

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

БАКАЛАВРСКАЯ РАБОТА

на тему «**Информационная система для проведения онлайн олимпиады по программированию**»

Студент	_____ А.В. Луговой _____
Руководитель	_____ Е.А. Ерофеева _____
Консультант по аннотации	_____ Н.В. Яценко _____

Допустить к защите
Заведующий кафедрой к.т.н., доцент, А.В. Очеповский _____

« _____ » _____ 2017 г.

АННОТАЦИЯ

ОЛИМПИАДА ПО ПРОГРАММИРОВАНИЮ, ИНФОРМАТИЗАЦИЯ ОБРАЗОВАНИЯ, ОНЛАЙН ОЛИМПИАДА, ИНТЕЛЛЕКТУАЛЬНОЕ РАЗВИТИЕ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ИНФОРМАЦИОННАЯ СИСТЕМА.

Настоящая выпускная квалификационная работа выполнена Луговым Алексеем Васильевичем, студентом института математики, физики и информационных технологий Тольяттинского государственного университета.

Тема работы: «Информационная система для проведения онлайн олимпиады по программированию».

Актуальность данной темы обусловлена необходимостью разработки информационной системы, позволяющей расширить аудиторию участников олимпиады по программированию путем проведения заочной (онлайн) формы олимпиады в дополнение к уже существующей очной.

Целью выпускной квалификационной работы является разработка информационной системы для проведения онлайн олимпиады по программированию.

Структура выпускной квалификационной работы представлена введением, тремя главами, заключением, списком литературы и приложением.

Во введении описана сформирована цель и поставлены задачи, которые необходимо достигнуть в результате выполнения работы.

В первой главе проведена идентификация предметной области автоматизации, выбрана и обоснована методология и технология концептуального моделирования, разработаны модели бизнес-процессов «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ», проведен анализ родственных работ в выбранной предметной области.

Во второй главе проведено проектирование информационной системы, в частности выбрана и обоснована методология и технология логического

моделирования, разработана объектная модель информационной системы, а на ее основе – логическая модель.

В третьей главе представлены основные этапы развертывания информационной системы, представлен ход тестирования разработанного программного продукта.

Результатом настоящей выпускной квалификационной работы является разработанная информационная система для проведения онлайн олимпиады по программированию.

Принято решение о внедрении разработанной информационной системы.

В работе использовано 10 таблиц, 32 рисунка, список используемой литературы содержит 20 источников, в том числе 5 на иностранном языке. Объем выполненной выпускной квалификационной работы – 51 страница.

ABSTRACT

The title of the graduation work is “Information System for the Online Programming Olympiad”.

Programming Olympiad is a competition between programmers. This kind of Olympiad can be hold in intramural or extramural (online) form. The goal of this graduation work is to develop information system that can hold online programming Olympiad with automated check of the participants’ programs. The approach involves designing and developing of full-fledged information system that includes modules for test and practice stages of the Olympiad.

The object is the processes which take place during online programming Olympiad.

The subject is the online programming Olympiad business process automation.

The first part describes the functional modeling of the information system. This part describes technical and economical characteristics of organization that nowadays hold the intramural form of programming Olympiad. It also contains “AS-IS” business model which shows current state of affair in this organization and “TO-BE” business model which demonstrates how the new information system could improve this state. There is also analysis of similar systems here which reveals the necessity of information system development.

The second part is focused on the logical modeling of the information system. Here we did the object modeling of the information system which includes Use Case and Class diagrams constructing. And we also developed the logical model of the information system which is based on object model.

The third part describes the development of the information system. There are information system database development, information system software development and information system testing in this part.

Developed information system will expand the audience of programming Olympiad participants.

The volume of graduation work is 51 pages, including 10 tables, 32 pictures and 20 literature sources, 5 of which are written in foreign language.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ГЛАВА 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ	7
1.1 Техничко-экономическая характеристика предметной области	7
1.2 Концептуальное моделирование предметной области	8
1.2.1 Обоснование выбора технологии концептуального моделирования предметной области	8
1.2.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»	9
1.2.3 Выявление недостатков существующего бизнес-процесса и рекомендации по его усовершенствованию с помощью информационной системы	12
1.2.4 Формирование требований к новой информационной системе	13
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям	15
1.4 Постановка задачи на разработку проекта создания/внедрения информационной системы	16
1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»	16
Выводы по главе 1	18
ГЛАВА 2 ЛОГИЧЕСКОЙ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	19
2.1 Выбор технологии логического моделирования	19
2.2 Объектное моделирование информационной системы	20
2.3 Информационное обеспечение информационной системы	25
2.3.1 Используемые классификаторы и системы кодирования	25
2.3.2 Характеристика нормативно-справочной и входной оперативной информации	27
2.3.3 Характеристика выходной информации	27
2.4 Логическое моделирование информационной системы	28
2.5 Требования к аппаратно-программному обеспечению информационной системы	29

Выводы по главе 2.....	30
ГЛАВА 3 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	31
3.1 Выбор архитектуры информационной системы	31
3.2 Выбор среды разработки программного обеспечения	32
3.3 Разработка физической модели данных информационной системы	33
3.4 Разработка программного обеспечения информационной системы	34
3.5 Описание функциональности информационной системы	38
3.6 Тестирование программного проекта	45
Выводы по главе 3.....	48
ЗАКЛЮЧЕНИЕ	49
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	50
ПРИЛОЖЕНИЕ А	52
ПРИЛОЖЕНИЕ Б.....	59

ВВЕДЕНИЕ

Понятие «олимпиада» зародилось в Древней Греции. Именно так назывался промежуток в 4 года между олимпийскими играми. В настоящее время это слово несколько изменило свое значение. «Олимпиада» в современном понимании означает некие спортивные, театральные и иного вида соревнования, состязания [9].

В частности, олимпиада по программированию подразумевает соревнование между участниками, знакомыми с основными правилами написания программ, за звание лучшего программиста года. Данная олимпиада может проводиться как очно (для этого необходимо личное присутствие участников), так и заочно, дистанционно. Для более удобной организации дистанционного проведения олимпиады необходима специальная информационная система, способная давать оценку способностям участников к программированию.

В наши дни информация является одним из наиболее важных ресурсов, а информационные системы давно уже стали необходимым инструментом практически во всех сферах человеческой деятельности. Важным этапом на пути к созданию информационной системы является ее проектирование. В общем случае, под проектированием понимается процесс, направленный на создание реальных объектов и/или эффектов с заданными функциональными, технико-экономическими, экологическими и потребительскими качествами [10].

Таким образом, **актуальность** темы настоящей выпускной квалификационной работы обусловлена необходимостью разработки информационной системы, позволяющей расширить аудиторию участников олимпиады по программированию путем проведения заочной (онлайн) формы олимпиады в дополнение к уже существующей очной.

Несмотря на наличие родственных работ в выбранной предметной области, проектирование и последующая разработка информационной системы для проведения онлайн олимпиады по программированию имеет смысл,

поскольку ее полного и общедоступного аналога на данный момент не существует. Более подробно данный вопрос будет рассмотрен в параграфе 1.4.

Целью настоящей выпускной квалификационной работы является разработка информационной системы для проведения онлайн олимпиады по программированию.

Для достижения цели необходимо выполнение следующих **задач**:

- проведение идентификации предметной области;
- описание проблем, связанных с отсутствием информационной системы;
- составление моделей бизнес-процессов «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- анализ родственных работ в выбранной предметной области;
- формулировка функциональных и нефункциональных требований к информационной системе;
- разработка объектной и логической модели данных для выбранной предметной области;
- непосредственная реализация информационной системы;
- тестирование информационной системы на предмет соответствия сформулированным требованиям.

Объектом исследования являются процессы, происходящие во время проведения онлайн олимпиады по программированию, такие, как:

- процесс написания и последующей компиляции исходного кода программ;
- процесс контроля выполнения участником условий заданий;
- процесс начисления участникам баллов и последующего определения победителя олимпиады.

Предмет исследования – автоматизация бизнес-процессов онлайн олимпиады по программированию.

При выполнении выпускной квалификационной работы предполагаются к использованию такие **методы исследования**, как CASE-технологии структурного и объектно-ориентированного анализа и проектирования.

ГЛАВА 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Техничко-экономическая характеристика предметной области

Предметной областью автоматизации выпускной квалификационной работы является олимпиада по программированию «CodeUp», которую ежегодно проводит тольяттинский офис компании Netcracker совместно с Институтом математики, физики и информационных технологий Тольяттинского государственного университета.

В настоящее время данная олимпиада проводится исключительно в очной форме, предполагающей личное присутствие участников в установленном организатором месте, в определенное время. Однако, как известно, не у всех потенциальных участников имеется возможность лично посетить это мероприятие. Поэтому возникает необходимость организации заочной его формы (онлайн олимпиады) с целью охвата большей аудитории участников.

Вне зависимости от формы проведения, основными задачами упомянутой олимпиады по программированию являются:

- развитие и пропаганда информационных технологий, как важного направления интеллектуального развития участников;
- профессиональная ориентация участников на инженерные специальности, связанные с информационными технологиями;
- выявление среди участников трех победителей, обладающих наилучшим знанием информационных технологий и основ программирования.

Таким образом, для проведения заочной формы олимпиады необходимо разработать информационную систему, располагающую следующими функциями:

- теоретический опрос участников на первом этапе олимпиады;
- компиляции исходного кода программ, написанных участниками на втором этапе олимпиады согласно заданиям;
- контроль выполнения участником условий заданий;

- начисление участникам баллов с последующим определением победителя олимпиады.

Теперь, когда определены функции, которыми должна обладать информационная система, перейдем к ее непосредственному проектированию. Проектирование любой информационной системы проходит в два этапа [2]. На первом этапе осуществляется ее концептуальное моделирование, на втором – логическое. Перед созданием концептуальной модели системы необходимо определить, по какой технологии эта модель будет создана.

1.2 Концептуальное моделирование предметной области

1.2.1 Обоснование выбора технологии концептуального моделирования предметной области

Проведем сравнительный анализ наиболее известных нотаций концептуального моделирования информационных систем. Результаты анализа представим в таблице 1.1.

Таблица 1.1 – Сравнительный анализ методологий

Критерий сравнения	Нотация IDEF0	Нотация ARIS	Нотация UML
Легкость изучения	Легко	Очень сложно	Сложно
Область применения	Программное обеспечение, бизнес-процессы	Бизнес-процессы	Программное обеспечение, бизнес-процессы
Формальное описание	Жесткое	Очень жесткое	Менее жесткое
Подход к проектированию	Функциональный	Процессный	Объектно-ориентированный

Таким образом, наиболее подходящей представляется нотация IDEF0, поскольку ARIS чрезвычайно сложна в изучении, а UML обладает не слишком точным формальным описанием, в связи с чем ее конструкции недостаточно жестко регламентированы [15].

IDEF0 – нотация графического моделирования бизнес-процессов [4]. С помощью наглядного графического языка IDEF0, изучаемая система предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций.

1.2.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

Модель «КАК ЕСТЬ» («AS-IS») – модель существующего состояния организации. Данная модель позволяет систематизировать протекающие в данный момент процессы, а также используемые информационные объекты [7]. На основе этого выявляются узкие места в организации и взаимодействиях бизнес-процессов, определяется необходимость тех или иных изменений в существующей структуре. Такую модель часто называют функциональной и выполняют с использованием различных графических нотаций и CASE-средств. На этапе построения функциональной модели AS-IS считается, что очень важно создать максимально приближенную к действительности модель, основанную на реальных потоках процессов, а не на их идеализированном представлении [5].

Контекстная модель AS-IS процесса «Проведение олимпиады по программированию» в нотации IDEF0, выполненная с использованием CASE-средства Ramus, представлена на рисунке 1.1.

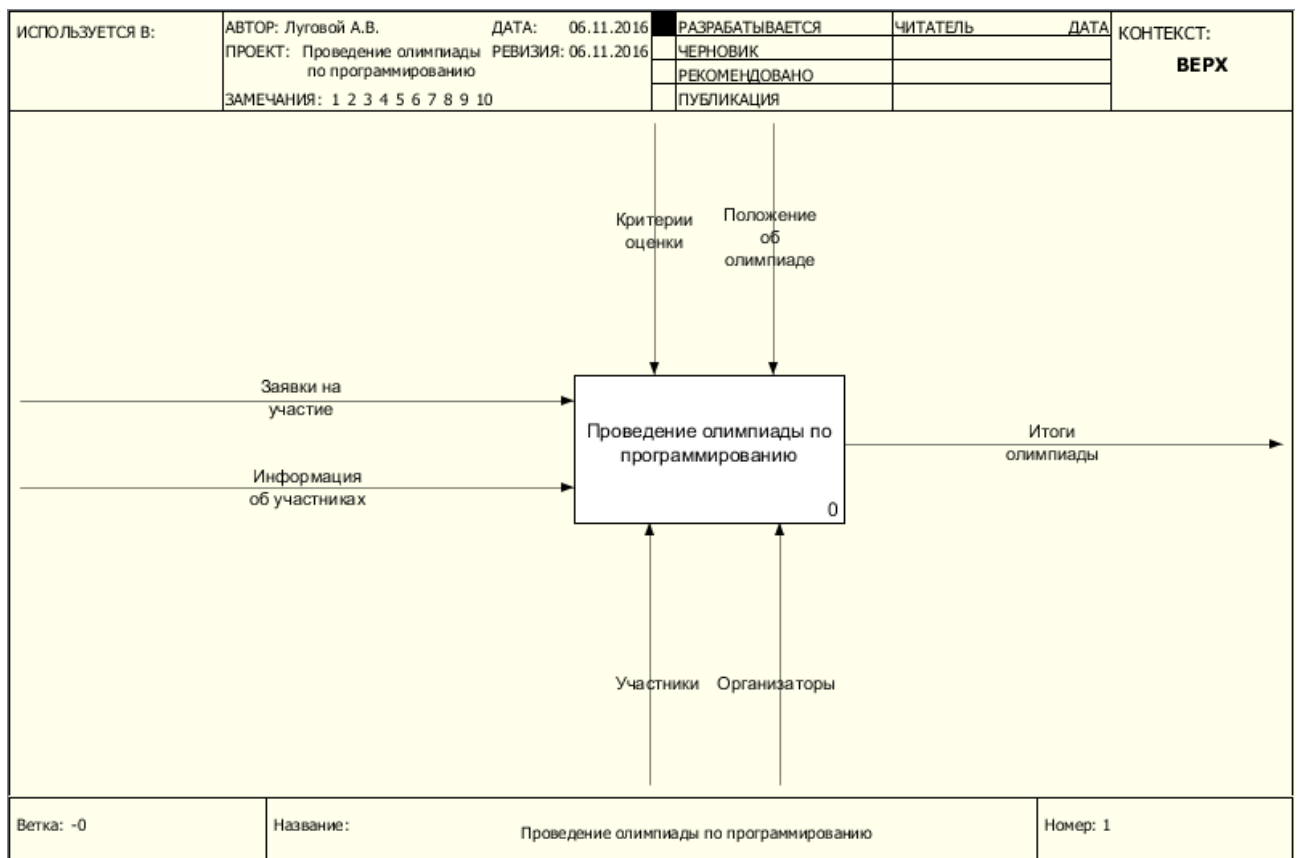


Рисунок 1.1 – Контекстная модель AS-IS

Для лучшего понимания особенностей протекания главного процесса следует выполнить его декомпозицию [8], то есть разделение целого на части (в данном случае на так называемые процессы верхнего уровня). Модель проведенной декомпозиции главного процесса, также выполненная с использованием CASE-средства Ramus, изображена на рисунке 1.2.

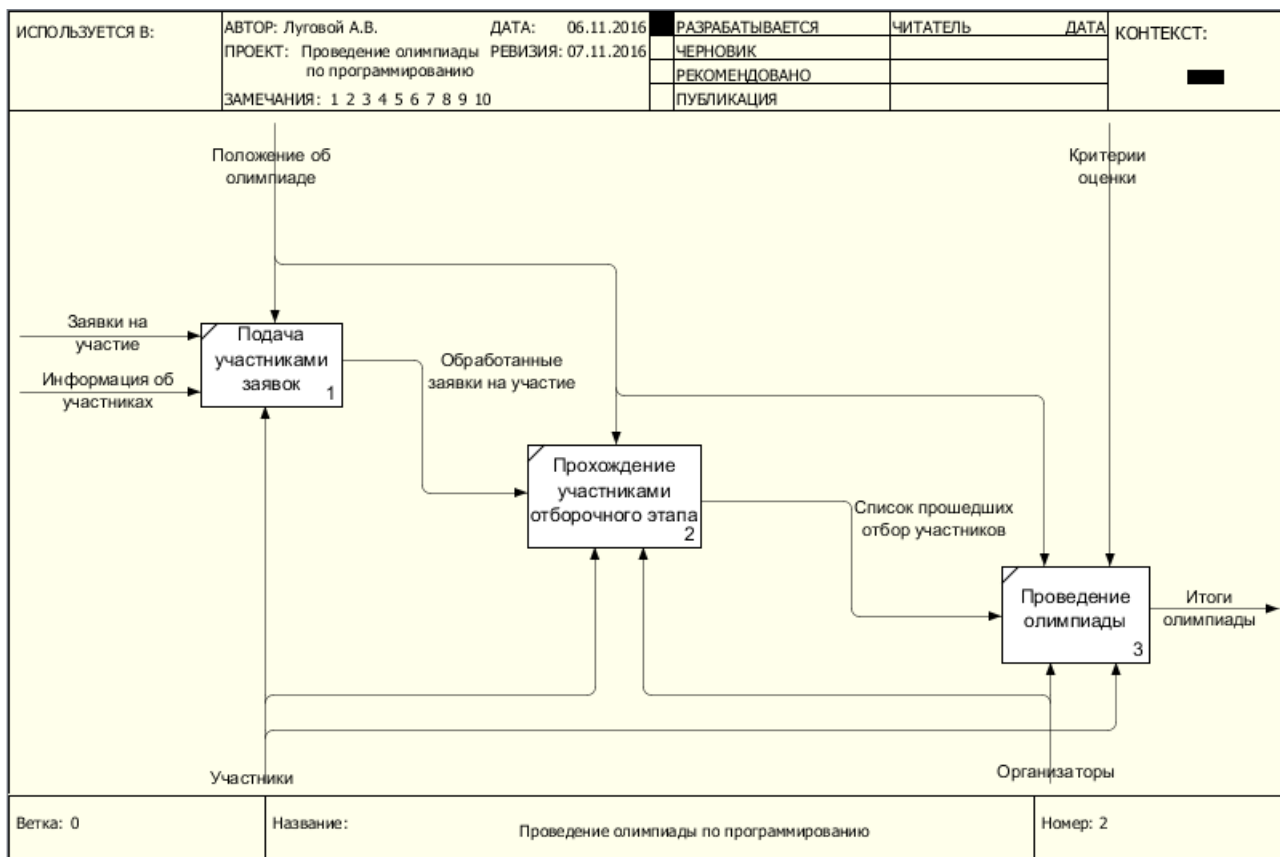


Рисунок 1.2 – Декомпозиция контекстной модели AS-IS

Рассмотрим более подробно каждый из процессов верхнего уровня:

1. Процесс «Подача участниками заявок» подразумевает, что каждый потенциальный участник олимпиады изъявляет свое желание участвовать в ней путем подачи заявки в электронном виде.

2. Процесс «Прохождение участниками отборочного этапа» включает в себя теоретический опрос участников. Проводится в письменной форме на бумажном носителе в установленное организатором время, в указанном им месте.

3. Процесс «Проведение олимпиады» подразумевает выполнение заданий олимпиады теми участниками, которым удалось пройти отборочный

этап. Участники выполняют задания в исполнительной среде на персональных компьютерах, предоставленных организаторами олимпиады, демонстрируя им результаты выполнения.

Из описания процессов 2 и 3 видно, что возможна их автоматизация с целью предоставления некоторым участникам возможности дистанционного прохождения олимпиады и уменьшения затрат ресурсов (трудовых и материальных) со стороны организаторов.

Проведем декомпозицию процессов, подлежащих автоматизации. Это необходимо для лучшего понимания особенностей их протекания [1].

Проведем декомпозицию процесса верхнего уровня под названием «Прохождение участниками отборочного этапа». Результат представлен на рисунке 1.3.

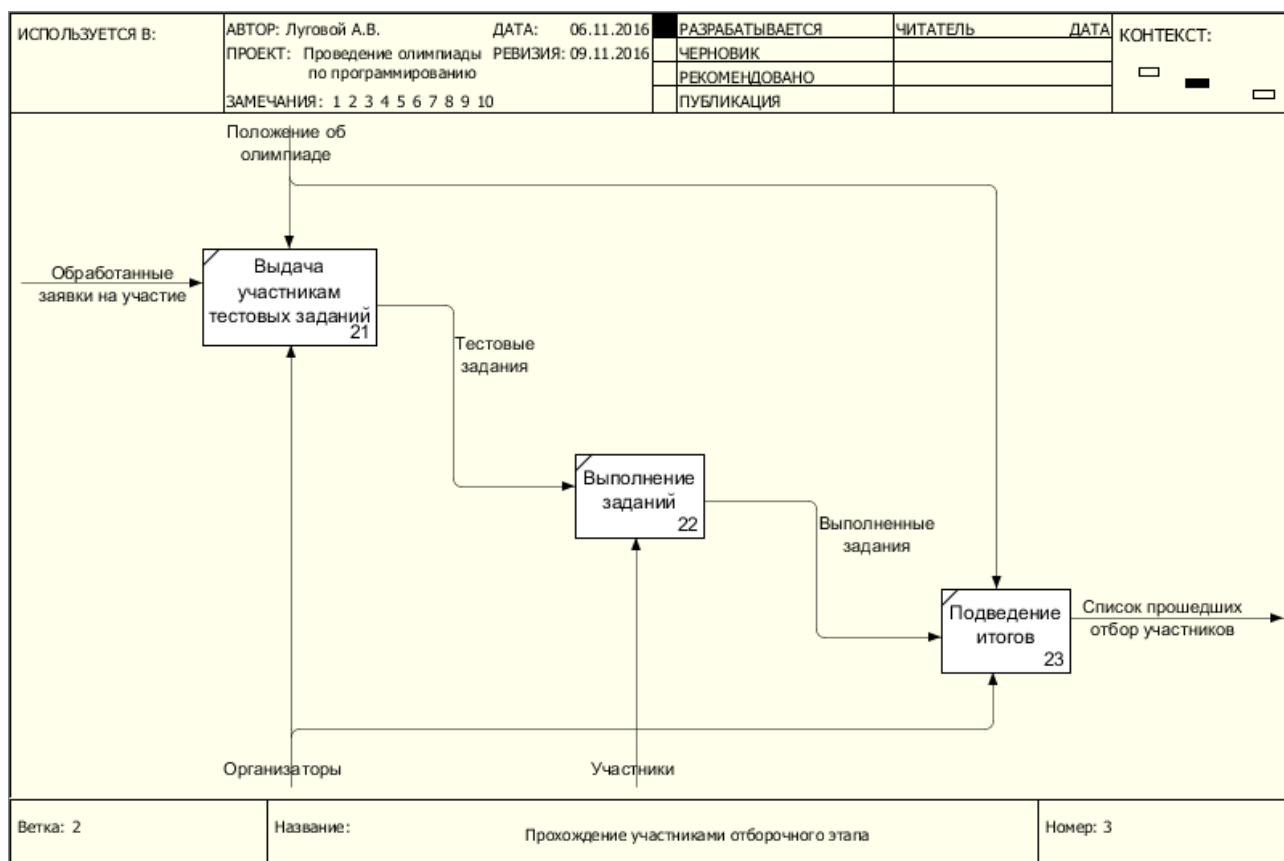


Рисунок 1.3 – Декомпозиция процесса «Прохождение участниками отборочного этапа»

Далее построим модель декомпозиции другого процесса верхнего уровня, называющегося «Проведение олимпиады», представив результат на рисунке 1.4.

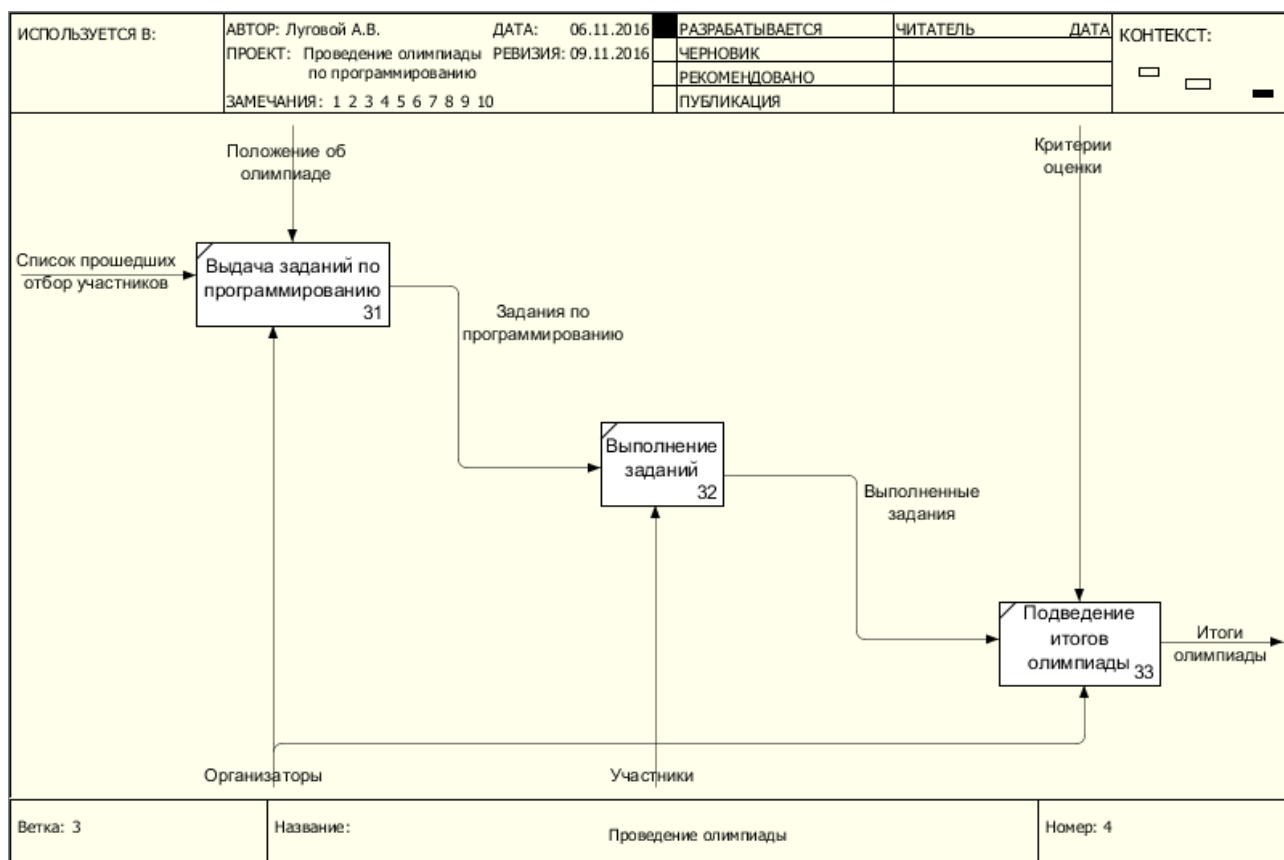


Рисунок 1.4 – Декомпозиция процесса «Проведение олимпиады»

Прежде, чем переходить к рассуждениям об автоматизации процессов «Прохождение участниками отборочного этапа» и «Проведение олимпиады», необходимо выявить их недостатки.

1.2.3 Выявление недостатков существующего бизнес-процесса и рекомендации по его усовершенствованию с помощью информационной системы

Процессы верхнего уровня «Прохождение участниками отборочного этапа» и «Проведение олимпиады», входящие в состав бизнес-процесса «Проведение олимпиады по программированию», обладают следующими недостатками:

1. Неэффективный документооборот, выражающийся в необходимости ручной проверки организаторами ответов на вопросы отборочного этапа. Данный недостаток может быть устранен путем автоматизации проведения отборочного этапа при помощи добавления в проектируемую информационную систему соответствующих функций.

2. Со стороны участников: необходимость демонстрации выполнения заданий организаторам. Данный недостаток проявляется на этапе непосредственного выполнения заданий по программированию и так же, как и предыдущий, может быть устранен путем добавления в информационную систему функций автоматизированного тестирования и компиляции программного кода.

3. Необходимость очного присутствия участников, как на отборочном этапе, так и на этапе непосредственного проведения олимпиады. Данный недостаток может быть устранен путем проведения заочной формы олимпиады в дополнение к очной. Для этого необходимо внедрение информационной системы в веб-сайт олимпиады по программированию.

После того, как выявлены все недостатки существующего бизнес-процесса, перейдем к формулировке требований, которым должна соответствовать будущая информационная система.

1.2.4 Формирование требований к новой информационной системе

Для определения требований к внедряемой информационной системе воспользуемся простой и понятной классификацией FURPS+. Данная классификация представляет собой аббревиатуру, которая расшифровывается следующим образом:

- F – Functionality (функциональность);
- U – Usability (удобство использования);
- R – Reliability (надежность);
- P – Performance (производительность);
- S – Supportability (сопровождаемость).

В таблице 1.2 представлены функциональные требования, выявленные при анализе вышеперечисленных контекстов, а в таблице 1.3 – нефункциональные.

Таблица 1.2 – Функциональные требования к системе

№ п/п	Требование	Статус	Полезность	Риск
1	Проводить отборочный опрос участников	Одобрённые	Критичное	Средний
2	Принимать от участников выполненные задания по программированию	Одобрённые	Критичное	Средний
3	Проверять выполненные задания	Одобрённые	Критичное	Средний
4	Сообщать участникам о результатах выполнения заданий	Предложенные	Критичное	Средний

Таблица 1.3 – Нефункциональные требования к системе

№ п/п	Требование	Статус	Полезность	Риск
Требования к удобству использования				
5	Кегль основного шрифта не ниже 12 пунктов	Одобрённые	Критичное	Низкий
6	Дизайн интерфейса в едином стиле и цветовой гамме	Одобрённые	Критичное	Низкий
7	Элементы интерфейса должны иметь четкую структуру	Одобрённые	Критичное	Низкий
8	Базовое разрешение экрана 1366x768 с возможностью масштабирования	Предложенные	Критичное	Низкий
Требования к надежности				
9	Время восстановления системы после отказа – не более 24 часов	Предложенные	Критичное	Средний
Требования к производительности				
10	Система должна обеспечивать одновременную работу как минимум 50 пользователям	Одобрённые	Критичное	Средний
11	Время отклика информационной системы должно составлять не более 5 секунд	Предложенные	Критичное	Средний
Требования к сопровождаемости				
12	Устранение возникающих проблем в течение 24 часов	Предложенные	Критичное	Средний

Далее необходимо произвести анализ уже существующих ИТ-решений на предмет соответствия описанным требованиям.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

Проведем анализ наиболее известных информационных систем, имеющих то или иное отношение к оценке выполнения заданий по программированию. Это необходимо для того, чтобы более точно определить дальнейшие направления проектирования, а также не повторить чужих ошибок при разработке. Результаты анализа систем на соответствие установленным в параграфе 1.5 требованиям представлены в таблице 1.4. Номера критериев соответствуют таковым из таблицы 1.2 и таблицы 1.3.

Таблица 1.4 – Анализ известных ИТ-решений

№ критерия	Название информационной системы и ее адрес в сети		
	JavaRush (www.javarush.ru)	Programmr (www.programmr.com)	Code School (www.codeschool.com)
1	Не соответствует	Не соответствует	Не соответствует
2	Частично соответствует	Соответствует	Соответствует
3	Соответствует	Соответствует	Соответствует
4	Соответствует	Соответствует	Соответствует
5	Соответствует	Соответствует	Не соответствует
6	Соответствует	Не соответствует	Соответствует
7	Соответствует	Частично соответствует	Соответствует
8	Частично соответствует	Частично соответствует	Не соответствует
9	Не соответствует	Не соответствует	Не соответствует
10	Соответствует	Соответствует	Соответствует
11	Соответствует	Соответствует	Соответствует
12	Не соответствует	Не соответствует	Не соответствует
Итого	8 из 12	7 из 12	7 из 12

По итогам проведенного анализа необходимо определить целесообразность проектирования и разработки новой информационной системы в рассматриваемой предметной области. Если подобная целесообразность будет очевидна, то следует поставить задачу на разработку проекта информационной системы.

1.4 Постановка задачи на разработку проекта создания/внедрения информационной системы

Согласно проведенному в параграфе 1.3 анализу, ни одна из рассмотренных информационных систем не удовлетворяет всем установленным требованиям. Таким образом, проектирование и разработка информационной системы для проведения онлайн олимпиады по программированию являются целесообразными, поскольку данная система не будет целиком копировать функционал других известных решений. Система станет самостоятельным программным продуктом.

Предлагаемая к разработке информационная система должна обладать функциями, описанными в параграфе 1.1 настоящей работы. Кроме этого, система должна соответствовать требованиям, описанным в параграфе 1.2.4.

Система должна состоять из двух модулей. Первый модуль включает в себя функционал, необходимый для проведения отборочного этапа олимпиады. Второй модуль предназначен для проведения основного (практического) этапа олимпиады. Согласно требованиям заказчика информационной системы, оба модуля должны быть реализованы на языке Java.

Теперь перейдем к разработке модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ».

1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»

Для выделения и последующего описания основных требований, которым должна удовлетворять проектируемая информационная система, была построена контекстная модель «КАК ДОЛЖНО БЫТЬ» («ТО-ВЕ»), которая отображает будущее предполагаемое состояние предметной области. Данная модель представлена на рисунке 1.5. В ней, по сравнению с моделью AS-IS, добавился новый механизм – «Информационная система».

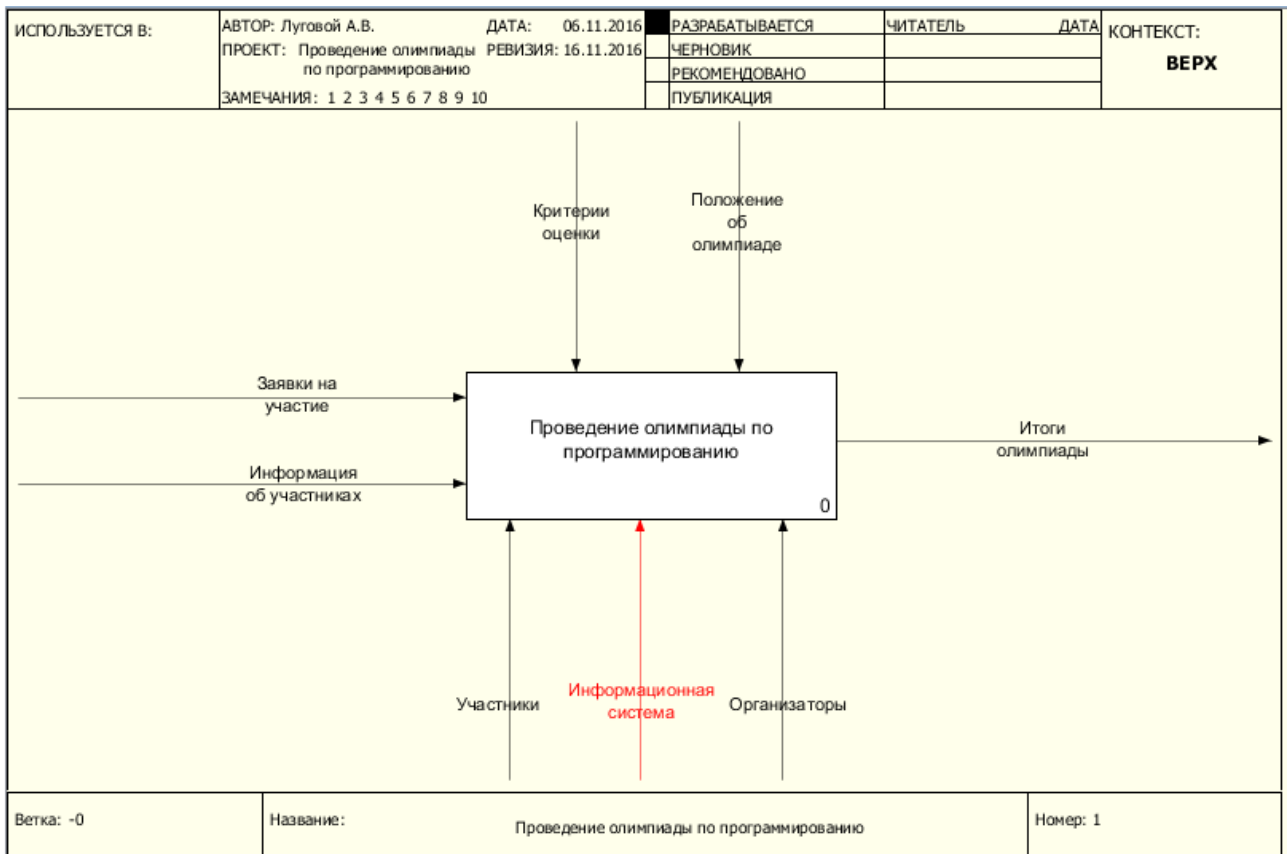


Рисунок 1.5 – Контекстная модель ТО-ВЕ

Декомпозиция контекстной модели представлена на рисунке 1.6.

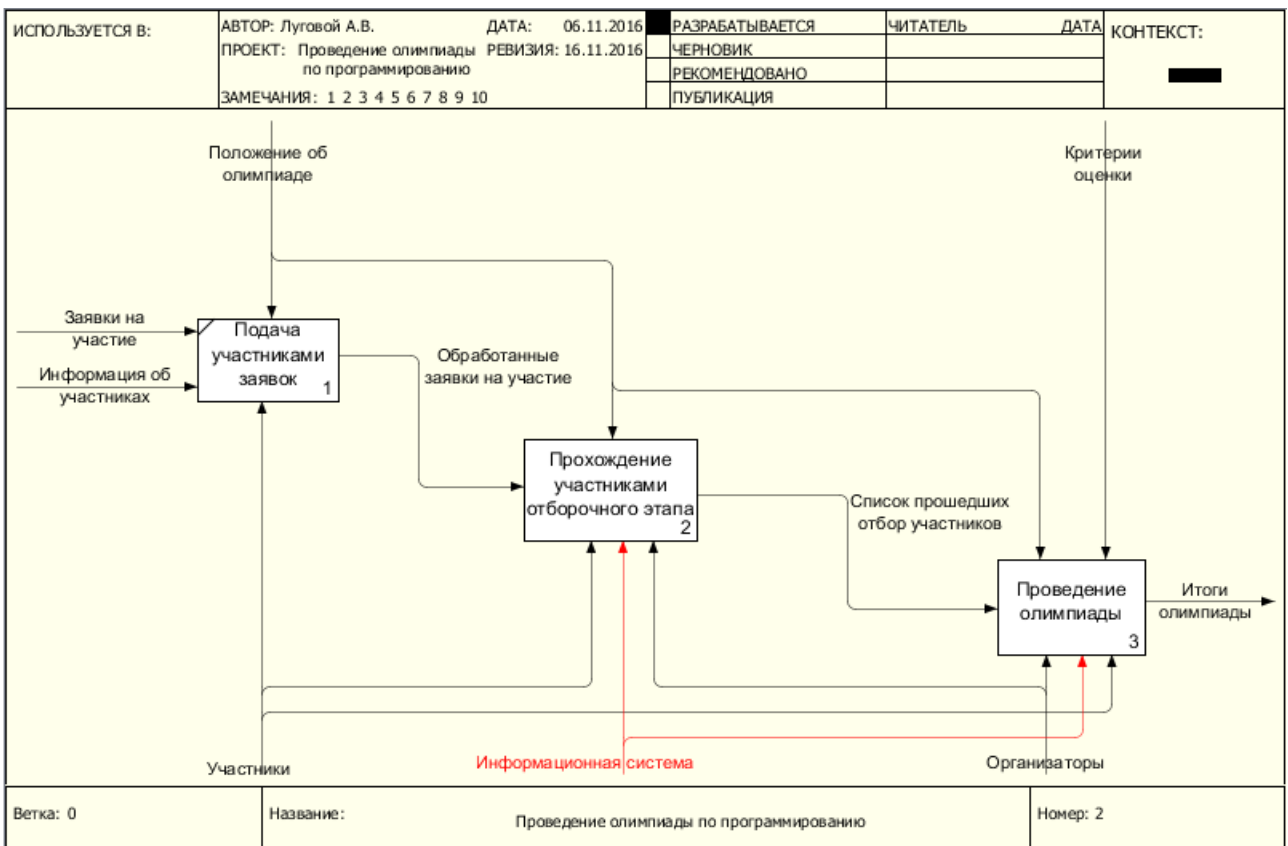


Рисунок 1.6 – Декомпозиция контекстной модели ТО-ВЕ

Как можно заметить, теперь на отборочном этапе и на этапе проведения олимпиады применяется информационная система.

Перейдем к подведению итогов написания первой главы выпускной квалификационной работы.

Выводы по главе 1

Была проведен анализ бизнес-процессов олимпиады по программированию «CodeUp», который позволил определить, что часть таких процессов нуждается в автоматизации. Были также рассмотрены уже существующие информационные системы и проведен их сравнительный анализ, благодаря которому стало очевидно, что разработка новой информационной системы в данной области действительно необходима.

ГЛАВА 2 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Выбор технологии логического моделирования

Логическое моделирование подразумевает создание схемы базы данных, основанной на конкретной модели данных, например, реляционной или объектной [19].

Согласно установленным заказчиком требованиям для управления базой данных должна использоваться система управления базами данных (СУБД) MySQL. Для более удобной работы с СУБД MySQL может быть использован визуальный клиент. При создании базы данных информационной системы для проведения онлайн олимпиады по программированию предполагается использование визуального клиента phpMyAdmin, включающего CASE-средство логического моделирования баз данных (БД), основанное на использовании методологии IDEF1X. Таким образом, в целях сокращения трудозатрат на дальнейшее проектирование целесообразным представляется использование именно данного встроенного средства вместо любого другого стороннего, и именно данной методологии.

Стартовая страница клиента phpMyAdmin показана на рисунке 2.1.

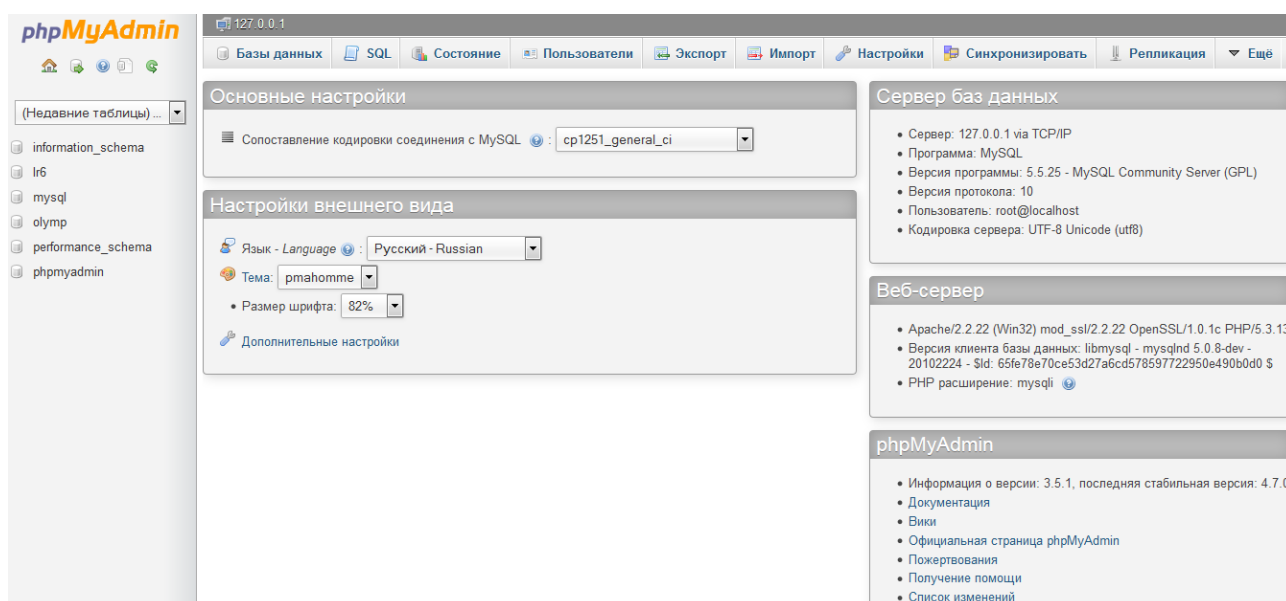


Рисунок 2.1 – Стартовая страница phpMyAdmin

Логическая модель БД информационной системы для проведения онлайн олимпиады по программированию будет основана на объектной модели.

2.2 Объектное моделирование информационной системы

В качестве объектной модели информационной системы может выступать диаграмма вариантов использования [20]. Данная диаграмма позволяет выявить основные процессы, происходящие в системе и их взаимосвязь. Кроме того, диаграмма вариантов использования способствует выделению функциональной структуры информационной системы. На базе созданной диаграммы составляется план дальнейшей разработки системы [11].

Разработанная диаграмма вариантов использования представлена на рисунке 2.2.



Рисунок 2.2 – Диаграмма вариантов использования

На диаграмме представлены следующие действующие лица (актеры):

- клиент – абстрактный участник олимпиады, использует информационную систему для регистрации в олимпиаде и дальнейшего участия в ней;
- организатор – абстрактный организатор олимпиады, ответственный за составление заданий олимпиады и подведение ее итогов;

- администратор – должностное лицо, ответственное за корректную работу компьютерного оборудования и программного обеспечения, необходимого для проведения онлайн олимпиады.

В таблице 2.1 приведем краткую характеристику прецедентов (т.е. самих вариантов использования) диаграммы.

Таблица 2.1 – Характеристика прецедентов

Актер	Прецедент	Характеристика
Клиент	Регистрация	Регистрация участника на сайте
Клиент	Выполнение тестовых заданий олимпиады	Выполнение клиентом-участником тестовых заданий отборочного этапа олимпиады
Клиент	Выполнение практических заданий олимпиады	Выполнение клиентом-участником практических заданий по программированию, входящих в основной этап олимпиады (практическую часть)
Клиент	Ознакомление с результатами олимпиады	Получение участником информации о результатах олимпиады
Организатор	Добавление заданий олимпиады	Добавление организатором в систему как тестовых, так и практических заданий олимпиады
Организатор	Объявление итогов олимпиады	Подведение и объявление итогов олимпиады
Администратор	Внесение изменений в систему	Внесение изменений в систему, необходимых для корректировки ее работы
Администратор	Предоставление пользователям информации об олимпиаде	Предоставление пользователям информации о дате проведения олимпиады, порядке и правилах участия в олимпиаде

В дополнение к диаграмме вариантов использования построим диаграммы классов модуля отборочного этапа (рисунок 2.3) и модуля практического этапа олимпиады (рисунок 2.4), входящего в состав информационной системы, которая отражает сами классы, их атрибуты, а также связи между ними.

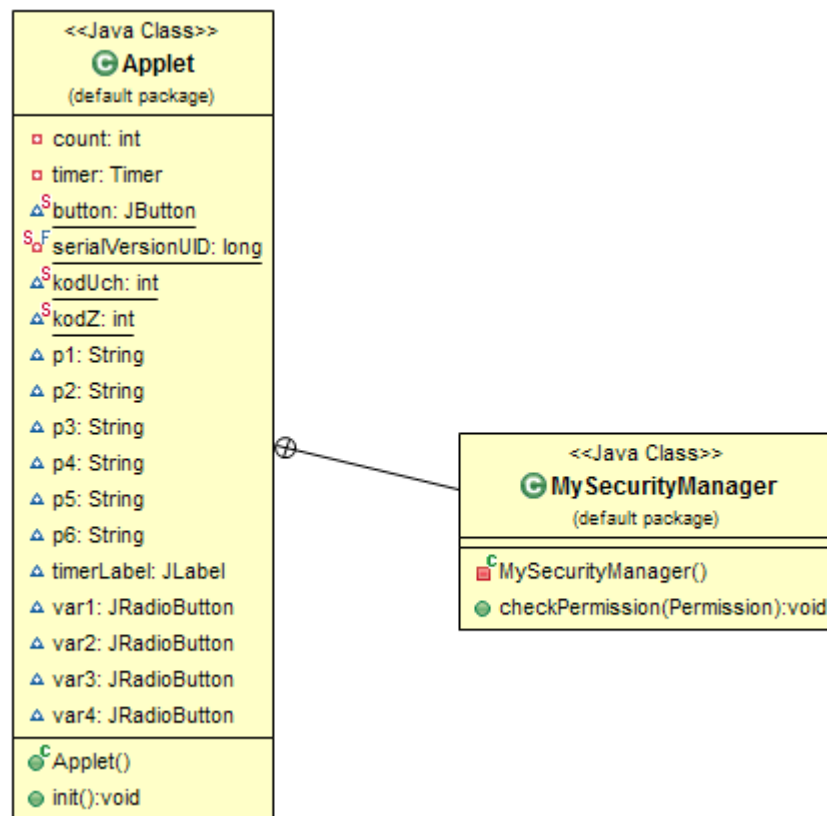


Рисунок 2.3 – Диаграмма классов модуля отборочного этапа

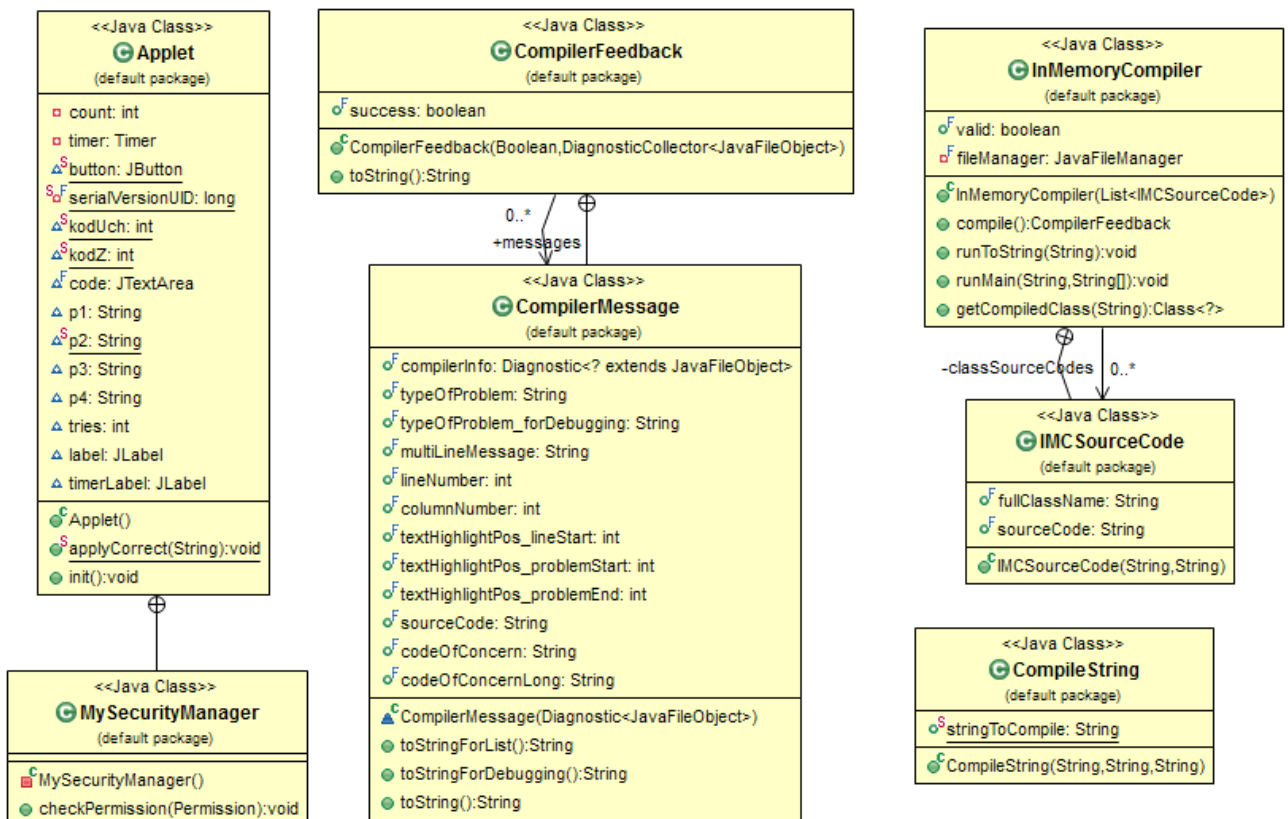


Рисунок 2.4 – Диаграмма классов модуля практического этапа

На диаграммах представлены следующие классы: *Applet*, *CompilerFeedback*, *InMemoryCompiler*, *MySecurityManager*, *CompilerMessage*, *IMCSourceCode* и *CompileString*.

Класс *Applet* отвечает за работу графического интерфейса пользователя. Кроме того, данный класс содержит точку входа в программу. В случае с тестовым модулем, класс *Applet* содержит также функционал для работы с базой данных тестовых заданий.

Класс *CompilerFeedback* отвечает за вывод ответа компилятора исходного кода в виде текстовой строки. Исходный код собирается и отправляется классом *Applet*.

Класс *InMemoryCompiler* осуществляет непосредственную компиляцию кода «на лету» (JIT-компиляцию).

Класс *MySecurityManager* отвечает за запрос необходимых привилегий безопасности у клиента.

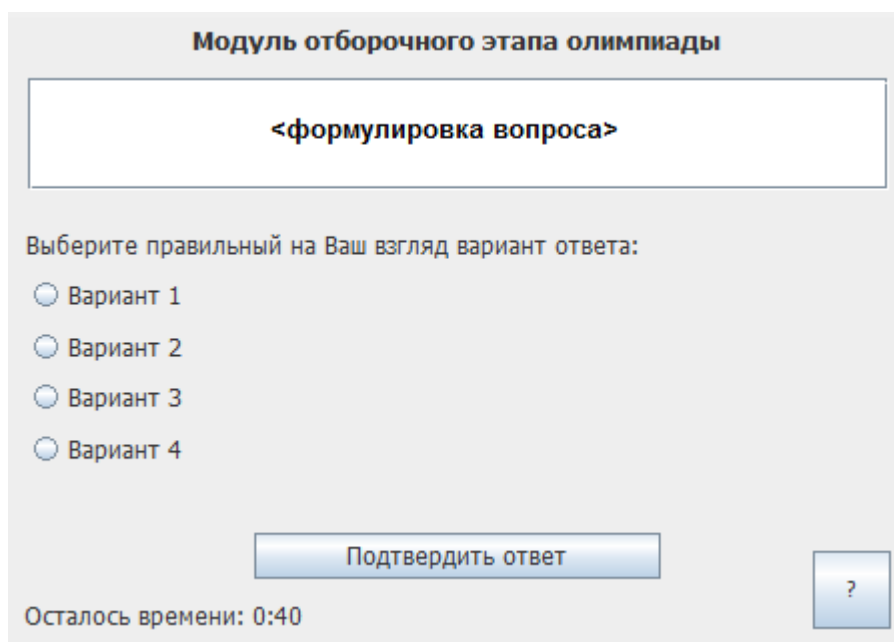
Класс *CompilerMessage* необходим для вывода сообщений о невозможности компиляции кода в связи с отсутствием установленной на стороне клиента библиотеки JDK.

Класс *IMCSourceCode* отвечает за сохранение исходного кода написанных клиентами программ на сервере. Исходные коды сохраняются в виде текстовых файлов.

Класс *CompileString* анализирует исходный код на наличие требуемых заданием ключевых слов. Данная процедура необходима для предотвращения попыток обмана системы клиентами. При помощи проверки ключевых слов исключается возможность вывода только требуемого заданием ответа, без непосредственного выполнения условия задания.

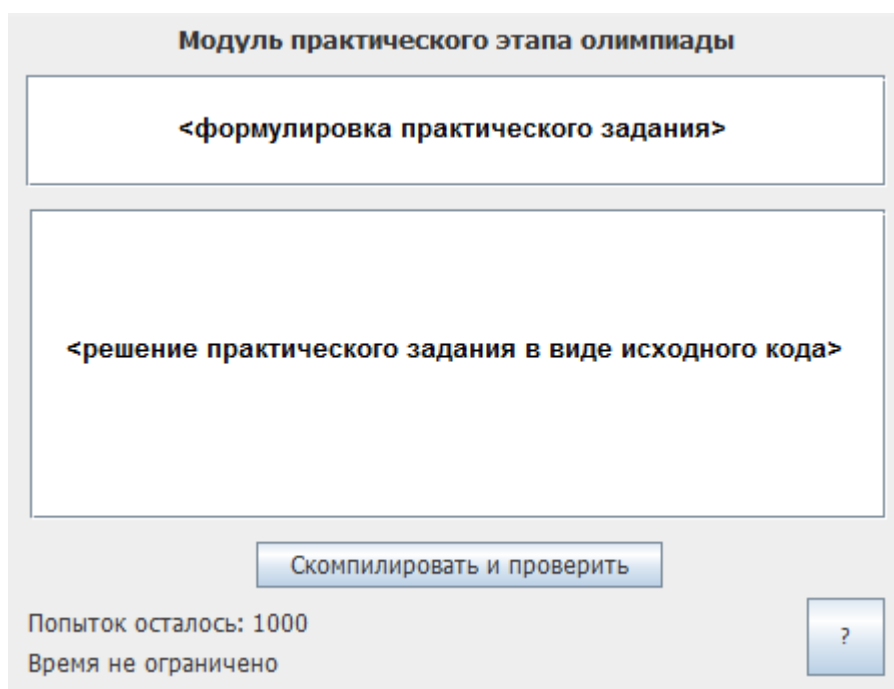
На основании диаграммы вариантов использования и диаграммы классов можно построить логическую модель данных информационной системы. Однако, прежде чем переходить к построению данной модели, следует определить, какое информационное обеспечение будет использоваться в информационной системе и как должны выглядеть интерфейсы модулей.

На рисунке 2.5 представлен проект интерфейса модуля отборочного этапа, а на рисунке 2.6 – проект интерфейса модуля практического этапа.



The screenshot shows a web interface titled "Модуль отборочного этапа олимпиады". At the top, there is a text box containing the placeholder "<формулировка вопроса>". Below this, the instruction "Выберите правильный на Ваш взгляд вариант ответа:" is followed by four radio button options: "Вариант 1", "Вариант 2", "Вариант 3", and "Вариант 4". At the bottom, there is a "Подтвердить ответ" button, a "?" button, and a timer showing "Осталось времени: 0:40".

Рисунок 2.5 – Проект интерфейса модуля отборочного этапа



The screenshot shows a web interface titled "Модуль практического этапа олимпиады". At the top, there is a text box containing the placeholder "<формулировка практического задания>". Below this is a larger text box containing the placeholder "<решение практического задания в виде исходного кода>". At the bottom, there is a "Скомпилировать и проверить" button, a "?" button, and the text "Попыток осталось: 1000" and "Время не ограничено".

Рисунок 2.6 – Проект интерфейса модуля практического этапа

Примечание: по нажатию кнопки «?» выводится информация о версии модуля и его разработчике.

Перейдем к определению информационного обеспечения, которое будет использоваться в информационной системе для проведения онлайн олимпиады по программированию.

2.3 Информационное обеспечение информационной системы

2.3.1 Используемые классификаторы и системы кодирования

При решении задачи проведения онлайн олимпиады по программированию используется ряд классификаторов и кодов.

Классификаторы разрабатываются как на уровне организации (предприятия), так и на уровне целых государств. Существуют следующие уровни классификаторов [2]:

- международные – используются по всему миру;
- общегосударственные – используются в пределах государств, но не могут противоречить международным классификаторам;
- отраслевые – используются в рамках отдельной отрасли, не могут противоречить общегосударственным классификаторам;
- локальные – используются на уровне отдельного предприятия или организации.

В таблице 2.2 приведен перечень уровней классификаторов и их краткие обозначения.

Таблица 2.2 – Перечень обозначений уровней классификаторов

Вид классификатора	Краткое обозначение
Международный	М
Общегосударственный	ОГ
Отраслевой	О
Локальный	Л

Под кодированием понимают присвоение кода объекту классификации. Назначение кодирования – формализованное описание наименований аспектов данных. Кодирование осуществляется по определенной системе. В таблице 2.3 приведен перечень систем кодирования и их краткие обозначения.

Таблица 2.3 – Перечень обозначений систем кодирования

Система кодирования	Краткое обозначение
Порядковая	П
Серийно-порядковая	СП
Разрядная (параллельная)	Р
Комбинированная (последовательная)	К

При использовании порядковой системы кодирования каждый объект множества кодируется с помощью текущего номера по порядку. Каждый последующий номер больше предыдущего на единицу.

При использовании серийно-порядковой системы в качестве кодов выступают числа натурального ряда с закрепленной отдельной серией таких чисел за классификационными объектами, обладающими одинаковыми признаками.

При кодировании разрядным методом признаки классификации кодируются определенными разрядами вне зависимости друг от друга.

При использовании комбинированной системы кодирования в кодовом обозначении знаки на каждой ступени деления зависят от результатов разбиения на всех предыдущих ступенях. Код группировки предоставляет информацию о последовательности характерных для нее признаков.

В таблице 2.4 приведены классификаторы и коды, использование которых предполагается при разработке информационной системы.

Таблица 2.4 – используемые классификаторы и коды

Имя объекта кодируемого множества	Значимость кода	Система кодирования	Вид классификатора	Код	Значение
Код участника	4	П	Л	1-9999	Порядковые номера участников
Код организатора	4	П	Л	1-9999	Порядковые номера организаторов
Код задания (тестового, практического)	7	Р	Л	2017001	2017 – год проведения олимпиады 001 – порядковый номер задания
Код результата (тестового, практического)	5	П	Л	1-99999	Порядковые номера результатов

Использование общей системы классификаторов и кодов облегчает построение и заполнение таблиц базы данных.

2.3.2 Характеристика нормативно-справочной и входной оперативной информации

Под входной информацией следует понимать всю информацию, необходимую для решения конкретной задачи. В частности, входная оперативная информация информационной системы для проведения олимпиады по программированию представляет собой заполнение и последующее сохранение в базу данных (либо в текстовые файлы) экранных форм, содержащих решения задач по программированию.

В качестве входной информации используются:

- исходные коды программ, написанных участниками в ходе выполнения заданий практической части олимпиады;
- ответы участников на тестовые задания.

2.3.3 Характеристика выходной информации

Участнику олимпиады доступны следующие виды выходной информации:

- решено ли тестовое задание правильно и не превышено ли допустимое время его выполнения;
- правильно ли решено практическое задание, а именно совпадает ли результат компиляции исходного кода с ожидаемым результатом, обнаружены ли при этом все необходимые ключевые слова и не превышено ли допустимое время выполнения данного задания;
- сколько баллов набрано в ходе прохождения практического этапа олимпиады и какое место занимает данный участник в общем рейтинге.

Организатору олимпиады доступны следующие виды выходной информации:

- какое количество баллов набрано каждым участником и какое место занимает он в общем рейтинге;
- исходные коды программ, написанных участниками (протокол FTP);
- контактные данные участников.

Информация выводится в соответствующих разделах сайта, входящего в состав информационной системы, либо непосредственно внутри модулей.

На данный момент собрано достаточно информации для перехода к проектированию используемой в информационной системе базы данных.

2.4 Логическое моделирование информационной системы

Логическое моделирование информационной системы представляет собой проявление интеграции реляционной и объектной технологий [16].

Сегодня реляционные СУБД являются одним из основных средств реализации баз данных. Основой для проектирования реляционных БД служит методология IDEF1X. По методологии IDEF1X в качестве логической модели данных информационной системы выступает диаграмма «сущность-связь». Представим данную диаграмму на рисунке 2.7.

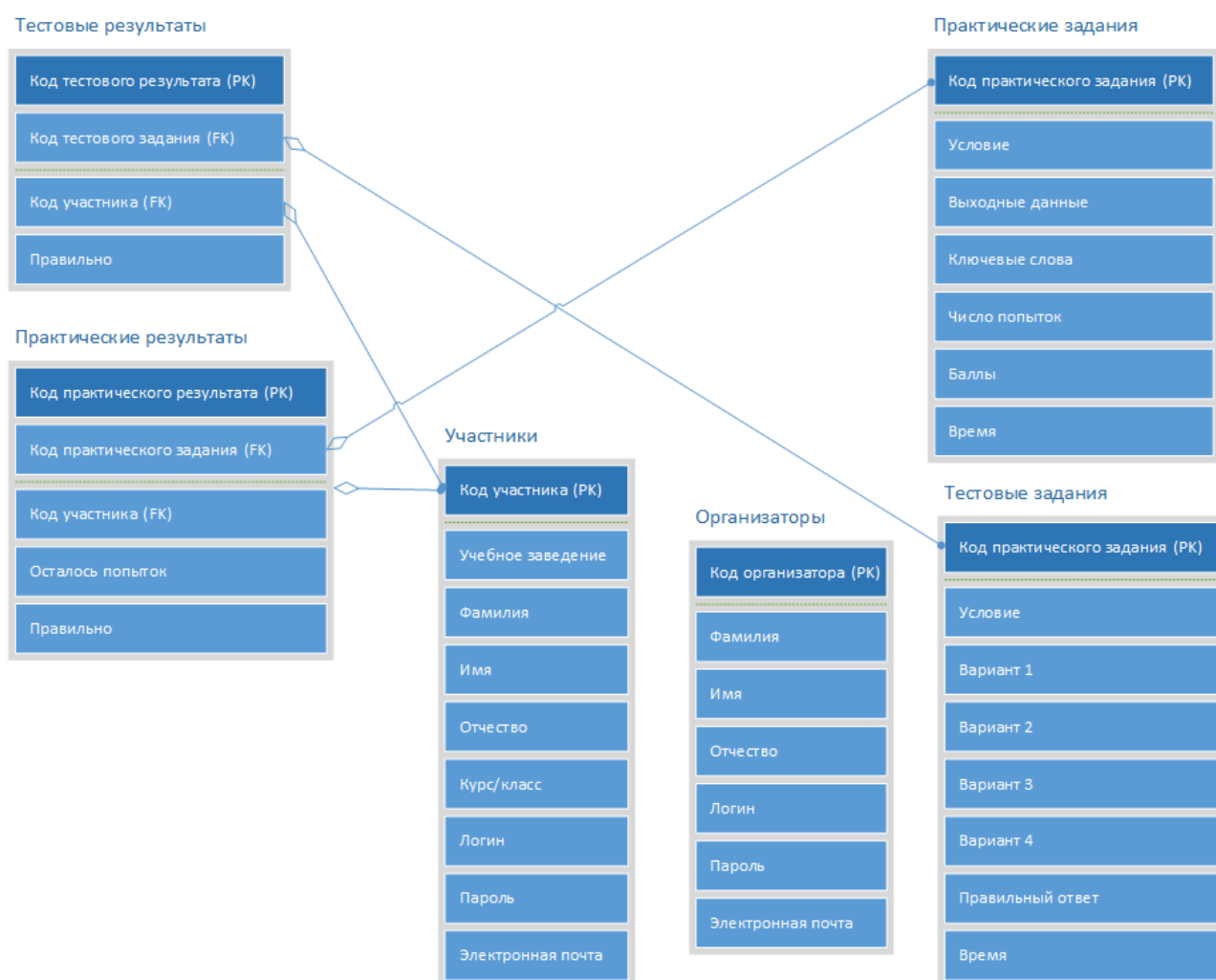


Рисунок 2.7 – Логическая модель данных

На диаграмме представлены такие сущности, как «Организаторы», «Участники», «Практические задания», «Практические результаты», «Тестовые задания», «Тестовые результаты». Отдельные сущности взаимосвязаны друг с другом.

Данный подход к моделированию служит для анализа информационной структуры систем различной природы. Модель является отображением логической структуры объектов информационной системы. На основании данной модели может быть создана база данных.

2.5 Требования к аппаратно-программному обеспечению информационной системы

Аппаратно-программное обеспечение информационной системы должно максимально эффективно использовать имеющиеся на предприятии технические средства.

Технический комплекс информационной системы состоит из следующих технических средств:

- Сервер БД и сервер приложений;
- Персональные компьютеры (ПК) пользователей;
- ПК администраторов.

Согласно переговорам, проведенным с заказчиком, предполагается размещение информационной системы на бесплатном хостинге. Следовательно, требования к серверам можно опустить, поскольку их соблюдение не зависит от возможностей автора настоящей работы.

Минимальные требования к техническим характеристикам ПК участника, ПК администратора и ПК организатора определяются минимальными системными требованиями наиболее требовательного компонента, необходимого для работы системы, которым является среда Java Runtime Environment (JRE). Ее минимальные системные требования таковы:

- Процессор – Intel Pentium II 266 МГц;
- Объем оперативной памяти – 128 Мб;

- Свободное место на жестком диске – 15 Мб (необходимо для размещения библиотеки *InMemoryCompiler*);

- Браузер Internet Explorer версии 9+, Safari x64, Opera или Firefox.

Под библиотекой *InMemoryCompiler* здесь понимается отдельная библиотека среды JRE, необходимая и достаточная для работы всей информационной системы.

Выводы по главе 2

Была построена объектная модель информационной системы – диаграмма вариантов использования. Также была построена диаграмма классов. На основании объектной модели и диаграммы классов разработана логическая модель данных согласно выбранной методологии. Определены требования к аппаратно-программному обеспечению информационной системы.

ГЛАВА 3 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

3.1 Выбор архитектуры информационной системы

Оба модуля информационной системы предполагается реализовать в виде web-приложений с целью упрощения доступа со стороны участников.

Web-приложения состоят из клиентской и серверной частей, тем самым реализуя технологию «клиент-сервер». Данная архитектура в общем случае может быть представлена в таких вариантах, как:

- двухзвенная клиент-серверная архитектура, состоящая из «толстого клиента» и сервера баз данных (СБД);
- трехзвенная клиент-серверная архитектура, состоящая из «тонкого клиента», сервера приложений и СБД.

Однако, web-приложения не могут быть реализованы в двухзвенной архитектуре, поскольку web-браузер («тонкий клиент») не способен работать напрямую с СБД. Необходимо остановить выбор на трехзвенной архитектуре.

Web-приложения, написанные на языке Java, называются Java-апплетами [12].

На рисунке 3.1 представлена архитектура разрабатываемой информационной системы.

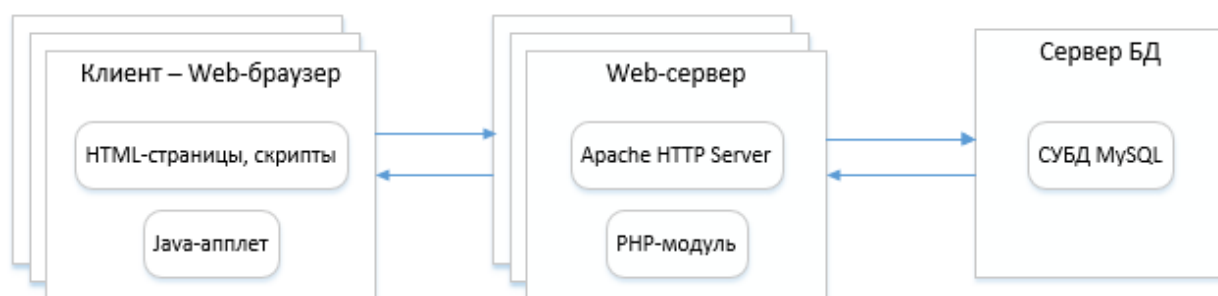


Рисунок 3.1 – Архитектура разрабатываемой информационной системы

Подобная архитектура предоставляет клиентам возможность удаленного доступа без установки на персональный компьютер дополнительного программного обеспечения и отличается безопасностью использования [17].

Перейдем к выбору среды разработки программного обеспечения.

3.2 Выбор среды разработки программного обеспечения

Для разработки программного обеспечения на языке Java используются интегрированные среды разработки (ИСР) [18].

Среди наиболее известных ИСР для разработки на языке Java можно выделить следующие:

- Eclipse;
- NetBeans;
- JBuilder.

Проведем сравнительный анализ выделенных ИСР. Результаты представим в таблице 3.1.

Таблица 3.1 – Сравнительный анализ ИСР

Критерий сравнения	Название интегрированной среды разработки		
	Eclipse	NetBeans	JBuilder
Популярность среди разработчиков [14]	Высокая (20%)	Средняя (7%)	Низкая (2%)
База сторонних расширений	Очень большая	Большая	Маленькая
Автоматизация кодирования	Возможно в полуавтоматическом режиме подключать недостающие библиотеки. Среда предлагает варианты исправления ошибок в коде	Возможно в полуавтоматическом режиме подключать недостающие библиотеки	Автоматизация кодирования отсутствует
Поддерживаемые операционные системы	Windows, Linux, macOS, Solaris	Windows, Linux, macOS, Solaris	Windows
Возможности настройки среды	Очень широкие	Широкие	Широкие
Лицензия	Eclipse Public License (EPL), предполагает свободное (бесплатное) использование ИСР	GNU General Public License (GPL), предполагает свободное (бесплатное) использование ИСР, при условии, что автор передает созданный при помощи данной ИСР программный код в общественную собственность	Проприетарная – за использование данной ИСР с разработчика взимается плата

Согласно проведенному анализу, наиболее предпочтительной средой разработки программного обеспечения информационной системы является Eclipse. Данная среда наиболее полно отвечает всем заданным критериям сравнения.

Теперь, когда выбрана среда разработки информационной системы, перейдем к разработке ее физической модели данных.

3.3 Разработка физической модели данных информационной системы

Для описания физической реализации БД следует выполнить физическое моделирование, которое подразумевает описание конкретной реализации базы данных, которая размещена во внешней памяти компьютера, описание структуры хранения данных [3].

На рисунке 3.2 представлена физическая модель данных информационной системы для проведения онлайн олимпиады по программированию.

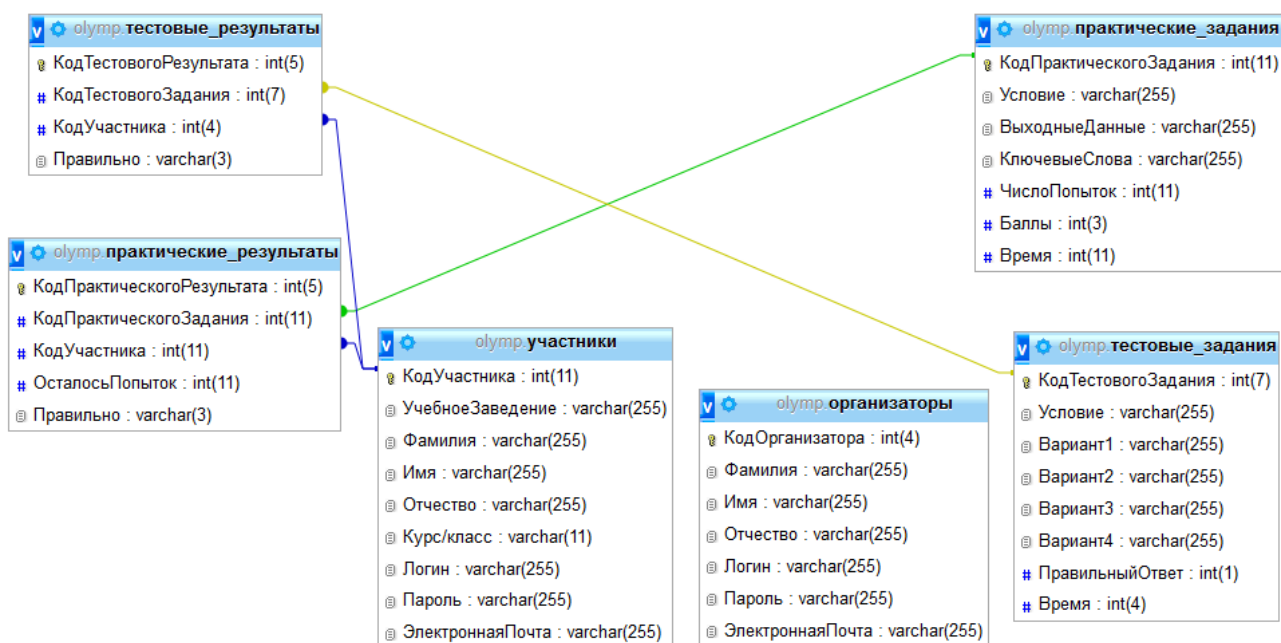


Рисунок 3.2 – Физическая модель данных

На данной модели представлены шесть таблиц: «тестовые_результаты», «практические_результаты», «тестовые_задания», «практические_задания», «участники», «организаторы». Каждый атрибут таблиц имеет свой

определенный тип данных. Отдельные таблицы связаны между собой при помощи атрибутов – первичного и внешнего ключей.

Перейдем к разработке программного обеспечения.

3.4 Разработка программного обеспечения информационной системы

Перед тем, как перейти к непосредственной разработке модулей, входящих в состав информационной системы, необходимо определить логику их работы [17]. Для этого следует построить блок-схемы, отражающие последовательность функционирования этих модулей.

На рисунке 3.3 представлена блок-схема для модуля отборочного этапа.

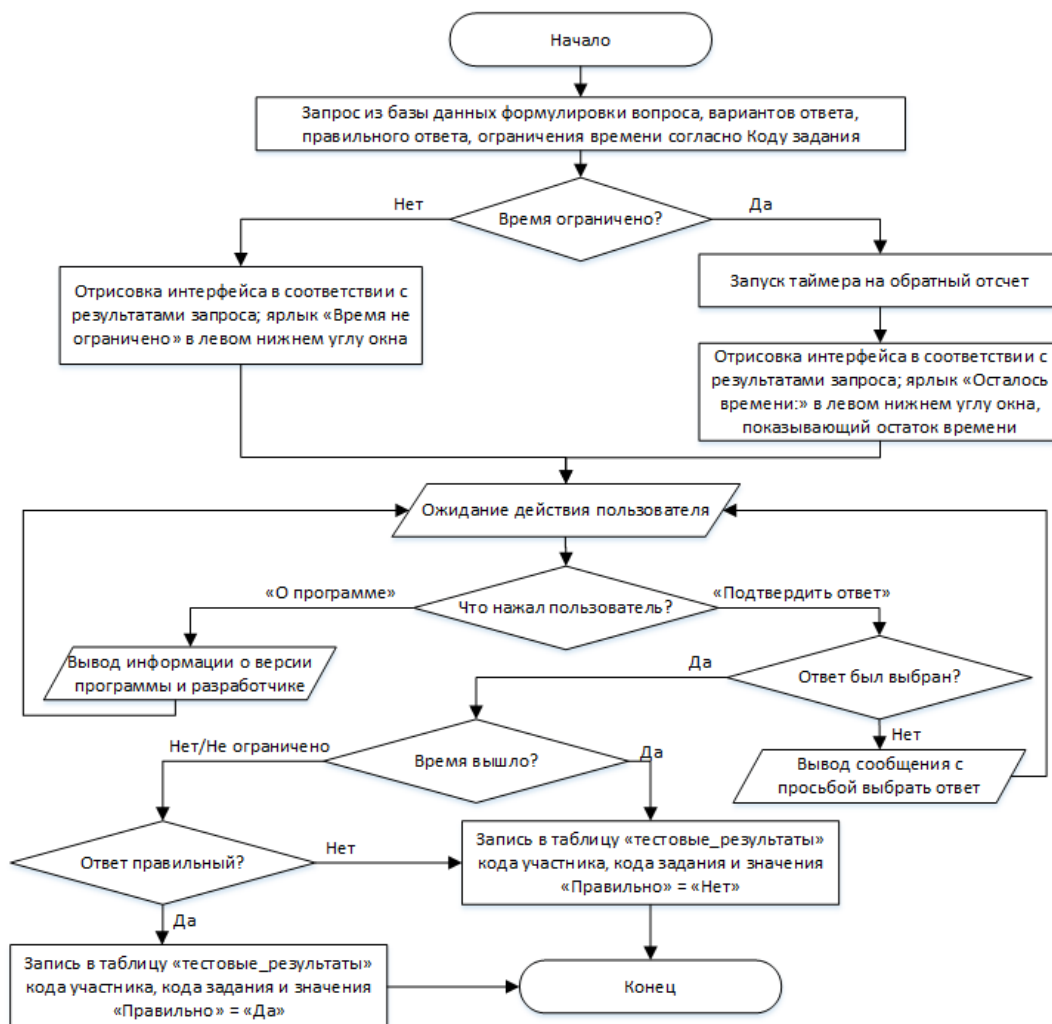


Рисунок 3.3 – Блок-схема для модуля отборочного этапа олимпиады

На рисунке 3.4 представлена блок-схема для модуля основного (практического) этапа олимпиады.

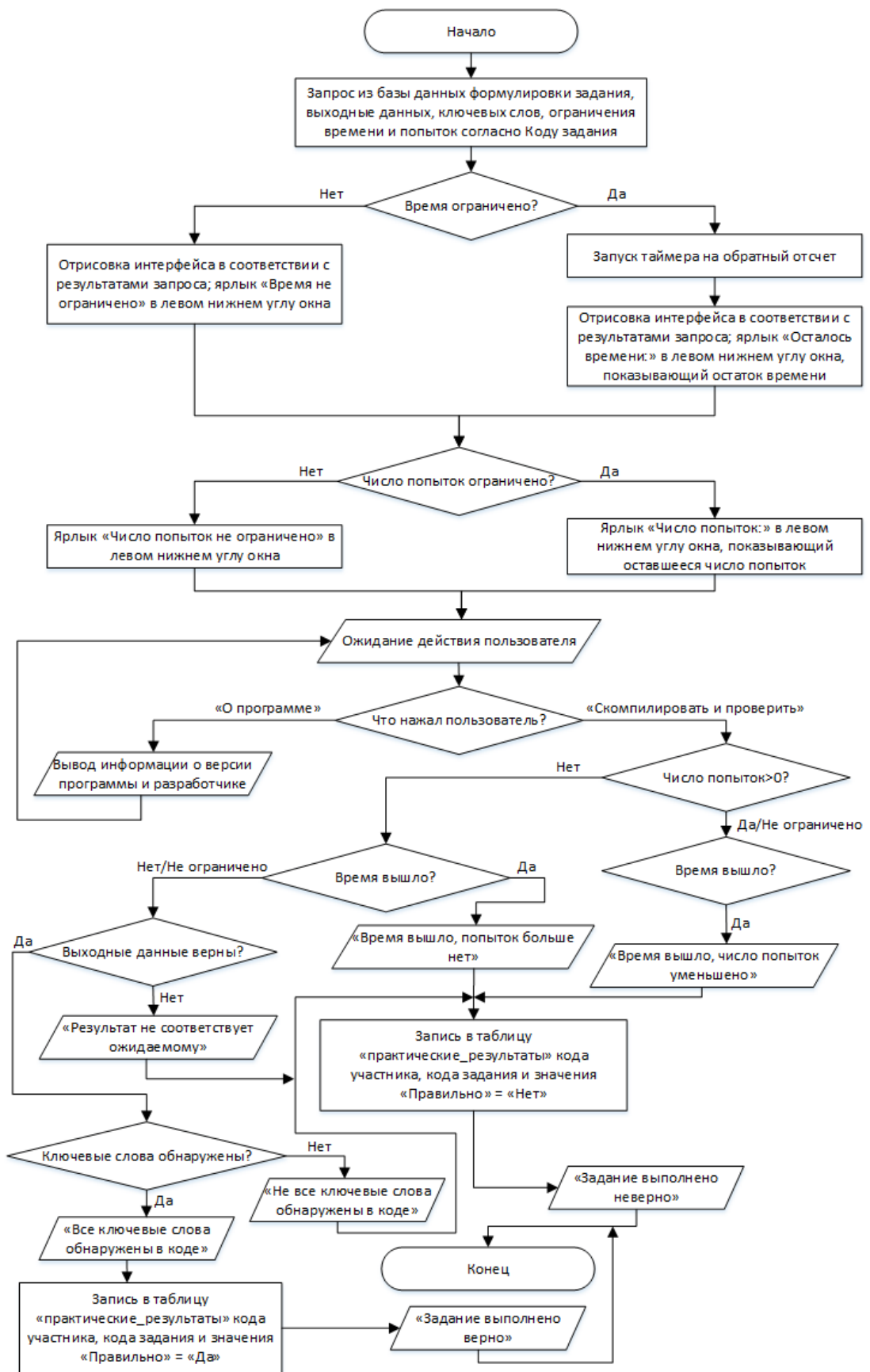


Рисунок 3.4 – Блок-схема для модуля практического этапа олимпиады

Перейдем к непосредственной реализации модулей в выбранной ИСР. При разработке будем учитывать построенные ранее блок-схемы и диаграммы классов.

Исходные коды разработанных модулей представлены в приложениях А (модуль тестового этапа) и Б (модуль практического этапа). Кроме того, поскольку модули являются web-приложениями, необходима разработка web-страниц, в которые модули будут встроены и которые будут передавать в модули переменные кода участника и кода задания. Также понадобятся страницы с выбором заданий, с авторизацией участника олимпиады, с регистрацией (т.е. подачей заявок на участие) новых участников и т.д. Таким образом, необходима разработка небольшого web-сайта, в состав которого войдут следующие web-страницы:

- index.php – содержит форму авторизации участников, а также ссылку на страницу registration.html;
- organizer.php – содержит форму авторизации организаторов олимпиады;
- login.php и loginOrganizer.php – страницы, осуществляющие непосредственную авторизацию пользователей и организаторов соответственно;
- results.php – содержит таблицу с результатами олимпиады;
- help.html – страница помощи участникам олимпиады;
- test.php – страница, в которую встроены модуль тестового этапа олимпиады;
- task.php – страница, в которую встроены модуль практического этапа олимпиады;
- registration.html – страница, содержащая форму регистрации участников олимпиады;
- registration.php – страница, осуществляющая регистрацию участников олимпиады – добавление введенных ими сведений в БД.

Для реализации связующего слоя между модулями информационной системы и базой данных необходима также разработка следующих web-страниц:

- `select.php` – страница, выполняющая запрос к базе данных, позволяющий узнать, сколько у данного участника осталось попыток на выполнение данного практического задания (при условии, что тот совершал хотя бы одну попытку);

- `selectAll.php` - страница, выполняющая запрос к базе данных, позволяющий узнать условие практического задания, число попыток, выходные данные, ключевые слова, ограничение времени;

- `selectAllTest.php` - страница, выполняющая запрос к базе данных, позволяющий узнать формулировку тестового вопроса, варианты ответов, правильный ответ, ограничение времени;

- `selectTest.php` - страница, выполняющая запрос к базе данных, позволяющий узнать, предпринималась ли данным участником попытка ответить на данное тестовое задание, и чем она завершилась;

- `sc.php` – страница, сохраняющая верные решения практических заданий в папку «`sourceCodes`», размещенную в корневой папке сайта;

- `testCorrect.php` – страница, сохраняющая в базе данных сведения о том, что данный участник верно ответил на данный вопрос;

- `testIncorrect.php` – страница, сохраняющая в базе данных сведения о том, что данный участник неверно ответил на данный вопрос;

- `completed.php` – страница, сохраняющая в базе данных сведения о том, что данный участник верно решил данное практическое задание;

- `tries.php` – страница, сохраняющая в базе данных сведения о том, что данный участник неверно решил данное практическое задание (при этом число оставшихся попыток у данного участника уменьшается на единицу).

Перейдем к более детальному описанию принципов функционирования разработанных модулей и web-страниц.

3.5 Описание функциональности информационной системы

Для наглядного представления основных этапов функционирования информационной системы можно разработать схему технологического процесса обработки информации. Данная схема представлена на рисунке 3.5.



Рисунок 3.5 – Схема технологического процесса обработки информации

Разберем каждый представленный на схеме этап более подробно. Так, на первом этапе будущий участник олимпиады проходит регистрацию, вводя в соответствующую web-форму личные данные. Данная форма размещена на странице registration.html. Вид формы представлен на рисунке 3.6.

Регистрация участников

Фамилия

Имя

Отчество

Имя пользователя

Пароль

Учебное заведение

№ курса или класса

© 2017. Автор и администратор сайта: [Алексей Луговой](#), группа ПИ6-1301.

Рисунок 3.6 – Форма регистрации участников

Далее участник проходит процедуру авторизации на сайте олимпиады (страница index.php). Вид формы авторизации показан на рисунке 3.7.

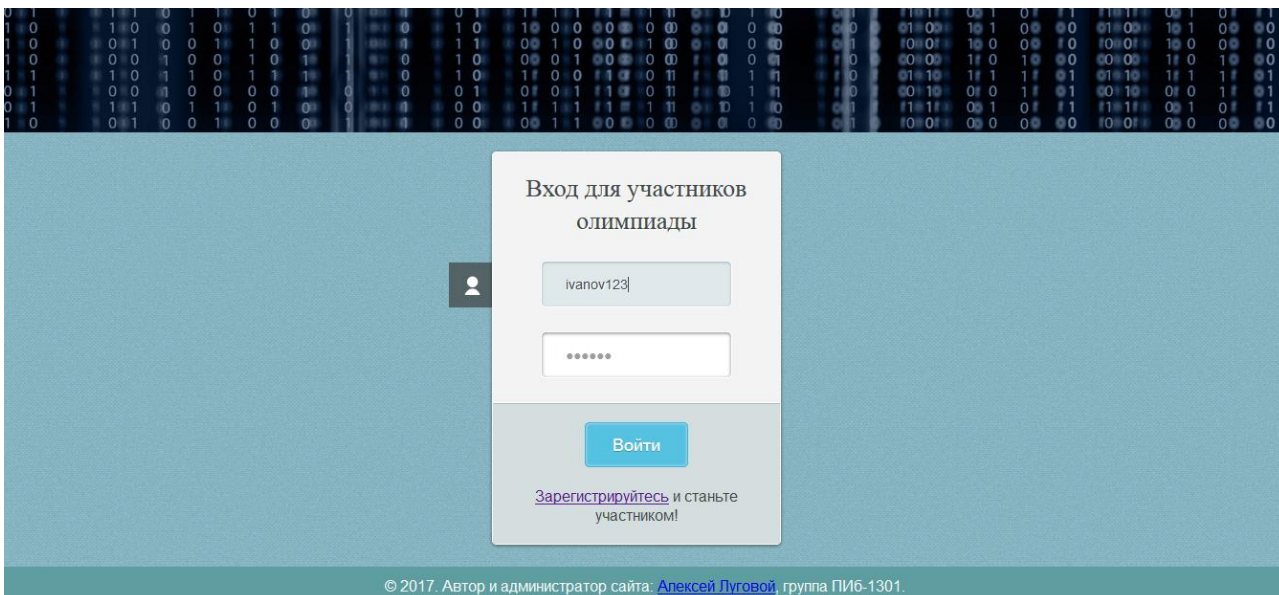


Рисунок 3.7 – Форма авторизации участников

Успешно авторизовавшийся участник получает доступ к отборочному (тестовому) этапу олимпиады. Авторизация сохраняется в форме сессии. Участник имеет право выполнять тестовые задания в любом порядке, однако для перехода к основному этапу олимпиады необходимо выполнить все предложенные задания, причем нужно ответить верно не менее, чем на 80% вопросов. Выбор заданий все так же происходит на странице index.php. Вид данной страницы с авторизовавшимся участником представлен на рисунке 3.8.

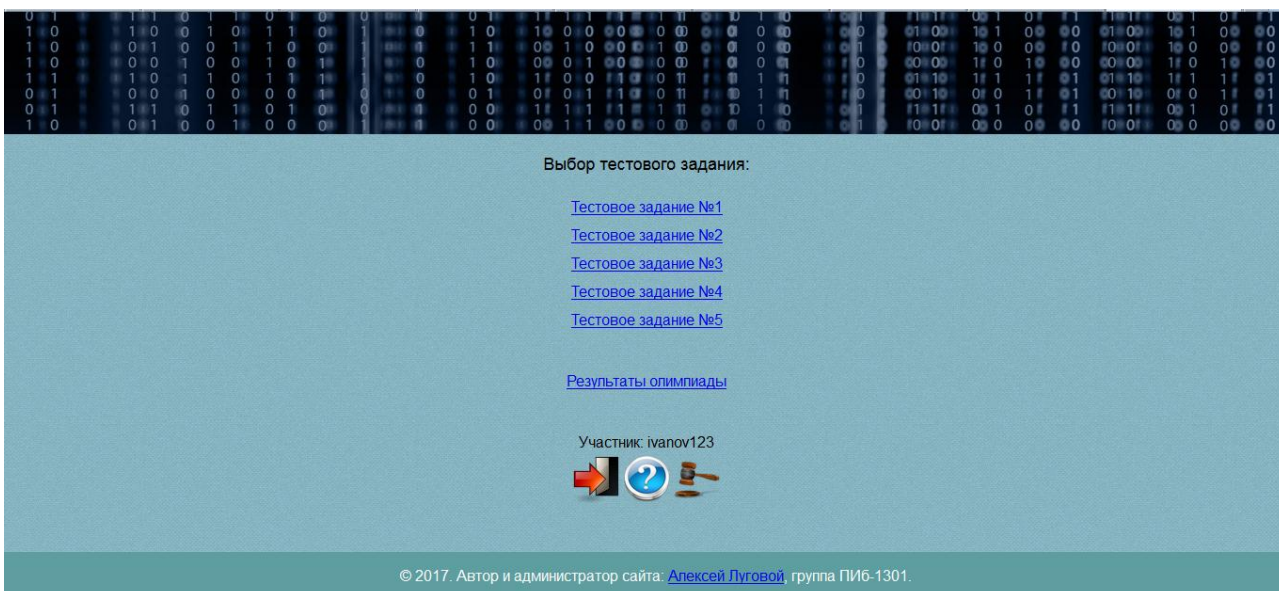


Рисунок 3.8 – Выбор тестовых заданий

После того, как участник выберет задание, он попадает на страницу test.php, на которой размещен модуль отборочного этапа олимпиады. В модуль

передаются переменные кода задания и кода участника. Вид страницы с модулем отборочного этапа представлен на рисунке 3.9.

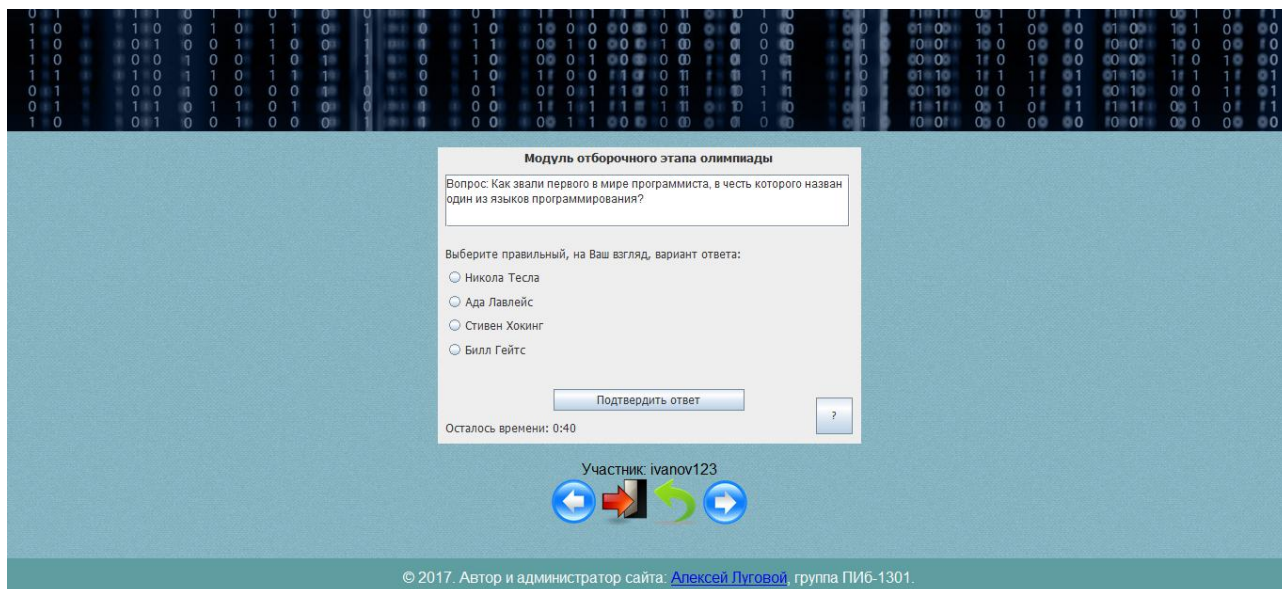


Рисунок 3.9 – Модуль отборочного этапа

После того, как участник выберет вариант ответа и нажмет кнопку «Подтвердить ответ», он получит сообщение о том, правильный ли выбран вариант. Результат будет отражен в базе данных олимпиады (таблица «тестовые_результаты»). Вид сообщения показан на рисунке 3.10.

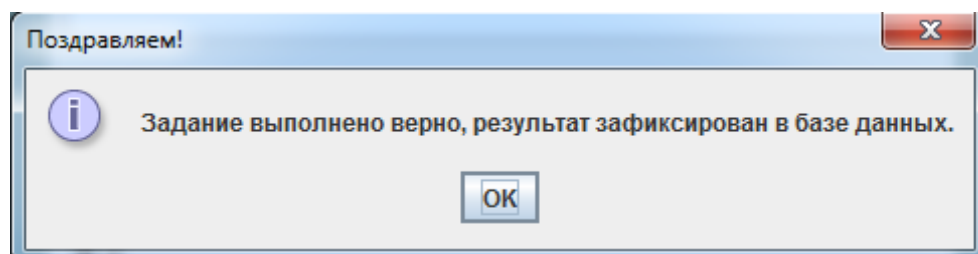


Рисунок 3.10 – Результат выполнения тестового задания

Повторное выполнение одного и того же задания не допускается, при последующих попытках участник получит соответствующее предупреждение.

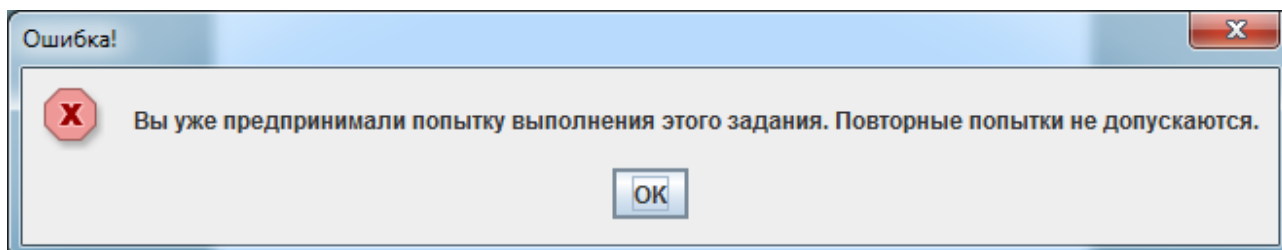


Рисунок 3.11 – Предупреждение о недопустимости повторных попыток

При помощи кнопок в нижней части страницы участник может перейти к выполнению следующего или предыдущего тестового задания, вернуться к выбору задания или выйти с сайта.

После того, как участник выполнит все задания отборочного этапа, производится подсчет процента правильно выполненных заданий. Если этот процент больше либо равен 80%, то на странице index.php вместо заданий отборочного этапа появляются задания основного этапа (рисунок 3.12).

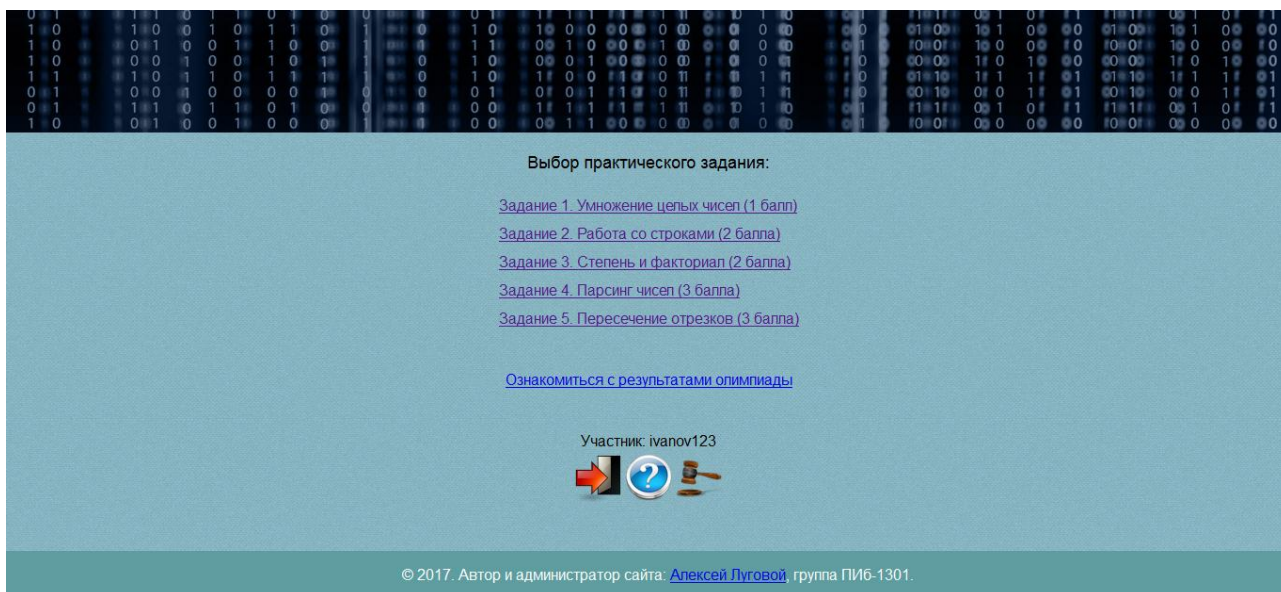


Рисунок 3.12 – Выбор практических заданий

После того, как участник выберет задание, он попадает на страницу task.php, где размещен модуль основного этапа олимпиады (рисунок 3.13).

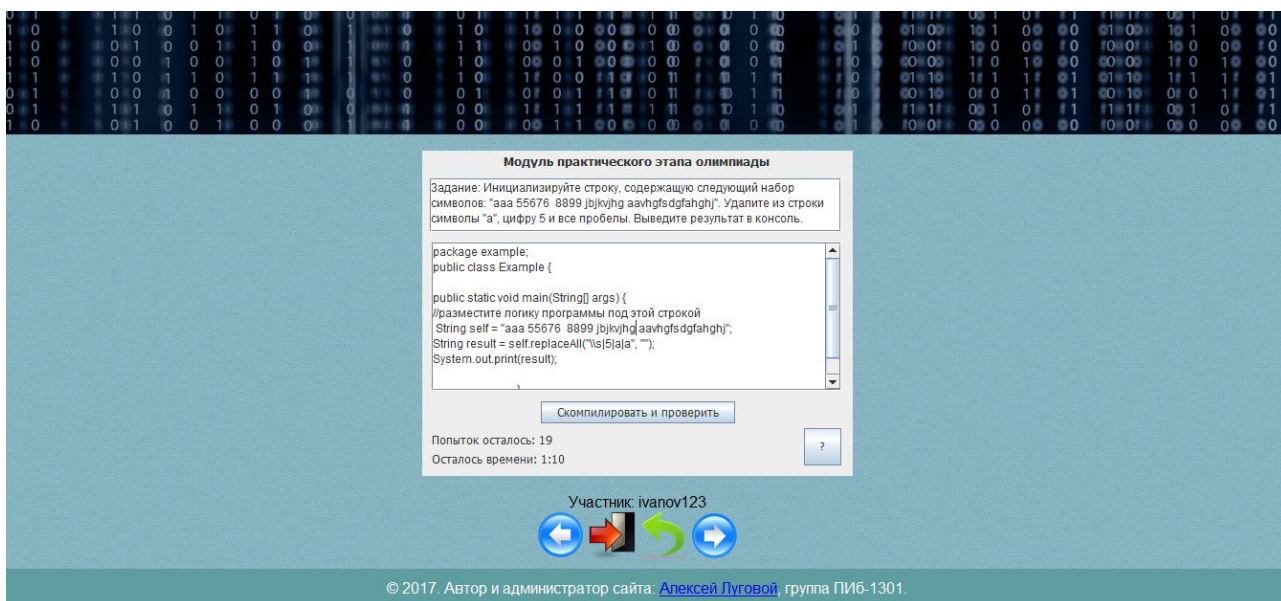


Рисунок 3.13 – Модуль основного (практического) этапа олимпиады

После того, как участник допишет недостающий программный код в соответствии с формулировкой задания и нажмет кнопку «Скомпилировать и проверить», он получит несколько сообщений. В первом сообщении содержится результат компиляции только что написанной участником программы, а также ожидаемый результат компиляции. Вид сообщения показан на рисунке 3.14.

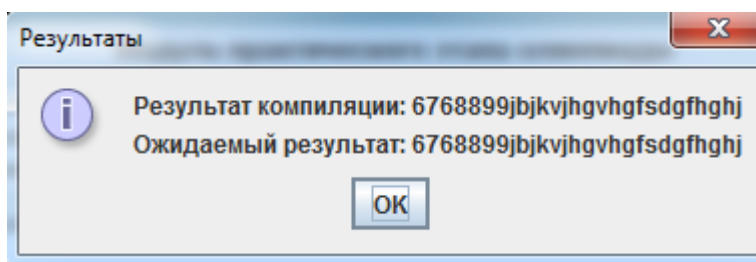


Рисунок 3.14 – Результаты компиляции

Во втором сообщении (рисунок 3.15) участник получает информацию о том, соответствует ли результат компиляции его программы ожидаемому результату.

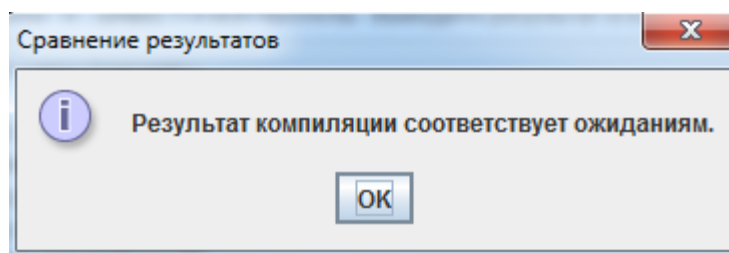


Рисунок 3.15 – Сравнение результатов

В третьем сообщении (рисунок 3.16) участник получает информацию о том, все ли требуемые ключевые слова были обнаружены в коде. В данном задании набор ключевых слов был таким: «String», «replace» и «aaa 55676 8899 jbjkvjhg aavhgfsdgfhghj»). Все три ключевых слова были употреблены в коде программы, о чем и сообщается участнику.

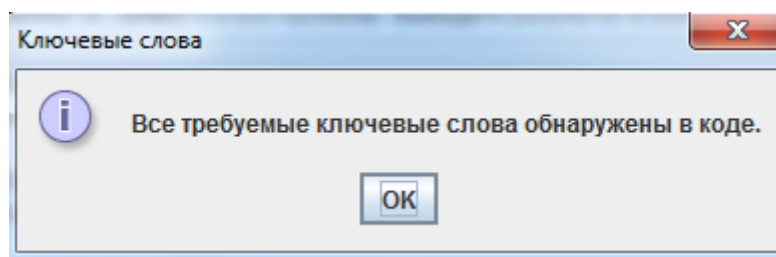


Рисунок 3.16 – Ключевые слова

В последнем сообщении участнику сообщается о том, засчитано его задание или нет. Задание засчитывается только при условии равенства фактических результатов компиляции ожидаемым и одновременном присутствии всех требуемых ключевых слов. В данном случае задание выполнено верно (рисунок 3.17).

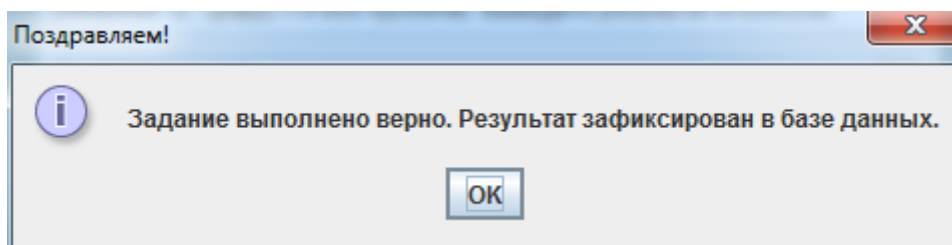


Рисунок 3.17 – Конечный результат выполнения задания

Одновременно происходит занесение результатов выполнения задания в таблицу «практические_результаты» базы данных. Повторное выполнение правильно выполненных заданий не допускается, при последующих попытках участник получит аналогичное изображенному на рисунке 3.11 предупреждение. Кроме того, исходный код программы, написанной участником, сохраняется на сервере в виде текстового файла.

Если в ходе выполнения тестового или практического задания у участника заканчивается отведенное время, появляется соответствующее предупреждение. Тестовое задание автоматически считается решенным неверно. Количество попыток на решение практических заданий уменьшается на единицу, аналогичная ситуация происходит и при неправильном выполнении практических заданий.

Участник имеет возможность ознакомиться с результатами олимпиады независимо от того, на каком этапе олимпиады он находится в данный момент. Для этого на странице с выбором задания размещена ссылка на таблицу результатов олимпиады (рисунок 3.18). В данной таблице показан рейтинг участника, т.е. место, которое он занимает в настоящее время.

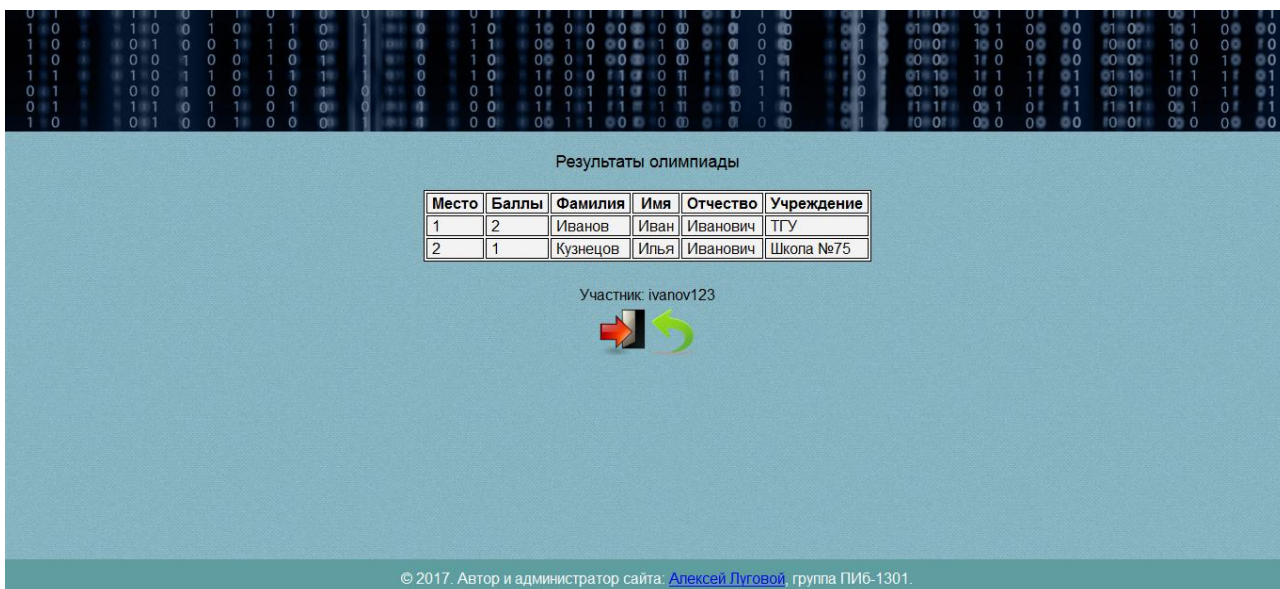


Рисунок 3.18 – Результаты олимпиады

В установленный организатором олимпиады день происходит окончательное подведение итогов олимпиады. Исходные коды программ, написанных участниками, подвергаются дополнительной проверке на отсутствие элементов нечестного выполнения. По результатам такой проверки победа отдельных участников может быть признана нелегитимной.

В конечном итоге организатор выделяет трех победителей олимпиады и оповещает их о победе посредством электронной почты. В отличие от участника, организатор имеет доступ к электронной почте участника и может увидеть, на каком курсе (в каком классе) обучается участник (рисунок 3.19).

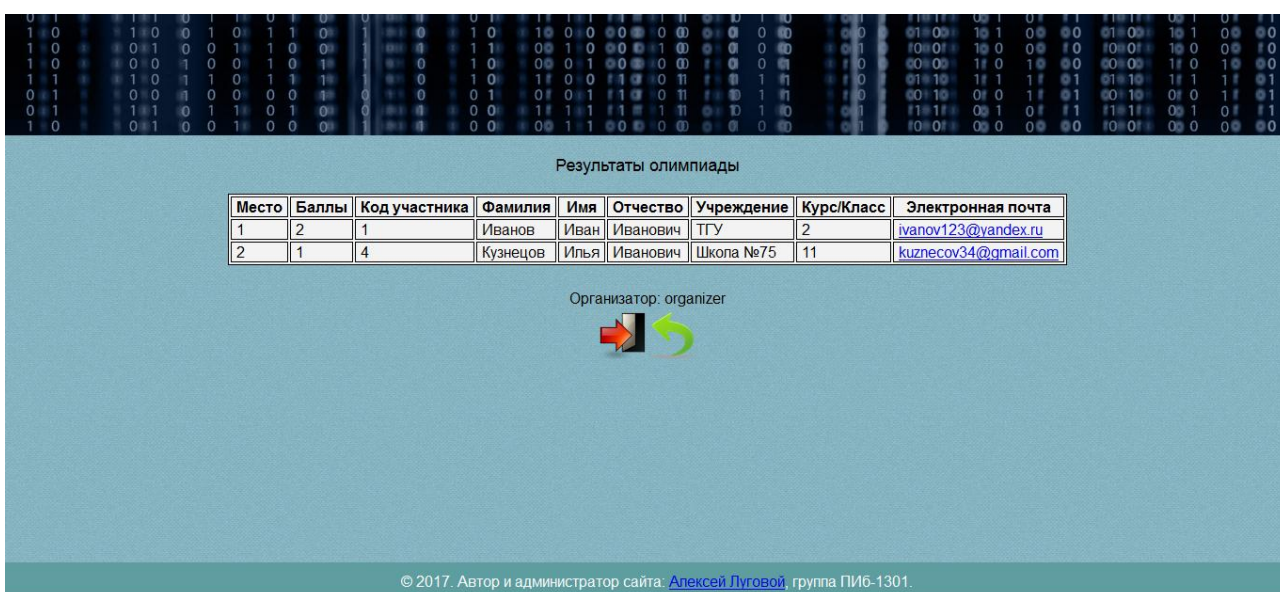


Рисунок 3.19 – Результаты олимпиады с точки зрения организатора

Перейдем к тестированию функционала получившегося программного проекта.

3.6 Тестирование программного проекта

Согласно рекомендациям заказчика, основным методом проверки качества полученного программного проекта будет ручное функциональное тестирования с использованием метода «Черного ящика». Тестирование методом «Черного ящика» — стратегия (метод) тестирования функционального поведения программы или системы с точки зрения внешнего мира, при котором не используется знание о внутреннем устройстве тестируемого объекта [6].

Тестирование методом «Черного ящика» предполагает следующие тестовые шаги [13]:

1. Проверка наличия в разработанном приложении основных функциональных требований и тестирование основных функций системы.
2. Тестирование основных функциональных возможностей системы.
3. Тестирование бизнес-логики.
4. Тестирование дополнительных функций системы.

В таблицу 3.2 поместим все необходимые для тестирования тест-кейсы согласно тестовым шагам.

Таблица 3.2 – Тест-кейсы

№	Название	Инструкции	Ожидаемый результат
1	Проверка наличия в разработанном приложении основных функциональных требований и тестирование основных функций системы		
1.1	Наличие сайта олимпиады, на котором могут быть размещены модули для ее проведения	В браузере перейти по адресу сайта олимпиады: http://olympnc.16mb.com/	По указанному адресу расположена стартовая страница сайта олимпиады по программированию
1.2	Наличие на сайте олимпиады модуля для проведения отборочного этапа олимпиады	Зайти на сайт олимпиады, убедиться в наличии модуля отборочного этапа	Модуль отборочного этапа присутствует на сайте
1.3	Наличие на сайте олимпиады модуля для проведения основного этапа олимпиады	Зайти на сайт олимпиады, убедиться в наличии модуля основного этапа	Модуль основного этапа присутствует на сайте

1.4	Наличие на сайте олимпиады таблицы результатов	Зайти на сайт олимпиады, убедиться в наличии таблицы результатов	Таблица результатов присутствует на сайте
2	Тестирование основных функциональных возможностей системы		
2.1	Корректность работы модуля тестового этапа олимпиады (в том числе и с базой данных)	Зайти на сайт олимпиады, выбрать любое доступное тестовое задание. Дать заведомо неверный ответ. Выбрать другое задание. Дать заведомо верный ответ	Сообщение, которое видит пользователь, по смыслу соответствует корректности ответа. В базе данных отражается правильный по смыслу результат
2.2	Корректность компиляции кода модулем основного этапа олимпиады	Написать программу, выводящую в консоль строчный формат данных. Затем написать программу, выводящую в консоль целочисленный формат данных. Наконец, написать программу, выводящую в консоль формат числа с плавающей точкой	Все три вывода в консоль соответствуют исходному коду
2.3	Корректность сравнения фактического результата компиляции с ожидаемым (модуль основного этапа)	Вывести в консоль сообщение, равное ожидаемому результату	Пользователь получает сообщение о том, что фактический результат компиляции соответствует ожидаемому
2.4	Корректность обнаружения ключевых слов в коде (модуль основного этапа)	Употребить в коде программы все требуемые заданием ключевые слова; употребить часть слов; не употреблять вообще	Только в случае употребления в коде всех ключевых слов пользователь получает сообщение о том, что все ключевые слова были обнаружены
2.5	Корректность оценки правильности выполнения заданий основного этапа олимпиады	Написать программу, выдающую результат, который соответствует ожидаемому заданием. При этом в программе должны содержаться все требуемые заданием ключевые слова	Пользователь получает сообщение «Задание выполнено верно» только при условии наличия всех ключевых слов и равенства фактического результата компиляции ожидаемому
2.6	Корректность работы модуля основного этапа с базой данных	Выполнить задание неверно; выполнить другое задание верно	В базе данных отражается правильный по смыслу результат
3	Тестирование бизнес-логики		
3.1	Недопустимость повторных попыток выполнить тестовое задание	Выполнить любое тестовое задание; попытаться выполнить его повторно	При повторной попытке появляется соответствующее предупреждение, модуль не запускается

3.2	Корректность работы таймера при выполнении тестовых заданий	Начать выполнение тестового задания; дождаться истечения отведенного времени	По истечении отведенного времени пользователь получает сообщение об этом, в базу вносится запись
3.3	Недопустимость повторных попыток выполнить практическое задание	Выполнить любое практическое задание; попытаться выполнить его повторно	При повторной попытке появляется соответствующее предупреждение, модуль не запускается
3.4	Корректность работы таймера при выполнении практических заданий	Начать выполнение практического задания; дождаться истечения отведенного времени	По истечении отведенного времени пользователь получает сообщение об этом, в базу вносится запись
3.5	Корректность работы ограничителя числа попыток при выполнении практических заданий	Исчерпать все отведенные на выполнение попытки; предпринять еще одну	Пользователь получает сообщение о том, что все попытки были исчерпаны, задание не проверяется
3.6	Корректность начисления и суммирования баллов за выполнение практических заданий	Правильно выполнить несколько практических заданий; открыть таблицу результатов	В таблице результатов отражена сумма баллов за выполненные задания
4	Тестирование дополнительных функций системы		
4.1	Корректность работы обратной связи с администратором	Попытаться связаться с администратором через ссылку в футере сайта	Происходит вызов почтового клиента по умолчанию
4.2	Корректность работы функции добавления заданий организатором	Зайти на сайт от имени администратора; добавить любое задание	Добавленное задание корректно отображается в базе данных

После того, как сформированы тест-кейсы, необходимо на их основе произвести тестирование информационной системы. Все найденные недочеты оформим в виде баг-репорта (таблица 3.3).

Таблица 3.3 – Баг-репорт

№	Описание бага	Компонент, приоритет	Шаги	Фактический результат	Ожидаемый результат
1	Множественное начисление баллов за правильное выполнение заданий с бесконечным числом попыток	Функциональность, средний	Правильно выполнить практическое задание, число попыток на выполнение которого не ограничено; повторно правильно выполнить это же задание	Участнику дважды начисляются баллы за выполнение задания	Участник получает баллы только за первое выполнение задания

Во избежание появления описанного в таблице 3.3 бага организаторам олимпиады рекомендуется устанавливать нулевое количество баллов за выполнение заданий с неограниченным числом попыток. Подобные задания следует расценивать как тренировочные.

На данный момент программный продукт готов к использованию. Перейдем к подведению итогов данной главы.

Выводы по главе 3

Была выбрана архитектура информационной системы для проведения онлайн олимпиады по программированию и среда для разработки ее модулей. Построена физическая модель данных информационной системы. Разработана сама информационная система. Проведено тестирование компонентов информационной системы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была разработана информационная система для проведения онлайн олимпиады по программированию. Для этого были предприняты следующие шаги:

- проведена идентификация предметной области;
- описаны проблемы, связанные с отсутствием информационной системы;
- составлены модели бизнес-процессов «КАК ЕСТЬ» и «КАК ДОЛЖНО БЫТЬ»;
- проведен анализ родственных работ в выбранной предметной области;
- сформулированы функциональные и нефункциональные требования к информационной системе;
- разработаны объектная и логическая модели данных для выбранной предметной области;
- произведена непосредственная реализация информационной системы;
- проведено тестирование информационной системы на предмет соответствия сформулированным требованиям.

В результате достигнута цель выпускной квалификационной работы и выполнены поставленные задачи.

Разработанная информационная система позволит расширить аудиторию участников олимпиады по программированию и частично автоматизирует ее проведение в заочной форме.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Бодров, О.А. Предметно-ориентированные экономические информационные системы: учеб. пособие / О.А. Бодров. – М.: Гор. линия-Телеком, 2013. – 244 с.
2. Варфоломеева, А.О. Информационные системы предприятия: Учебное пособие / А.О. Варфоломеева, А.В. Коряковский, В.П. Романов. – М.: НИЦ ИНФРА-М, 2013. – 283 с.
3. Голицина, О.Л., Максимов Н.В., Попов И.И. Базы данных: Учебное пособие. – М.: Форум: ИНФРА-М, 2013. – 352 с.
4. Заботина, Н.Н. Проектирование информационных систем – М.: ДРОФА, 2013. – 336 с.
5. Мезенцев, К.Н. Автоматизированные информационные системы: Учебник для студентов учреждений среднего профессионального образования / К.Н. Мезенцев. – М.: ИЦ Академия, 2013. – 176 с.
6. Олейник, П.П. Корпоративные информационные системы: Учебник для вузов. Стандарт третьего поколения / П.П. Олейник. – СПб.: Питер, 2012. – 176 с.
7. Парамонов Ф.И., Колесниченко О.В. Основы проектирования АСУП: Учебное пособие. – М.: Изд-во МАИ, 2015. – 92 с.
8. Рудинский, И.Д. Технология проектирования автоматизированных систем обработки информации и управления: учеб. пособие / И. Д. Рудинский. – М.: Горячая линия – Телеком, 2014. – 304 с.
9. Солганик, Г.Я. Толковый словарь: Язык газеты, радио, телевидения: около 10 000 слов и выражений / Г. Я. Солганик. – М.: АСТ, 2013. – 751 с.
10. Тарасова, О.П. Организация проектной деятельности дизайнера: учеб. пособие / О.П. Тарасова. – Оренбург: ОГУ, 2013. – 51 с.
11. Чистов, Д.В. Проектирование информационных систем. Учебник и практикум / Д.В. Чистов, П.П. Мельников, А.В. Золотарюк, Н.Б. Ничепорук. – М.: Юрайт, 2016. – 260 с.

Электронные ресурсы

12. Cope, G. The Future of Java Applets, 2017 // Algosome Software Design [Электронный ресурс]: сайт о разработке программного обеспечения: <http://www.algosome.com/articles/future-of-java-applets.html>

13. Parag. D. Writing Effective Test Cases for Any Application, 2015 // uTest – The Professional Network for Testers [Электронный ресурс]: сайт о тестировании программного обеспечения: <https://www.utest.com/articles/writing-effective-test-cases-for-any-application>

14. Ладик Ю.В. Обзор IDE средств для программирования на Java, 2013 // JavaPortal.ru – Все о Java и JavaScript [Электронный ресурс]: сайт о разработке программного обеспечения: <http://www.javaportal.ru/projects/taidej/results.html>

15. Репин, В.В. Сравнительный анализ нотаций ARIS/IDEF и поддерживающих их продуктов, 2014 // iTeam.Ru – технологии корпоративного управления [Электронный ресурс]: сайт консалтинговой компании: https://iteam.ru/publications/it/section_51/article_2518

Литература на иностранном языке

16. Bailis, P. Readings in Database Systems – 5th edition, MIT Press, 2015.

17. Esposito, D. Architecting Applications for the Enterprise – 2nd edition, Microsoft Press, 2014.

18. Fusco, M. Java 8 in action – 2st edition, Pearson PLC, 2014.

19. Haridi, S. Concepts, Techniques, and Models of Computer Programming – 3rd edition, MIT Press, 2014.

20. Kalloniatis C. Innovative Information Systems Modelling Techniques – InTech Press, 2016.

ПРИЛОЖЕНИЕ А

Листинг исходного кода модуля отборочного этапа олимпиады

```
public class Applet extends JApplet {
    private int count;
    private Timer timer;
    public Applet() {
        JPanel panel1 = new JPanel(new GridLayout(1, 2));

    }

    static JButton button = new JButton("Подтвердить ответ");
    private static final long serialVersionUID = 1L;
    static int kodUch; //Переменные, получаемые апплетом извне
    static int kodZ; //Переменные, получаемые апплетом извне
    String p1="", p2="", p3="", p4="", p5="", p6="";
    JLabel timerLabel = new JLabel("Осталось времени:");
    JRadioButton var1 = new JRadioButton("Вариант 1");
    JRadioButton var2 = new JRadioButton("Вариант 2");
    JRadioButton var3 = new JRadioButton("Вариант 3");
    JRadioButton var4 = new JRadioButton("Вариант 4");

    private class MySecurityManager extends SecurityManager {
        @Override
        public void checkPermission(Permission perm) {
            return;
        }
    }

    public void init() {
        ButtonGroup bG = new ButtonGroup();
        bG.add(var1);
        bG.add(var2);
        bG.add(var3);
        bG.add(var4);

        SecurityManager sm = new MySecurityManager();
        System.setSecurityManager(sm);

        kodZ=Integer.parseInt(getParameter("kodZ"));
        kodUch=Integer.parseInt(getParameter("kodUch"));
        getContentPane().setLayout(null);

        String
strTries="http://www.olymp.tla/selectTest.php?kodUch="+kodUch+"&kodZ="+kod
Z;
```

```

String inputLine="0";
try {
    //Адрес подключения
    URL url=new URL(strTries);
    //HTTP-подключение
    HttpURLConnection conn=(HttpURLConnection)url.openConnection();
    //подключаемся
    conn.connect();

    //поток чтения данных
    BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream()));

    //получаем строку-ответ
    inputLine=in.readLine().trim();

    //закрываем поток
    in.close();
    //отключаемся
    conn.disconnect();
    conn=null;

    String strAll="http://www.olymp.tla/selectAllTest.php?kodZ="+kodZ;
    url=new URL(strAll);
    //HTTP-подключение
    conn=(HttpURLConnection)url.openConnection();
    //подключаемся
    conn.connect();

    //поток чтения данных
    in=new BufferedReader(new
InputStreamReader(conn.getInputStream(),"UTF-8"));

    //получаем строку-ответ
    String inputLine2=in.readLine().trim();

    //закрываем поток
    in.close();
    //отключаемся
    conn.disconnect();
    conn=null;
    String[] inputLineAll=inputLine2.split("~@@~@~",7);
    p1 = inputLineAll[0]; //Условие
    p2 = inputLineAll[1]; //Вариант ответа 1
    p3 = inputLineAll[2]; //Вариант ответа 2

```



```

    p4 = inputLineAll[3]; //Вариант ответа 3
    p5 = inputLineAll[4]; //Вариант ответа 4
    p6 = inputLineAll[5]; //Правильный вариант ответа
    var1.setText(p2);
    var2.setText(p3);
    var3.setText(p4);
    var4.setText(p5);
    count = Integer.parseInt(inputLineAll[6]); //Осталось времени
}
catch (Exception f) {JOptionPane.showMessageDialog(null, "Ошибка: нет
соединения с базой данных.", "Ошибка!", JOptionPane.ERROR_MESSAGE);}
if (!inputLine.equals("nf")) {
    JOptionPane.showMessageDialog(null, "Вы уже предпринимали
попытку выполнения этого задания. Повторные попытки не
допускаются.", "Ошибка!", JOptionPane.ERROR_MESSAGE );
    System.exit(0);
}
else {
    try {
        String strAll="http://www.olymp.tla/selectAllTest.php?kodZ="+kodZ;
        URL url=new URL(strAll);
        //HTTP-подключение
        HttpURLConnection conn=(HttpURLConnection)url.openConnection();
        //подключаемся
        conn.connect();

        //поток чтения данных
        BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream(),"UTF-8"));

        //получаем строку-ответ
        inputLine=in.readLine().trim();

        //закрываем поток
        in.close();
        //отключаемся
        conn.disconnect();
        conn=null;
    }
    catch (Exception f) {JOptionPane.showMessageDialog(null, "Ошибка: нет
соединения с базой данных.", "Ошибка!", JOptionPane.ERROR_MESSAGE);}
    String[] inputLineAll=inputLine.split("~@@~",4);
    p1 = inputLineAll[0]; //Условие
    p2 = inputLineAll[1]; //Число попыток (начальное)
    p3 = inputLineAll[2]; //Выходные данные

```

```

        p4 = inputLineAll[3]; //Ключевые слова
    }
    if (count==0) timerLabel.setText("Время не ограничено");
    else {
        timer = new Timer(1000, new ActionListener(){
            @Override
            public void actionPerformed(ActionEvent e) {
                int minutes = count / 60;
                int seconds = count % 60;
                count--;
                if (count == -1) {
                    JOptionPane.showMessageDialog(null, "Время вышло. Задание не
засчитано.", "Внимание!", JOptionPane.WARNING_MESSAGE);
                    timer.stop();
                    try {
                        String
strAll="http://www.olymp.tla/testIncorrect.php?kodZ="+kodZ+"&kodUch="+kodUc
h;
                        URL url=new URL(strAll);
                        //HTTP-подключение
                        HttpURLConnection
conn=(HttpURLConnection)url.openConnection();
                        //подключаемся
                        conn.connect();
                        BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                        in.close();
                        //отключаемся
                        conn.disconnect();
                        conn=null;
                    } catch (Exception g) {}
                    System.exit(0);
                }
                if (seconds>=10) timerLabel.setText("Осталось времени:
"+String.valueOf(minutes)+":"+String.valueOf(seconds));
                else timerLabel.setText("Осталось времени:
"+String.valueOf(minutes)+":0"+String.valueOf(seconds));
            }
        });
        timer.start();
    }
    final String[] result=new String[2];
    result[0]=p3;
    result[1]=p4;

```

```

        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent arg0) {
                if
                (!var1.isSelected()&&!var2.isSelected()&&!var3.isSelected()&&!var4.isSelected())
                JOptionPane.showMessageDialog(null, "Вы не выбрали ни один из вариантов
                ответа!", "Внимание", JOptionPane.WARNING_MESSAGE);
                else if
                ((p6.equals("1")&&var1.isSelected())||(p6.equals("2")&&var2.isSelected())||(p6.equal
                s("3")&&var3.isSelected())||(p6.equals("4")&&var4.isSelected())) {
                    try {
                        String
                strAll="http://www.olymp.tla/testCorrect.php?kodZ="+kodZ+"&kodUch="+kodUch;
                        URL url=new URL(strAll);
                        //HTTP-подключение
                        HttpURLConnection
                conn=(HttpURLConnection)url.openConnection();
                        //подключаемся
                        conn.connect();
                        BufferedReader in=new BufferedReader(new
                InputStreamReader(conn.getInputStream()));
                        in.close();
                        //отключаемся
                        conn.disconnect();
                        conn=null;
                        JOptionPane.showMessageDialog(null, "Задание выполнено верно,
                результат зафиксирован в базе
                данных.", "Поздравляем!", JOptionPane.INFORMATION_MESSAGE);
                        System.exit(0);
                    } catch (Exception e) {JOptionPane.showMessageDialog(null, "Ошибка
                при добавлении данных в базу.", "Ошибка!", JOptionPane.ERROR_MESSAGE);}
                }
                else {
                    try {
                        String
                strAll="http://www.olymp.tla/testIncorrect.php?kodZ="+kodZ+"&kodUch="+kodUc
                h;
                        URL url=new URL(strAll);
                        //HTTP-подключение
                        HttpURLConnection
                conn=(HttpURLConnection)url.openConnection();
                        //подключаемся
                        conn.connect();
                        BufferedReader in=new BufferedReader(new
                InputStreamReader(conn.getInputStream()));

```

```

        in.close();
        //отключаемся
        conn.disconnect();
        conn=null;
        JOptionPane.showMessageDialog(null, "Задание выполнено
неверно, результат зафиксирован в базе
данных.", "Увы!", JOptionPane.INFORMATION_MESSAGE);
        System.exit(0);

    } catch (Exception e) {JOptionPane.showMessageDialog(null,
"Ошибка при добавлении данных в базу.", "Ошибка!",
JOptionPane.ERROR_MESSAGE);}
    }
}
});
button.setFont(new Font("Tahoma", Font.PLAIN, 12));
button.setBounds(123, 257, 203, 23);
getContentPane().add(button);

JScrollPane scrollPane_1 = new JScrollPane();
scrollPane_1.setBounds(8, 29, 430, 56);
getContentPane().add(scrollPane_1);

JTextArea zadanie = new JTextArea();
zadanie.setWrapStyleWord(true);
zadanie.setLineWrap(true);
scrollPane_1.setViewportView(zadanie);
zadanie.setEditable(false);
zadanie.setText("Вопрос: "+p1);

JLabel title = new JLabel("Модуль отборочного этапа олимпиады");
title.setHorizontalAlignment(SwingConstants.CENTER);
title.setFont(new Font("Tahoma", Font.BOLD, 13));
title.setBounds(10, 4, 428, 14);
getContentPane().add(title);

timerLabel.setFont(new Font("Tahoma", Font.PLAIN, 12));
timerLabel.setBounds(8, 291, 180, 14);
getContentPane().add(timerLabel);

JLabel advice = new JLabel("Выберите правильный, на Ваш взгляд,
вариант ответа:");
advice.setFont(new Font("Tahoma", Font.PLAIN, 12));
advice.setBounds(8, 106, 320, 14);
getContentPane().add(advice);

```

```
var1.setFont(new Font("Tahoma", Font.PLAIN, 12));
var1.setBounds(8, 127, 109, 23);
getContentPane().add(var1);
```

```
var2.setFont(new Font("Tahoma", Font.PLAIN, 12));
var2.setBounds(8, 153, 109, 23);
getContentPane().add(var2);
```

```
var3.setFont(new Font("Tahoma", Font.PLAIN, 12));
var3.setBounds(8, 178, 109, 23);
getContentPane().add(var3);
```

```
var4.setFont(new Font("Tahoma", Font.PLAIN, 12));
var4.setBounds(8, 204, 109, 23);
getContentPane().add(var4);
```

```
JButton aboutBtn = new JButton("?");
aboutBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        JOptionPane.showMessageDialog(null, "Информационная система
для проведения онлайн олимпиады по программированию\nМодуль
отборочного этапа олимпиады v1.0, 2017г.\nАвтор: Луговой А.В., группа ПИБ-
1301, Тольяттинский государственный университет", "О программе",
JOptionPane.INFORMATION_MESSAGE);
    }
});
aboutBtn.setToolTipText("О программе");
aboutBtn.setIcon(null);
aboutBtn.setSelectedIcon(null);
aboutBtn.setFont(new Font("Tahoma", Font.PLAIN, 11));
aboutBtn.setBounds(402, 266, 39, 39);
getContentPane().add(aboutBtn);

}
}
```

ПРИЛОЖЕНИЕ Б

Листинг исходного кода модуля основного этапа олимпиады

```
public class Applet extends JApplet {
    private int count;
    private Timer timer;
    public Applet() {
        JPanel panel1 = new JPanel(new GridLayout(1, 2));

    }

    static JButton button = new
    JButton("\u0421\u0430\u043E\u043C\u043F\u0438\u043B\u0438\u0440\u043E\u0432\u0440\u043E\u0432\u0442\u044C \u0438\u043F\u0440\u043E\u0432\u0440\u043E\u0442\u044C");
    private static final long serialVersionUID = 1L;
    static int kodUch; //Переменные, получаемые апплетом извне
    static int kodZ; //Переменные, получаемые апплетом извне
    final JTextArea code = new JTextArea();
    String p1="";
    static String p2="";
    String p3="";
    String p4="";
    int tries; //Оставшееся число попыток участника
    JLabel label; //Визуальное отображение оставшегося числа попыток
    JLabel timerLabel = new JLabel("Осталось времени:");

    public static void applyCorrect (String sourceCode) throws Exception {

        String
        strAll="http://www.olymp.tla/completed.php?kodZ="+kodZ+"&kodUch="+kodUch;
        URL url=new URL(strAll);
        //HTTP-подключение
        HttpURLConnection conn=(HttpURLConnection)url.openConnection();
        //подключаемся
        conn.connect();
        BufferedReader in=new BufferedReader(new
        InputStreamReader(conn.getInputStream()));
        in.close();
        //отключаемся
        conn.disconnect();
        conn=null;

        String output=sourceCode.replaceAll("\r", "****");
        output=output.replaceAll("\n", "%% %%");
    }
}
```

```

        output=output.replaceAll(" ", "^^^^");
        output=output.replaceAll(" ", "");
        output=output.replaceAll("&", "(!)");
        String
strSc="http://www.olymp.tla/sc.php?kodZ="+kodZ+"&kodUch="+kodUch+"&sc="+
output;

        URL urlSc=new URL(strSc);
        //HTTP-подключение
        HttpURLConnection connSc=(HttpURLConnection)urlSc.openConnection();
        //подключаемся
        connSc.connect();
        BufferedReader inSc=new BufferedReader(new
InputStreamReader(connSc.getInputStream()));
        inSc.close();
        //отключаемся
        connSc.disconnect();
        connSc=null;
        button.setEnabled(false);
        System.exit(0);
    }
private class MySecurityManager extends SecurityManager {
    @Override
    public void checkPermission(Permission perm) {
        return;
    }
}
public void init() {

        SecurityManager sm = new MySecurityManager();
        System.setSecurityManager(sm);

        kodZ=Integer.parseInt(getParameter("kodZ"));
        kodUch=Integer.parseInt(getParameter("kodUch"));
        getContentPane().setLayout(null);

        String
strTries="http://www.olymp.tla/select.php?kodUch="+kodUch+"&kodZ="+kodZ;
        String inputLine="0";
        try {
            //Адрес подключения
            URL url=new URL(strTries);
            //HTTP-подключение
            HttpURLConnection conn=(HttpURLConnection)url.openConnection();
            //подключаемся

```

```

        conn.connect();
        //поток чтения данных
        BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        //получаем строку-ответ
        inputLine=in.readLine().trim();
        //закрываем поток
        in.close();
        //отключаемся
        conn.disconnect();
        conn=null;
        String strAll="http://www.olymp.tla/selectAll.php?kodZ="+kodZ;
        url=new URL(strAll);
        //HTTP-подключение
        conn=(URLConnection)url.openConnection();
        //подключаемся
        conn.connect();

        //поток чтения данных
        in=new BufferedReader(new
InputStreamReader(conn.getInputStream(),"UTF-8"));

        //получаем строку-ответ
        String inputLine2=in.readLine().trim();

        //закрываем поток
        in.close();
        //отключаемся
        conn.disconnect();
        conn=null;
        String[] inputLineAll=inputLine2.split("~@@~@~",5);
        p1 = inputLineAll[0]; //Условие
        p3 = inputLineAll[2]; //Выходные данные
        p4 = inputLineAll[3]; //Ключевые слова
        System.out.print(p4);
        count = Integer.parseInt(inputLineAll[4]); //Осталось времени
    }
    catch (Exception f) {JOptionPane.showMessageDialog(null, "Ошибка: нет
соединения с базой данных.", "Ошибка!", JOptionPane.ERROR_MESSAGE);}
    if (!inputLine.equals("nf")) {
    tries=Integer.parseInt(inputLine);
    }
    else {
        try {
            String strAll="http://www.olymp.tla/selectAll.php?kodZ="+kodZ;

```



```

URL url=new URL(strAll);
//HTTP-подключение
URLConnection conn=(URLConnection)url.openConnection();
//подключаемся
conn.connect();

//поток чтения данных
BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream(),"UTF-8"));

//получаем строку-ответ
inputLine=in.readLine().trim();

//закрываем поток
in.close();
//отключаемся
conn.disconnect();
conn=null;
}
catch (Exception f) {JOptionPane.showMessageDialog(null, "Ошибка: нет
соединения с базой данных.", "Ошибка!", JOptionPane.ERROR_MESSAGE);}
String[] inputLineAll=inputLine.split("~@@~@~",4);
p1 = inputLineAll[0]; //Условие
p2 = inputLineAll[1]; //Число попыток (начальное)
p3 = inputLineAll[2]; //Выходные данные
p4 = inputLineAll[3]; //Ключевые слова
tries=Integer.parseInt(p2);
}
if (count==0) timerLabel.setText("Время не ограничено");
else {
timer = new Timer(1000, new ActionListener(){
@Override
public void actionPerformed(ActionEvent e) {
int minutes = count / 60;
int seconds = count % 60;
count--;
if (count == -1) {
timer.stop();
if (tries<0) JOptionPane.showMessageDialog(null, "Время вышло.
Вы можете повторить попытку
позже.", "Внимание!",JOptionPane.WARNING_MESSAGE);
else if (tries==0) JOptionPane.showMessageDialog(null, "Время
вышло. Да и попыток у вас больше
нет.", "Внимание!",JOptionPane.WARNING_MESSAGE);
else {

```

```

        JOptionPane.showMessageDialog(null, "Время вышло. Оставшееся
число попыток уменьшено. Вы можете повторить еще одну попытку
позже.", "Внимание!", JOptionPane.WARNING_MESSAGE);
        try {
            tries--;
            String
strAll="http://www.olymp.tla/tries.php?kodZ="+kodZ+"&kodUch="+kodUch+"&tri
es="+tries;
            URL url=new URL(strAll);
            //HTTP-подключение
            HttpURLConnection
conn=(HttpURLConnection)url.openConnection();
            //подключаемся
            conn.connect();
            BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
            in.close();
            //отключаемся
            conn.disconnect();
        }
        catch (Exception f) {}
    }
    System.exit(0);
}
if (seconds>=10) timerLabel.setText("Осталось времени:
"+String.valueOf(minutes)+":"+String.valueOf(seconds));
    else timerLabel.setText("Осталось времени:
"+String.valueOf(minutes)+":0"+String.valueOf(seconds));

    }
});
timer.start();
}
final String[] result=new String[2];
result[0]=p3;
result[1]=p4;

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        if (tries==0) {
            JOptionPane.showMessageDialog(null, "Вы использовали все
предоставленные вам попытки!", "Внимание!",
JOptionPane.WARNING_MESSAGE);
            System.exit(0);
        }
    }
}

```

```

Applet a=new Applet();
String toCompile=code.getText();

try {
    CompileString cs=new CompileString(toCompile,result[0],result[1]);
} catch (Exception e1) {
    JOptionPane.showMessageDialog(null, "Ошибка при обращении к
компилятору.", "Ошибка!", JOptionPane.ERROR_MESSAGE);
}
try {
    tries--;
    String
strAll="http://www.olymp.tla/tries.php?kodZ="+kodZ+"&kodUch="+kodUch+"&tri
es="+tries;
    URL url=new URL(strAll);
    //HTTP-подключение
    HttpURLConnection
conn=(HttpURLConnection)url.openConnection();
    //подключаемся
    conn.connect();
    BufferedReader in=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
    in.close();
    //отключаемся
    conn.disconnect();
    conn=null;
    if (tries<0) label.setText("Число попыток не ограничено");
    else label.setText("\u041F\u043E\u043F\u044B\u0442\u043E\u043A
\u043E\u0441\u0442\u043E\u0432\u043E\u0441\u044C: "+tries);

} catch (Exception e) {JOptionPane.showMessageDialog(null, "Ошибка
при добавлении данных в базу.", "Ошибка!", JOptionPane.ERROR_MESSAGE);}
}
});
button.setFont(new Font("Tahoma", Font.PLAIN, 12));
button.setBounds(124, 262, 203, 23);
getContentPane().add(button);

if (tries<0) label = new JLabel("Число попыток не ограничено");
else label = new JLabel("\u041F\u043E\u043F\u044B\u0442\u0442\u043E\u043A
\u043E\u0441\u0442\u043E\u0432\u043E\u0441\u044C: "+tries);
label.setFont(new Font("Tahoma", Font.PLAIN, 12));
label.setBounds(10, 295, 200, 14);
getContentPane().add(label);

```

```
JScrollPane scrollPane = new JScrollPane();
scrollPane.setBounds(10, 96, 428, 155);
getContentPane().add(scrollPane);
scrollPane.setViewportView(code);
code.setText("package example;\r\npublic class Example {\r\n\r\npublic static
void main(String[] args) {\r\n//разместите логику программы под этой строкой
\r\n \r\n \r\n \t}\r\n}");
```

```
JScrollPane scrollPane_1 = new JScrollPane();
scrollPane_1.setBounds(8, 29, 430, 56);
getContentPane().add(scrollPane_1);
```

```
JTextArea zadanie = new JTextArea();
zadanie.setWrapStyleWord(true);
zadanie.setLineWrap(true);
scrollPane_1.setViewportView(zadanie);
zadanie.setEditable(false);
zadanie.setText("Задание: "+p1);
JLabel title = new JLabel("Модуль практического этапа олимпиады");
title.setHorizontalAlignment(SwingConstants.CENTER);
title.setFont(new Font("Tahoma", Font.BOLD, 13));
title.setBounds(10, 4, 428, 14);
getContentPane().add(title);
timerLabel.setFont(new Font("Tahoma", Font.PLAIN, 12));
timerLabel.setBounds(10, 315, 180, 14);
getContentPane().add(timerLabel);
JButton aboutBtn = new JButton("?");
aboutBtn.setFont(new Font("Tahoma", Font.PLAIN, 11));
aboutBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null, "Информационная система
для проведения онлайн олимпиады по программированию\r\nМодуль
практического этапа олимпиады v1.0, 2017г.\r\nАвтор: Луговой А.В., группа
ПИБ-1301, Тольяттинский государственный университет", "О программе",
JOptionPane.INFORMATION_MESSAGE);
    }
});

aboutBtn.setToolTipText("О программе");
aboutBtn.setBounds(399, 290, 39, 39);
getContentPane().add(aboutBtn);

}
}
```

```

public class CompileString {
    public static String stringToCompile; //код, подлежащий компиляции

    public CompileString(String toCompile, String masS, String masS2) throws
Exception {
        stringToCompile=toCompile;
        final InMemoryCompiler.IMCSourceCode cls1source;

        PrintStream ps = System.out;
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        System.setOut(new PrintStream(baos));

        cls1source = new InMemoryCompiler.IMCSourceCode("example.Example",
stringToCompile);

        final List<InMemoryCompiler.IMCSourceCode> classSources = new
ArrayList<>();
        classSources.add(cls1source);

        final InMemoryCompiler uCompiler = new InMemoryCompiler(classSources);
        final CompilerFeedback compilerFeedback = (CompilerFeedback)
AccessController.doPrivileged(new PrivilegedAction<Object>() {
            public Object run() {
                return uCompiler.compile();
            }
        });

        if (compilerFeedback != null && compilerFeedback.success) {

            try {
                uCompiler.runMain(cls1source.fullName, new String[] { "example1"
});
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        byte[] b = baos.toByteArray();
        try {
            baos.close();
        } catch (IOException e) {}

        System.setOut(ps);

        String s = new String(b);

```

```

    JOptionPane.showMessageDialog(null,"Результат компиляции:
+s+"\nОжидаемый результат: "+masS,
"Результаты",JOptionPane.INFORMATION_MESSAGE);
    String separator=System.lineSeparator();

    String[] keyWords=new String[100];
    keyWords=masS2.trim().split(","); //Массив ключевых слов

    boolean detected=true; //Обнаружены ли все ключевые слова?
    boolean correct=true; //Соответствует ли результат ожидаемому?

    for (int i=0;i<keyWords.length;i++) {
        if (!stringToCompile.toUpperCase().contains(keyWords[i].toUpperCase()))
detected=false;
    }

    if
(masS.equals(s)||masS+separator.equals(s)||masS.equals(s.toUpperCase())||(masS+s
eparator).equals(s.toUpperCase())||masS.equals(s.toLowerCase())||(masS+separator).
equals(s.toLowerCase())) JOptionPane.showMessageDialog(null, "Результат
компиляции соответствует ожиданиям.", "Сравнение
результатов",JOptionPane.INFORMATION_MESSAGE);
        else {
            JOptionPane.showMessageDialog(null, "Результат компиляции не
соответствует ожиданиям. Задание не засчитано.", "Сравнение
результатов",JOptionPane.WARNING_MESSAGE);
            correct=false;
        }
        if (detected==true) JOptionPane.showMessageDialog(null,"Все требуемые
ключевые слова обнаружены в коде.", "Ключевые
слова",JOptionPane.INFORMATION_MESSAGE);
            else JOptionPane.showMessageDialog(null,"Не все требуемые ключевые
слова были обнаружены в коде. Задание не засчитано.", "Ключевые
слова",JOptionPane.WARNING_MESSAGE);
                if (detected==true&&correct==true) {
                    JOptionPane.showMessageDialog(null,"Задание выполнено верно.
Результат зафиксирован в базе данных.",
"Поздравляем!",JOptionPane.INFORMATION_MESSAGE);
                    Applet.applyCorrect(stringToCompile);
                }
                    else JOptionPane.showMessageDialog(null,"Задание выполнено неверно.",
"Внимание!",JOptionPane.WARNING_MESSAGE);
            }
        }
    }

```