

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: **Разработка web-портала для проведения online олимпиад по программированию**

Студент	Ф.Ф. Микаилов	
Руководитель	Е.А. Ерофеева	

Допустить к защите

Заведующий кафедрой к.тех.н, доцент, А.В. Очеповский

« _____ » 20 ____ г.

Тольятти 2016

ОГЛАВЛЕНИЕ

Введение.....	3
Глава 1 Анализ предметной области АИС	5
1.1 Характеристика предприятия и его деятельности.....	5
1.2 Выбор технологии анализа проектирования.....	6
1.3 Анализ процессов проводимой олимпиады, нуждающихся в автоматизации.....	7
Глава 2 Проектирование web-портала для проведения online олимпиад по программированию	13
2.1 Определение требований к системе	13
2.2 Обоснование и постановка задачи на разработку новой АИС.....	16
2.3 Логическое моделирование web-портала для проведения online олимпиад по программированию	20
2.3.1 Разработка диаграммы вариантов использования web-портала	20
2.3.2 Диаграмма классов web-портала	26
2.3.3 Разработка логической модели данных web-портала	27
2.4 Физическое моделирование автоматизированной системы для проведения online олимпиад по программированию	29
2.4.1 Выбор архитектуры web-портала.....	29
2.4.2 Выбор сервера базы данных для реализации АИС для проведения online олимпиад по программированию.	31
2.4.3 Выбор языка программирования для реализации web-портала.....	32
2.4.4 Выбор web-сервера для web-портала.....	34
2.4.6 Диаграмма размещения компонентов web-портала	36
Глава 3 Практическая реализация автоматизированной системы для проведения online олимпиад по программированию	38
3.1 Реализация функциональных требований	38
Заключение	47
Список используемой литературы	48

Введение

Особую роль компьютерные технологии играют в образовательных процессах, связанных с решением задач. Здесь они могут стать не только средством обучения, но и важным ресурсом в стимулировании интереса к точным, естественным наукам и информатике.

В настоящее время проведение олимпиад по информационным технологиям очень востребовано и олимпиады по программированию наиболее популярны среди них.

В России одним из наиболее популярных центров проведения олимпиад по программированию является Таганрогский государственный университет, имеющий собственную автоматизированную систему проверки выполненных задач. Эта система открыта для новых участников и характеризуется регулярными виртуальными соревнованиями.

В настоящее время уровень программирования в России сильно растет, поэтому программирование начинают изучать уже «со школьной скамьи», ежегодно проводятся школьные, городские и другие олимпиады по информационным технологиям и программированию.

Цель: разработка web-портала для проведения online олимпиад по программированию.

Объект: процесс проведения online олимпиад по программированию.

Предмет исследования: web-портал с системой автоматизированной проверки решений задач по программированию.

Для достижения поставленной цели в работе необходимо решить следующие задачи:

- провести обследование и выполнить идентификацию предметной области автоматизации;
- обосновать выбор методологии и технологии концептуального моделирования АИС;
- исследовать существующие IT аналоги;

- сформулировать требования к АИС
- спроектировать базу данных;
- составить техническое задание.

В ходе работы должна быть спроектирована и реализована автоматизированная информационная система для проведения online олимпиад по программированию.

Методы исследования:

- структурный подход к анализу и проектированию ИС;
- методология объектно-ориентированного анализа и проектирования ИС;
- методология онтологического подхода к моделированию ИС;
- методология объектно-структурного подхода к моделированию ИС;
- CASE-технологии структурного и объектно-ориентированного анализа и проектирования.

В качестве метода исследования выбрано моделирование по CASE-технологии с использованием SADT-методологии моделирования.

Практическая значимость работы состоит в том, что проведенное концептуальное моделирование является основой для разработки программного обеспечения АИС.

Глава 1 Анализ предметной области АИС

1.1 Характеристика предприятия и его деятельности

Бакалаврская работа выполнена по результатам преддипломной практики в Тольяттинском государственном университете г. Тольятти. Тольяттинский государственный университет создан распоряжением Правительства Российской Федерации от 29 мая 2001 г. № 873-р на базе Тольяттинского политехнического института и Тольяттинского филиала Самарского государственного педагогического университета и является их правопреемником.

На рисунке 1.1 изображена организационная структура Тольяттинского государственного университета (ТГУ).

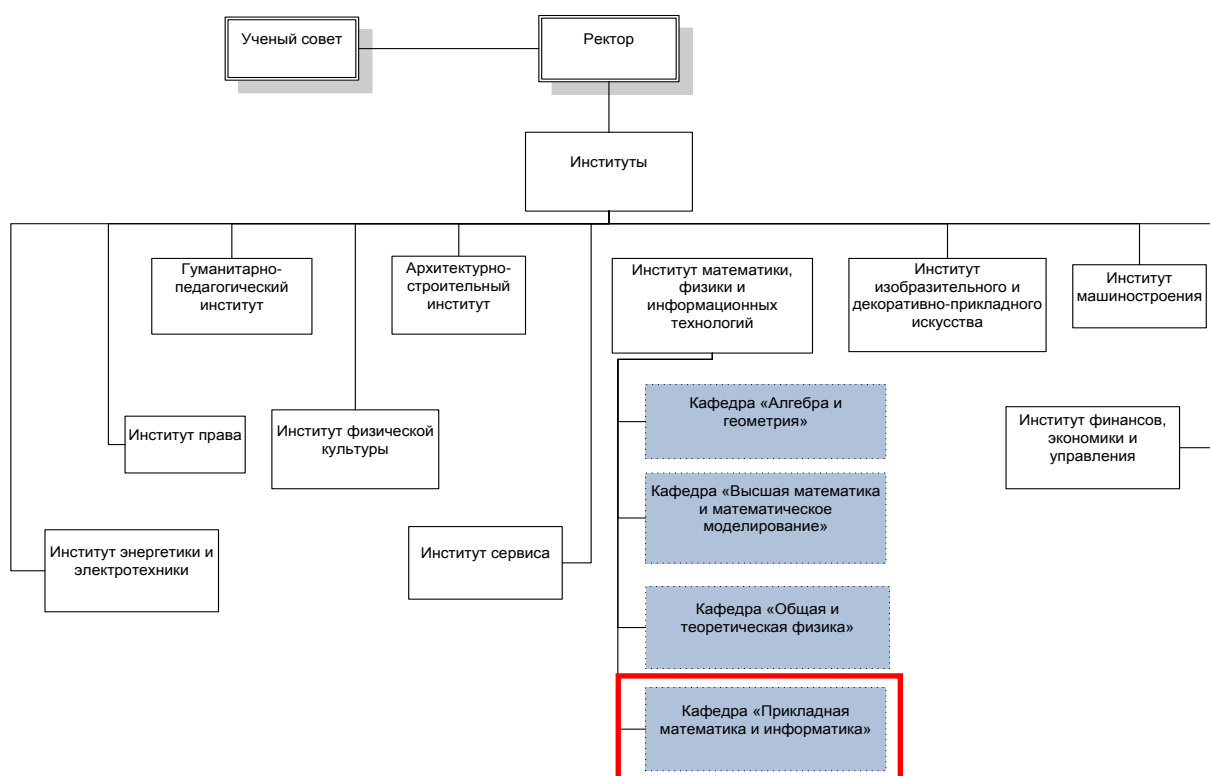


Рисунок 1.1- Организационная структура университета ТГУ

Немного описания о моем институте: институт математики, физики и информационных технологий (ИМФИТ) готовит бакалавров и магистров по нескольким направлениям подготовки, которые можно объединить в две основные группы:

1) математики-прикладники и программисты, специализирующиеся на создании математических моделей, информационных систем, а также разработке информационных технологий в различных областях знания;

2) преподаватели вузов и учителя по таким направлениям знаний, как математика, физика, информатика. Все эти направления имеют стратегическое значение для развития города Тольятти и вне всяких сомнений являются крайне востребованными сегодня.

На рисунке 1.1 представлены кафедры этого института.

У каждой кафедры имеется свой учебный персонал, каждый член которого носит степень и звание занимаемой должности. С кафедры «Прикладная математика и информатика» преподаватели поставили перед собой задачу о разработке web-портала для проведения online олимпиад по программированию, где я должен был организовать саму олимпиаду и процесс ее прохождения.

1.2 Выбор технологии анализа проектирования

Проектирование будущей информационной системы начинается с выбора методологии проектирования.

Рассмотрим основную информацию по нескольким наиболее известным и распространённым методологиям:

- IDEF0;
- RUP;
- ICONIX.

Методология IDEF0 широко используется благодаря простой и понятной для понимания графической нотации, применение которой для построения модели очень удобное.

Главное место в методологии отводится диаграммам. На диаграммах отображают функции системы посредством геометрических прямоугольников, а также имеющиеся связи между функциями и внешней средой. Связи отображаются с помощью стрелок.

Графический язык IDEF0 удивительно прост и гармоничен. Основу методологии IDEF0 составляет графический язык описания процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Rational Unified Process (RUP) представляет собой целостную методологию организации процессов разработки программного обеспечения. Процессы, организуемые на основе RUP, варьируются от наиболее простых - предназначенных для небольших проектов с коротким циклом разработки - до более сложных процессов, покрывающих более широкий спектр потребностей больших, возможно даже распределенных, групп разработчиков. RUP успешно применяется в проектах любых типов и объемов.

Методология ICONIX представляет собой нечто среднее между очень громоздким рациональным унифицированным процессом (Rational Unified Process - RUP) и весьма компактной методологией программирования extreme (XP). Процесс ICONIX, как и RUP, основан на прецедентах, но не характеризуется множеством его недостатков. В этом процессе тоже применяется язык моделирования UML (Unified Modeling Language), однако основное внимание уделяется анализу требований.

Итак, для проектирования информационной системы принято решение использовать методологию IDEF0.

1.3 Анализ процессов проводимой олимпиады, нуждающихся в автоматизации

Для обоснования необходимости использования web-портала следует выделить и проанализировать процессы, нуждающиеся в автоматизации. Наиболее наглядное представление об этих процессах дает его IDEF0-диаграмма.

Концептуально-функциональная модель деятельности «Проведение online олимпиады по программированию» IDEF0 (AS-IS) представлена на рисунке 1.2.



Рисунок 1.2 – Концептуально-функциональная модель деятельности «Проведение online олимпиады по программированию» IDEF0 (AS-IS)

Входными потоками данных является:

- данные об участнике;
- данные об олимпиаде;
- заявки на участие.

Выходными потоками данных являются:

- результаты олимпиады.

Потоками управления являются:

- положение о порядке проведения олимпиады;

- критерии оценивания.

Потоками механизмов системы являются:

- организатор олимпиады;
- участник олимпиады.

Следующая диаграмма представляет собой декомпозицию главного процесса, диаграмма представлена на рисунке 1.3. Все блоки тесно связаны между собой, каждый последующий блок начинается после завершения предыдущего и является его логическим продолжением.

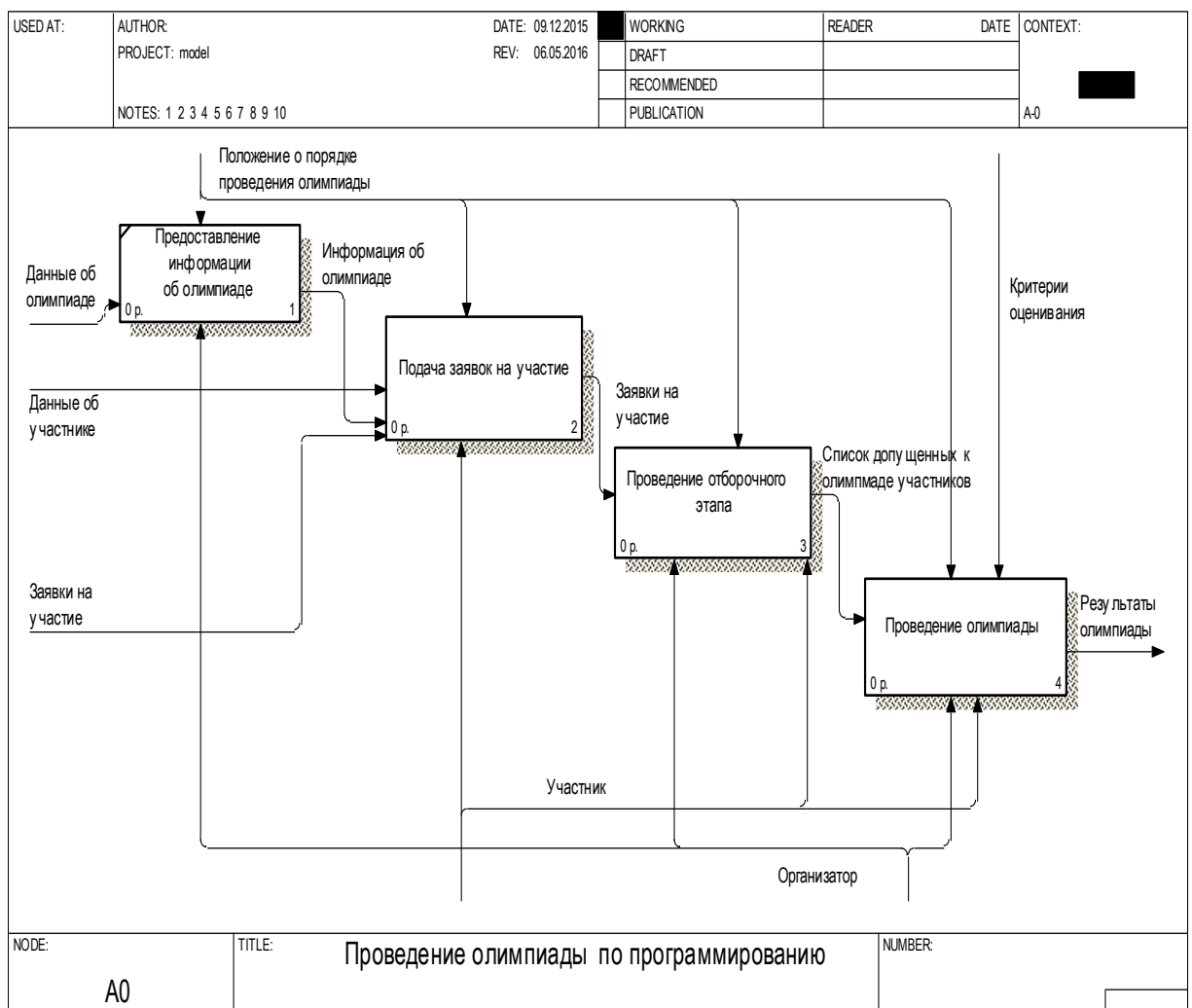


Рисунок 1.3 - Функционально-ориентированная модель декомпозиции контекстной диаграммы «Проведение online олимпиады по программированию» IDEF0 (AS-IS)

На рисунке видно, что происходят четыре основных процесса:

- предоставление информации об олимпиаде;
- подача заявки на участие;
- проведение отборочного этапа;
- проведение олимпиады.

Процесс «Подача заявок на участие», декомпозиция которого показана на рисунке 1.4, связан с организационной частью олимпиады, где для участия и прохождения олимпиады в электронной форме Участник должен зарегистрироваться, заполнить специальные поля формы на сайте и далее подать заявку на участие.

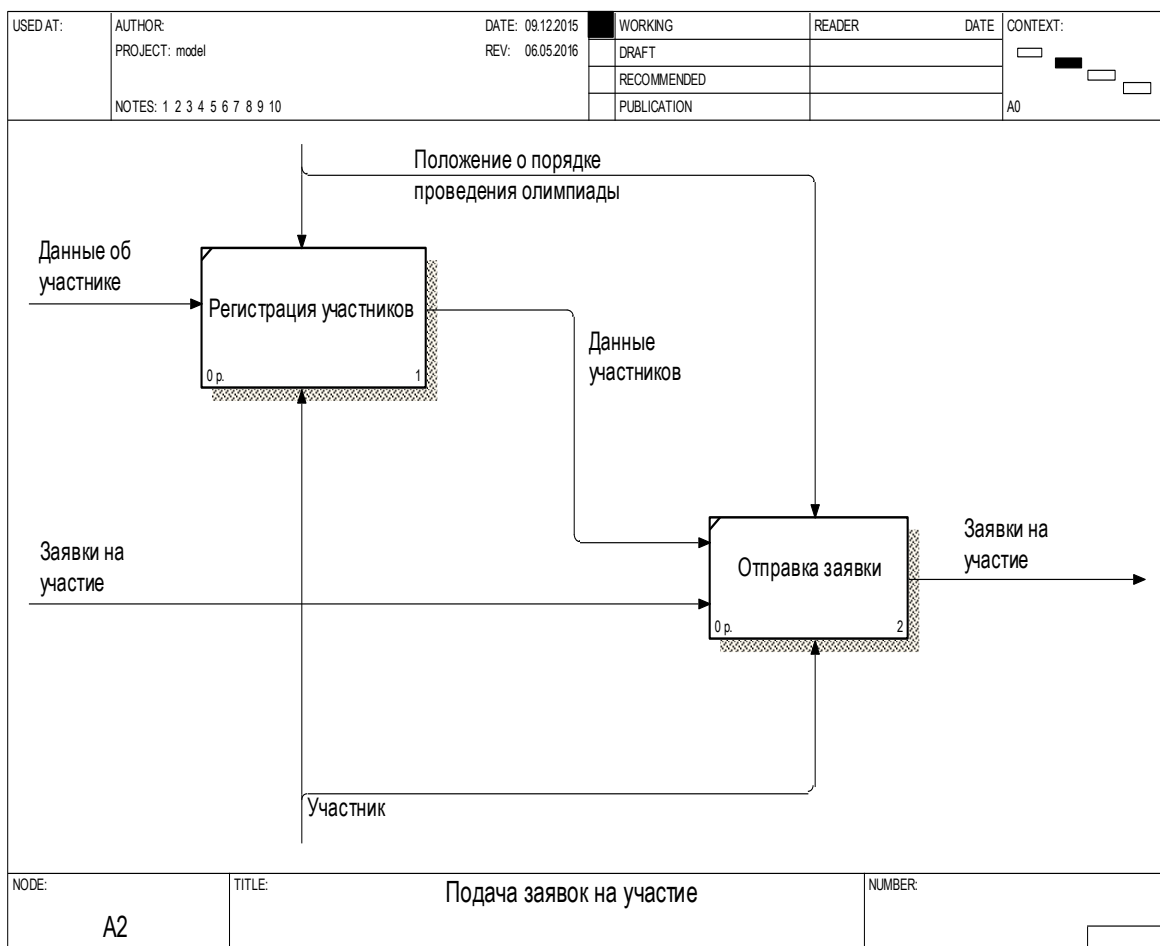


Рисунок 1.4 - Функционально-ориентированная модель декомпозиции процесса «Подача заявок на участие» IDEF0 (AS-IS)

Процесс «Проведение отборочного этапа» (Рисунок 1.5) подразумевает следующее: участник олимпиады может выполнять задания отборочного этапа

с любого компьютера, имеющего доступ в сеть Интернет. По итогам отборочного этапа олимпиады определяются победители данного тура, которые смогут участвовать в заключительном этапе олимпиады.

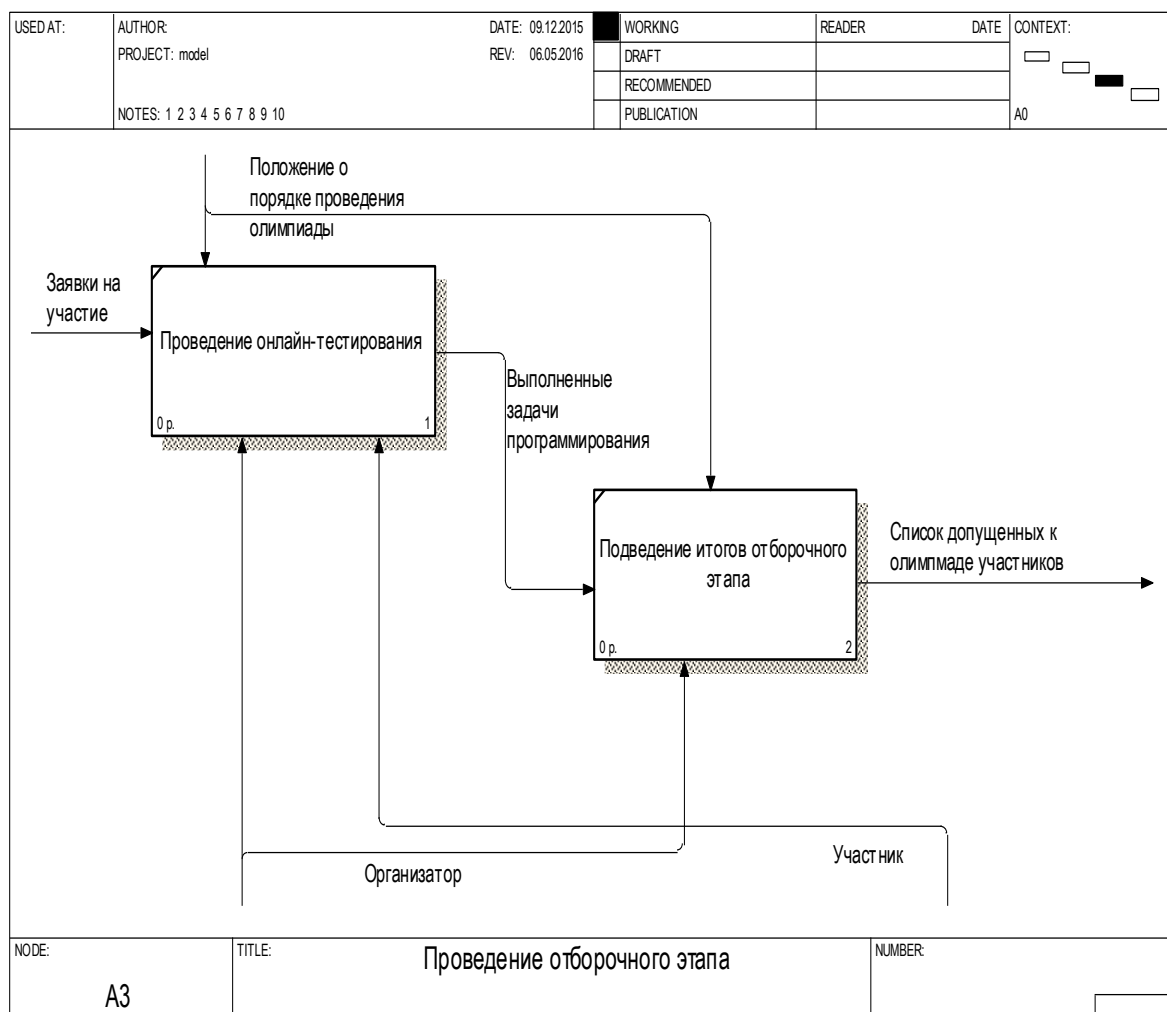


Рисунок 1.5 - Функционально-ориентированная модель декомпозиции процесса «Проведение отборочного этапа» IDEF0 (AS-IS)

Процесс «Проведение олимпиады» (Рисунок 1.6) подразумевает следующее: участники, успешно прошедшие отборочный этап, допускаются к прохождению олимпиады.

На этом этапе данные участники, которые успешно прошли отборочный этап, выполняют задания самой олимпиады.

После всех решенных заданий олимпиады, каждый участник может ознакомиться со своими результатами и увидеть список правильно решенных заданий.

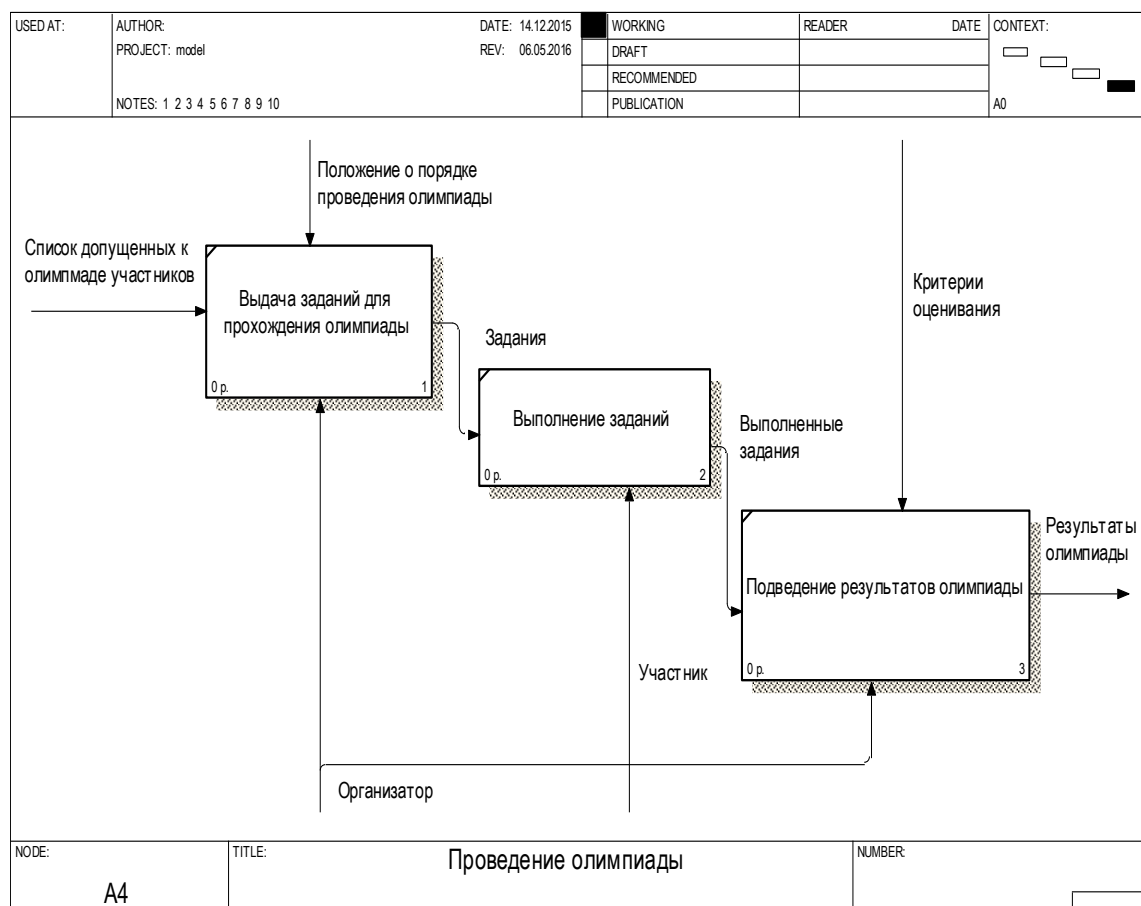


Рисунок 1.6 - Функционально-ориентированная модель декомпозиции процесса «Проведение олимпиады» IDEF0 (AS-IS)

Исходя из описания процессов организации олимпиады, можно сделать вывод, что практически на каждом этапе есть работа с различного вида документацией (результаты олимпиады, подача заявок), которую можно автоматизировать.

Вывод по первой главе

В результате проведения анализа предметной области было произведено сравнение методологий и технологии концептуального моделирования АИС.

Были выявлены процессы, нуждающиеся в автоматизации, такие как подача заявок на участие, формирование задач, проверка задач.

Глава 2 Проектирование web-портала для проведения online олимпиад по программированию

2.1 Определение требований к системе

Определим основные требования, предъявляемые к будущей системе. Требования к системе описаны по классификации FURPS+(Таблица 2.1).

Классификация требований к системе FURPS+ была разработана Робертом Грэйди (Robert Grady) из Hewlett-Packard. Сокращение FURPS расшифровывается следующим образом:

1. Functionality, функциональность.
2. Usability, удобство использования.
3. Reliability, надежность.
4. Performance, производительность.
5. Supportability, поддерживаемость.
6. Также необходимо помнить о возможных ограничениях, таких как:
 - ограничения проектирования, design;
 - ограничения разработки, implementation;
 - ограничения на интерфейсы, interface;
 - физические ограничения, physical.

Требование можно определить как «подробное описание того, что должно быть реализовано». Существует два основных типа требований:

1. Функциональные требования – означает, какое поведение должна предлагать система. Функциональное требование – это формулировка того, что должна делать система, это описание назначения системы.

2. Нефункциональные требования – особое свойство или ограничение, накладываемое на систему.

При анализе всех контекстов были выявлены следующие требования к разрабатываемой системе (Таблица 2.1).

Таблица 2.1 - Требования к системе

ID	Требование	Статус	Полезность	Риск	Стабильность
Функциональные требования					
1	Предоставлять информацию о турнире на web-портале	Одобренные	Критичное	Средний	Низкая
2	Регистрировать участников соревнования	Одобренные	Критичное	Средний	Низкая
3	Принимать заявки участников соревнования	Одобренные	Критичное	Средний	Низкая
4	Отправлять результаты соревнования участникам на web-портале	Предложенные	Критичное	Средний	Средняя
5	Предоставлять возможность ознакомиться с работами участников соревнования на web-портале	Предложенные	Критичное	Средний	Средняя
Требования к удобству использования					
8	Названия разделов и страниц сайта понятные и четко соответствуют их содержанию.	Одобренные	Критичное	Низкий	Низкая
9	Навигация должна отражать	Одобренные	Критичное	Низкий	Низкая

Продолжение таблицы 2.1

ID	Требование	Статус	Полезность	Риск	Стабильность
	информационную структуру сайта и быть простой и понятной в использовании				
10	Информация на сайте должна быть четко структурирована	Одобренные	Критичное	Низкий	Низкая
11	Разрешение экрана 1366x768	Одобренные	Критичное	Низкий	Низкая
12	Размер основного шрифта должен быть не меньше 12 px	Одобренные	Критичное	Низкий	Низкая
13	Дизайн сайта должен быть выполнен в единой цветовой гамме и едином стиле.	Одобренные	Критичное	Низкий	Низкая
Требования к надежности					
14	Среднее время восстановления работоспособного состояния не более 1-2 суток	Предложенные	Критичное	Низкий	Низкая
15	организация бесперебойного питания технических средств	Предложенные	Критичное	Средний	Средняя

Продолжение таблицы 2.1

ID	Требование	Статус	Полезность	Риск	Стабильность
Требования к производительности					
16	Загрузка страницы сайта не должна быть более 60 секунд	Предложенные	Критичное	Средний	Средняя
Требования к поддержке					
16	Время устранения возникших проблем с серверной частью 30-60 мин	Предложенные	Критичное	Средний	Средняя
Ограничения реализации					
17	Реализация на языках php, javascript, html и css	Одобренные	Критичное	Средний	Низкая

2.2 Обоснование и постановка задачи на разработку новой АИС

Была построена контекстная диаграмма "КАК ДОЛЖНО БЫТЬ" (ТО ВЕ) представленная на рисунке 2.1.

Модель ТО-ВЕ описывает возможное будущее состояние предметной области, в которое она перейдёт в результате оптимизации существующей системы и внедрения новых технологий.

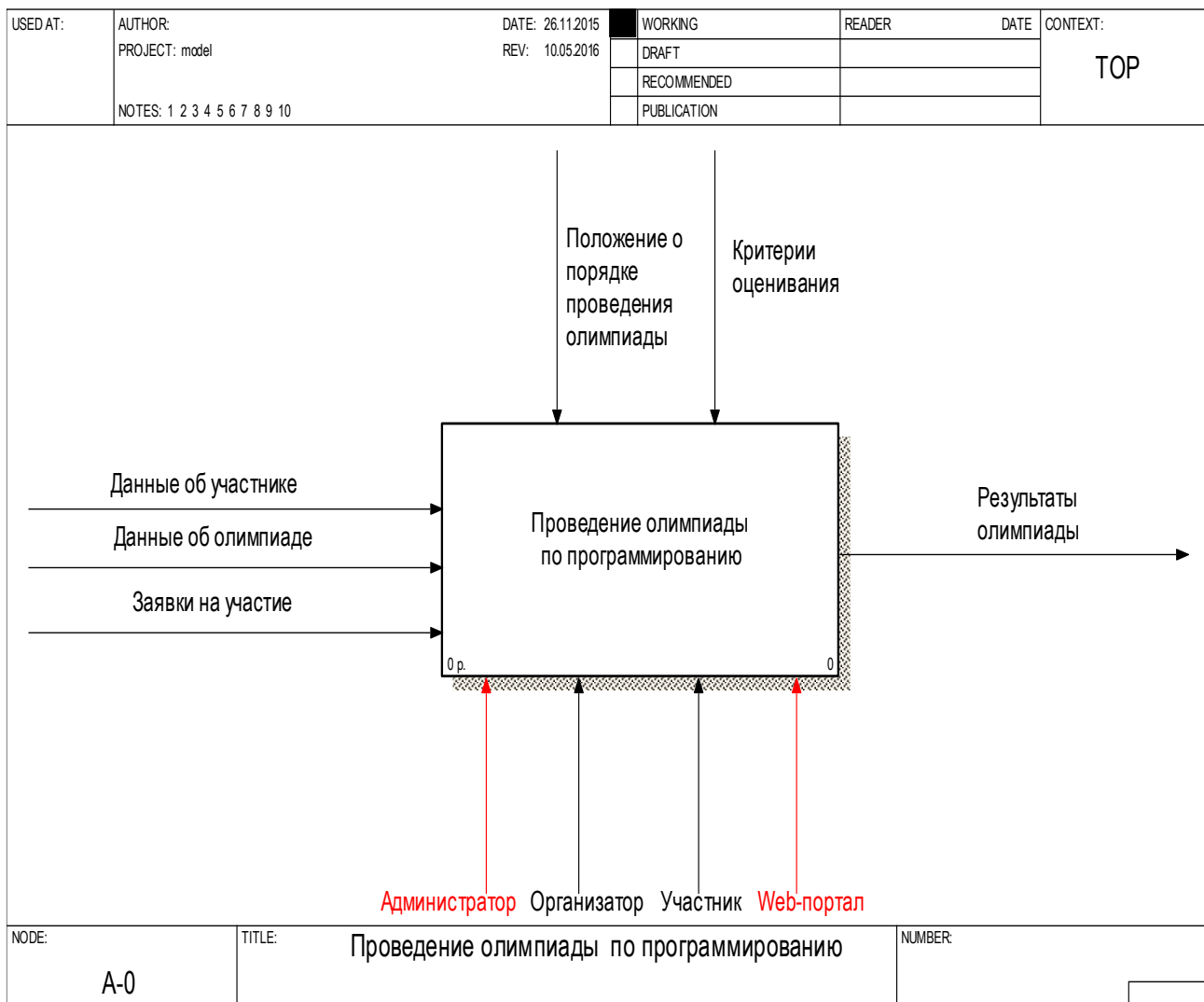


Рисунок 2.1 - «Проведение online олимпиады по программированию IDEF0 (TO BE)»

Если сравнивать контекстную диаграмму "КАК ДОЛЖНО БЫТЬ", представленную на рисунке 2.1 с диаграммой "КАК ЕСТЬ", изображенной на рисунке 1.2, можно заметить, что механизмами стали четыре потока: администратор, участник, организатор и web-портал.

Декомпозиция контекстной диаграммы представлена на рисунке 2.2. Теперь основная информация об олимпиаде размещается на web-портале.

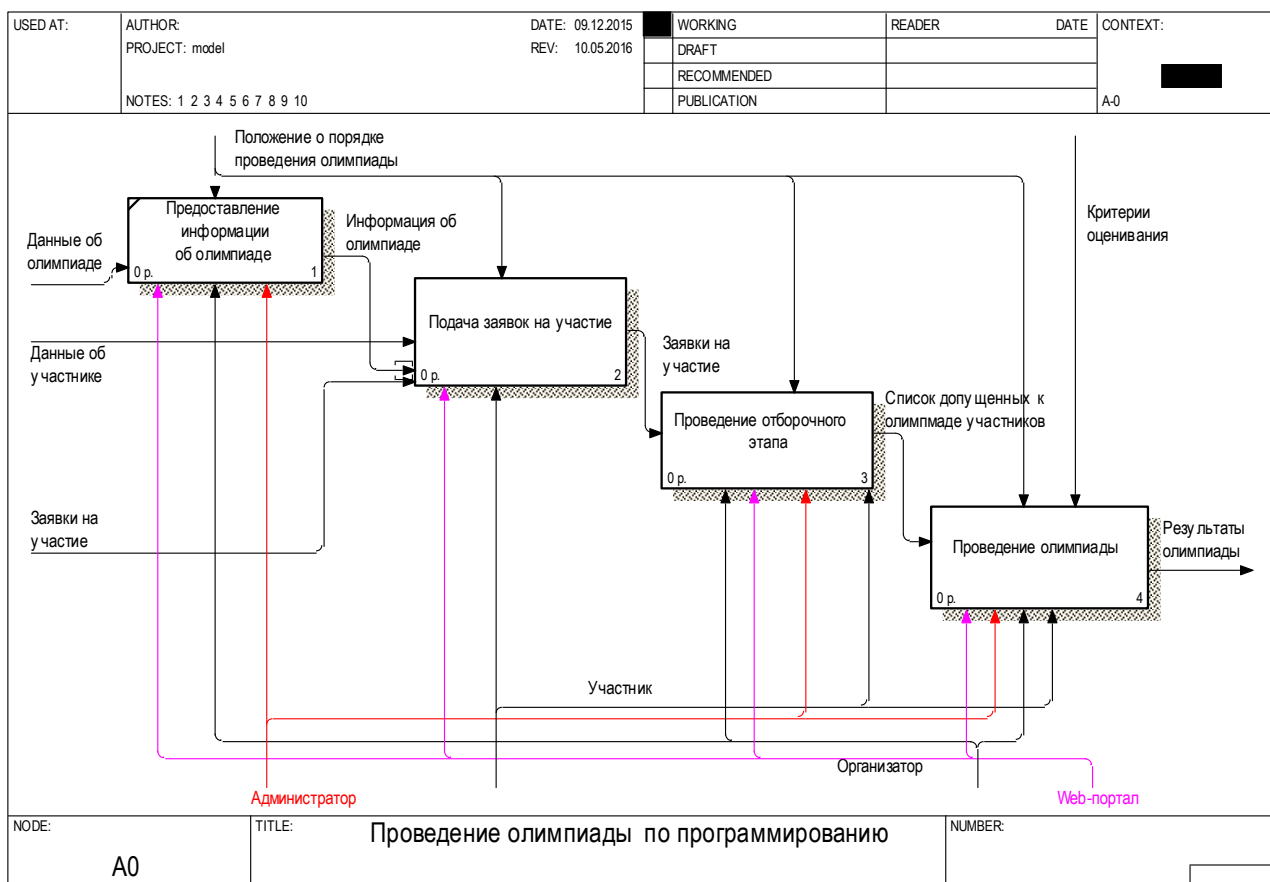


Рисунок 2.2 – «Проведение online олимпиады по программированию»

На данном этапе были продемонстрирована декомпозиция главного процесса «Проведение online олимпиады по программированию»

Автоматизированная информационная система должна обеспечивать следующие функции:

- добавление, редактирование и удаление соревнований и задач для проведения online олимпиады по программированию;
- подача заявок на участие;
- выполнение программы на основе тестовых входных данных программного кода;
- подведение итогов в результате сравнения программного кода с правильным вариантом реализации.

На начальном этапе пользователь проходит регистрацию, после чего подает заявку на участие в online олимпиаде по программированию.

После этого сформированная заявка поступает для подтверждения администратору (организатору). Он формирует задачу, и уже готовые задачи поступают данным участникам. На рисунке 2.3 изображена модель декомпозиции контекстной диаграммы «Проведение online олимпиады по программированию» DFD (TO-BE).

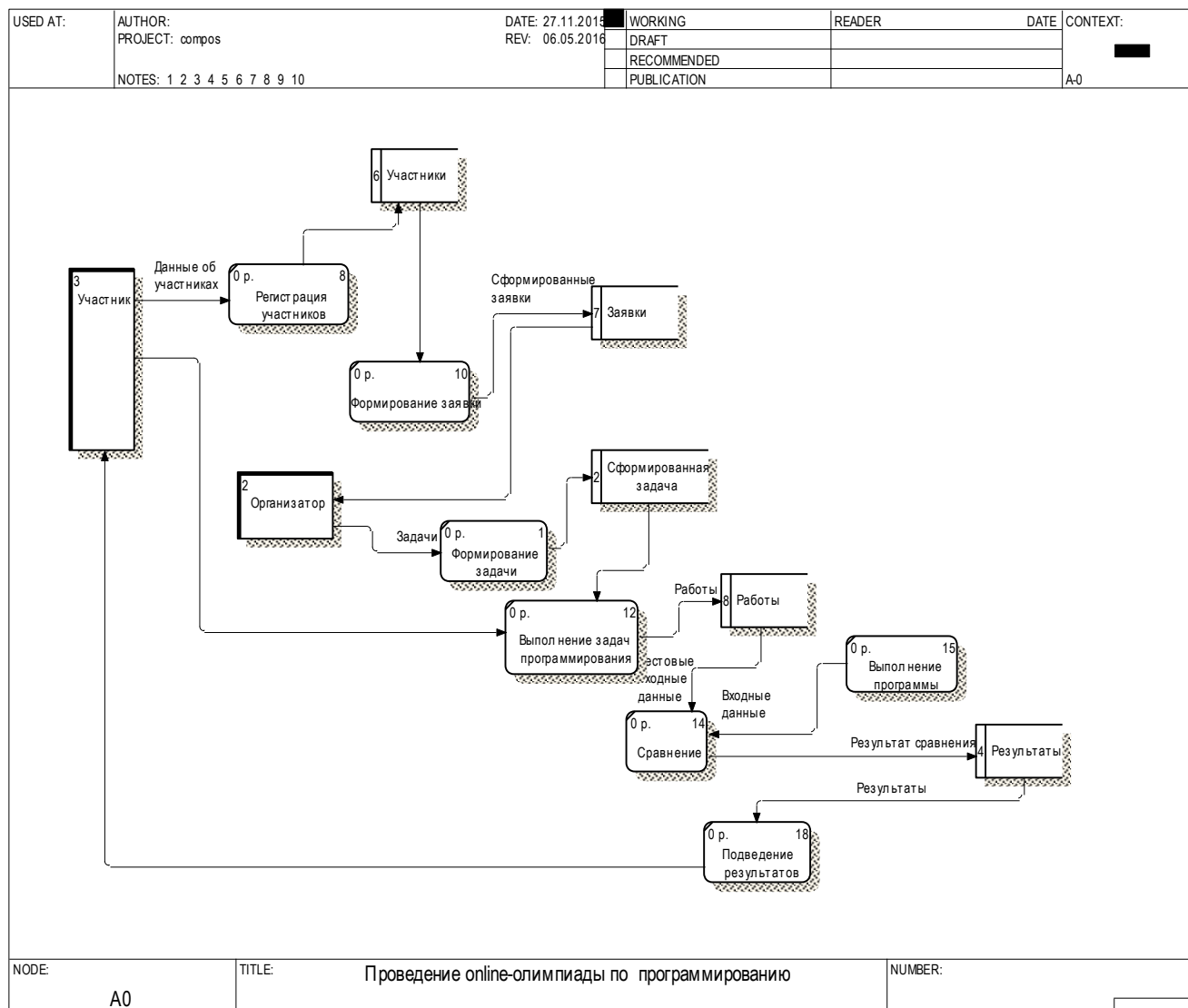


Рисунок 2.3 - Функционально-ориентированная модель декомпозиции контекстной диаграммы «Проведение online олимпиады по программированию» DFD (TO-BE)

Диаграммы потоков данных DFD иллюстрируют иерархию функциональных процессов, связанных потоками данных. Это представление отображает, как каждый процесс преобразует входные данные в выходные, а

также обнаруживает отношения между этими процессами. Модель описывает процесс преобразования информации, начиная от ее ввода в систему и до выдачи в итоге пользователю. Внешние сущности в системе порождают информационные потоки данных, потоки переносят информацию к подсистемам, процессам или потребителям информации — внешним сущностям.

Диаграмма потоков данных DFD показывает организацию добавления задач, реализации их программного кода и их сравнений в автоматизированной системе. На ней отображены две сущности: Организатор и Участник. Участник загружает свой программный код решения задачи в автоматизированную систему, потом эта информация обрабатывается и добавляется в информационную базу. Информацию о задачах и контрольных решениях заносит Организатор олимпиады, на основании чего производится сопоставление решений Участника с контрольными образцами.

2.3 Логическое моделирование web-портала для проведения online олимпиад по программированию

Результатом логического моделирования является объектная модель и логическая модель данных web-портала.

На стадии логического моделирования web-портала используется методология объектно-ориентированного проектирования и анализа, основанная на языке визуального моделирования UML.

2.3.1 Разработка диаграммы вариантов использования web-портала

Рассмотрим проектируемую систему организации олимпиады с точки зрения диаграммы вариантов использования. С помощью диаграммы вариантов использования, на которой показана совокупность прецедентов и актеров, а также отношения между ними, можно выделить внешние системы, контактирующие с системой, основные процессы и их взаимосвязь. Диаграммы вариантов использования дают возможность выделить функциональную структуру системы, не вдаваясь в детали ее реализации. Кроме того,

производится предварительное выделение объектов системы и их классификация. На основании построенной модели составляется план разработки системы.

Для данной предметной области выделим следующих актеров:

- пользователь;
- администратор системы;
- организатор олимпиады.

Рассмотрим, какие возможности должна предоставлять разрабатываемая система:

- пользователь использует систему для регистрации и участия в олимпиаде;
- администратор системы реализует контроль функционирования системы, предоставляет информацию об олимпиаде, редактирует данные;
- организатор олимпиады занимается проведением олимпиады, а также выявлением ее победителей.

Автоматизированная информационная система (АИС) реализует хранение данных, доступ к ним по компьютерной сети с возможностью автоматизации поиска. На основании вышеперечисленных пунктов, можно выделить следующие прецеденты (таблица 2.2), которые должны быть реализованы в новой системе.

Таблица 2.2 - Краткое описание прецедентов

Прецеденты	Краткое описание
1	2
Регистрация	Регистрация пользователя на сайте
Добавление олимпиады	Добавление олимпиады с описанием: теоретическим материалом и практическими заданиями
Решение заданий	Решение заданий олимпиады
Просмотр результата	Позволяет пользователю получить сведения о результатах выполненных заданий

Продолжение таблицы 2.2

Прецеденты	Краткое описание
Предоставление информации	Предоставление необходимой информации об олимпиаде
Редактирование данных	Редактирование данных олимпиады
Подведение итогов	Подведение итогов олимпиады

Разработанная диаграмма вариантов использования для основных прецедентов проектируемой системы представлена на рисунке 2.4.



Рисунок 2.4 - Диаграмма вариантов использования

Рассмотрим спецификации для основных прецедентов в виде табличного представления. Далее в таблицах 2.3 – 2.8 представлены спецификации основных прецедентов.

Таблица 2.3 - Описание прецедента "Регистрация"

Прецедент: Регистрация
ID: 1
Краткое описание: Регистрация участников олимпиады
Главные актеры: Пользователь
Второстепенные актеры: Нет

Продолжение таблицы 2.3

Предусловия:
1. Прецедент начинается по инициативе пользователя
Основной поток:
1. Пользователь заходит на web-портал.
2. Пользователь регистрируется как Участник.
3. Вводит свои данные.
4. Если все поля заполнены правильно.
4.1. Нажимает кнопку "Зарегистрироваться".
Постусловия:
1. Данные введены и актуальны

Таблица 2.4 - Описание прецедента "Решение заданий олимпиады"

Прецедент: Решение заданий олимпиады
ID: 2
Краткое описание: Пользователь просматривает задания олимпиады и отправляет их решение на сайт
Главные актеры: Пользователь
Второстепенные актеры: Нет
Предусловия:
1. Прецедент начинается по инициативе пользователя
Основной поток:
1. Пользователь заходит на web-портал.
2. Вводит свои данные.
3. Прикрепляет файл с решением задания.
Постусловия:
1. Данные введены и актуальны
Основной поток:
Нет

Таблица 2.5 - Описание прецедента "Предоставление информации об олимпиаде"

Прецедент: Предоставление информации об олимпиаде
ID: 3
Краткое описание: Администратор добавляет всю информацию об олимпиаде на web-портал
Главные актеры: Администратор
Второстепенные актеры: Нет
Предусловия: 1. Прецедент начинается по инициативе администратора
Основной поток: 1. Администратор заходит на web-портал. 2. Вводит свой логин и пароль. 3. Добавляет всю нужную информацию на web-портал.
Постусловия: 1. Данные предоставлены

Таблица 2.6 - Описание прецедента "Просмотр результата"

Прецедент: Просмотр результата
ID: 4
Краткое описание: Позволяет пользователю получить сведения о результатах выполненных заданий
Главные актеры: Пользователь системы
Второстепенные актеры: нет
1. Предусловие: Прецедент начинается по инициативе пользователя

Продолжение таблицы 2.6

Основной поток:
1. Пользователь выполняет задания.
2. Подведение итогов соревнования осуществляется системой автоматически
Постусловие:
1. Пользователь получает сведения о результатах выполненных заданий
Альтернативные потоки: Нет

Таблица 2.7 - Описание прецедента "Редактирование данных"

Прецедент:
Редактирование данных
ID: 5
Краткое описание:
Администратор добавляет, изменяет и удаляет данные с web-портала
Главные актеры:
Администратор
Второстепенные актеры:
Нет
Предусловия:
1. Прецедент начинается по инициативе администратора
Основной поток:
1. Администратор заходит на web-портал.
2. Заходит на форму авторизации администратора.
3. Вводит свой логин и пароль.
4. Добавляет, изменяет или удаляет нужные данные с web-портала.
5. Сохраняет данные.
Постусловия:
1. Данные web-портала отредактированы
Основной поток:
Нет

Таблица 2.8 - Описание прецедента "Добавление олимпиады"

Прецедент: Добавление олимпиады
ID: 6
Краткое описание: Добавление нового соревнования с описанием: теоретическим и практическим материалом
Главные актеры: Организатор олимпиады
Второстепенные актеры: нет
1. Предусловие: Прецедент начинается по инициативе пользователя
Основной поток: 1. Нажимает на кнопку “Добавить новое соревнование ” 2. Заполняет данные о соревновании 3. Создает материал и выставляет критерии оценивания
Постусловие: 1. Данные о пользователе добавлены в базу данных
Альтернативные потоки: Нет

На данном этапе были выделены основные роли, которые помогут ограничить доступ к набору функций и прав пользователей.

2.3.2 Диаграмма классов web-портала

На основе диаграммы вариантов использования была построена диаграмма классов (Рисунок 2.5) отображающая классы, их атрибуты и операции, а также связи между классами с указанием кратности.

На диаграмме представлены классы: *School*, *Participant*, *Administrator*, *Tournament*, *Task* и *Decision*.

Класс *School* (Школа) включает название и номер образовательного учреждения.

Класс *Participant* (Участник) содержит Ф.И.О ученика, название группы и номер класса.

Класс *Administrator* (Администратор) включает Ф.И.О администратора.

Класс *Tournament* (Турнир) включает название турнира, время открытия и закрытия доступа к прохождению соревнования.

Класс *Task* включает условие задачи, входные и выходные данные программ участников.

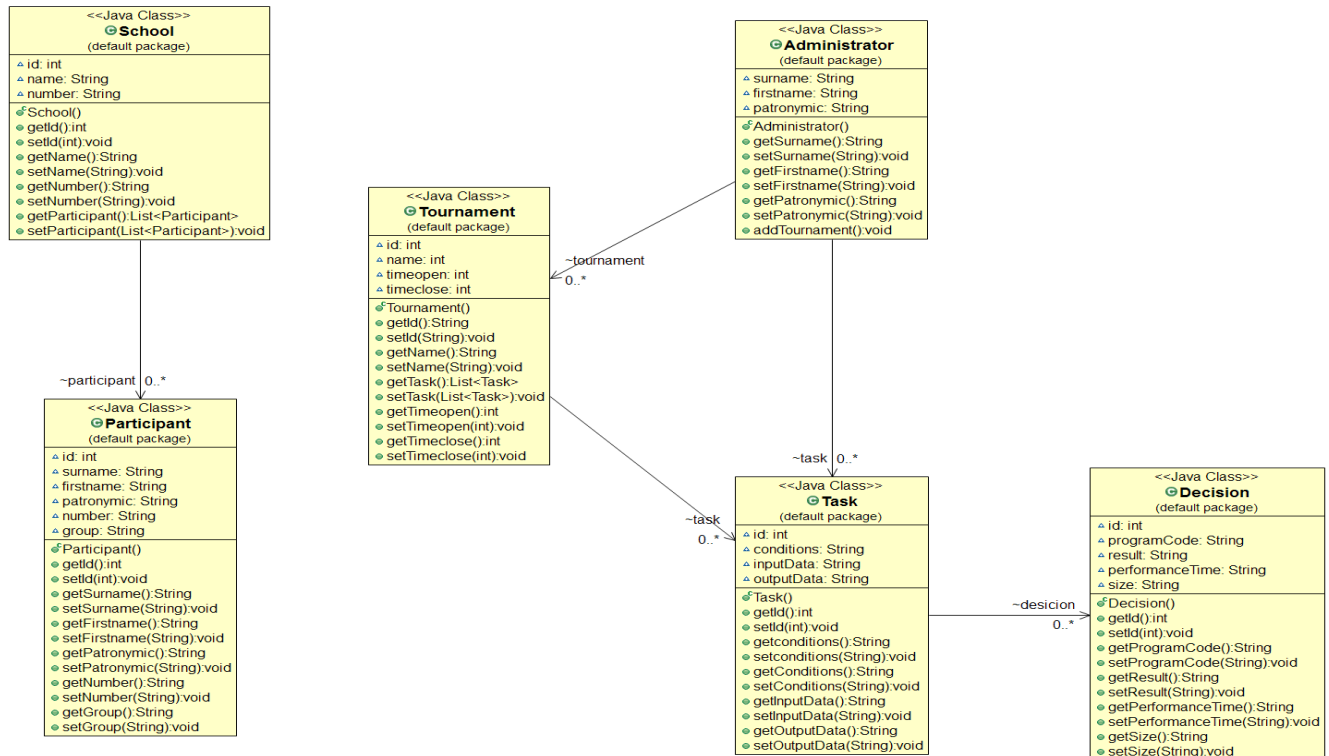


Рисунок 2.5 – Диаграмма классов

Последний класс *Decision* содержит непосредственно программный код, результат выполнения программы, время выполнения программы и размер программного кода (количество строк).

2.3.3 Разработка логической модели данных web-портала

Процесс логического моделирования данных информационной системы является проявлением взаимосвязи и интеграции объектной и реляционной технологий.

В настоящее время реляционные системы управления базами данных СУБД являются основным средством реализации баз данных и хранилищ данных. Реляционная технология используется достаточно долго, освоена большим количеством пользователей и разработчиков.

Для проектирования реляционных БД используется методология IDEF1X.

В методологии IDEF1X модель данных представляется на двух уровнях: логическом и физическом.

Логическая модель данных является независимым логическим представлением концептуальной модели данных информационной системы.

Физическая модель данных содержит определения всех реализуемых объектов в конкретной БД и для конкретной СУБД.

Для создания логической модели данных диаграмма классов web-портала для проведения online олимпиад трансформируется в ER-модель (диаграмму «сущность-связь»), в которую добавлены новые сущности, необходимые для обеспечения необходимого уровня нормализации данных.

На рисунке 2.6 изображена логическая модель данных, построенная на основе диаграммы классов.



Рисунок 2.6 – Логическая модель данных

Данный подход к моделированию данных базируется на концепции «сущность-связь» и используется для анализа информационной структуры систем разной природы. Информационная модель изображает логическую структуру информации об объектах системы.

Сущность - некоторый обособленный объект или событие моделируемой системы, имеющий конкретный набор параметров – атрибутов.

На данной диаграмме представлены следующие сущности: *Учебное заведение*, *Участник*, *Решение*, *Турнир* и *Задача* со связями различной кратности.

От одного учебного заведения может участвовать несколько учеников, что показано связью типа “один-ко-многим” между сущностями *Учебное заведение* и *Участник*.

Аналогично с предыдущей ситуацией сущность *Участник* относится к сущности *Решение* как “один-ко-многим”, потому что один участник олимпиады может решать несколько задач по программированию.

Такая же связь будет между сущностями *Задача* и *Решение*, поскольку одна задача может иметь множество разных решений.

Турнир состоит из множества разных задач, поэтому в данном случае необходима связь кратностью “один-ко-многим”.

2.4 Физическое моделирование автоматизированной системы для проведения online олимпиад по программированию

2.4.1 Выбор архитектуры web-портала

Современные web-приложения реализуются в архитектуре «клиент-сервер».

Архитектура «клиент-сервер» может быть представлена в следующих вариантах:

1) двухзвенная архитектура «клиент - сервер».

Первое звено - клиентские компьютеры с прикладными программами, с помощью которых пользователи обращаются по сети к базе данных («толстый клиент»).

Второе звено - сервер баз данных (СБД), также участвующий в обработке данных.

Благодаря двухзвенной архитектуре снижается нагрузка на информационную сеть, поскольку передаются только запросы и ответы на них.

Преимущества:

- относительная простота реализации;
- относительно невысокая стоимость серверного оборудования.

Недостатки:

- использование устаревших ИТ-решений;
- 2) трехзвенная архитектура «клиент - сервер».

Первое звено – web-браузер (IE, Mozilla, Chrome и др.) – «тонкий клиент».

Второе звено – сервер приложений – программное обеспечение промежуточного уровня. Синхронизирует работу компонентов всей системы и организует связь между ними. Обычно такую роль играют сетевые службы и приложения, в том числе web-серверы.

Третье звено – СБД. Данный сервер не работает напрямую с клиентскими программами, что повышает безопасность информации в системе.

Преимущества:

- масштабируемость;
- конфигурируемость - изолированность уровней друг от друга позволяет (при правильном развертывании архитектуры) быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней.

Недостатки:

- относительно высокая сложность создания приложений;
- высокие требования к производительности серверов приложений и сервера базы данных.

Результаты сравнительного анализа архитектур реализации web-портала приведены в таблице 2.9.

Таблица 2.9- Сравнительный анализ архитектур системы

Характеристика	Клиент-сервер, двухзвенная модель	Клиент-сервер, трехзвенная модель
Масштабируемость	-	+
Конфигурируемость	-	+
Минимальные затраты на разработку и эксплуатацию	+	-
Минимальные требования к оборудованию	+	-
Поддержка современных web-технологий	-	+
Сумма баллов	2	3

По результатам анализа предлагается использовать при реализации web-портала трехзвенную архитектуру «клиент-сервер».

2.4.2 Выбор сервера базы данных для реализации АИС для проведения online олимпиад по программированию.

Рассмотрим бесплатно распространяемые СУБД MySQL, PostGreSQL, FireBird и произведем их сравнительный анализ.

PostgreSQL - объектно-реляционная СУБД, которая поддерживает архитектуру «клиент-сервер» с распределением процессов между пользователями. Недостатком данной СУБД является уменьшение скорости работы команд манипулирования данными при больших массивах информации.

MySQL – это реляционная СУБД, которая характеризуется большой скоростью, устойчивостью и легкостью в использовании, является решением для малых и средних приложений. Это одна из самых быстрых современных СУБД.

Firebird – компактная реляционная, кросс-платформенная СУБД. Основным недостатком является отсутствие кэша результатов запросов, полнотекстовых индексов, значительное падение производительности при росте внутренней фрагментации базы.

Результаты сравнительного анализа СБД для web-портала приведены в таблице 2.10.

Таблица 2.10- Сравнительный анализ СБД

Характеристики	PostgreSQL	MySQL	FireBird
Быстродействие	-	+	–
Поддержка транзакций	+	+	+
Поддержка реляционных баз данных	+	+	+
Сумма баллов	2	3	2

Из сравнительной таблицы можно сделать вывод, что СУБД MySQL, обеспечивает лучшую производительность, поэтому выбираем ее в качестве СБД web-портала.

2.4.3 Выбор языка программирования для реализации web-портала

При подходе к разработке программного обеспечения в рамках спиральной модели его жизненного цикла при выборе средств разработки важно учесть следующие критерии:

- поддержка технологии быстрой разработки приложений RAD (Rapid Application Development), основанной на объектно-ориентированном программировании;
- полномасштабная поддержка трехзвенной архитектуры «клиент-сервер»;
- знание языка разработчиком;
- поддержка СУБД MySQL;

В настоящее время для разработки динамических сайтов широко применяются такие технологии web-программирования, как ASP.NET, Java и PHP.

ASP.NET – технология создания Web-приложений компании Майкрософт. Она является составной частью платформы Microsoft.NET.

Для получения высокой производительности данную технологию рекомендуется использовать совместно с другими продуктами компании Майкрософт.

Java – популярный объектно-ориентированный язык программирования, разработанный компанией Sun (в последующем приобретённой компанией Oracle). К недостаткам относится возможное снижение скорости выполнения приложений.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки динамических web-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических web-приложений.

Результаты сравнительного анализа технологий web-программирования для web-портала приведены в таблице 2.11

Таблица 2.11 - Сравнительный анализ технологий web- программирования

Характеристики (баллы)	ASP.NET	Java	PHP
Объектно-ориентированный подход к разработке ПО	+	+	+
Поддержка трехзвенной архитектуры «клиент-сервер»	+	+	+
Знание языка разработчиком	-	-	+
Поддержка СУБД MySQL	-	+	+
Сумма баллов	2	3	5

По результатам анализа выбираем язык программирования PHP.

Принимая во внимание вышеперечисленные доводы, выбираем PHP в качестве основного средства разработки динамического приложения web-портала для проведения online-олимпиад по программированию.

Для разработки ПО использована локальная среда разработки PHP – Денвер (<http://denwer.ru/>).

Как следует из описания, среда разработки Денвер позволяет запустить эмулятор web-сервера с поддержкой технологии PHP+ MySQL на машине, работающей под управлением MS Windows.

Инструкции по установке дистрибутива Денвер можно прочитать на сайте разработчиков. Отметим только, что его установка достаточно проста и не требует особых навыков. Этот дистрибутив можно рекомендовать для начинающих программистов, желающих научиться работать с PHP. Денвер отличается такими качествами, как модульность, допускающая загрузку дополнительных расширений, отсутствие необходимости конфигурирования, возможность работать с PHP как с модулем Apache, так и используя CGI. Следует учесть, что Денвер не заменяет полностью традиционный вариант установки Apache для MS Windows.

После запуска все компоненты Денвера размещаются на виртуальном диске.

2.4.4 Выбор web-сервера для web-портала

Сравним свободно распространяемые web-серверы Apache, Nginx и Tornado.

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux и Microsoft Windows. Основными достоинствами Apache считаются надёжность и гибкость конфигурации.

Lighttpd – web-сервер, разрабатываемый с расчётом на быстроту и защищённость, а также соответствие стандартам. Это свободное программное обеспечение, распространяемое по лицензии BSD. К недостатком можно отнести некоторые проблемы с конфигурированием.

Tornado – расширяемый, не блокирующий web-сервер и фреймворк, написанный на языке Python. Он создан для использования в проекте FriendFeed. Отличается сложностью настройки и конфигурирования.

Результаты сравнительного анализа web-серверов представлены в таблице 2.12.

Таблица 2.12 - Сравнительный анализ web-серверов

Характеристики	Apache	Lighttpd	Tornado
потребление ресурсов	+	-	-
поддержка PHP	+	+	+
простота настройки	+	-	-
распространенность	+	-	-
Сумма баллов	4	1	1

Как следует из таблицы 2.12, наилучшими характеристиками обладает web-сервер Apache.

2.4.5 Разработка физической модели данных для web-портала

Физическая модель данных зависит от конкретной СУБД, фактически являясь отображением системного каталога.

В физической модели содержится информация обо всех объектах БД.

На рисунке 2.7 представлена физическая модель БД web-портала, построенная для СУБД MySQL.

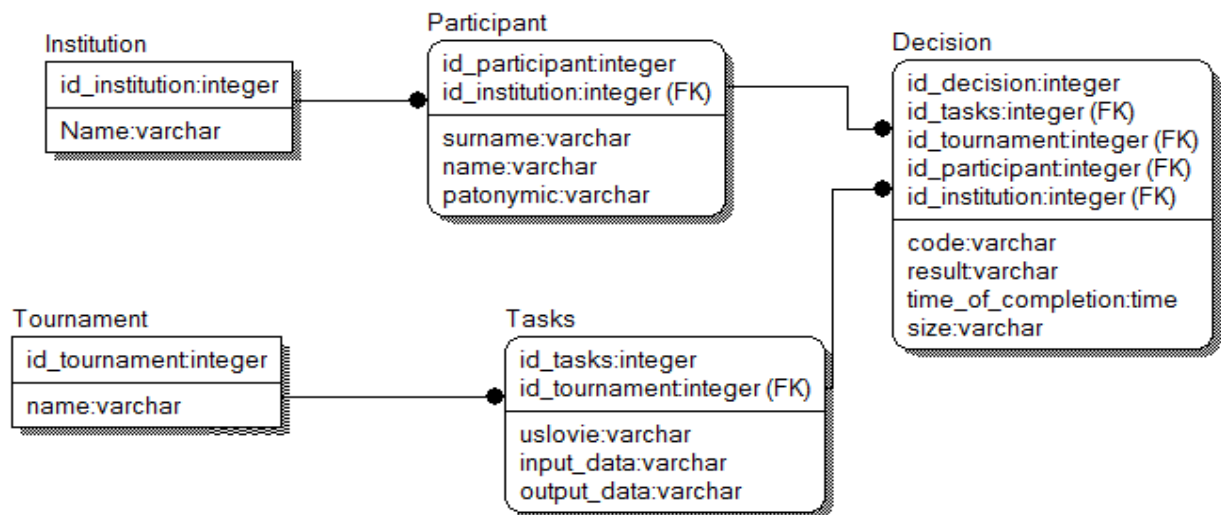


Рисунок 2.7 – Физическая модель данных

Физическая модель данных разработана на основе логической модели данных системы.

2.4.6 Диаграмма размещения компонентов web-портала

Информационная система состоит из следующих компонентов:

- 1) клиент – Web –браузер (IE, Mozilla, Chrome);
- 2) web-сервер – построен на базе ПО Apache;
- 3) сервер баз данных на основе СУБД MySQL.

Доступ к серверам web-портала осуществляется через Интернет.

Диаграмма компонентов в отличие от ранее рассмотренных диаграмм, описывает особенности физического представления системы. Она позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный и исполняемый код. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и зависимости между ними.

Разработанная диаграмма компонентов представлена на рисунке 2.8.

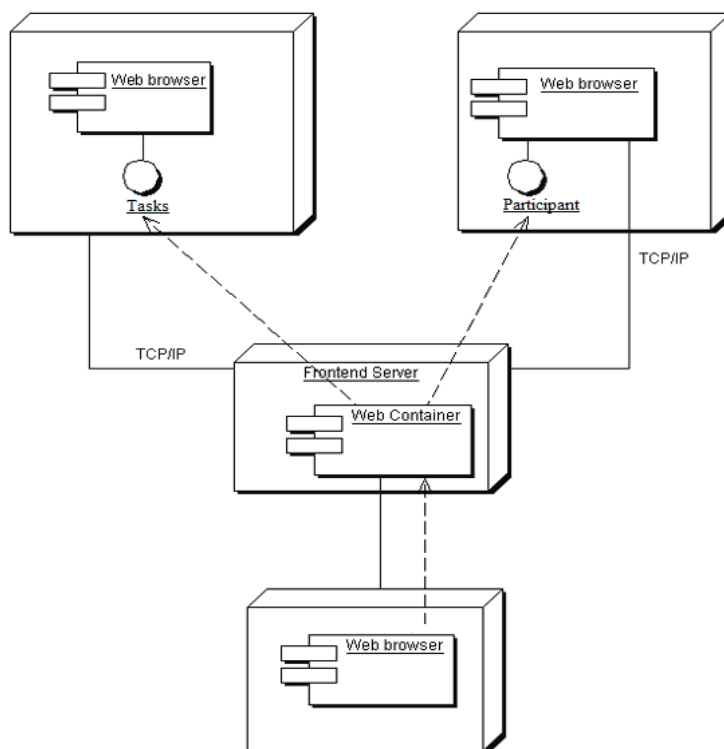


Рисунок 2.8 - Диаграмма компонентов

Таким образом, была разработана диаграмма компонентов, показывающая совокупность модулей аппаратного и программного уровней, и взаимосвязи их друг с другом.

Вывод по второй главе

В главе было проведено проектирование автоматизированной информационной системы, где были рассмотрены логическое и физическое моделирования базы данных. Содержание этих пунктов раскрывает структуру базы данных, что необходимо для её создания.

Построена концептуальная модель потоков данных для процесса проведения online олимпиады. На ее основе были сформулированы функциональные требования, предъявляемые к АИС для проведения online олимпиад, связанные с созданием соревнования по программированию, хранением данных об участниках, их заявках на участие, проверка результатов решений задач и подведение итогов прохождения олимпиады.

Глава 3 Практическая реализация автоматизированной системы для проведения online олимпиад по программированию

3.1 Реализация функциональных требований

В процессе разработки автоматизированной системы были выполнены следующие задачи:

Для начала заходим на веб-сервер phpmyadmin (Рисунок 3.1).

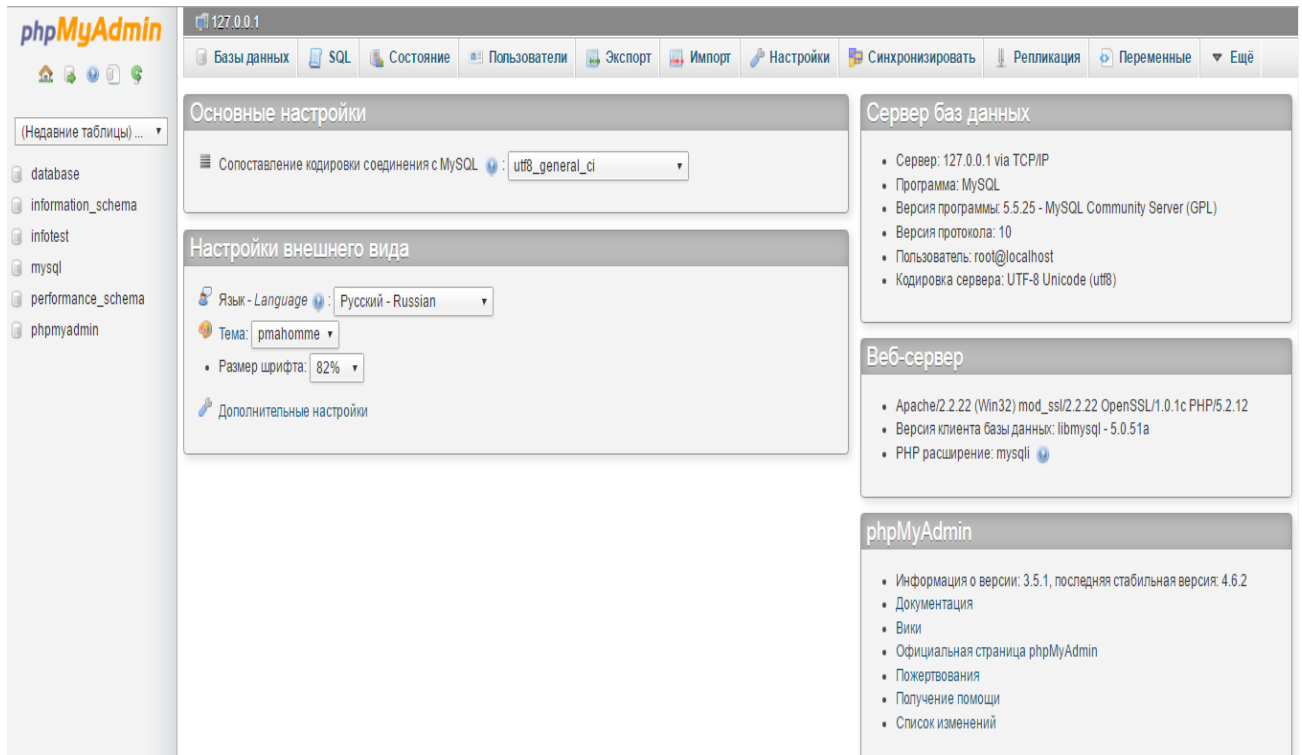


Рисунок 3.1 – Веб-сервер phpmyadmin

Создаем таблицу users в базе данных database. Таблица будет содержать данные пользователя: логин и пароль (Рисунок 3.2).

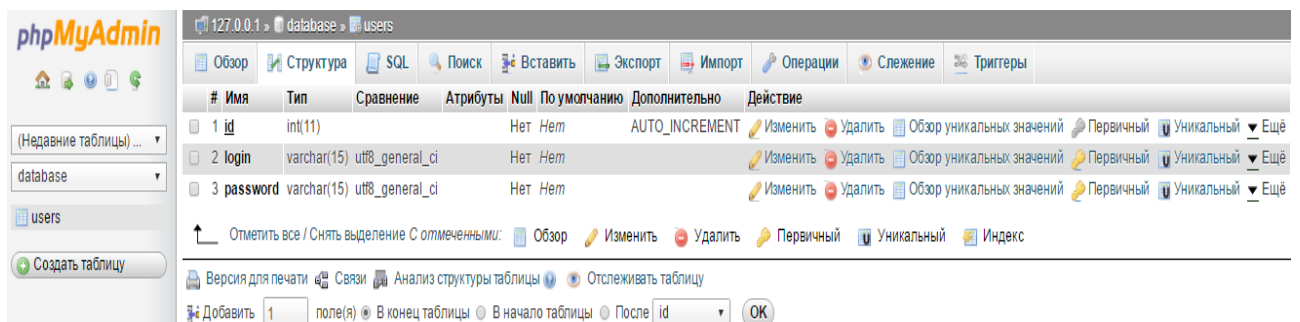


Рисунок 3.2 – Таблица users

Чтобы установить соединение с этой таблицей создаем файл bd.php (Рисунок 3.3).

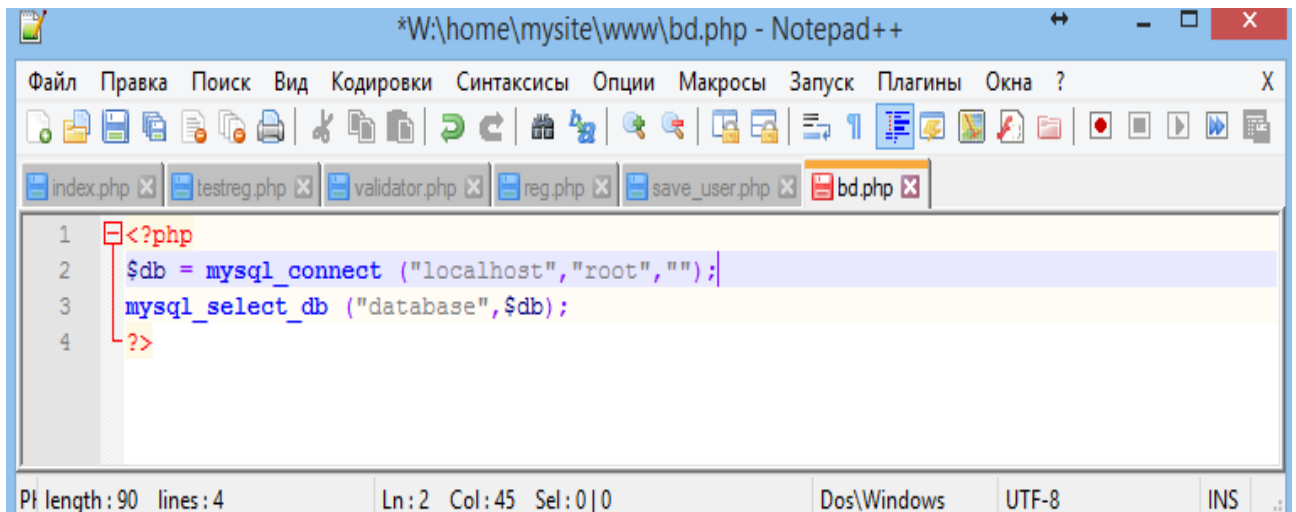


Рисунок 3.3 – Соединение файла с таблицей users

Создаем страницу, на которой пользователи смогут оставлять свои данные. Для этого создаем файл reg.php (Рисунок 3.4).

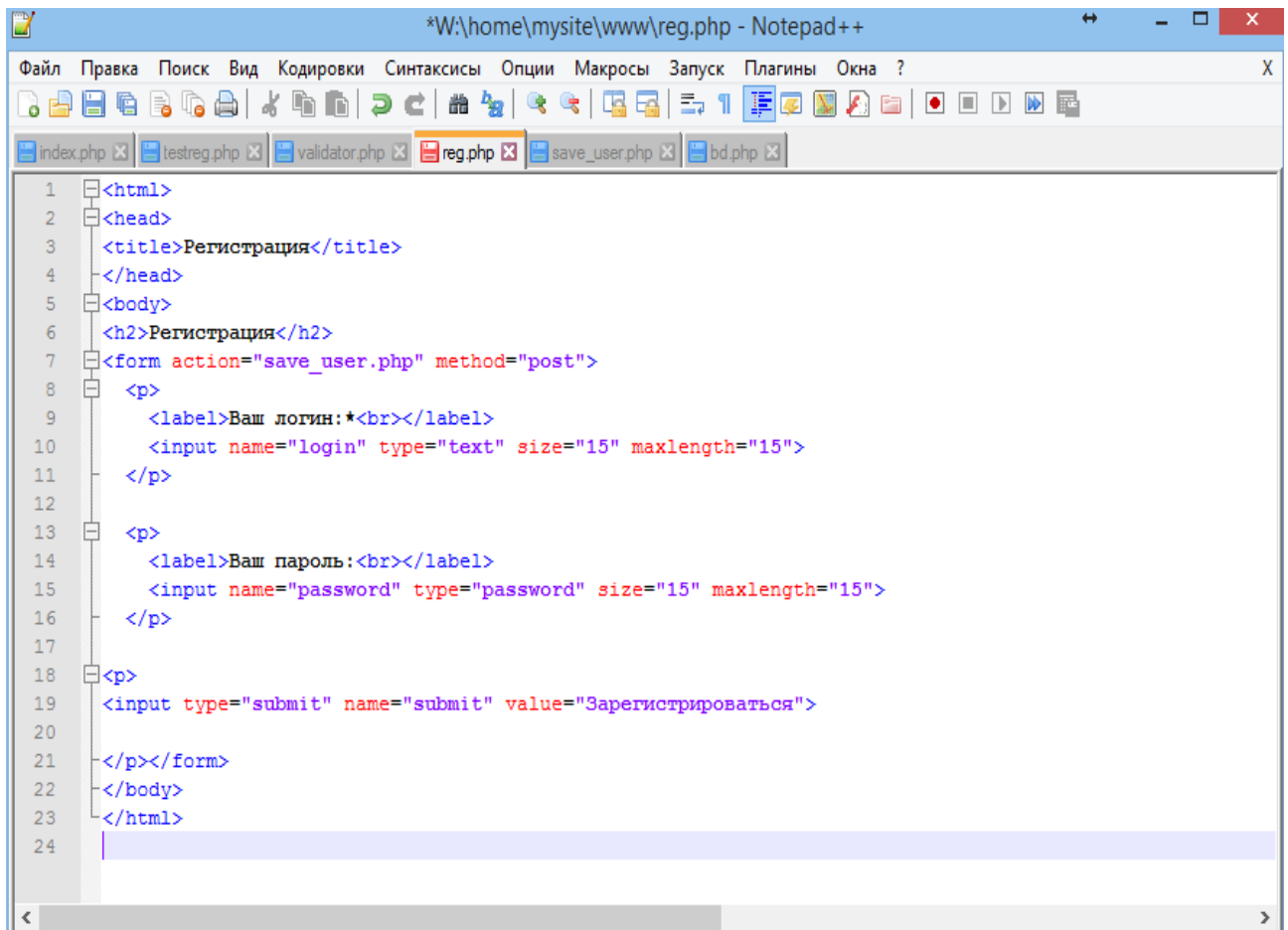
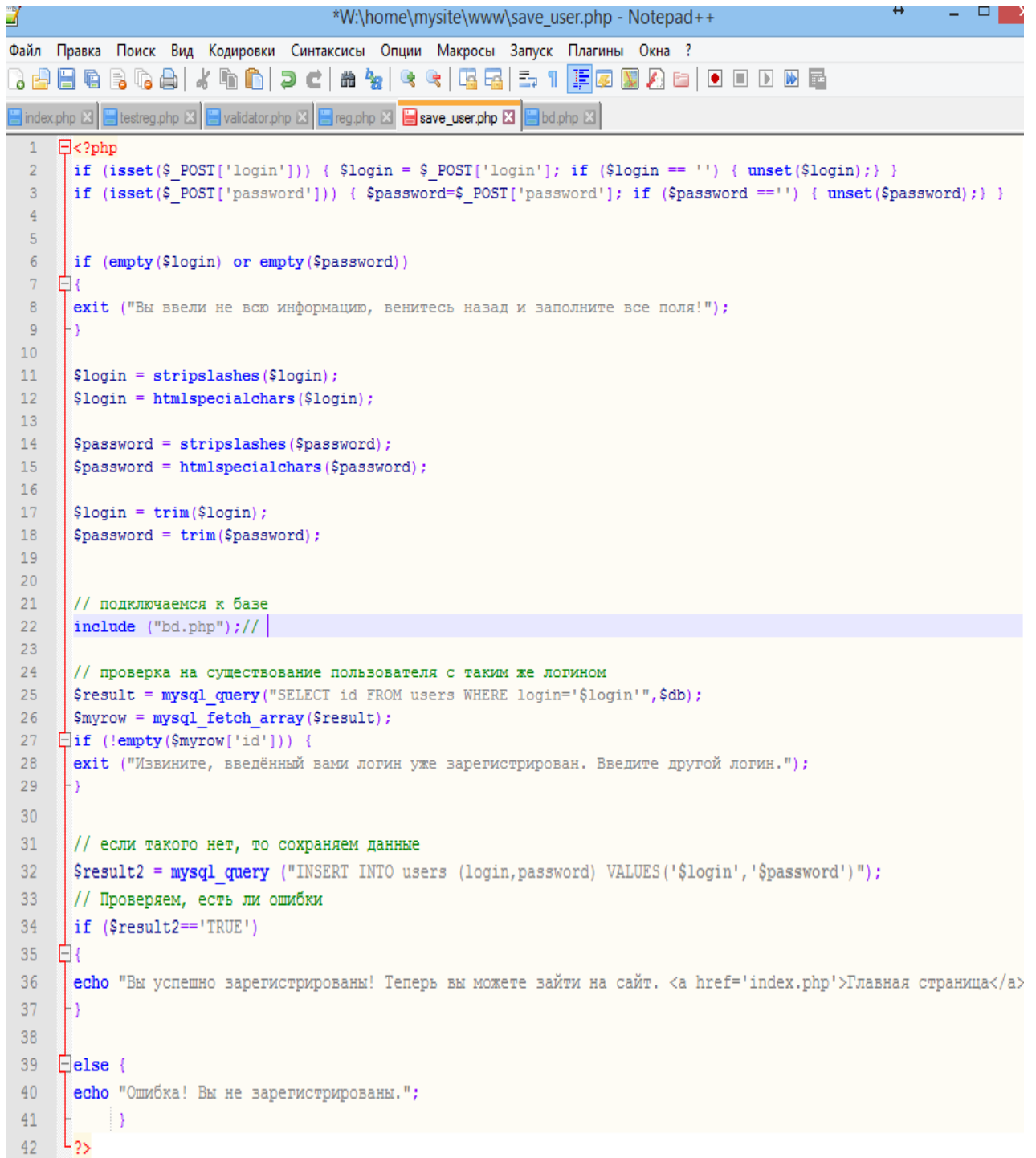


Рисунок 3.4– Регистрация пользователей

После нажатия на кнопку "Зарегистрироваться", данные из полей отправляются на страницу save_user.php методом "post".

Теперь, чтобы занести эти данные в базу данных в файле save_user.php прописываем следующий код (Рисунок 3.5).



```

1 <?php
2 if (isset($_POST['login'])) { $login = $_POST['login']; if ($login == '') { unset($login);} }
3 if (isset($_POST['password'])) { $password=$_POST['password']; if ($password == '') { unset($password);} }
4
5
6 if (empty($login) or empty($password))
7 {
8     exit ("Вы ввели не всю информацию, вернитесь назад и заполните все поля!");
9 }
10
11 $login = stripslashes($login);
12 $login = htmlspecialchars($login);
13
14 $password = stripslashes($password);
15 $password = htmlspecialchars($password);
16
17 $login = trim($login);
18 $password = trim($password);
19
20
21 // подключаемся к базе
22 include ("bd.php");//
23
24 // проверка на существование пользователя с таким же логином
25 $result = mysql_query("SELECT id FROM users WHERE login='$login'", $db);
26 $myrow = mysql_fetch_array($result);
27 if (!empty($myrow['id'])) {
28     exit ("Извините, введенный вами логин уже зарегистрирован. Введите другой логин.");
29 }
30
31 // если такого нет, то сохраняем данные
32 $result2 = mysql_query ("INSERT INTO users (login,password) VALUES('$login','$password')");
33 // Проверяем, есть ли ошибки
34 if ($result2=='TRUE')
35 {
36     echo "Вы успешно зарегистрированы! Теперь вы можете зайти на сайт. <a href='index.php'>Главная страница</a>";
37 }
38
39 else {
40     echo "Ошибка! Вы не зарегистрированы.";
41 }
42 ?>

```

Рисунок 3.5 – Сохранение пользователей в базе данных

В файле index.php описываем процедуру входа на сайт уже зарегистрированных пользователей (Рисунок 3.6).

```

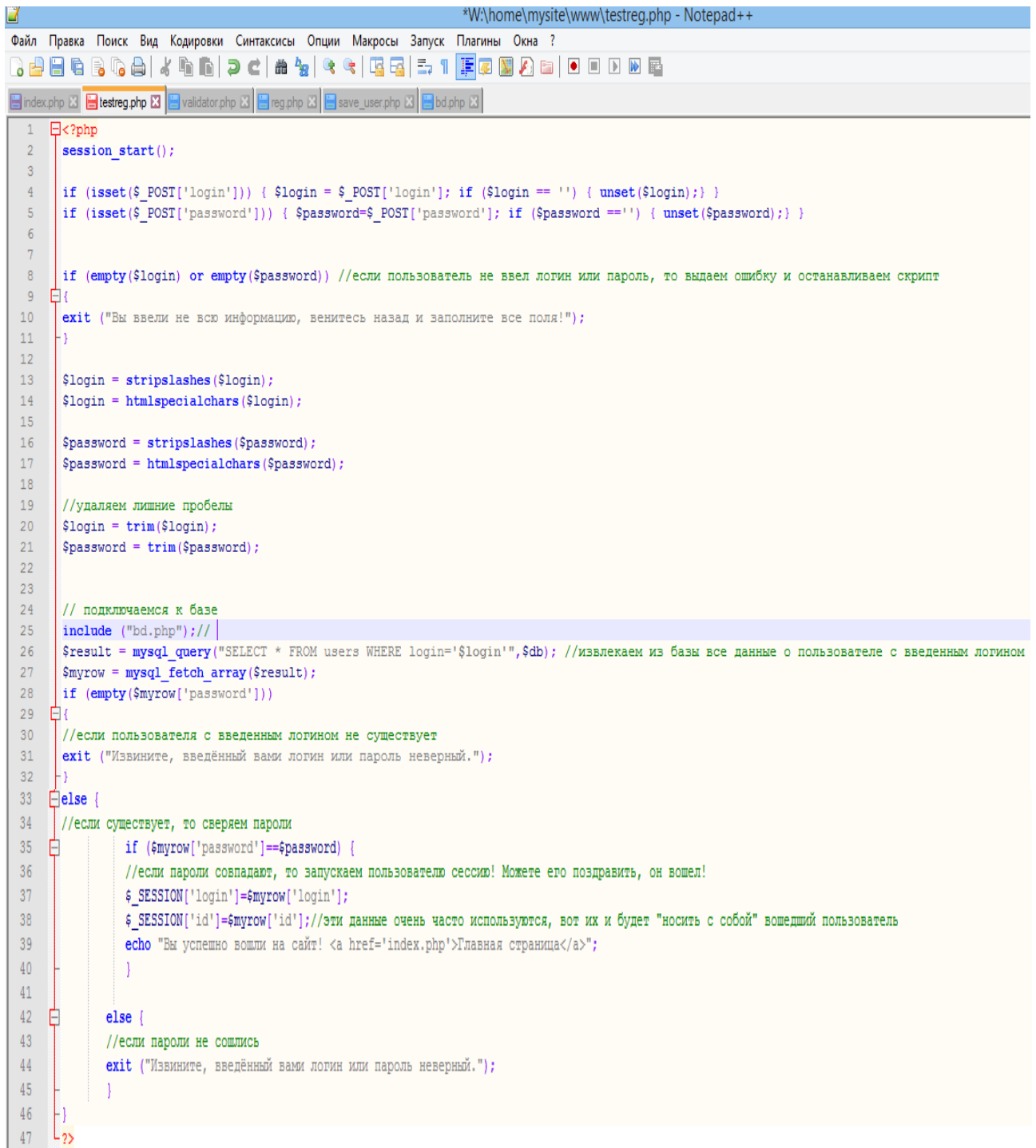
1  <?php
2
3      session_start();
4      ?>
5
6      <html>
7      <head>
8      <title>Главная страница</title>
9      </head>
10     <body>
11     <h2>Главная страница</h2>
12     <form action="testreg.php" method="post">
13
14     <p>
15         <label>Ваш логин:<br></label>
16         <input name="login" type="text" size="15" maxlength="15">
17     </p>
18
19     <p>
20         <label>Ваш пароль:<br></label>
21         <input name="password" type="password" size="15" maxlength="15">
22     </p>
23
24     <p>
25         <input type="submit" name="submit" value="Войти">
26     </p>
27
28     <br>
29     <a href="reg.php">Зарегистрироваться</a>
30 </p></form>
31 <br>
32 <?php
33 // Проверяем, пусты ли переменные логина и id пользователя
34 if (empty($_SESSION['login']) or empty($_SESSION['id']))
35 {
36     // Если пусты, то мы не выводим ссылку
37     echo "Вы вошли на сайт, как гость<br><a href='#'>Эта ссылка доступна только зарегистрированным пользователям</a>";
38 }
39 else
40 {
41     // Если не пусты, то мы выводим ссылку
42     echo "Вы вошли на сайт, как ".$_SESSION['login']. "<br><a href=''>Эта ссылка доступна только зарегистрированным пользователям</a>";
43 }
44 ?>
45 </body>
46 </html>

```

Рисунок 3.6 – Вход на сайт

Вся процедура работает на сессиях. Именно в ней хранятся данные пользователя, пока он находится на сайте.

Создаем файл testreg.php для проверки введенного логина и пароля (Рисунок 3.7).



```

1 <?php
2 session_start();
3
4 if (isset($_POST['login'])) { $login = $_POST['login']; if ($login == '') { unset($login); } }
5 if (isset($_POST['password'])) { $password = $_POST['password']; if ($password == '') { unset($password); } }
6
7
8 if (empty($login) or empty($password)) //если пользователь не ввел логин или пароль, то выдаем ошибку и останавливаем скрипт
9 {
10 exit ("Вы ввели не всю информацию, вернитесь назад и заполните все поля!");
11 }
12
13 $login = stripslashes($login);
14 $login = htmlspecialchars($login);
15
16 $password = stripslashes($password);
17 $password = htmlspecialchars($password);
18
19 //удаляем лишние пробелы
20 $login = trim($login);
21 $password = trim($password);
22
23
24 // подключаемся к базе
25 include ("bd.php"); //
26 $result = mysql_query("SELECT * FROM users WHERE login='$login'", $db); //извлекаем из базы все данные о пользователе с введенным логином
27 $myrow = mysql_fetch_array($result);
28 if (empty($myrow['password']))
29 {
30 //если пользователя с введенным логином не существует
31 exit ("Извините, введенный вами логин или пароль неверный.");
32 }
33 else {
34 //если существует, то сверяем пароли
35 if ($myrow['password']==$password) {
36 //если пароли совпадают, то запускаем пользователю сессию! Можете его поздравить, он вошел!
37 $_SESSION['login']=$myrow['login'];
38 $_SESSION['id']=$myrow['id']; //эти данные очень часто используются, вот их и будет "носить с собой" вошедший пользователь
39 echo "Вы успешно вошли на сайт! <a href='index.php'>Главная страница</a>";
40 }
41
42 else {
43 //если пароли не сошлись
44 exit ("Извините, введенный вами логин или пароль неверный.");
45 }
46 }
47

```

Рисунок 3.7 – Проверка логина и пароля

На рисунке 3.8 продемонстрирован мой внешний вид сайта, а также его главное меню.

Дизайн сайта выглядит более привлекательным и удобным для

отображения.

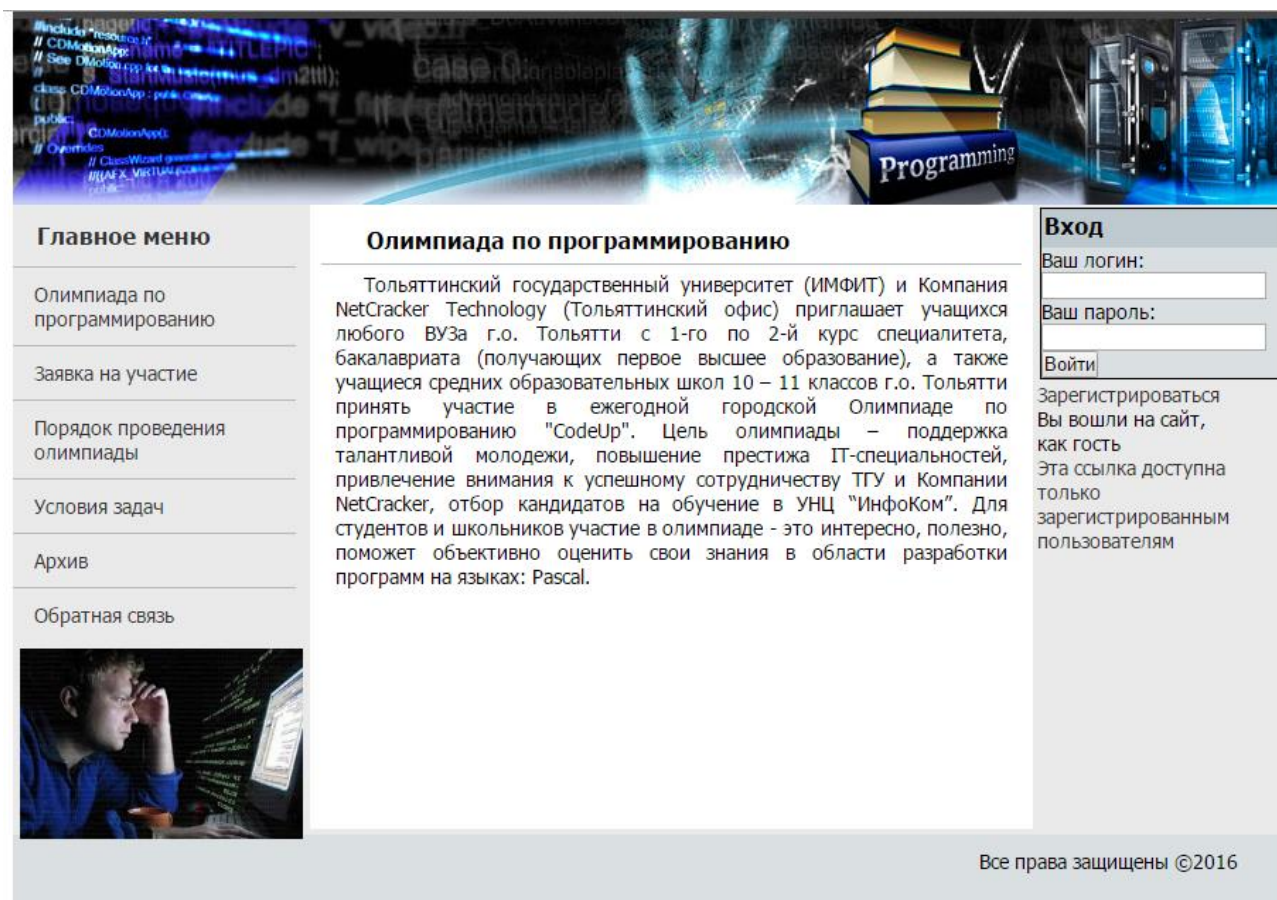


Рисунок 3.8 – Главное меню

Защита информационной системы (Рисунок 3.9):

- *идентификация* (именование и опознавание), *аутентификация* (подтверждение подлинности) пользователей системы;
- *разграничение доступа* пользователей к ресурсам системы и авторизация (присвоение полномочий) пользователям;

Вход

Ваш логин:

Ваш пароль:

Войти

Зарегистрироваться
Вы вошли на сайт,
как гость
Эта ссылка доступна
только
зарегистрированным
пользователям

Рисунок 3.9 – Вход на сайт

В случае прохождения успешной проверки пользователя производится авторизация пользователя с использованием сессии.

Разграничение прав доступа происходит разделением пользователей на три группы:

1. Гость – не авторизовавшийся или не зарегистрированный пользователь (посетитель)
2. Пользователь – авторизовавшийся пользователь
3. Администратор – специалист занимающийся обслуживанием сайта

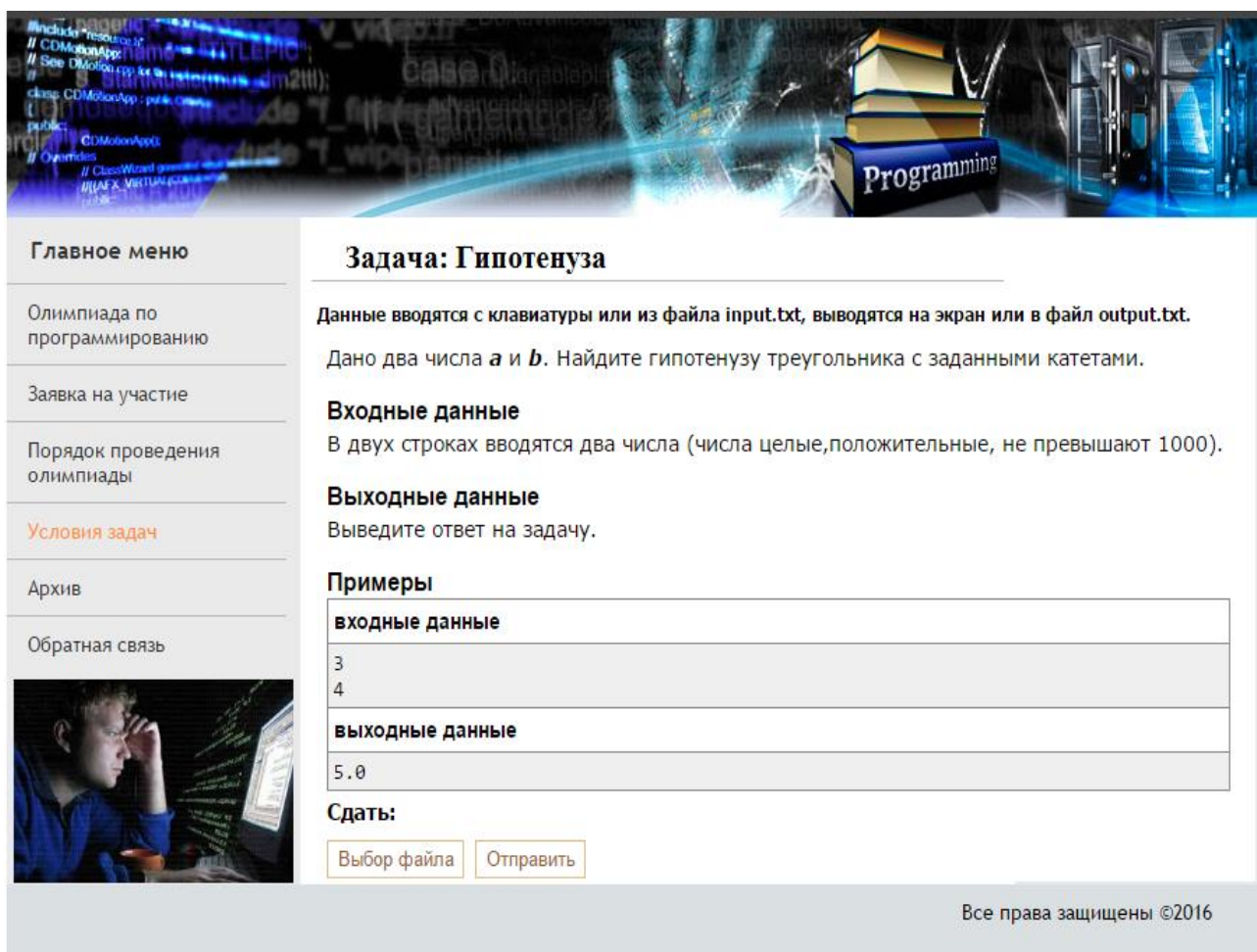
На начальном этапе пользователь проходит регистрацию, после чего подает заявку на участие в online олимпиаде (Рисунок 3.10).

Оформление заявки

Фамилия*	
Имя*	
Отчество*	
Дата рождения*	дд.мм.гггг
Город*	
Вы учащийся*	☐ Школы ☐ Лицея ☐ Колледжа ☐ Университета
Телефон	
E-Mail*	
Отправить заявку на участие	

Рисунок 3.10 – Подача заявки на участие в олимпиаде

После этого сформированная заявка поступает для подтверждения организатору олимпиады. Он формирует задачу, и уже готовые задачи поступают данным участникам. На рисунке 3.11 представлен пример одной из задач олимпиады.



Главное меню

- Олимпиада по программированию
- Заявка на участие
- Порядок проведения олимпиады
- Условия задач**
- Архив
- Обратная связь

Задача: Гипотенуза

Данные вводятся с клавиатуры или из файла input.txt, выводятся на экран или в файл output.txt.

Дано два числа a и b . Найдите гипотенузу треугольника с заданными катетами.

Входные данные

В двух строках вводятся два числа (числа целые, положительные, не превышают 1000).

Выходные данные

Выведите ответ на задачу.

Примеры

входные данные	выходные данные
3 4	5.0

Сдать:

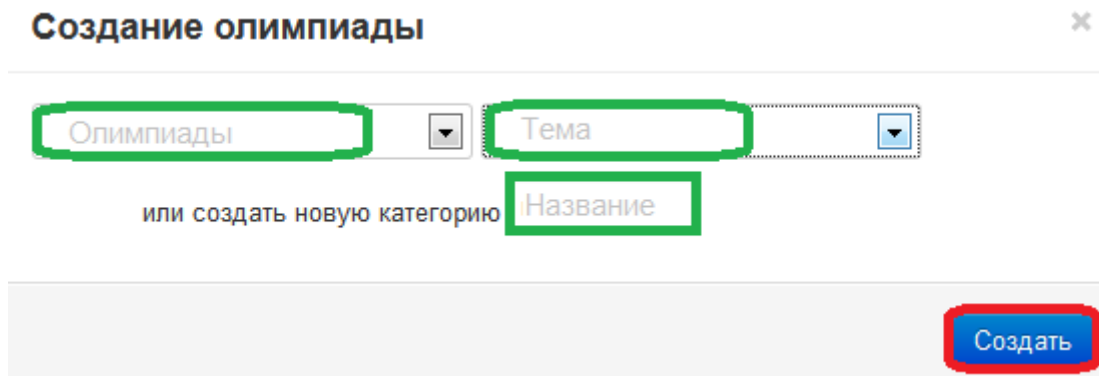
Все права защищены ©2016

Рисунок 3.11 – Пример задачи

После того как участнику олимпиады открыт доступ к заданиям олимпиады, он их решает и отправляет на проверку.

Для создания олимпиады пользователь системы должен быть авторизован, причем обладать правами на создание олимпиад (Рисунок 3.12).

После заполнения всех полей нужно нажать на кнопку «Создать».



Создание олимпиады ×

Олимпиады Тема

или создать новую категорию

Рисунок 3.12 – Создание олимпиады

После этого откроется окно по созданию или редактированию олимпиад. Мы видим несколько вкладок. На вкладке «Итоги» можно будет объявлять результаты по проведению олимпиады.

Во вкладке «Статистика» будет отображаться статистика по проведению олимпиады.

Во вкладке «Общие сведения» можно опубликовать олимпиаду, то есть сделать её открытой для всех пользователей, так как на момент подготовки олимпиада будет видима только тем, кто её создал. Так же здесь можно удалить олимпиаду и изменить к ней описание (Рисунок 3.13).

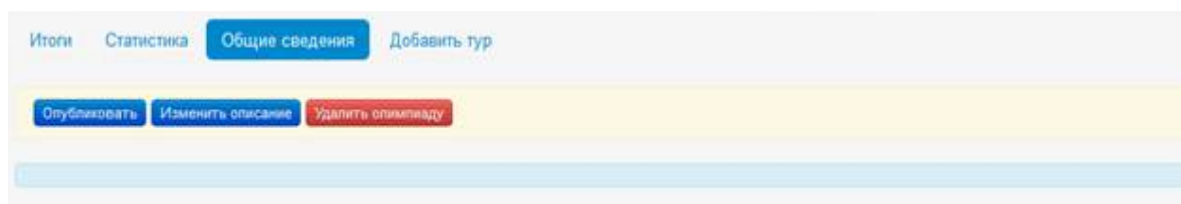


Рисунок 3.13 – Создание и редактирование олимпиад

Вывод по третьей главе

Была разработана автоматизированная информационная система, в которой были реализованы механизмы защиты системы, а также возможность добавлять, удалять и редактировать соревнования пользователем системы.

Заключение

В ходе работы было проведено моделирование системы по проведению online олимпиад по программированию.

Исследование предметной области проводилось на основе концептуальной модели по методологиям IDEF0, DFD и UML с использованием CASE-технологий, которые позволили упростить процесс проектирования концептуальной модели.

Концептуальная модель предметной области, помогла выявить основные элементы будущей системы, их атрибуты, характеристики и способы их взаимодействия.

На основе диаграммы вариантов использования были составлены общие требования к функциональному поведению проектируемой системы, разработана исходная концептуальная модель системы для ее последующей детализации в форме логической модели.

На основании модели «Как должно быть» и анализа существующих программных решений для выбранной предметной области была реализована постановка задачи на проектирование АИС.

Пройденные этапы моделирования помогли построить модель системы, полностью удовлетворяющую всем требованиям, предъявленным в задании.

Для проектируемой системы была построена информационная модель, построены физическая и логическая схема базы данных.

С помощью технологии PHP+MySQL разработана автоматизированная информационная система для проведения online олимпиад по программированию.

Список используемой литературы

1. ГОСТ-34 РД 50-680-88. Руководящий документ по стандартизации. Методические указания. Автоматизированные системы. Основные положения.
2. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Термины и определения.
3. ГОСТ 34.320-96. Информационная технология. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы.
4. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.
5. Агальцов, В.П. Базы данных. Локальные базы данных: учеб. [для студ. вузов, обуч. по спец. "Информатика и вычислительная техника"] / В. П. Агальцов. - 3-е изд., перераб.; Гриф УМО. - М. : ФОРУМ - ИНФРА-М, 2011. - 349 с.
6. Босуэлл, Д. Читаемый код, или Программирование как искусство / Д. Босуэлл, Т. Фаучер. – СПб.: Питер, 2012. – 208 с.
7. Гупта А. Java EE 7. Основы. / А. Гупта – Вильямс, 2014. – 336 с.
8. Джим Арлоу, Айла Нейштадт. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование - Символ-Плюс 2012. - 624 с.
9. Дронов, Владимир PHP, MySQL, HTML5 и CSS 3. Разработка современных динамических web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, 2016. - 688 с.
10. Дэнни Гудман, Майкл Моррисон. JavaScript. Библия пользователя. - СПб: Вильямс, 2012. - 1184 с.
11. Заботина, Н.Н. Проектирование информационных систем: учебное пособие / Н.Н. Заботина – М.: ИНФРА - М, 2013. – 331 с.
12. Зандстра, М. PHP. Объекты, шаблоны и методики программирования / Мэтт Зандстра. - М.: Вильямс, 2015. - 576 с.

13. Золотов С.Ю. Проектирование информационных систем [Электронный ресурс]: учеб. пособие / С.Ю. Золотов; Томский гос. ун-т систем управления и радиоэлектроники. – Томск: Эль Контент, 2013.- 86 с.
14. Зудилова, Т.В. Web-программирование PHP: учебное пособие / М.Л. Буркова - СПб.: НИУ ИТМО, 2012. - 79 с.
15. Информационные системы и технологии. Проектирование систем с использованием SADT-методологии. Методические указания. – М., 2013 г.
16. Гайкович В.Ю., Ершов Д.В. Основы безопасности информационных технологий - М.: МИФИ, 2013. – 96 с.
17. Карвин, Б. Программирование баз данных SQL. Типичные ошибки и их устранение / Б. Карвин, пер. с англ. Райтман - М.: Рид Групп, 2012.-338 с.
18. Киселев. А. EJB 3 в действии. / А. Киселев. - ДМК Пресс, 2016 . – 600 с. Колисниченко, Денис PHP и MySQL. Разработка веб-приложений / Денис Колисниченко. - М.: БХВ-Петербург, 2015. - 592 с.
19. Конверс PHP 5 и MySQL. Библия пользователя / Конверс, др. Т. и. - М.:Вильямс, 2014.-814 с.
20. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2014. – 206 с.
21. Ломов, А.Ю. Apache, Perl, MySQL: практика создания динамических сайтов. Самоучитель. /А.Ю. Ломов. - СПб: БХВ-Петербург, 2014. –368 с.
22. Маклафлин, Б. Объектно-ориентированный анализ и проектирование / Б. Маклафлин, Г. Поллайс, Д. Уэст. — СПб.: Питер, 2013. — 608 с.
23. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScriptи CSSи HTML5-3-е изд. / Р. Никсон - СПб.: Питер, 2015. - 688 с.
24. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А. Прохоренок — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 912 с.

25. Рудинский, И. Д. Технология проектирования автоматизированных систем обработки информации и управления: учеб. пособие / И. Д. Рудинский. – М. : Горячая линия - Телеком, 2011. - 304 с.
26. Томас А. Пауэлл. Ajax. Настольная книга программиста / Томас А. Пауэлл. – М: Эксмо, 2010. - 720 стр.
27. Шелухин, О. И. Моделирование информационных систем: учеб. пособие. 004 / О. И. Шелухин. - 2-е изд., перераб. и доп. – М. : Горячая линия - Телеком, 2012. - 516 с.
28. Назарова, О.Б. Разработка реляционных баз данных с использованием CASE-средства All Fusion Data Modeler [Электронный ресурс]: учеб.- метод. пособие / О.Б. Назарова, О.Е. Масленникова. — 2-е изд., стер. — М.: ФЛИНТА, 2013. — 74 с.
29. Суркова Н.Е. Проектирование информационных систем [Электронный ресурс]: метод. указания к курсовому проекту / Н.Е. Суркова. — Москва: Рос. новый ун-т, 2012. — 59 с.
30. Justin Ferriman - Какова роль вашей СУО? [Электронный ресурс] // LearnDash [Официальный веб-сайт]: <http://www.learndash.com/what-is-the-role-of-your-lms/Justin Ferriman>.
31. Alan Mark Davis. Just Enough Requirements Management: Where Software Development Meets Marketing. — Dorset House, 2015.
32. Gilmore W.J. Beginnig PHP and MySQL. - 3th Edition, 2012.
33. John Brock, Arun Gupta, Geertjan Wielenga. PHP and HTML5 Enterprise Application Development. - 1st Edition, Oracle Press, 2014.
34. Mario Fusco, Alan Mycroft. PHP in Action: Lambdas, Streams, and functional-style programming. - 1st Edition, Manning Publications, 2014.
35. Robbins J.R. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. - 4th Edition, 2012.