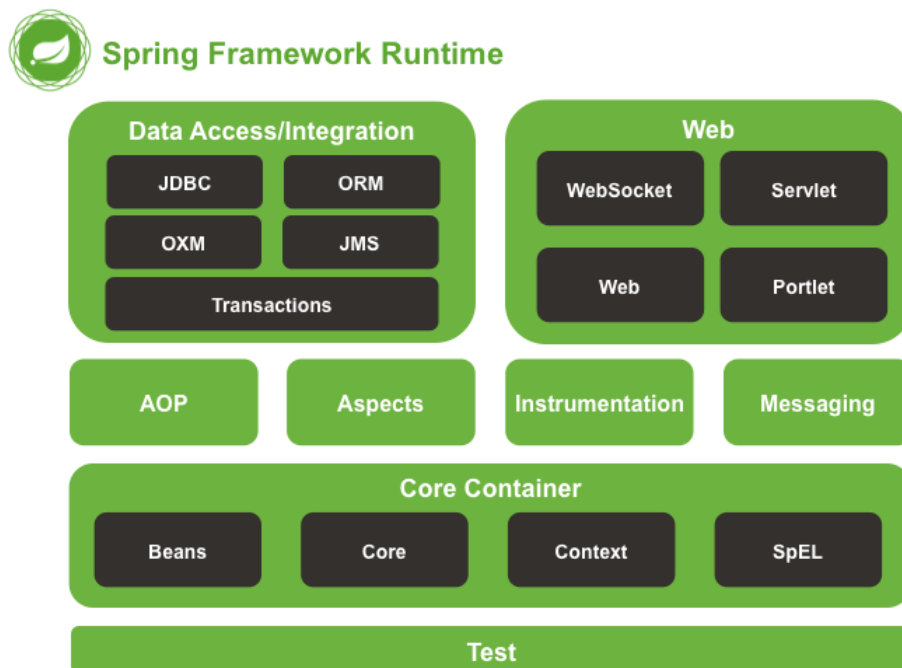


Рис. 1.1. Модули Spring Framework [22].

Ядро Spring Framework построено на принципе инверсии управления (Inversion of Control). Согласно этому принципу, управление объектами осуществляется внешним элементом — IoC-контейнером. Объекты, которыми он управляет, называются «beans». Реализацией принципа IoC является внедрение зависимостей (Dependency Injection), при которой инвертируемым элементом управления являются зависимости объекта. Внедрение зависимостей значительно упрощает разработку и тестирование приложения, так как делает код модульным.

Для корректной работы приложения необходимо обязательно проверять данные, которые отправляет пользователь. Осуществлять такую проверку можно на стороне клиента и на стороне сервера. Проверка на стороне клиента хороша скоростью отклика и отсутствием необходимости обращаться к серверу. Но при этом такую проверку легко обойти через редактор кода страницы в веб-браузере. Поэтому проверка введённых

пользователем данных на стороне сервера является обязательной. Для такой проверки необходимо осуществить привязку данных с веб-страницы к полям объектов на сервере. Привязка данных осуществляется с помощью модуля Spring Web MVC.

Далее производится проверка введённых данных с помощью реализации спецификации Bean Validation — Hibernate Validator [23]. Согласно данной спецификации, проверка данных происходит на основе аннотаций, которые необходимо вставить перед определением поля проверяемых данных. Существует набор базовых встроенных аннотаций, но для более сложных проверок есть возможность создавать собственные.

Архитектура обработки запросов в Spring Web MVC строится на проектном шаблоне MVC (Model View Controller), что представляет собой ещё один пример принципа разделения ответственности. В данном случае система делится на три компонента: модель (содержит объекты с полями данных и методами), представление (отвечает за визуализацию данных модели) и контроллер (управляет взаимодействием между моделью и представлением). В Spring Web MVC также присутствует диспетчер (DispatcherServlet), который принимает все HTTP-запросы вместо контроллера, что делает данную архитектуру более совершенной по сравнению с базовой MVC.

Такой подход позволяет разделить код визуализации и код бизнес-логики приложения, позволяя вносить изменения в одну его часть не меняя другой. Также это способствует упрощённой разработке и отладке приложения.

### **1.2.3. Слой управления данными**

В программном комплексе distolymp в качестве источника данных используется реляционная база данных.

Для взаимодействия с информацией в реляционной базе данных чаще всего используется язык запросов SQL (Structured Query Language). Этот

язык позволяет довольно просто управлять даже относительно большой базой данных.

Но для управления базой данных одного языка недостаточно, необходима система управления базой данных (СУБД). В программном комплексе distolymp используется одна из самых популярных СУБД MySQL [24]. В качестве подсистемы хранения, которая отвечает за выполнение SQL-запросов, используется InnoDB.

Чтобы приложение могло «общаться» с базой данных ему необходимо генерировать SQL-запросы. Так как Java является объектно-ориентированным языком программирования, то логично будет использовать объектно-ориентированный подход в данном вопросе. Для этого используется спецификация JPA (Java Persistence API), которая инкапсулирует SQL-запросы. JPA отображает таблицы из базы данных в классы-сущности (Entity), которые связаны между собой отношениями, задаваемыми структурой базы данных.

В программном комплексе distolymp используется одна из самых популярных реализаций спецификации — Hibernate JPA [25]. Инкапсуляция SQL-запросов позволяет избежать ручного написания SQL-кода, но также порождает некоторые проблемы, связанные с автоматической генерацией запросов. Так, запросы могут генерироваться в избыточном количестве или вовсе не верно. Эти моменты необходимо тщательно отслеживать для корректной и быстрой работы приложения.

Hibernate JPA инкапсулирует SQL-запросы и позволяет даже без знаний SQL разрабатывать методы доступа к данным, но зачастую эти методы представляют собой шаблонный код, в котором различаются только имена переменных. Этот недостаток призван решить Spring Data JPA. Данный механизм предоставляет средства для реализации CRUD (Create Read Update Delete) операций. А так как CRUD-операции представляют собой подавляющее большинство обращений к базе данных, то такая автоматизация ускоряет и упрощает разработку.

## Выводы по главе 1

- Популярные LMS предоставляют множество возможностей для обучения, но не приспособлены для проведения интернет-олимпиад.
- Имеющиеся в мире виртуальные лаборатории, в отличие от системы distolymr с моделями виртуальных лабораторий BARSIC, не имеют подсистем генерации псевдослучайных физически осмысленных значений параметров моделей, проверки правильности результатов, полученных при выполнении, и интеграции этих подсистем с серверной системой.
- Связка программных комплексов distolymr и BARSIC является уникальным и мощным инструментом тестирования теоретических знаний и практических навыков в области физики.
- Для разработки программного комплекса distolymr были выбраны: интегрированная среда разработки IntelliJ IDEA, комплект разработки на платформе Java Oracle JDK 8, автоматический сборщик приложения Maven, система контроля версий Git, сайт для хостинга исходного кода GitHub.
- Концепция многослойной архитектуры используется для разделения приложения на модули, которые проще разрабатывать и отлаживать. В рамках данной концепции приложение было разделено на три слоя: визуализации, логики и данных.
- Для реализации слоя визуализации используются: язык разметки HTML, таблицы стилей CSS, библиотека готовых стилей Bootstrap, язык программирования JavaScript с библиотекой jQuery, шаблонизатор Thymeleaf.
- Для реализации слоя логики используется: фреймворк Spring, а конкретно Spring Core, предоставляющий IoC-контейнер и инструменты для проверки данных, и Spring Web MVC, предоставляющий усовершенствованный проектный шаблон MVC.

- Для проверки отправленных на сервер данных используется Hibernate Validator.
- Для реализации слоя данных используются: Hibernate JPA, Spring Data JPA, предоставляющие классы-сущности (Entity) и готовые интерфейсы и методы для работы с ними.
- Технологии, выбранные для разработки Java-версии программного комплекса distolymp, полностью соответствуют современным тенденциям в области серверного программирования.

## 2. Программный комплекс distolymp

Программный комплекс distolymp был разработан для проведения «Интернет-олимпиады школьников по физике», а также используется в СПбГУ для проведения практических занятий.

### 2.1. PHP-версия программного комплекса distolymp

Рассмотрим работу существующей на данный момент PHP-версии программного комплекса. Как уже было сказано, серверная часть distolymp работает на связке программного обеспечения LAMP (Linux, Apache, MySQL, PHP).

На данный момент используется операционная система на ядре Linux — CentOS 7 [26].

В качестве веб-сервера используется Apache HTTP Server [27]. Это один из самых популярных бесплатных программных веб-серверов с открытым исходным кодом.

Для хранения данных в комплексе используется реляционная база данных, управляемая СУБД MySQL.

Основным языком программирования является PHP5. PHP является высокоуровневым языком программирования и очень часто используется для веб-разработки.

Пользовательский интерфейс работает на связке: язык гипертекстовой разметки HTML, таблицы стилей CSS, язык программирования JavaScript. Последний также используется для предобработки и проверки некоторой информации, вводимой пользователем.

Рассмотрим структуру каталогов PHP-версии программного комплекса distolymp. В корневом каталоге находятся папки:

- «class» — папка, содержащая служебные классы;
- «css» — папка, содержащая файлы таблиц стилей CSS;
- «data» — папка, содержащая файлы, относящиеся к заданиям олимпиады;

- «db» — папка, содержащая скрипты для работы с базой данных;
- «func» — папка, содержащая часто используемые функции и процедуры;
- «img» — папка, содержащая служебные изображения;
- «inc» — папка, содержащая инициализаторы и финализаторы для объектов, которые работают с базой данных;
- «js» — папка, содержащая файлы на языке программирования JavaScript;
- «language» — папка, содержащая скрипты для настройки языка программного комплекса;
- «lib» — папка, содержащая вспомогательные библиотеки;
- «module» — папка, содержащая PHP-файлы, относящиеся к бизнес-логике комплекса;
- «sys» — папка, содержащая файлы, необходимые для поддержки старой версии комплекса;
- «template» — папка, содержащая PHP-файлы, относящиеся к визуализации комплекса.

## **2.2. База данных программного комплекса distolymr**

Для хранения информации в программном комплексе distolymr используется реляционная база данных. Для управления базой данных используется СУБД MySQL. На данный момент используется система хранения данных MyISAM.

В 2011 году студентом Л. В. Басовым была разработана документация по структуре базы данных PHP-версии программного комплекса distolymr [28]. База данных состоит из 48 таблиц. Все таблицы объединены в 8 групп по характеру содержащейся в них информации:

- 1) группы пользователей;
- 2) списки заданий и настройки видимости;
- 3) страны, регионы РФ, города, институты и школы;



- 4) задачи, тесты и модели;
- 5) информация о пользователях;
- 6) учётные записи пользователей, преподавателей и регистраторов;
- 7) активность пользователей в системе;
- 8) ответы и дополнительное время.

### **2.3. Программный комплекс BARSIC**

В программном комплексе distolymp существуют три типа заданий: задачи, тесты и виртуальные лабораторные работы (модели) [1]. Последние реализованы с помощью программного комплекса BARSIC [2], который позволяет как создавать модели, так и запускать их. Он включает в себя язык программирования, среду разработки, исполняющую среду (проигрыватель).

Язык программирования BARSIC является высокоуровневым интерпретируемым объектным языком программирования со строгой типизацией. Синтаксис BARSIC построен на принципе максимальной читаемости, поэтому в программах, написанных на этом языке, практически не требуется вставлять комментарии. Также по этой же причине диагностика синтаксических ошибок значительно упрощается.

Среда разработки BARSIC включает в себя поддержку работы с пользовательским интерфейсом «из коробки», что значительно ускоряет разработку программ. Также в среду разработки уже встроена система перехвата и обработки исключительных ситуаций, что делает программы, написанные на BARSIC, более надёжными [29].

Для выполнения виртуальных лабораторных работ пользователю необходимо скачать проигрыватель BARSIC. В проигрыватель встроены веб-браузер и калькулятор для удобства. Задачи и тесты при этом можно выполнять как во внешнем веб-браузере, так и в проигрывателе BARSIC.

### **2.4. Структура Java-версии программного комплекса distolymp**

Студенткой Д. С. Усовой была разработана актуальная архитектура будущей Java-версии программного комплекса distolymp [6]. Как уже было

сказано в первой главе, за основу структуры приложения была взята трёхслойная архитектура. Технологии, используемые при разработке и создании архитектуры, также были подробно описаны и выбраны в первой главе. Эти технологии полностью соответствуют современным тенденциям в области серверного программирования на языке Java.

В основе полученной структуры каталогов приложения лежит платформа автоматизации сборки Maven [15]. Рассмотрим полученную структуру поподробней. В корневом каталоге («src/main») расположены три папки: «java» (папка, содержащая весь код на языке программирования Java), «resources» (папка, содержащая параметрические файлы для настройки приложения), «webapp» (папка, содержащая всё, что относится к слою визуализации, например, HTML-файлы, CSS-файлы и тому подобное).

Теперь рассмотрим архитектуру ключевых каталогов, разработанную Д. С. Усовой [6], которые расположены в папке «java».

- «common» — папка, в которой содержатся вспомогательные классы (парсеры, генераторы файлов, работа с файлами и тому подобное) для работы бизнес-логики.
- «config» — папка, содержащая конфигурационные классы настроек приложения.
- «controller» — папка, содержащая контроллеры из парадигмы MVC.
- «dto» — папка, содержащая DTO (Data Transfer Object), которые используются для передачи нужных данных между слоями архитектуры.
- «entity» — папка, содержащая классы-сущности (Entity), служащие для отображения таблиц из базы данных.
- «exception» — папка, содержащая классы-исключения, а также глобальный обработчик исключительных ситуаций.
- «mapper» — папка, содержащая классы (мапперы), которые служат для отображения Entity в DTO и обратно.

- «repository» — папка, содержащая реализации интерфейсов из фреймворка Spring Data (репозитории), которые позволяют использовать уже написанные методы доступа к базе данных.
- «service» — папка, содержащая классы-сервисы. Являются основным слоем бизнес-логики. В текущей архитектуре приложения существует два типа сервисов: CRUD-сервисы (содержат в себе более низкоуровневую логику взаимодействия с репозиториями) и просто сервисы (содержат в себе основную бизнес-логику).
- «util» — папка, содержащая конвертеры фильтров, необходимых для реализации постраничного поиска и сортировки данных из базы данных.
- «validation» — папка, содержащая аннотации для проверки введённых данных с помощью Hibernate Validator и классы-валидаторы для этих аннотаций.

## **Выводы по главе 2**

- Последующее расширение и усовершенствование РНР-версии программного комплекса distolymr осложняется выбранными технологиями разработки. Несмотря на то что РНР прост в изучении и использовании, данный язык обладает рядом недостатков, например, слабая динамическая типизация и отсутствие обратной совместимости. Данные недостатки неприемлемы при разработке крупных проектов. Соответственно, разработка Java-версии программного комплекса distolymr является важнейшей задачей.
- В течение эксплуатации РНР-версии программного комплекса distolymr в базе данных выявились неиспользуемые таблицы и поля, в связи с чем необходимо провести анализ базы данных и модернизировать её.
- На данный момент выполнение моделей BARSIC возможно только с ПК, на котором установлен проигрыватель BARSIC. Для удобства

пользователей и расширения числа участников олимпиады идёт разработка Android-версии проигрывателя BARSIC.

- Структура Java-версии программного комплекса, выполненная с помощью современного средства автоматизации сборки Maven, основана на принципах модульности и разделения ответственности. Во-первых, это позволяет заниматься разработкой комплекса несколькими программистами параллельно. Во-вторых, отладка и поиск ошибок в коде значительно упрощаются.

### 3. Полученные результаты и их обсуждение

Перед началом работы на личный ПК была установлена виртуальная машина Oracle VM VirtualBox [30] с 64-разрядной ОС CentOS 7. На неё была установлена PHP-версия программного комплекса distolymp, при помощи установщика, разработанного Д. С. Усовой [6]. Также на виртуальную машину была установлена структурная копия базы данных без реальных данных.

#### 3.1. Создание актуальной схемы базы данных

В процессе использования PHP-версии программного комплекса выявились неиспользуемые таблицы и поля базы данных. Также в Java-версии комплекса некоторые поля автоматизированы и более не требуют ручного ввода. Исходя из вышесказанного, решено было проанализировать существующую базу данных и, учитывая опыт использования, создать новую схему базы данных.

Как уже было описано ранее в пункте 2.2, текущая база данных состоит из 48 таблиц, которые объединены в 8 групп. Благодаря модернизации и оптимизации программного комплекса, активная база данных теперь состоит из 39 таблиц, объединённых в 10 групп:

##### 1. Groups:

- 1.1. groups — таблица групп пользователей;
- 1.2. constraints — таблица ограничений по времени выполнения олимпиады;
- 1.3. user\_groups — связывает таблицу groups с таблицей users;
- 1.4. staff\_groups — связывает таблицу groups с таблицей staff;
- 1.5. managing\_group — таблица административных групп;
- 1.6. history\_mgroup — связывает таблицу managing\_group с таблицей users;

##### 2. Lists:

- 2.1. categories — таблица категорий групп, использующихся в разделах;
- 2.2. divisions — таблица разделов системы;

- 2.3. lists — таблица списков заданий;
- 2.4. list\_problems — связывает таблицу lists с таблицей p\_type;
- 2.5. classes — таблица классов (параллелей обучения участников олимпиады);
- 2.6. staff\_lists — связывает таблицу staff с таблицей lists;
- 2.7. settings — таблица настроек;
- 3. Logs:
  - 3.1. user\_log\_list — таблица входа пользователя в список заданий;
  - 3.2. user\_log — таблица с историей действий пользователя в системе;
- 4. Tasks:
  - 4.1. p\_type — таблица с основными параметрами заданий;
  - 4.2. p\_task — таблица задач;
  - 4.3. p\_sim — таблица с моделями BARSIC;
  - 4.4. p\_test — таблица тестов;
- 5. Education:
  - 5.1. institutes — таблица предпочитаемых вузов;
  - 5.2. places — таблица площадок очного тура;
  - 5.3. schools — таблица школ;
  - 5.4. school\_types — таблица типов школ;
- 6. Geography:
  - 6.1. countries — таблица стран;
  - 6.2. federals — таблица федеральных округов;
  - 6.3. regions — таблица регионов Российской Федерации;
  - 6.4. towns — таблица городов (населённых пунктов);
- 7. Diploma:
  - 7.1. diplom\_types — таблица видов дипломов;
  - 7.2. diplom\_strings — таблица с надписями, наносимыми на дипломы;
- 8. Users:
  - 8.1. staff — таблица преподавателей;
  - 8.2. staff\_divisions — связывает таблицу staff с таблицей divisions;

- 8.3. users — таблица пользователей;
9. User info:
- 9.1. user\_institutes — связывает таблицу users с таблицей institutes;
- 9.2. user\_places — связывает таблицу users с таблицей places;
- 9.3. passports — таблица паспортных данных;
- 9.4. informations — таблица с дополнительной информацией;
- 9.5. contacts — таблица контактов для связи;
10. Others:
- 10.1. answers — таблица ответов пользователей на задания;
- 10.2. user\_extra\_time — таблица с дополнительным временем работы пользователя в списке заданий.

Оставшиеся девять таблиц, которые не вошли в классификацию, не удалены. В базе данных они все ещё существуют для обеспечения совместимости версий. В более новой Java-версии программного комплекса эти таблицы не используются, так как в процессе эксплуатации комплекса выяснилось, что данные таблицы не применяются на практике.

В соответствии с вышеописанной классификацией была разработана схема базы данных. Часть этой схемы приведена на Рис. 3.1.

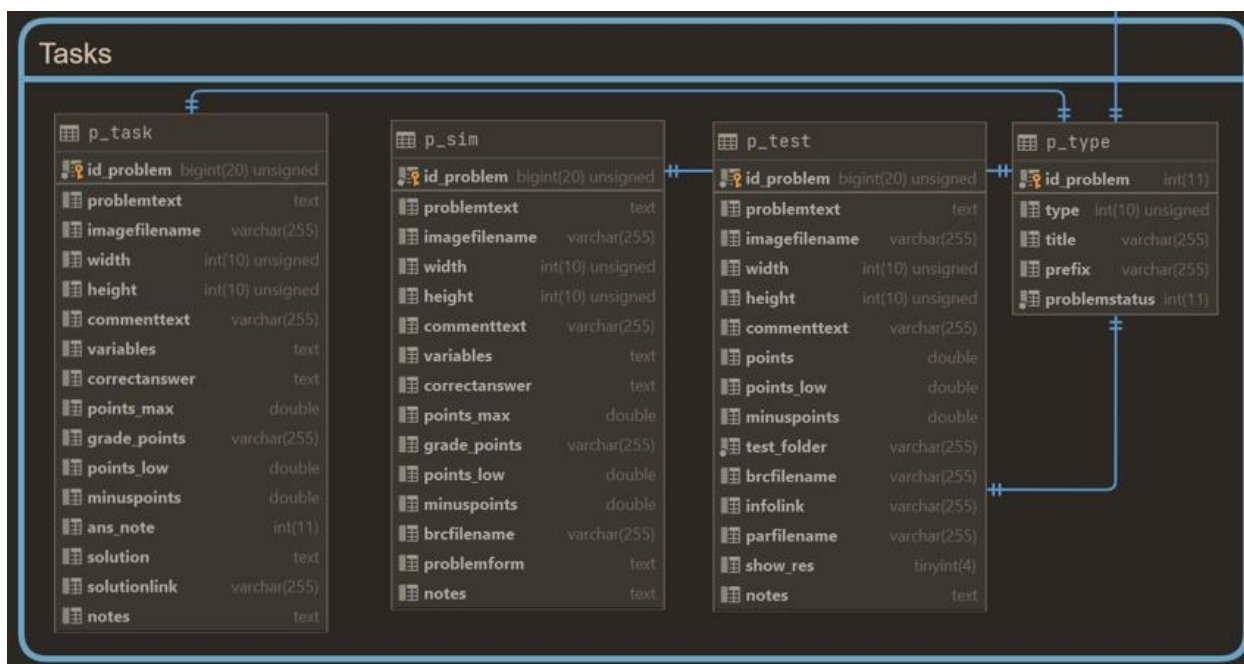


Рис. 3.1. Часть схемы базы данных, отвечающая за задания.

## 3.2. Реализация справочников

Справочники (CRUD-справочники) по своей сути являются удобным пользовательским интерфейсом взаимодействия с данными в базе данных. Поэтому их реализация является одной из самых простых задач в разработке Java-версии программного комплекса distolymp. В рамках данной работы было разработано два справочника: справочник классов (параллелей) обучения участников олимпиады и справочник городов.

### 3.2.1. Справочник классов (параллелей) обучения участников олимпиады

Справочник классов отображает информацию обо всех классах, которые есть в базе данных. На Рис. 3.2 показан вид справочника классов на PHP-версии программного комплекса distolymp.

Списки и преподаватели, автоматически назначаемые группам при регистрации							
Класс	Список	Преподаватель	Регистрация	Кодовая фраза	Письмо участнику от имени (email)	Служебная почта	Действия
11	Список 1	admin	закрыта	Не назначена	distolympinfo@mail.ru	mail.@mail.ru	изменить переименовать удалить
10	Не назначен	Не назначен	закрыта	Не назначена	distolympinfo@mail.ru	mail.@mail.ru	изменить переименовать удалить
9	Не назначен	Не назначен	закрыта	Не назначена	distolympinfo@mail.ru	mail.@mail.ru	изменить переименовать удалить
8	Не назначен	Не назначен	закрыта	Не назначена	distolympinfo@mail.ru	mail.@mail.ru	изменить переименовать удалить
7	Не назначен	Не назначен	закрыта	Не назначена	distolympinfo@mail.ru	mail.@mail.ru	изменить переименовать удалить

Рис. 3.2. Справочник классов на PHP-версии программного комплекса distolymp.

При разработке Java-версии комплекса основная задача была в первую очередь воссоздать существующую функциональность. Поэтому в Java-версии были лишь исправлены ошибки стилей HTML-страниц (ширина тела страницы была меньшей, чем ширина меню), а также добавлена небольшая новая функциональность в виде кнопок «Открыть регистрацию» и «Закрыть регистрацию», которые открывают и закрывают возможность



регистрации всем классам соответственно. На Рис. 3.3 приведён вид того же справочника на Java-версии.

Открыть регистрацию		Закреть регистрацию					
Класс	Список	Преподаватель	Регистрация	Кодовая фраза	Письмо участнику от имени (email)	Служебная почта	Действия
11	spisok 2	admin	закрота	Не назначена	.@...	qq@11	переименовать изменить удалить
10	Список 1	monachov	закрота	клоун	distolympinfo@mail.ru	qq@11	переименовать изменить удалить
7	spisok 2	monachov	закрота	Не назначена	Не назначен	Не назначен	переименовать изменить удалить
7_old	Не назначен	Не назначен	закрота	Не назначена	Не назначен	Не назначен	переименовать изменить удалить
3	Не назначен	Не назначен	закрота	q	qq@11	qq@11	переименовать изменить удалить
12	spisok 3	monachov	закрота	123	distolympinfo@mail.ru	qq@11	переименовать изменить удалить
2	Не назначен	Не назначен	закрота	Не назначена	distolympinfo@mail.ru	qq@11	переименовать изменить удалить

Добавить новый класс

Рис. 3.3. Справочник классов на Java-версии программного комплекса *distolymp*.

На Рис. 3.4 отображён вид страницы редактирования класса на PHP-версии комплекса. Страница редактирования появляется при нажатии на кнопку «изменить».

Значения по умолчанию для класса

Класс:

Список:

Преподаватель:

Регистрация:

Кодовая фраза:

Письмо участнику от имени (email):

Письмо участнику от имени (name):

Служебная почта:

Текст до:

Текст после:

Рис. 3.4. Форма редактирования класса на PHP-версии программного комплекса *distolymp*.

Так как основная задача была реализация с прежней функциональностью, то вид формы редактирования на Java-версии программного комплекса практически не изменился (Рис. 3.5). Был изменён тип поля имени класса. Раньше переименовать класс можно было только на главной странице справочника классов, теперь это можно сделать также на форме редактирования. Это поле проверяется на сервере соответствующим образом во избежание пустого имени класса.

The image shows a web form titled "Значения по умолчанию для класса" (Default values for the class). The form contains several input fields and dropdown menus:

- Класс:** Text input field containing "11".
- Список:** Dropdown menu with "spisok 2" selected.
- Преподаватель:** Dropdown menu with "admin" selected.
- Регистрация:** Dropdown menu with "закрыта" (closed) selected.
- Кодовая фраза:** Empty text input field.
- Письмо участнику от имени (email):** Text input field containing ".@...".
- Письмо участнику от имени (name):** Text input field containing "Оргкомитет Интернет-олимпиады по физике".
- Служебная почта:** Empty text input field.
- Текст до:** Text area containing "здравствуйте".
- Текст после:** Text area containing "до свидания".

Рис. 3.5. Форма редактирования класса на Java-версии программного комплекса *distolymp*.

Удаление и переименование было реализовано в точности так же как и в РНР-версии при нажатии на одноимённые кнопки. Нажатие на кнопку «Добавить новый класс» вызывает пустую форму редактирования, заполненный вид которой был показан выше.

В итоге была реализована улучшенная версия справочника классов из РНР-версии комплекса. Добавлена новая функциональность и исправлены некоторые ошибки.

### 3.2.2. Справочник городов

Справочник городов отображает все города, которые есть в базе данных, согласно фильтру, который задаётся в панели «Поиск города». Фильтр работает следующим образом. В поле «Город» можно искать город по его названию или части названия. В выпадающем списке «Страна» можно выбрать страну, города которой будут отображены в списке. Выпадающий список «На странице» указывает количество городов, которые будут отображены в списке. Чекбокс «Показать скрытые», соответственно, отображает города с видимостью «скрытый». Список городов всегда сортируется по названию города в алфавитном порядке. На Рис. 3.6 приведён вид справочника городов на РНР-версии программного комплекса distolymp. На данном изображении отображаются первые 10 не скрытых городов.

№	<input type="checkbox"/>	Город
1	<input type="checkbox"/>	Волосово
2	<input type="checkbox"/>	Волхов
3	<input type="checkbox"/>	Всеволожск
4	<input type="checkbox"/>	Выборг
5	<input type="checkbox"/>	Высоцк
6	<input type="checkbox"/>	Гатчина
7	<input type="checkbox"/>	Ивангород
8	<input type="checkbox"/>	Ингушэ
9	<input type="checkbox"/>	Каменногорск
10	<input type="checkbox"/>	Кингисепп

Зарегистрировать новый город

Назначить видимость выбранным городам

Удалить выбранные города

Рис. 3.6. Справочник городов на Java-версии программного комплекса distolymp.

Основная задача при разработке справочника городов так же как и при разработке справочника классов — первоначально реализовать функциональность PHP-версии. Поэтому Java-версия справочника выглядит точно так же и имеет такую же функциональность (Рис. 3.6).

При нажатии на город открывается вкладка, на которой отображены учебные заведения (школы) данного города (Рис. 3.7) и вкладка, на которой отображена информация о городе (Рис. 3.8).

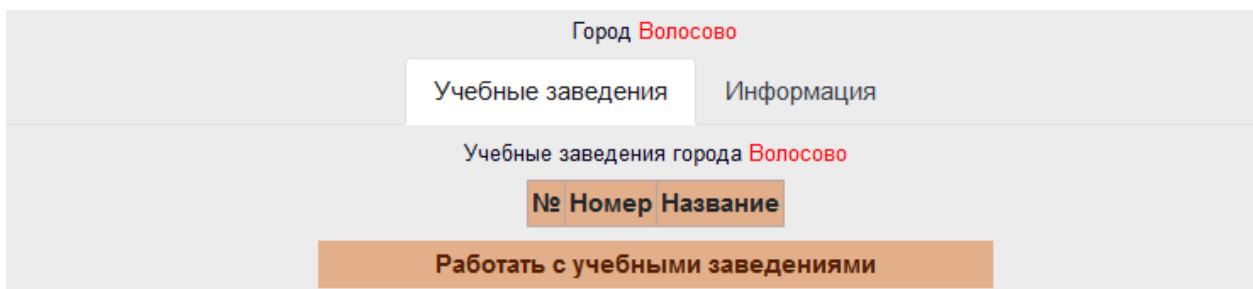


Рис. 3.7. Учебные заведения города на Java-версии программного комплекса *distolymp*.

На Рис. 3.7 кнопка «Работать с учебными заведениями» перенаправит нас на страницу справочника учебных заведений с фильтром по нашему городу.

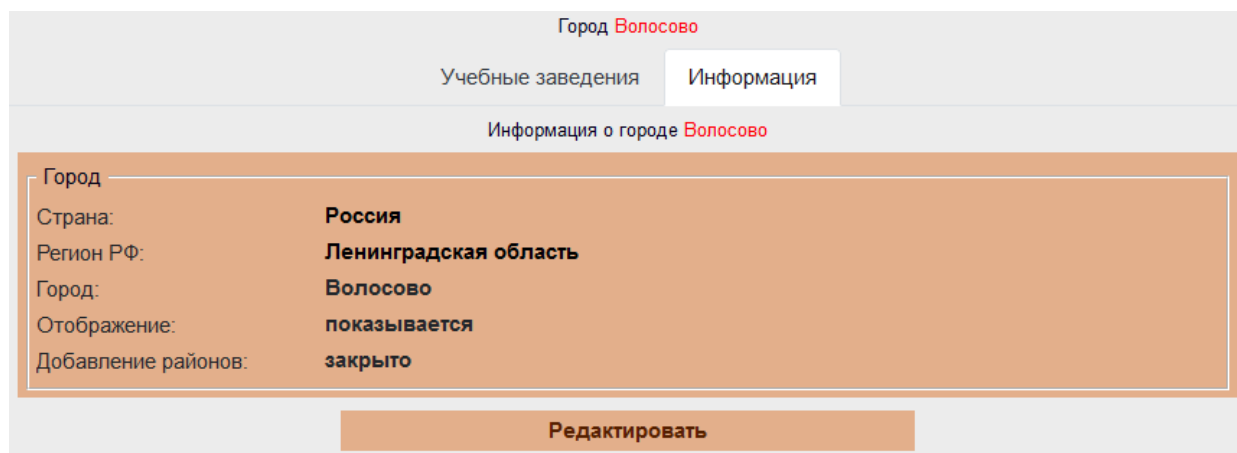


Рис. 3.8. Информация о городе на Java-версии программного комплекса *distolymp*.

На Рис. 3.8 кнопка редактировать откроет форму редактирования города, которая показана на Рис. 3.9.

Город Волосово

Город	
Страна:	Россия
Регион РФ:	47. Ленинградская область
Город:	Волосово
Отображение:	показывается
Добавление районов:	закрыто

Рис. 3.9. Форма редактирования города на Java-версии программного комплекса *distolymp*.

В форме редактирования при выборе страны «Россия» динамически появляется поле «Регион РФ». Если же выбрана любая другая страна, этого поля нет. В поле «Город» необходимо ввести название города, это поле проходит серверную валидацию, чтобы не возникало городов с пустым именем.

Кнопка «Зарегистрировать новый город» на главной странице вызывает вышеупомянутую форму редактирования, но без заполнения. Кнопка «Назначить видимость выбранным городам», соответственно, назначает городам, выбранным с помощью чекбоксов, видимость «показывается» или «скрыт». Кнопка «Удалить выбранные города» удаляет города, выбранные с помощью чекбоксов.

Таким образом, справочник городов был реализован полностью как на PHP-версии программного комплекса *distolymp*. Основная задача по реализации уже имеющейся функциональности была достигнута.

### 3.3. Реализация модуля решения задач

Всего в программном комплексе *distolymp* существует три типа заданий: задачи, модели BARSIC и тесты. Задачи с числовым ответом (задачи типа «Только ответы», Рис. 3.10) в программном комплексе представляют собой аналог обычных школьных задач, но с задаваемой допустимой погрешностью. В задаче имеется условие, и в качестве решения

необходимо дать численные ответы, вычислив необходимые значения физических величин. Также существуют задачи типа «Только комментарий» (ответ в виде произвольного текста), «Ответы и комментарий», «Только файл» (загрузка ответа в виде файла произвольного формата) и «Файл и комментарий» — Рис. 3.10.

На Рис. 3.10 представлен вид вкладки «Задачи» PHP-версии программного комплекса *distolymp*.

Префикс	Образец задачи
Заголовок	Определить импульс и энергию тел ( 10 баллов)
Задание	<p>&lt;h3 align=center&gt;Олимпиада 2 тур: Определите импульс и энергию тел (10 баллов)&lt;/h3&gt;                  Два тела движутся по взаимно перпендикулярным направлениям.                  &lt;br&gt;                 &lt;b&gt;Масса первого тела {m1} кг, а скорость {v1} м/с. Масса второго {m2} кг и скорость {v2} м/с. Определите модуль полного импульса P данной системы тел и кинетическую энергию E системы.</p> <p>&lt;p&gt;Ответ вводить с точностью до целых.</p>
Изображение	Обзор... Файл не выбран. Width: 0 Height: 0
Комментарий	
Значения переменных (?)	v1=(2..6.1); v2=(5..10.1); m1=(7..11.1); m2=(2..6.1); p1=m1*v1; p2=m2*v2
Тип ответа	<input type="radio"/> Только файл <input type="radio"/> Файл и комментарий <input type="radio"/> Только комментарий <input checked="" type="radio"/> Ответы и комментарий <input type="radio"/> Только ответы
Ответы	h1=0
Количество баллов:	5
Штраф за неправильный ответ:	2
Минимальное количество баллов за правильный ответ:	1
<input type="button" value="Сохранить"/> <input type="button" value="Отменить"/>	

№	Заголовок	Статус	Действия
1	Образец задачи : Определить импульс и энергию тел ( 10 баллов)	Опубликована	просмотр   редактировать   копировать удалить   архивировать

Поиск

Рис. 3.10. Главная страница задач на PHP-версии программного комплекса *distolymp*.

Как можно видеть по Рис. 3.10 в РНР-версии комплекса на главной странице задач смешаны форма редактирования и список всех задач. Было принято решение в Java-версии разделить эту страницу на две: список задач с поиском и форма редактирования. В результате вид полученного списка задач представлен на Рис. 3.11.

№	Заголовок	Статус	Действия
1	Файл	Новая	просмотр редактировать копировать удалить
2	Файл и комментарий	Новая	просмотр редактировать копировать удалить
3	Комментарий	Новая	просмотр редактировать копировать удалить
4	Образец задачи: Определите импульс и энергию теп (10 баллов)	Новая	просмотр редактировать копировать удалить
5	11 класс: Найти объем	Новая	просмотр редактировать копировать удалить

**Создать новую задачу**

Рис. 3.11. Список всех задач на Java-версии программного комплекса *distolymp*.

Поиск по заголовку и префиксу задач был перемещён в верхнюю часть страницы для удобства. Задачи отсортированы в порядке добавления в базу данных (сначала новые). Была убрана кнопка «архивировать», так как она на практике не использовалась (но кнопку «разархивировать» было решено оставить, так как в системе могут оставаться случайно заархивированные задачи).

В РНР-версии комплекса кнопка «редактировать» заполняла форму редактирования на этой же странице, теперь эта кнопка открывает форму редактирования, вид которой представлен на Рис. 3.12.

Префикс	Образец задачи ✓
Заголовок	Определите импульс и энергию тел (10 баллов) ✓
Задание	<p>&lt;h3 align=center&gt;Олимпиада 2 тур: Определите импульс и энергию тел (10 баллов)&lt;/h3&gt;</p> <p>▼ <b>Специальные символы</b></p> <p>&lt; &gt; ° ± · × ÷ ² ³ x<sup>y</sup> x<sub>y</sub> √ " ' § _ \ Ж К Ц Ъ  α β γ δ ε η θ λ μ ν ξ π ρ σ τ φ ω Δ Φ Ψ Ω</p> <p>Файл не выбран <span>Обзор</span></p>
Изображение	<p>Ширина: 200 ✓</p> <p>Высота: 0 ✓</p> <p><input type="checkbox"/> Удалить</p> <p><b>Просмотр</b></p>
Значения переменных <sup>(?)</sup>	<p>v1=(2..6,1);  v2=(5..10,1);</p>
Тип ответа	<p><input checked="" type="radio"/> Только ответы</p> <p><input type="radio"/> Ответы и комментарий</p> <p><input type="radio"/> Только комментарий</p> <p><input type="radio"/> Файл и комментарий</p> <p><input type="radio"/> Только файл</p>
Ответы	<p>p=sqrt(p1*p1+p2*p2)+-2;  e=(m1*v1*v1/2+m2*v2*v2/2)+-2</p>
Количество баллов	5; 5
Штраф за неправильный ответ	<p>Штрафной балл должен быть от 0 до 65535</p>
Минимальное количество баллов за правильный ответ	2.0 ✓

Рис. 3.12. Форма редактирования задачи на Java-версии программного комплекса distolymp.

Вид формы редактирования также значительно изменился. Теперь поля «Заголовок», «Задание», «Штраф за неправильный ответ» и «Минимальное



количество баллов за правильный ответ» являются обязательными для ввода. Появился выпадающий блок «Специальные символы» с HTML-символами, которые используются для оформления текста задачи и ответов. Поле «Комментарий» было убрано, так как не использовалось на практике. Теперь на форме присутствует валидация как на стороне клиента, так и на стороне сервера. Поля, отмеченные зелёной галочкой (либо красным восклицательным знаком), проверяются на стороне клиента и на стороне сервера. Неотмеченные поля, в силу сложности валидации, проходят проверку только на сервере. Сообщения об ошибках в Java-версии комплекса появляются рядом с полем для ввода вместо всплывающих сообщений в PHP-версии.

Так же как и в PHP-версии комплекса при выборе типа ответа «Только ответы» или «Ответы и комментарий», поле «Ответы» появится, иначе исчезнет, так как при этом в нём не будет необходимости.

Поля «Значения переменных» и «Ответы» заполняются специальным образом. Логика заполнения этих полей построена на присвоении переменным числовых или текстовых значений для дальнейшего их использования в тексте задачи, тексте ответа, вычитывании ответа. Для использования введённых переменных был разработан парсер (parser — синтаксический анализатор, который преобразует текст в структуру данных) на основе JavaScript, так как он обладает простыми и понятными практически любому конструкциями языка программирования. При этом была обеспечена полная совместимость с PHP-версией проекта, так как это одна из важнейших задач.

Помимо простого задания числовых и текстовых значений переменным, существуют специальные конструкции, такие как:

- генерация псевдослучайного числа в заданном массиве чисел вида  
`var1 = [val1, val2, ..., valN];`

- генерация псевдослучайного числа в диапазоне с некоторым шагом вида `var2 = (val1..val2, step)`;
- наиболее часто используемые математические выражения с синтаксисом языка РНР.

Для удобства пользователя специальные конструкции добавлены во всплывающую подсказку рядом с надписью «Значения переменных».

Переменные можно использовать в качестве условий задачи и/или для расчёта правильного ответа. Для включения переменных в условия задачи необходимо указать их имя в тексте задачи в фигурных или квадратных скобках. А для использования в расчётах необходимо просто включить их в формулу расчёта ответа.

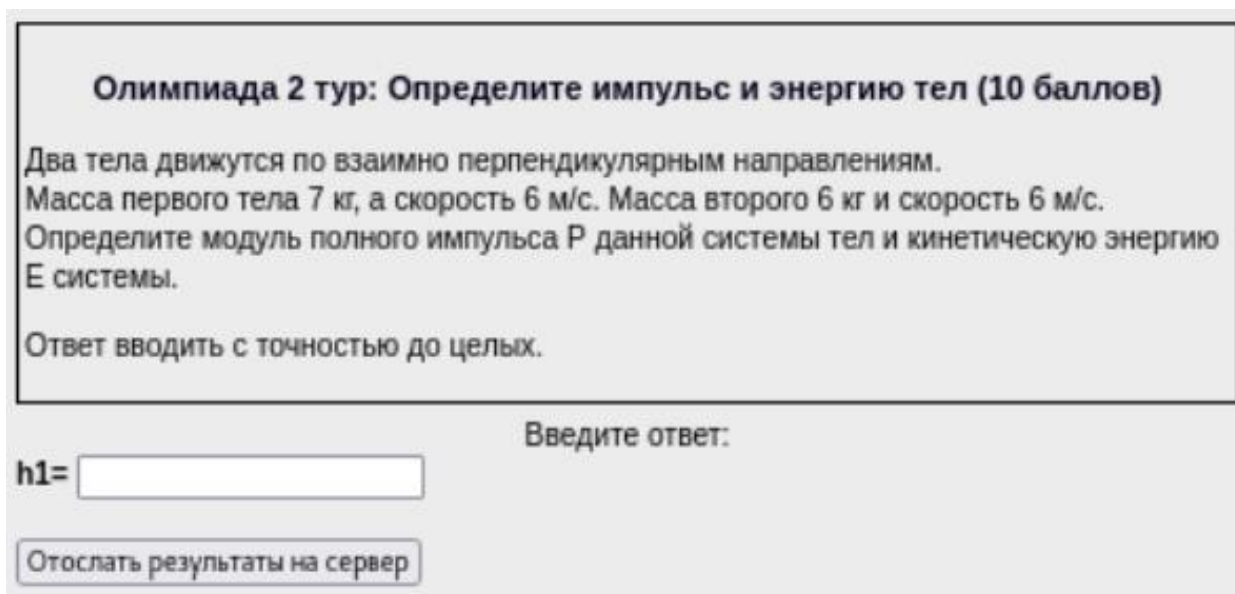
У ответов можно указать абсолютную или относительную погрешность (с помощью сочетания символов «+-»), которая будет учитываться при автоматической проверке ответа, введённого пользователем. Если погрешность не указана, то используется абсолютная погрешность равная «0,05».

Если имя переменной начинается с «comment\_» или «commentAfter\_» и заканчивается именем другой переменной, записанной в поле «Ответ», то эти переменные используются только в качестве пояснений для ввода ответа, до поля ответа и после, соответственно. Назовём такие переменные комментариями.

Для проверки поля «Количество баллов» также был написан парсер, так как для корректной работы комплекса необходимо, чтобы число баллов совпадало с числом ответов.

В Java-версии при загрузке изображения к задаче нет необходимости задавать ширину и высоту, достаточно только одного параметра, изображение будет масштабировано по нему (можно, как и в РНР-версии, задать оба параметра). Просмотр загруженного изображения теперь производится во всплывающем окне вместо отдельной вкладки.

Протестировать созданную или отредактированную задачу можно при нажатии на кнопку «просмотр» на главной странице задач (Рис. 3.13).



The screenshot shows a problem box with the following text:

**Олимпиада 2 тур: Определите импульс и энергию тел (10 баллов)**

Два тела движутся по взаимно перпендикулярным направлениям.  
Масса первого тела 7 кг, а скорость 6 м/с. Масса второго 6 кг и скорость 6 м/с.  
Определите модуль полного импульса  $P$  данной системы тел и кинетическую энергию  $E$  системы.

Ответ вводить с точностью до целых.

Введите ответ:

h1=

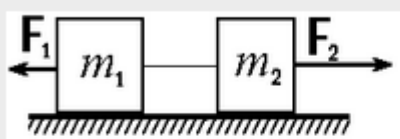
Рис. 3.13. Просмотр задачи на PHP-версии программного комплекса *distolymp*.

При просмотре задачи значения переменных генерируются псевдослучайным образом и заменяют собой имена этих переменных в фигурных или квадратных скобках. Число полей для ввода ответа равно числу переменных, заданных в поле «Ответ» на форме редактирования задачи. Если в задаче предусмотрен текстовый ответ, то появляется поле для текстового ответа. А если в задаче предусмотрен ответ в виде файла, то, соответственно, появляется поле для загрузки и отправки файла на сервер.

Когда в задаче присутствует числовой ответ, то пользователю выводится информация о правильности введённых ответов. Если в задаче отсутствует числовой ответ, то пользователь просто информируется о том, что его ответ отправлен на сервер для дальнейшей проверки.

В PHP-версии комплекса информирование пользователя происходило посредством всплывающих сообщений, что было неудобно, так как можно было легко забыть, где была допущена ошибка. В Java-версии информирование пользователя происходит посредством экранных подписей к ответам и над задачей (Рис. 3.14).

Частично правильно



**Олимпиада 2 тур: Определите импульс и энергию тел (10 баллов)**

Два тела движутся по взаимно перпендикулярным направлениям.

Масса первого тела 10.0 кг, а скорость 6.0 м/с. Масса второго 6.0 кг и скорость 10.0 м/с. Определите модуль полного импульса  $P$  данной системы тел и кинетическую энергию  $E$  системы.

Ответ вводить с точностью до целых.

Введите ответ:

Импульс  $P$

Н·с [Правильно](#)

Энергия  $E$

Дж [Неправильно](#)

Рис. 3.14. Просмотр задачи на Java-версии программного комплекса *distolymp*.

Как и в PHP-версии программного комплекса *distolymp*, в Java-версии просмотр задач никак не влияет на значения в базе данных и реализован так, как задача будет выглядеть у пользователя при прохождении олимпиады. При этом так как пользователи проходят олимпиаду с помощью проигрывателя BARSIC, то просмотр задачи был адаптирован и протестирован для просмотра из данного проигрывателя.

После отправки ответов на сервер администратор может посмотреть таблицу результатов. Таблица на PHP-версии представлена на Рис. 3.15.

Название	Ответ	Правильный ответ	Правильность	Баллы
h1	2=2	0 ± 0.05	Неправильно	0
Итого:			0 (из 1)	0 (из 5)

Параметры задания  
v1=6; v2=6; m1=7; m2=6; p1=42; p2=36;

Рис. 3.15. Результаты решения задачи на PHP-версии программного комплекса *distolymp*.

На изображении выше видно, что выводится таблица с ответами, правильностью и баллами, а ниже поле с информацией о параметрах (переменных) для задачи.

Название	Ответ	Правильный ответ	Правильность	Баллы
p	84	84.8528137423857 ± 2	Правильно	5
e	13	480.0 ± 2	Неправильно	0
Итого:			1 (из 2)	5.0 (из 10.0)

Информация о задании

Параметры:  
p1 = 60.0;  
p2 = 60.0;  
m1 = 10.0;  
m2 = 6.0;  
v1 = 6.0;  
v2 = 10.0;

Ответы:  
p=sqrt(p1\*p1+p2\*p2)+-2;  
e=(m1\*v1\*v1/2+m2\*v2\*v2/2)+-2

Рис. 3.16. Результаты решения задачи на Java-версии программного комплекса *distolymp*.

В Java-версии (Рис. 3.16) было решено улучшить блок с информацией о задании. Теперь в нём выводятся не только значения переменных, но и ответы. Таким образом, можно полностью вручную проверить правильность

расчётов компьютера. Также теперь если задача не содержит числового ответа, то таблица с ответами не появляется, а вместо неё в блоке «Информация о задании» выводится возможное количество баллов за задачу. Так как остальная информация из таблицы в данном случае бессмысленна.

Копирование задачи реализовано так, что копируются не только строки в базе данных, но и изображения для задачи. Таким образом, скопированная задача никак не связана с копируемой. Удаление задачи также удаляет и изображение к задаче. Кнопка «Создать новую задачу» просто открывает пустую форму редактирования задачи.

В итоге Java-версия вкладки работы с задачами была реализована в полном объёме и значительно улучшена.

### **3.4. Реализация модуля выполнения моделей BARSIC**

Модели BARSIC являются главной особенностью программного комплекса distolymp. По своей сути они являются виртуальными лабораторными работами. Так как вся олимпиада выполняется с помощью проигрывателя BARSIC, то переход от задач и тестов к виртуальным лабораторным работам проходит незаметно для пользователей.

На Рис. 3.17 представлен вид страницы моделей BARSIC на PHP-версии программного комплекса distolymp.

Префикс	<input type="text" value="Образец модели"/>
Заголовок	<input type="text" value="Олимпиада, модель - Определите по графику путь и перемещение тележки (20 баллов)"/>
Задание	<pre>&lt;h3 align=center&gt;Олимпиада, модель: Определите по графику путь и перемещение тележки (20 баллов) &lt;/h3&gt; Измерьте начальную координату тележки, её координату в момент времени [t1] сек после начала движения (нажать кнопку &amp;quot;Пуск&amp;quot;), а также путь, пройденный тележкой до этого момента. и соответствующую величину перемещения тележки. Все</pre>
Изображение	<input type="button" value="Обзор..."/> Файл не выбран.      Width: <input type="text" value="400"/> Height: <input type="text" value="50"/> <input type="button" value="Просмотр"/> <input type="checkbox"/> Удалить
Комментарий	<input type="text"/>
Значения переменных (?)	<pre>lenRail=10; CarX0=(1.1..1.5,0.1); CarV0=(1.31..1.43,0.01); t1=(8..11,0.5); s1=CarV0*t1;</pre>
Ответы	<pre>result1=CarX0+0.021; result2= r1+0.021; result3= s1+0.042;</pre>
Количество баллов: (?)	<input type="text" value="5;5;5;5"/>
Штраф за неправильный ответ:	<input type="text" value="4"/>
Минимальное количество баллов за правильный ответ:	<input type="text" value="0"/>
Файл BARSIC	<input type="button" value="Обзор..."/> Файл не выбран.      View
Отчет	<pre>&lt;html&gt; &lt;head&gt; &lt;meta http-equiv="content-type" content="text/html; charset=UTF-8"&gt; &lt;Script&gt; function commaToDot(s) {var n=s.indexOf(","); var s_new</pre>
<input type="button" value="Сохранить"/> <input type="button" value="Отменить"/>	

№	Заголовок	Статус	Действия
1	prefix : title	Новая	просмотр редактировать копировать удалить архивировать
2	Образец модели : Олимпиада, модель - Определите по графику путь и перемещение тележки (20 баллов)	Опубликована	просмотр редактировать копировать удалить архивировать
3	Удалено : Удаленное задание	Опубликована	просмотр редактировать копировать удалить архивировать

Рис. 3.17. Главная страница моделей BARSIC на PHP-версии программного комплекса distolymp.

Как и в случае с задачами в PHP-версии комплекса на главной странице моделей BARSIC смешан список существующих моделей и форма редактирования/создания. Поэтому так же как и в случае с задачами было принято решение разделить данную страницу на две: список моделей и форма редактирования. Для удобства было решено добавить поиск. Полученный в результате список моделей показан на Рис. 3.18.

№	Заголовок	Статус	Действия
1	Образец модели: Олимпиада, модель - Определите по графику путь и перемещение тележки (20 баллов)	Новая	просмотр редактировать копировать удалить

**Создать новую модель**

*Рис. 3.18. Список моделей BARSIC на Java-версии программного комплекса distolymp.*

Кнопка «архивировать» была убрана, так как не использовалась на практике. Но обратную кнопку «разархивировать» было решено оставить, так как в базе данных могут оставаться случайно заархивированные модели. Был добавлен поиск по заголовку и префиксу модели. Модели в списке отсортированы по времени создания (сначала новые).

Вид формы редактирования на Java-версии комплекса представлен на Рис. 3.19.



Префикс	Образец модели ✓
Заголовок	Олимпиада, модель - Определите по графику путь и пере ✓
Задание	<p>&lt;h3 align=center&gt;Олимпиада, модель: Определите по графику путь и перемещение тележки (20 баллов)</p> <p>▼ <b>Специальные символы</b></p> <p>&lt; &gt; ° ± · × ÷ ² ³ x<sup>y</sup> x<sub>y</sub> √ " ' § ¶ \ Ж К Ц Ъ  α β γ δ ε η θ λ μ ν ξ π ρ σ τ φ ω Δ Φ Ψ Ω</p> <p>Файл не выбран <span>Обзор</span></p>
Изображение	<p>Ширина: 500 ✓</p> <p>Высота: 0 ✓</p> <p><input type="checkbox"/> Удалить</p> <p><b>Просмотр</b></p>
Значения переменных <sup>(?)</sup>	lenRail=10; CarX0=(1.1..1.5,0.1);
Ответы	result1=CarX0+0.021; result2= r1+0.021;
Количество баллов	5;5;5;5
Штраф за неправильный ответ	4.0 ✓
Минимальное количество баллов за правильный ответ	0.0 ✓
Файл BARSIC	Файл не выбран <span>Обзор</span>
	<b>Просмотр</b>

Рис. 3.19. Форма редактирования модели BARSIC на Java-версии программного комплекса distolymp.

Вид формы редактирования значительно изменился. Поля «Заголовок», «Задание», «Штраф за неправильный ответ» и «Минимальное количество баллов за правильный ответ» являются обязательными для ввода. Появился выпадающий блок «Специальные символы» с HTML-символами, которые

используются для оформления текста задачи и ответов. Поле «Комментарий» было убрано, так как не использовалось на практике. Теперь на форме присутствует валидация как на стороне клиента, так и на стороне сервера. Поля, отмеченные зелёной галочкой (либо красным восклицательным знаком), проверяются на стороне клиента и на стороне сервера. Неотмеченные поля, в силу сложности валидации, проходят проверку только на сервере. Сообщения об ошибках в Java-версии комплекса появляются рядом с полем для ввода вместо всплывающих сообщений в PHP-версии.

Поля «Значения переменных», «Ответы» и «Количество баллов» заполняются специальным образом. Логика заполнения и работы этих полей и связанной с ними функциональности такая же, как и во вкладке «Задачи» и была подробно описана в предыдущем пункте.

В Java-версии при загрузке изображения к задаче нет необходимости задавать ширину и высоту, достаточно только одного параметра, изображение будет масштабировано по нему (можно, как и в PHP-версии, задать оба параметра). Просмотр загруженного изображения теперь производится во всплывающем окне вместо отдельной вкладки.

Было убрано поле «Отчёт», которое в PHP-версии определяет страницу ответов, которую пользователь будет заполнять при выполнении лабораторной работы. В Java-версии данное поле полностью генерируется автоматически, на основе данных из полей «Значения переменных» и «Ответы». Но для обеспечения корректной работы PHP-версии данное поле из базы данных не удаляется.

Для сохранения модели необходимо загрузить файл модели BARSIC, так как без него модель не имеет смысла. Поэтому данное поле имеет валидацию по расширению файла и его размеру как на стороне клиента, так и на стороне сервера. Нажатием на кнопку «Просмотр» ниже поля для файла можно скачать BARSIC-файл, относящийся к данной модели.

Для просмотра созданной модели необходимо нажать на кнопку «просмотр» на главной странице моделей BARSIC. В случае с моделями

сначала появится предстартовая страница с условием выполнения, в котором так же как и в задачах, имена переменных в фигурных или квадратных скобках были заменены на сгенерированные значения (Рис. 3.20).

**Олимпиада, модель: Определите по графику путь и перемещение тележки (20 баллов)**

Измерьте начальную координату тележки, её координату в момент времени 8.5 сек после начала движения (нажать кнопку "Пуск"), а также путь, пройденный тележкой до этого момента, и соответствующую величину перемещения тележки. Все величины определите с точностью до сотых, занесите результаты в отчёт и отошлите его на сервер.

Вы можете воспользоваться следующими средствами, если решите, что они вам необходимы:

- Увеличительное стекло - позволяет увеличивать изображение выбранной области окна. Нажатие мышью в любой части того же окна восстанавливает первоначальный масштаб.
- Выделение мышью области графика (нажать кнопку мыши и вести вправо вниз, а затем отпустить кнопку)- позволяет увеличивать изображение выбранной области графика. При необходимости можно опять выбрать нужный участок графика для показа во всём окне. И так далее.

Движение в обратном направлении (справа налево снизу вверх) в любой части того же окна либо вызов правой кнопкой мыши всплывающего меню и выбор пункта "Восстановить масштаб" восстанавливает первоначальный масштаб графика.

Задание разрешено переделывать, но за повторную отсылку результатов на сервер начисляется 4 штрафных балла, вычитаемых из полученной за задание оценки.

**Этот файл может быть правильно просмотрен только из проигрывателя barsic.exe версии 11.85 или более поздней**

[Начать выполнение практического задания](#)

*Рис. 3.20. Предстартовая страница с условием выполнения модели BARSIC на Java-версии программного комплекса distolymp.*

Ниже блока с текстом модели расположена кнопка, которая запускает саму модель BARSIC с заданными в условии значениями. В PHP-версии комплекса запуск самой модели в режиме тестирования (просмотра) был невозможен, в отличие от Java-версии. Запущенная модель в проигрывателе BARSIC представлена на Рис. 3.21.

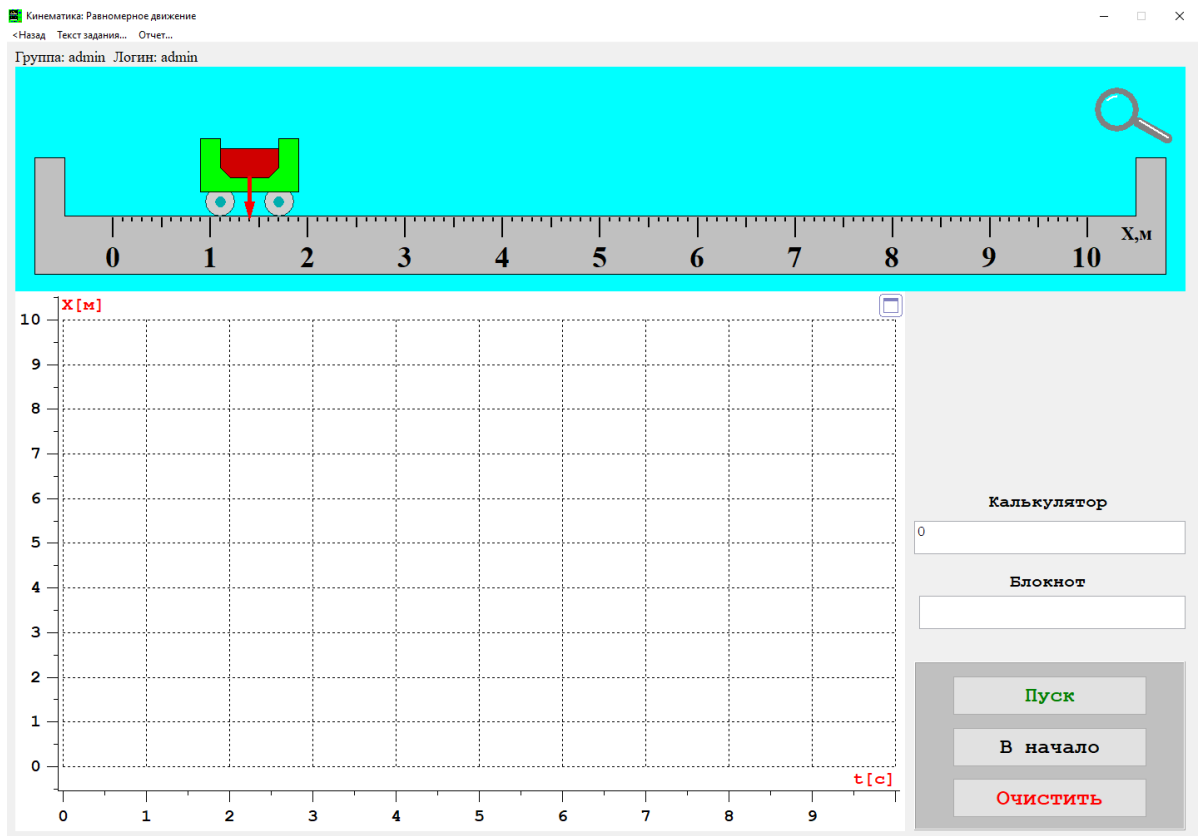


Рис. 3.21. Запущенная модель BARSIC в режиме просмотра на Java-версии программного комплекса distolymp.

В верхнем меню пункт «Назад» выходит из выполнения модели, пункт «Текст задания» открывает условие лабораторной работы (такое же как и на предстартовой странице модели), пункт «Отчёт» открывает страницу с полями для ответов, которые пользователю необходимо заполнить для выполнения работы (Рис. 3.22).

Отчет

Название величины	Ответ
X0	<input type="text"/> (см)
X1(t1)	<input type="text"/>
Путь	<input type="text"/>
S	<input type="text"/>

Рис. 3.22. Форма отчёта для выполнения модели BARSIC на Java-версии программного комплекса distolymp.

В PHP-версии данная форма полностью прописывалась вручную в виде HTML кода, что было неудобно, особенно для пользователей, которые незнакомы с языком разметки HTML. В Java-версии эта проблема была решена автоматической генерацией данной формы, что также позволяет избежать ошибок ручного ввода.

В процессе выполнения лабораторной работы данная форма заполняется и отправляется на сервер. После отправки формы пользователю возвращается таблица результатов (Рис. 3.23).

Название	Ответ	Результат	Баллы
X0	1.4	Правильно	5
X1(t1)	12	Неправильно	0
Путь	24.8	Неправильно	0
S	53	Неправильно	0
За текущую попытку:			5.0
Штрафных баллов:			-
Итого за задание:			5.0 (из 20.0)

Очистить      Закрыть

Рис. 3.23. Таблица результатов выполнения модели BARSIC на Java-версии программного комплекса *distolymp*.

Штрафные баллы не рассчитываются, так как в режиме просмотра (тестирования) модели никакие значения в базу данных не записываются, а значит, штрафные баллы рассчитать не представляется возможным. Кнопка «Очистить», соответственно, убирает таблицу с результатами и выводит обратно пустую форму для ответов.

Копирование моделей реализовано таким образом, что копируются не только значения в базе данных, но и файлы модели (изображение, BARSIC-файл). В итоге скопированная модель никак не связана с копируемой. Удаление модели также удаляет все файлы, связанные с

моделью. Кнопка «Создать новую модель» на главной странице просто открывает пустую форму редактирования модели.

Учитывая описанное в данном пункте, можно сказать, что вкладка «Модели» была не только полностью реализована, но и значительно улучшена по сравнению с PHP-версией программного комплекса distolymr.

### **3.5. Модуль выполнения тестов**

Тесты — это один из трёх типов заданий в программном комплексе distolymr. Они представляют собой классический тест с вопросами и вариантами ответа на них. В PHP-версии комплекса для создания теста необходимо было использовать не только сам программный комплекс, но и стороннюю программу «testEditor» («Редактор тестов»), написанную на языке программирования BARSIC. Данная программа использовалась для генерации PHP-файла, который потом вручную загружался в программный комплекс distolymr. В этом файле содержится информация о вопросах к тесту и некоторые его настройки. Другой блок настроек теста расположен непосредственно в базе данных комплекса distolymr.

Данный подход имеет несколько недостатков. Во-первых, такой способ создания тестов неудобен из-за необходимости использовать несколько программ. Во-вторых, из-за программы «Редактор тестов» создавать и редактировать вопросы в тесте возможно только с ПК. В-третьих, при ручной загрузке файлов на сервер есть вероятность ошибиться с именем файла, что приведёт к некорректной работе теста. Исходя из вышесказанного, было принято решение разработать и внедрить в программный комплекс distolymr функциональность программы «Редактор тестов».

#### **3.5.1. Создание редактора тестов**

«Редактор тестов» — это программа, написанная на языке программирования BARSIC и выполняемая в проигрывателе BARSIC. Она позволяет создавать тесты и настраивать некоторые их параметры, такие как: порядок вопросов в тесте (случайный или по порядку), наличие кнопки

«Пропустить вопрос», количество вопросов определённой сложности в тесте при выборе случайного их порядка. Также в этой программе редактируются вопросы теста. Вид данной программы показан на Рис. 3.24.

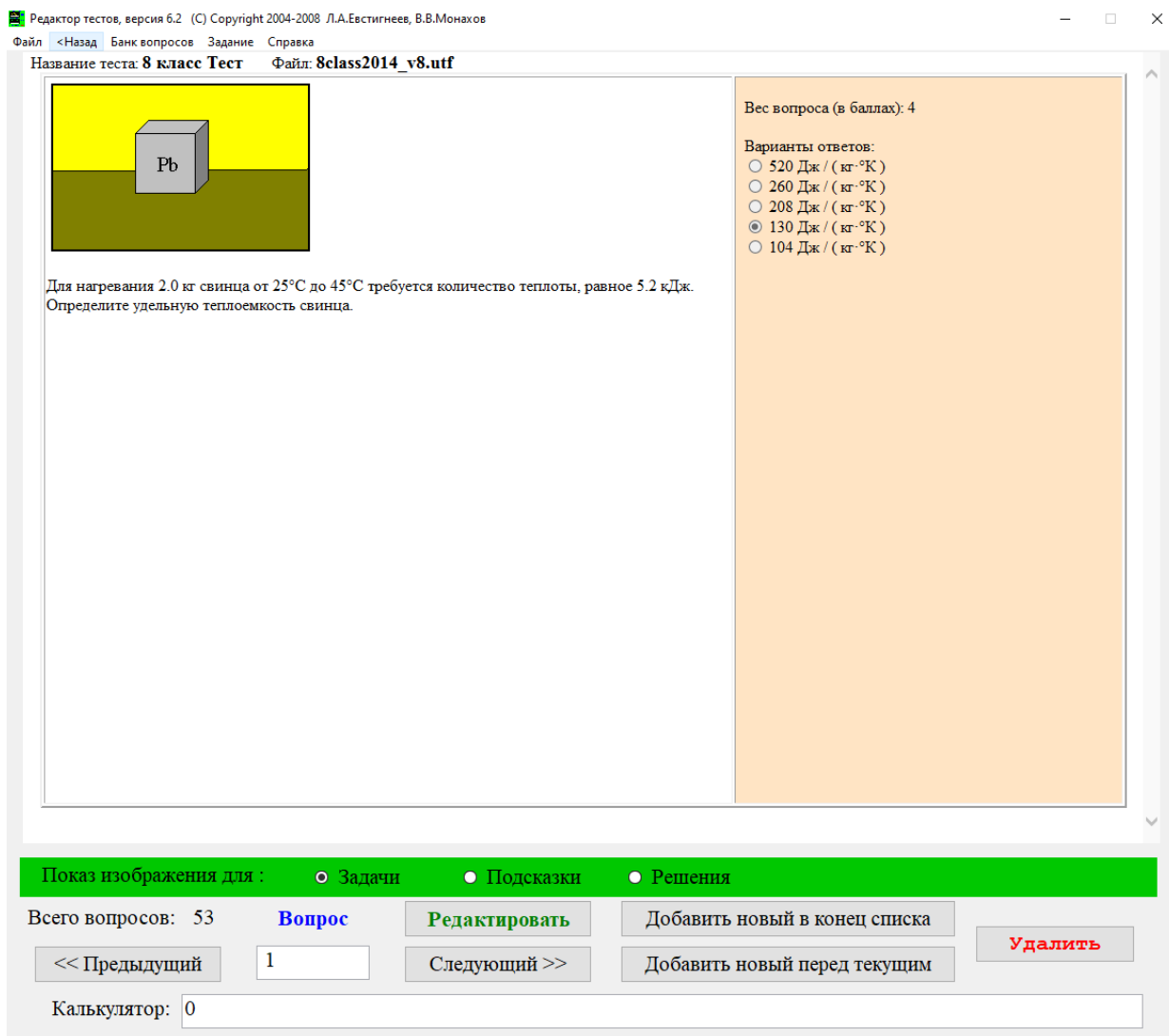


Рис. 3.24. Общий вид программы «Редактор тестов».

В программе «Редактор тестов» можно редактировать, добавлять и удалять вопросы в тесте, менять тип и сложность вопросов, а также загружать изображение для вопроса. Вид формы редактирования вопроса в этой программе показан на Рис. 3.25.

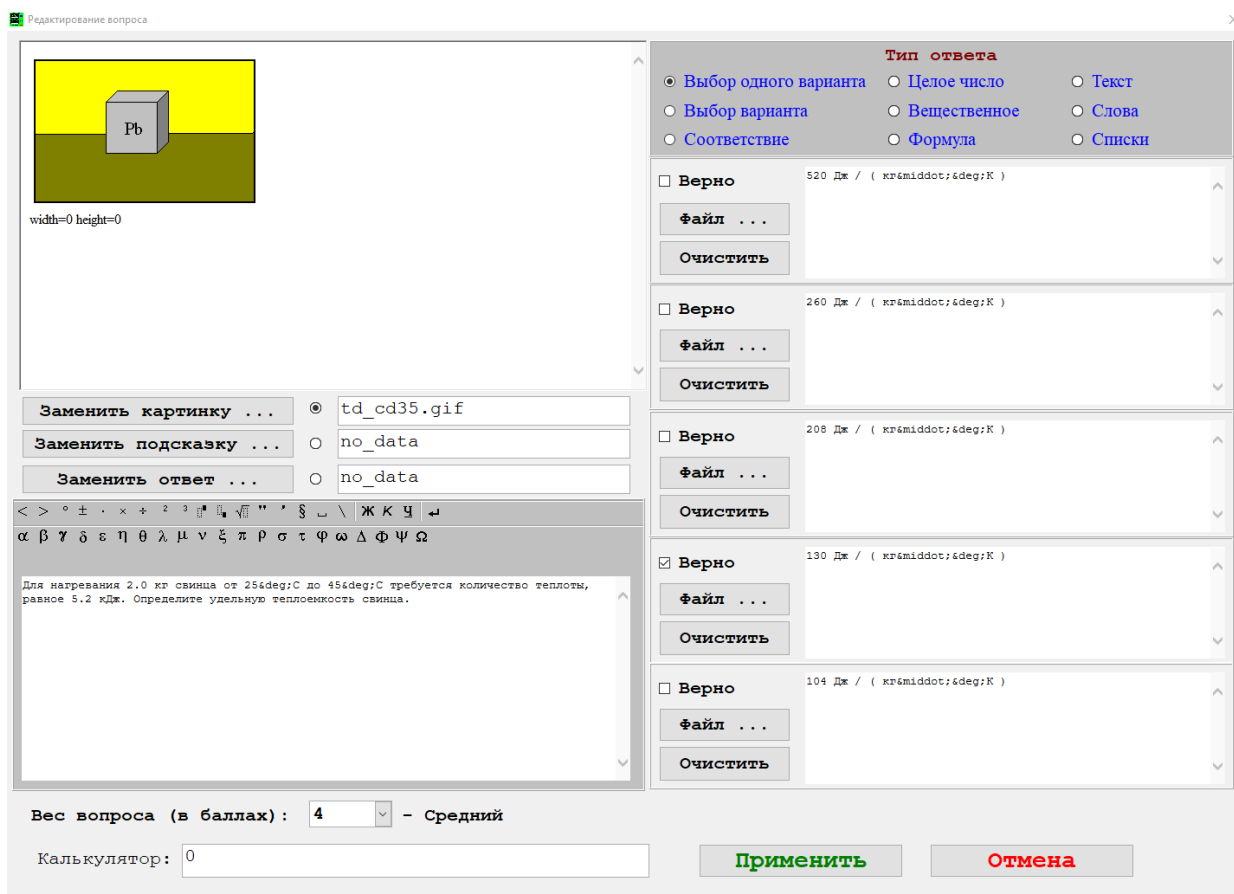


Рис. 3.25. Форма редактирования вопроса в программе «Редактор тестов».

При создании нового редактора тестов в программном комплексе distolymr было решено добавить некоторые улучшения, такие как:

- валидация текста вопроса и ответов на него;
- вырезание всей неиспользуемой функциональности (например, варианты ответа на вопрос в виде картинок, подсказки и так далее);
- возможность изменять порядковый номер вопроса в тесте.

По итогам разработки функциональность программы «Редактор тестов», которая отвечала за общие параметры теста, была вынесена на форму редактирования теста к другим его параметрам, хранящимся в базе данных. Также на форму редактирования теста был добавлен список всех вопросов теста и кнопки взаимодействия с ними (Рис. 3.26).



Добавить новый вопрос				
№	Вопрос	Тип	Сложность	Действия
1	Что такое физика?	Один ответ	Лёгкий	редактировать удалить
2	Выберите все варианты ответов, которые соответствуют вопросу	Несколько ответов	Лёгкий	редактировать удалить
3	Установите соответствие между физическими величинами и единицами измерения	Соответствие	Лёгкий	редактировать удалить
4	Длина экватора Земли?	Целочисленный ответ	Лёгкий	редактировать удалить
5	Рассчитайте число $\pi$	Вещественный ответ	Лёгкий	редактировать удалить

Рис. 3.26. Список вопросов на форме редактирования теста на Java-версии программного комплекса *distolymp*.

Кнопка «удалить» в данном списке удаляет вопрос из теста. Кнопка «редактировать» и кнопка «Добавить новый вопрос» открывают форму редактирования вопроса, только во втором случае она будет пустая. Вид формы редактирования вопроса в Java-версии редактора показан на Рис. 3.27.

Номер вопроса:

Тип вопроса:

Сложность вопроса:

Изображение:

Текст вопроса:

▼ Специальные символы

<	>	°	±	·	×	÷	²	³	x <sup>y</sup>	x <sub>y</sub>	√	"	'	§	⌂	\	Ж	К	Ц	↵
α	β	γ	δ	ε	η	θ	λ	μ	ν	ξ	π	ρ	σ	τ	φ	ω	Δ	Φ	Ψ	Ω

Ответ:

Сила	Н
Сила тока	А
Напряжение	В
Путь	м

Рис. 3.27. Форма редактирования вопроса на Java-версии программного комплекса *distolymp*.

В Java-версии ключевые особенности редактора были воссозданы, такие как: смена типа и сложности вопроса, палитра со специальными символами, возможность загрузки и просмотра загруженного изображения.

Также была добавлена такая функциональность как: валидация текста вопроса и ответов, валидация типа и размера изображения. При этом валидация производится как на стороне клиента, так и на стороне сервера. Возможность изменять порядковый номер вопроса, а новому вопросу устанавливать любой порядковый номер в тесте.

Для обеспечения работы редактора Java-версии комплекса был написан парсер PHP-файла, в котором хранятся параметры теста. Также был написан генератор, который создаёт данный PHP-файл по входным данным. Это необходимо не только для хранения параметров, но и для обеспечения корректной работы PHP-версии системы даже с теми тестами, которые были полностью созданы в редакторе на Java-версии программного комплекса.

По результатам работы, был разработан и внедрён в программный комплекс `distolymr` редактор тестов, написанный на языке программирования Java. Были решены все проблемы, выявленные в пункте 3.5.

- Отпала надобность использования нескольких программ.
- Так как редактор тестов теперь является частью программного комплекса `distolymr`, то это автоматически открывает возможность редактирования тестов не только с ПК, но и с мобильных устройств.
- Исчезла необходимость ручной загрузки файлов на сервер с контролем имени загружаемого файла.

### **3.5.2. Реализация модуля выполнения тестов**

Как уже было сказано ранее, тесты в программном комплексе `distolymr` представляют собой классические тесты с различными типами вопросов. Это делает тесты очень гибким инструментом, так как, комбинируя разные типы вопросов между собой, можно проверить как навыки счета и логики, так и сухие теоретические знания.

На Рис. 3.28 показано как выглядит основная вкладка тестов в PHP-версии программного комплекса distolymp.

Префикс	<input type="text" value="Образец теста"/>		
Заголовок	<input type="text" value="Тест"/>		
Задание	<div style="border: 1px solid #ccc; padding: 5px; min-height: 150px;">Выполните следующие задания.</div>		
Изображение	<input type="button" value="Обзор..."/> <small>Файл не выбран.</small>	Width: <input type="text" value="0"/>	Height: <input type="text" value="0"/>
Комментарий	<input type="text"/>		
	<input type="checkbox"/> Не показывать результат пользователю <input type="checkbox"/> Автоматически отправлять результаты на сервер		
Количество баллов за весь тест:	<input type="text" value="7"/>		
Штраф за неправильный ответ:	<input type="text" value="1"/>		
Минимальное количество баллов за правильный ответ:	<input type="text" value="1"/>		
Файл testNavigator (*.php)	<input type="button" value="Обзор..."/> <small>Файл не выбран.</small>	<a href="#">View</a>	
Файл с тестом (*_param.php)	<input type="button" value="Обзор..."/> <small>Файл не выбран.</small>	<a href="#">View</a>	<a href="#">Edit</a>
Добавить файл в папку с тестом	<input type="button" value="Обзор..."/> <small>Файл не выбран.</small>	<a href="#">View</a>	<a href="#">[+]</a>
	<input type="button" value="Сохранить"/>		<input type="button" value="Отменить"/>

№	Заголовок	Статус	Действия
1	Образец теста : Тест	Новая	<a href="#">просмотр</a> <a href="#">редактировать</a> <a href="#">копировать</a> <a href="#">удалить</a> <a href="#">архивировать</a>

Рис. 3.28. Главная страница тестов на PHP-версии программного комплекса distolymp.

Как в случае с задачами и моделями BARSIC главная страница тестов представляет собой форму редактирования теста и список существующих тестов. Поэтому, так же как и в предыдущих случаях разделим данную страницу на две. Вид страницы со списком тестов представлен на Рис. 3.29.

Фильтр тестов по заголовку и префиксу. <input type="text"/> <input type="button" value="Поиск"/>			
№	Заголовок	Статус	Действия
1	1	Новый	<a href="#">просмотр</a> <a href="#">редактировать</a> <a href="#">копировать</a> <a href="#">удалить</a>
2	Тест 8 класс: Контрольный тест	Новый	<a href="#">просмотр</a> <a href="#">редактировать</a> <a href="#">копировать</a> <a href="#">удалить</a>

Рис. 3.29. Список тестов на Java-версии программного комплекса *distolymp*.

Список существующих тестов был вынесен на отдельную страницу. В верхнюю часть страницы был добавлен поиск по заголовку и префиксу теста. В списке тесты отсортированы в порядке создания (сначала новые). Кнопка «архивировать» была убрана, так как не использовалась на практике. Но при этом кнопку «разархивировать» было решено оставить, так как в базе данных могут быть случайно заархивированные тесты.

Форма редактирования теперь так же представляет собой отдельную страницу. Чтобы попасть на неё, необходимо нажать на кнопку «редактировать», либо «Создать новый тест». В последнем случае форма редактирования откроется незаполненная. Вид данной формы представлен на Рис. 3.30.

Префикс	Тест 8 класс ✓																																										
Заголовок	Контрольный тест ✓																																										
Задание	Выполните следующие задания ✓ <b>▼ Специальные символы</b> <table border="1"> <tr> <td>&lt;</td><td>&gt;</td><td>°</td><td>±</td><td>·</td><td>×</td><td>÷</td><td>²</td><td>³</td><td>x<sup>y</sup></td><td>x<sub>y</sub></td><td>√</td><td>"</td><td>'</td><td>§</td><td>⌂</td><td>\</td><td>Ж</td><td>К</td><td>Ц</td><td>↵</td> </tr> <tr> <td>α</td><td>β</td><td>γ</td><td>δ</td><td>ε</td><td>η</td><td>θ</td><td>λ</td><td>μ</td><td>ν</td><td>ξ</td><td>π</td><td>ρ</td><td>σ</td><td>τ</td><td>φ</td><td>ω</td><td>Δ</td><td>Φ</td><td>Ψ</td><td>Ω</td> </tr> </table> Файл не выбран <span>Обзор</span>	<	>	°	±	·	×	÷	²	³	x <sup>y</sup>	x <sub>y</sub>	√	"	'	§	⌂	\	Ж	К	Ц	↵	α	β	γ	δ	ε	η	θ	λ	μ	ν	ξ	π	ρ	σ	τ	φ	ω	Δ	Φ	Ψ	Ω
<	>	°	±	·	×	÷	²	³	x <sup>y</sup>	x <sub>y</sub>	√	"	'	§	⌂	\	Ж	К	Ц	↵																							
α	β	γ	δ	ε	η	θ	λ	μ	ν	ξ	π	ρ	σ	τ	φ	ω	Δ	Φ	Ψ	Ω																							
Изображение	Ширина: 200 ✓ Высота: 0 ✓ <input type="checkbox"/> Удалить <b>Просмотр</b> <input checked="" type="checkbox"/> Показывать результаты пользователю <input checked="" type="checkbox"/> Случайный порядок вопросов в тесте Легких вопросов (2 балла): 6 Средних вопросов (4 балла): 6 Сложных вопросов (6 баллов): 4 Количество баллов в тесте: 60 <input checked="" type="checkbox"/> Кнопка "Пропустить"																																										
Параметры																																											
Количество баллов за весь тест	20.0 ✓																																										
Штраф за неправильный ответ	1.0 ✓																																										
Файл с тестом	Файл не выбран <span>Обзор</span> <b>Скачать файл с тестом</b>																																										

Рис. 3.30. Форма редактирования теста на Java-версии программного комплекса distolymp.

В Java-версии форма претерпела значительные изменения. Поля «Заголовок», «Задание» и «Штраф за неправильный ответ» являются обязательными для заполнения. Появился блок с HTML-символами для удобства оформления текста задания.

У изображения необязательно (но можно) задавать и ширину, и высоту. Достаточно будет ввести один параметр. А просмотр изображения теперь происходит во всплывающем блоке, а не в отдельном окне.

Появилась валидация всех полей как на стороне клиента, так и на стороне сервера, в том числе и для загружаемых файлов. А сообщения об ошибках появляются не во всплывающих сообщениях, а рядом с неверно заполненным полем.

Была убрана галочка «Автоматически отправлять результаты на сервер», так как на практике она всегда использовалась. Эта опция в Java-версии включена по умолчанию. Также было убрано поле «Комментарий», потому что оно не использовалось на практике. Поле «Минимальное количество баллов за правильный ответ» тоже было удалено, так как его значение всегда равно «1». Поле «Файл testNavigator» было решено удалить, так как этот файл для всех тестов одинаковый, и в Java-версии он копируется автоматически для каждого созданного теста. Кроме того, были убраны поля, отвечающие за загрузку изображений к тесту на сервер, потому что теперь это можно сделать в форме редактирования вопроса.

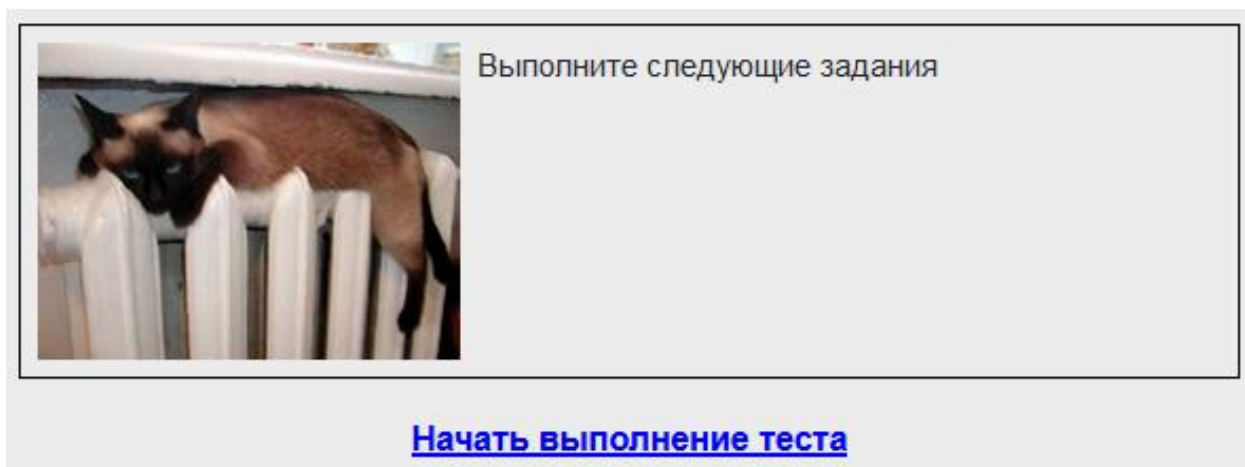
В пункте «Параметры» были добавлены галочки «Случайный порядок вопросов в тесте» и «Кнопка „Пропустить“». Данные опции появились из-за внедрения в программный комплекс distolymp редактора тестов. Ранее данные опции можно было настроить исключительно из стороннего редактора «Редактор тестов». Кроме того, если выбрать случайный порядок вопросов в тесте, то появятся выпадающие списки с настройкой количества вопросов определённой сложности в тесте. Ниже автоматически рассчитывается сумма баллов выбранных вопросов.

Поле «Файл с тестом» для большей гибкости комплекса было решено оставить. Таким образом, RHP-файл с параметрами и вопросами теста можно (но не обязательно) загрузить вручную, как и в RHP-версии. Если данный файл не загружать на сервер, то он сгенерируется автоматически. Кроме того,

данный файл можно скачать для использования в другом тесте или редактирования с помощью соответствующей кнопки.

Ниже формы редактирования теста расположен список вопросов в нём (Рис. 3.26). Но если тест новый, то список вопросов недоступен. Это объясняется тем, что у нового теста ещё нет своей папки и своего РНР-файла, в котором эти вопросы должны храниться. Таким образом, для редактирования вопросов необходимо сначала сохранить тест.

При нажатии на кнопку «просмотр» на главной странице тестов открывается предстартовая страница, как и в случае с моделями (Рис. 3.31). На ней расположено задание ко всему тесту и картинка для оформления.



*Рис. 3.31. Предстартовая страница теста на Java-версии программного комплекса distolymp.*

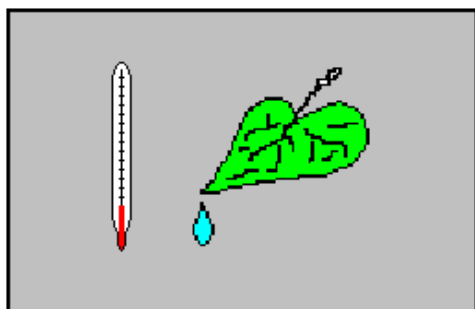
Ниже блока с заданием ко всему тесту расположена кнопка для старта тестирования, при нажатии на которую начинается прохождение теста. Этап прохождения теста на Java-версии отображён на Рис. 3.32.

Название теста: **Контрольный тест**

Задание: 1

Всего заданий: 16

Осталось заданий: 16



Выделяется в окружающую среду или поглощается из окружающей среды энергия при конденсации водяного пара?

Варианты ответов:

- Выделяется.
- Может выделяться, а может и поглощаться.
- Не выделяется и не поглощается.
- Выделяется только при  $T=373\text{ K}$ .
- Поглощается.

Ответить

Пропустить

Прервать работу

*Рис. 3.32. Просмотр теста на Java-версии программного комплекса distolymp.*

При прохождении теста пользователю показывается название теста, прогресс прохождения теста и сам вопрос. Кнопка «Пропустить» настраиваемая и её можно отключить. Кнопка «Прервать работу» засчитывает ответ на текущий вопрос и завершает тестирование, выводя пользователю результаты только по тем вопросам, на которые он дал ответ.

Варианты ответов при запуске теста псевдослучайным образом перемешиваются, что исключает заучивание правильного варианта ответа. Если в настройках теста был выбран случайный порядок вопросов в тесте, то сами вопросы также псевдослучайно выбираются и перемешиваются.



Так как пользователи проходят олимпиаду полностью с использованием проигрывателя BARSIC, то просмотр теста был адаптирован, в том числе для этого проигрывателя.

После завершения прохождения теста, если в его настройках был выбран пункт «Показывать результаты пользователю», будет отображена страница, показанная на Рис. 3.33.

Время начала теста: **11.05.2022 20:06**  
 Вы правильно ответили на **7** из **16**  
 Время прохождения теста: **00:15:27**  
 Набрано **43,75%** от максимального балла

№	Сложность	Вопрос	Правильность
1	Лёгкий	Выделяется в окружающую среду или поглощается из окружающей среды энергия при конденсации водяного пара? Варианты ответов: <input type="radio"/> Выделяется. <input type="radio"/> Может выделяться, а может и поглощаться. <input type="radio"/> Не выделяется и не поглощается. <input type="radio"/> Выделяется только при $T=373$ К. <input checked="" type="radio"/> Поглощается.	Неправильно
2	Средний	Пружина жесткостью $k = 240$ Н/м растянулась на 2 см в результате подвешивания некоторого груза. Оцените массу груза $m$ .	Неправильно
3	Лёгкий	На дне ведра намерз лёд. В ведро налили теплую воду и лёд на дне растаял. Изменится ли уровень воды в ведре после таяния льда? Изменится ли вес ведра (вместе с его содержимым)? Испарением можно пренебречь.	Правильно
4	Лёгкий	В каком колене U-образной трубки находится менее плотная жидкость ?	Правильно

Рис. 3.33. Страница результатов выполнения теста, при выбранном пункте «Показывать результаты пользователю» на Java-версии программного комплекса *distolymp*.

На вышеприведённой странице пользователь может посмотреть дату начала тестирования, время, затраченное на его прохождение, правильность

выполнения теста, а также таблицу с вопросами. В этой таблице отображены все вопросы теста в том порядке, в котором они были предложены пользователю, их сложность, текст и правильность ответа пользователя.

При нажатии на текст вопроса раскрывается блок со всеми ответами на него в том порядке, в котором они были представлены пользователю. Также там будет отмечен (вписан) ответ пользователя на данный вопрос. Таким образом, пользователь может найти и проанализировать свои ошибки в тесте.

Если же при настройке теста не выбирать пункт «Показывать результаты пользователю», то на странице результатов будет выведена та же информация, но без таблицы с вопросами.

Так как каждый тест на сервере хранится в своей папке, то при копировании, кроме значений в базе данных, копируется ещё и всё содержимое папки с тестом. В итоге скопированный тест никак не связан с копируемым. А удаление теста удаляет не только записи данного теста из базы данных, но и его папку со всем её содержимым.

В общей сложности модуль тестов был сильнее всего модернизирован по сравнению со своей PHP-версией. Главная причина — это встроенный редактор тестов в Java-версию программного комплекса distolymp. Помимо этого, в целом была улучшена как визуальная, так и функциональная составляющая вкладки «Тесты» по сравнению с PHP-версией.

### **3.6. Обсуждение результатов**

В ходе данной работы были решены все поставленные задачи, и, соответственно, достигнута цель работы — разработаны и реализованы модули выполнения заданий в Java-версии программного комплекса distolymp.

Помимо основной цели работы, были решены и другие задачи:

- разработка новой схемы структуры используемой базы данных;
- разработка справочников городов и классов (параллелей) обучения;

- разработка и внедрение в программный комплекс distolymp редактора тестов.

Для оценки качества написания кода в Приложении приведена его часть, отвечающая за основную логику в модуле решения задач. Предоставить код Java-версии программного комплекса в полном объёме не представляется возможным из-за соображений безопасности и размера.

Так как Java-версия комплекса находится в стадии активной разработки, то, безусловно, существуют моменты, которые необходимо улучшить. На данный момент в приложении полностью отсутствует тестирование, что в будущем сильно усложнит отладку и модернизацию комплекса. Поэтому, на мой взгляд, внедрение системы тестирования кода является одной из первостепенных задач. Поскольку на данный момент в разработке Java-версии комплекса принимают участие несколько человек, то считаю необходимым внедрение документации кода встроенными средствами Java. Это ускорит процесс включения новых разработчиков в работу и саму разработку.

По итогам проделанной работы Java-версия программного комплекса distolymp получила все возможности по управлению задачами, моделями BARSIC и тестами. Но остальные модули (аутентификация и авторизация пользователей, группы, кабинет пользователя и настройки) программного комплекса нуждаются в разработке.

### **Выводы по главе 3**

- Проанализирована структура базы данных программного комплекса distolymp. На основе полученных результатов была создана новая схема структуры базы данных, которая используется Java-версией комплекса.
- Реализованы справочники городов и классов (параллелей) обучения. Справочник классов (параллелей) обучения также был модернизирован.

- Реализована и значительно улучшена по сравнению с РНР-версией комплекса вся функциональность, отвечающая за управление задачами.
- Реализована и значительно улучшена по сравнению с РНР-версией комплекса вся функциональность, отвечающая за управление моделями BARSIC.
- Разработан и внедрён в программный комплекс distolymr редактор тестов.
- Реализована и значительно улучшена по сравнению с РНР-версией комплекса вся функциональность, отвечающая за управление тестами.
- В рамках упрощения дальнейшей разработки Java-версии программного комплекса предложено разработать и внедрить системы тестирования и документации кода.

## Выводы

- В результате проведённого анализа структуры имеющейся базы данных программного комплекса distolymp разработана новая схема структуры базы данных, которая используется Java-версией комплекса.
- Реализованы справочники городов и классов (параллелей) обучения. Модернизирован имевшийся в РНР-версии комплекса справочник классов (параллелей) обучения.
- Реализована и значительно улучшена по сравнению с РНР-версией комплекса вся функциональность, отвечающая за управление задачами, моделями BARSIC и тестами.
- Разработан и внедрён в программный комплекс distolymp редактор тестов.
- Для дальнейшей разработки Java-версии программного комплекса предложено разработать и внедрить систему тестирования и документации кода.

## Литература

1. Об Интернет-олимпиаде школьников по физике [Электронный ресурс] // Интернет-олимпиада школьников по физике: [сайт]. URL: [http://distolymp2.spbu.ru/olymp/index\\_about.html](http://distolymp2.spbu.ru/olymp/index_about.html) (дата обращения: 24.04.2022).
2. Монахов В.В., Кожедуб А.В., Науменко П.А., Евстигнеев Л.А., Крукелис М.А., Солодовников Д.В., Керницкий И.Б. BARSIC: Программный комплекс, ориентированный на физика-исследователя // Программирование. 2005. Т. 31. № 3. С. 68-80.
3. Интернет-олимпиада школьников по физике [Электронный ресурс] // Приложения языка программирования и среды BARSIC: [сайт]. URL: [http://distolymp2.spbu.ru/www/applicat\\_r.html?ysclid=l2emhm1uil](http://distolymp2.spbu.ru/www/applicat_r.html?ysclid=l2emhm1uil) (дата обращения: 24.04.2022).
4. PHP vs Java: Which is better for web development [Электронный ресурс] // Scand.com: [сайт]. URL: <https://scand.com/company/blog/php-vs-java-difference-comparison> (дата обращения: 24.04.2022).
5. Тимашов Д.А. Разработка структуры Java-версии программного комплекса distolymp. Бакалаврская работа СПбГУ, 2017. 40 с.
6. Усова Д.С. Разработка модулей работы с пользователями Java-версии программного комплекса distolymp. Бакалаврская работа СПбГУ, 2020. 46 с.
7. About Moodle [Электронный ресурс] // Moodle - Open-source learning platform: [сайт]. URL: [https://docs.moodle.org/400/en/About\\_Moodle](https://docs.moodle.org/400/en/About_Moodle) (дата обращения: 29.04.2022).
8. Sakai Features [Электронный ресурс] // Sakai LMS: [сайт]. URL: <https://www.sakailms.org/sakai-lms-features> (дата обращения: 29.04.2022).
9. 20 Free LMS Platforms [Электронный ресурс] // EdApp: The Mobile LMS: [сайт]. URL: <https://www.edapp.com/blog/top-lms-platforms> (дата обращения: 29.04.2022).
10. ChemCollective: Virtual Labs [Электронный ресурс] // ChemCollective: [сайт]. URL: <https://chemcollective.org/vlabs> (дата обращения: 29.04.2022).
11. What is PhET? A short introduction to the PhET simulations [Электронный ресурс] // PhET Interactive Simulations: [сайт]. URL: <https://phet.colorado.edu/en/about> (дата обращения: 29.04.2022).
12. ВЛР [Электронный ресурс] // ЦВМ: [сайт]. URL: [https://portal.tpu.ru/ceor/v\\_lab](https://portal.tpu.ru/ceor/v_lab) (дата обращения: 29.04.2022).

13. Виртуальные лабораторные работы для очного и дистанционного обучения [Электронный ресурс] // Professional Group: [сайт]. URL: <https://professionalgroup.ru/virtualnyye-laboratorii.html> (дата обращения: 29.04.2022).
14. IntelliJ IDEA [Электронный ресурс] // IntelliJ IDEA: [сайт]. URL: <https://www.jetbrains.com/ru-ru/idea/?ysclid=l2m3x9zlej> (дата обращения: 29.04.2022).
15. Maven [Электронный ресурс] // Maven - Introduction: [сайт]. URL: <https://maven.apache.org/what-is-maven.html> (дата обращения: 29.04.2022).
16. About - Git [Электронный ресурс] // Git: [сайт]. URL: <https://git-scm.com/about> (дата обращения: 29.04.2022).
17. About [Электронный ресурс] // GitHub: [сайт]. URL: <https://github.com/about> (дата обращения: 29.04.2022).
18. Bootstrap [Электронный ресурс] // Bootstrap: [сайт]. URL: <https://getbootstrap.com> (дата обращения: 29.04.2022).
19. jQuery [Электронный ресурс] // jQuery: [сайт]. URL: <https://jquery.com> (дата обращения: 29.04.2022).
20. Thymeleaf [Электронный ресурс] // Thymeleaf: [сайт]. URL: <https://www.thymeleaf.org> (дата обращения: 29.04.2022).
21. Spring | Home [Электронный ресурс] // Spring: [сайт]. URL: <https://spring.io> (дата обращения: 29.04.2022).
22. Introduction to the Spring Framework [Электронный ресурс] // Spring: [сайт]. URL: <https://docs.spring.io/spring-framework/docs/4.3.x/spring-framework-reference/html/overview.html> (дата обращения: 29.04.2022).
23. Hibernate Validator [Электронный ресурс] // Hibernate: [сайт]. URL: <https://hibernate.org/validator> (дата обращения: 30.04.2022).
24. MySQL [Электронный ресурс] // MySQL: [сайт]. URL: <https://www.mysql.com> (дата обращения: 29.04.2022).
25. Hibernate ORM [Электронный ресурс] // Hibernate: [сайт]. URL: <https://hibernate.org/orm> (дата обращения: 30.04.2022).
26. The CentOS Project [Электронный ресурс] // The CentOS Project: [сайт]. URL: <https://www.centos.org> (дата обращения: 29.04.2022).
27. Welcome! [Электронный ресурс] // The Apache HTTP Server Project: [сайт]. URL: <https://httpd.apache.org> (дата обращения: 29.04.2022).

28. Басов Л.В. Разработка элементов программного комплекса distolymp, предназначенного для проведения интернет-олимпиады по физике. Бакалаврская работа СПбГУ, 2011. 39 с.
29. Приложения языка программирования и среды BARSIC [Электронный ресурс] // Интернет-олимпиада школьников по физике: [сайт]. URL: [https://distolymp2.spbu.ru/www/applicat\\_r.html?ysclid=l2oxi2kwlt](https://distolymp2.spbu.ru/www/applicat_r.html?ysclid=l2oxi2kwlt) (дата обращения: 30.04.2022).
30. Oracle VM VirtualBox [Электронный ресурс] // Oracle VM VirtualBox: [сайт]. URL: <https://www.virtualbox.org> (дата обращения: 30.04.2022).



## Приложение

Число разработанных новых классов и интерфейсов Java: 175.

Число модифицированных классов и интерфейсов Java: 3.

Общий объём всего кода Java составляет 10668 непустых строк в 277 классах и интерфейсах.

Представленный ниже интерфейс «TaskService» и реализующий его класс «TaskServiceImpl» приведены в качестве примера небольшой части разработанного кода.

Интерфейс TaskService:

```
1 package ru.spbu.distolymp.service.admin.tasks.api;
2
3 import org.springframework.ui.ModelMap;
4 import ru.spbu.distolymp.dto.entity.answers.AnswerDto;
5 import ru.spbu.distolymp.dto.admin.tasks.TaskFilter;
6 import ru.spbu.distolymp.dto.admin.tasks.TaskListDto;
7 import ru.spbu.distolymp.dto.entity.tasks.TaskDto;
8
9 /**
10  * @author Vladislav Konovalov
11  */
12 public interface TaskService {
13     void fillShowAllTaskModelMap(ModelMap modelMap, int
numberTasksDisplayed);
14     void fillShowAllTaskByFilterModelMap(TaskFilter taskFilter, ModelMap
modelMap);
15     void fillShowAddPageModelMap(ModelMap modelMap);
16     void fillShowEditPageModelMap(Long id, ModelMap modelMap);
17     void fillShowPreviewPageModelMap(Long id, ModelMap modelMap);
18     void fillShowResultModelMap(AnswerDto answerDto, ModelMap modelMap);
19     void addTask(TaskDto taskDto);
20     void updateTask(TaskDto taskDto);
21     void deleteTaskAndImage(Long id);
22     void copyTask(TaskListDto taskDto);
23     void unarchiveTask(Long id);
24 }
```

Класс TaskServiceImpl:

```
1 package ru.spbu.distolymp.service.admin.tasks.impl;
2
3 import org.springframework.data.domain.PageRequest;
4 import org.springframework.data.domain.Pageable;
5 import org.springframework.data.domain.Sort;
```

```

6  import org.springframework.data.jpa.domain.Specification;
7  import org.springframework.stereotype.Service;
8  import org.springframework.transaction.annotation.Transactional;
9  import org.springframework.ui.ModelMap;
10 import org.springframework.web.multipart.MultipartFile;
11 import ru.spbu.distolymp.common.files.FileNameGenerator;
12 import ru.spbu.distolymp.common.files.FileUtils;
13 import ru.spbu.distolymp.common.tasks.parser.PointParser;
14 import ru.spbu.distolymp.common.tasks.resulthandler.TaskResultHandler;
15 import ru.spbu.distolymp.dto.entity.answers.AnswerDto;
16 import ru.spbu.distolymp.dto.admin.tasks.TaskFilter;
17 import ru.spbu.distolymp.dto.admin.tasks.TaskListDto;
18 import ru.spbu.distolymp.dto.admin.tasks.TaskResultDto;
19 import ru.spbu.distolymp.dto.admin.tasks.TaskViewDto;
20 import ru.spbu.distolymp.dto.entity.tasks.TaskDto;
21 import ru.spbu.distolymp.entity.tasks.Task;
22 import ru.spbu.distolymp.exception.common.ResourceNotFoundException;
23 import ru.spbu.distolymp.mapper.admin.tasks.api.TaskListMapper;
24 import ru.spbu.distolymp.mapper.admin.tasks.api.TaskViewMapper;
25 import ru.spbu.distolymp.mapper.entity.tasks.api.TaskMapper;
26 import ru.spbu.distolymp.repository.tasks.TaskRepository;
27 import ru.spbu.distolymp.service.admin.tasks.api.TaskService;
28 import
    ru.spbu.distolymp.service.crud.api.lists.ListingProblemCrudService;
29 import ru.spbu.distolymp.service.crud.impl.tasks.TaskCrudServiceImpl;
30 import ru.spbu.distolymp.util.admin.tasks.TaskSpecsConverter;
31 import java.util.List;
32
33 /**
34  * @author Vladislav Konovalov
35  */
36 @Service
37 public class TaskServiceImpl extends TaskCrudServiceImpl implements
TaskService {
38     private final TaskViewMapper taskViewMapper;
39     private static final Sort SORT_BY_ID_DESC =
Sort.by("id").descending();
40     private static final String TASKS_PARAM = "taskList";
41     private static final String TASK_PARAM = "task";
42
43     public TaskServiceImpl(TaskRepository taskRepository,
44                           ListingProblemCrudService
listingProblemCrudService,
45                           TaskListMapper taskListMapper,
46                           TaskMapper taskMapper,
47                           TaskViewMapper taskViewMapper) {
48         super(taskRepository, listingProblemCrudService, taskListMapper,
taskMapper);
49         this.taskViewMapper = taskViewMapper;
50     }

```

```

51
52     @Override
53     @Transactional(readOnly = true)
54     public void fillShowAllTaskModelMap(ModelMap modelMap, int
numberTasksDisplayed) {
55         List<TaskListDto> taskList = getTasks(numberTasksDisplayed);
56         modelMap.put(TASKS_PARAM, taskList);
57         modelMap.put("taskForCopy", new TaskListDto());
58     }
59
60     private List<TaskListDto> getTasks(int numberTasksDisplayed) {
61         if (numberTasksDisplayed <= 0) {
62             return getTasks(SORT_BY_ID_DESC);
63         }
64         Pageable pageable = getPageableSortedById(numberTasksDisplayed);
65         return getTasks(pageable);
66     }
67
68     private Pageable getPageableSortedById(int numberTasksDisplayed) {
69         return PageRequest.of(0, numberTasksDisplayed, SORT_BY_ID_DESC);
70     }
71
72     @Override
73     @Transactional(readOnly = true)
74     public void fillShowAllTaskByFilterModelMap(TaskFilter taskFilter,
ModelMap modelMap) {
75         int resultSize = taskFilter.getResultSize();
76         Specification<Task> specs =
TaskSpecsConverter.toSpecs(taskFilter);
77         List<TaskListDto> taskDtoList;
78         if (specs == null) {
79             taskDtoList = getTasks(resultSize);
80             modelMap.put(TASKS_PARAM, taskDtoList);
81             return;
82         }
83         if (resultSize == 0) {
84             taskDtoList = getTasksBySpec(specs, SORT_BY_ID_DESC);
85             modelMap.put(TASKS_PARAM, taskDtoList);
86             return;
87         }
88         Pageable sortedByIdDesc = getPageableSortedById(resultSize);
89         taskDtoList = getTasksBySpec(specs, sortedByIdDesc);
90         modelMap.put(TASKS_PARAM, taskDtoList);
91     }
92
93     @Override
94     @Transactional(readOnly = true)
95     public void fillShowAddPageModelMap(ModelMap modelMap) {
96         TaskDto taskDto = new TaskDto();
97         taskDto.setAnswerNote(2);

```

```

98         taskDto.setWidth(0);
99         taskDto.setHeight(0);
100        modelMap.put(TASK_PARAM, taskDto);
101    }
102
103    @Override
104    @Transactional(readonly = true)
105    public void fillShowEditPageModelMap(Long id, ModelMap modelMap) {
106        TaskDto taskDto = getTaskById(id)
107            .map(taskMapper::toDto)
108            .orElseThrow(ResourceNotFoundException::new);
109        modelMap.put(TASK_PARAM, taskDto);
110    }
111
112    @Override
113    @Transactional(readonly = true)
114    public void fillShowPreviewPageModelMap(Long id, ModelMap modelMap)
115    {
116        TaskViewDto taskDto = getTaskById(id)
117            .map(taskViewMapper::toDto)
118            .orElseThrow(ResourceNotFoundException::new);
119        AnswerDto answerDto = new AnswerDto();
120        String[] userAnswers = new
121        String[taskDto.getAnswerNameList().size()];
122        answerDto.setUserAnswers(userAnswers);
123        modelMap.put(TASK_PARAM, taskDto);
124        modelMap.put("answer", answerDto);
125        modelMap.put("result", new TaskResultDto());
126    }
127
128    @Override
129    @Transactional(readonly = true)
130    public void fillShowResultModelMap(AnswerDto answerDto, ModelMap
131    modelMap) {
132        Task task =
133        getTaskById(answerDto.getProblemId()).orElseThrow(ResourceNotFoundException::new);
134        TaskViewDto taskDto = taskViewMapper.toDto(task,
135        answerDto.getParam());
136        TaskResultDto resultDto = TaskResultHandler.toResultDto(
137        taskDto,
138        answerDto.getUserAnswers(),
139        task.getCorrectAnswer(),
140        PointParser.parsePoints(task.getGradePoints()),
141        task.getMaxPoints());
142        modelMap.put(TASK_PARAM, taskDto);
143        modelMap.put("result", resultDto);
144    }
145
146    @Override

```

```

142     @Transactional
143     public void addTask(TaskDto taskDto) {
144         Task task = taskMapper.toEntity(taskDto);
145         MultipartFile image = taskDto.getImage();
146         task.setStatus(1);
147         initFields(task);
148         if ("".equals(image.getOriginalFilename())) {
149             saveOrUpdate(task, false);
150         } else {
151             String imageExtension = FileUtils.getImageExtension(image);
152
153             task.setImageFileName(FileNameGenerator.generateFileName(imageExtension)
154 );
155
156             saveOrUpdate(task, FileUtils.getFileBytes(image));
157         }
158     }
159
160     @Override
161     @Transactional
162     public void updateTask(TaskDto taskDto) {
163         Task task = taskMapper.toEntity(taskDto);
164         initFields(task);
165         MultipartFile image = taskDto.getImage();
166         String oldImageName = task.getImageFileName();
167         if (oldImageName == null) {
168             if ("".equals(image.getOriginalFilename())) {
169                 saveOrUpdate(task, false);
170             } else {
171                 String imageExtension =
172                 FileUtils.getImageExtension(image);
173
174                 task.setImageFileName(FileNameGenerator.generateFileName(imageExtension)
175 );
176
177                 saveOrUpdate(task, FileUtils.getFileBytes(image));
178             }
179         } else {
180             if ("".equals(image.getOriginalFilename())) {
181                 saveOrUpdate(task, taskDto.isDeleteImage());
182             } else {
183                 String imageExtension =
184                 FileUtils.getImageExtension(image);
185
186                 task.setImageFileName(FileNameGenerator.generateFileName(imageExtension)
187 );
188
189                 saveOrUpdate(task, FileUtils.getFileBytes(image),
190 oldImageName, taskDto.isDeleteImage());
191             }
192         }
193     }

```

```

183     private void initFields(Task task) {
184         task.setType(3);
185         if (task.getWidth() == null)
186             task.setWidth(0);
187         if (task.getHeight() == null)
188             task.setHeight(0);
189
190         task.setMaxPoints(PointParser.calculatePoints(task.getGradePoints()));
191     }
192     @Override
193     @Transactional
194     public void deleteTaskAndImage(Long id) {
195         String imageName = getTaskById(id)
196             .orElseThrow(ResourceNotFoundException::new)
197             .getImageFileName();
198         deleteTaskById(id);
199         if (imageName != null) fileService.deleteFile(imageName);
200     }
201
202     @Override
203     @Transactional
204     public void copyTask(TaskListDto taskTitleDto) {
205         TaskDto taskDto = getTaskById(taskTitleDto.getId())
206             .map(taskMapper::toDto)
207             .orElseThrow(ResourceNotFoundException::new);
208         Task task = taskMapper.toEntity(taskDto);
209         task.setId(null);
210         task.setTitle(taskTitleDto.getTitle());
211         task.setType(3);
212         task.setStatus(1);
213
214         task.setMaxPoints(PointParser.calculatePoints(task.getGradePoints()));
215         String imageName = task.getImageFileName();
216         if (imageName != null) {
217             String extension =
218                 FileUtils.getExtensionFromFileName(imageName);
219             task.setImageFileName(FileNameGenerator.generateFileName(extension));
220             saveOrUpdate(task, fileService.getFileWithName(imageName));
221         } else {
222             saveOrUpdate(task, false);
223         }
224     }
225     @Override
226     @Transactional
227     public void unarchiveTask(Long id) {
228         Task task =
229             getTaskById(id).orElseThrow(ResourceNotFoundException::new);

```

```
228         task.setStatus(1);
229         saveOrUpdate(task, false);
230     }
231 }
```