

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение  
высшего образования «Самарский национальный исследовательский  
университет имени академика С.П. Королева (Самарский университет)»

Институт \_\_\_\_\_ информатики и кибернетики \_\_\_\_\_

Кафедра \_\_\_\_\_ программных систем \_\_\_\_\_

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

\_\_\_\_\_ к курсовой работе по дисциплине «Программная инженерия» \_\_\_\_\_

\_\_\_\_\_ по теме «Клавиатурный тренажёр с функциями администратора» \_\_\_\_\_

Обучающийся \_\_\_\_\_ С.А. Атякшев

Обучающийся \_\_\_\_\_ Д.О. Колбанов

Обучающийся \_\_\_\_\_ Р.К. Лукашевич

Руководитель \_\_\_\_\_ Л.С. Зеленко

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение  
высшего образования «Самарский национальный исследовательский  
университет имени академика С.П. Королева (Самарский университет)»

Институт \_\_\_\_\_ информатики и кибернетики \_\_\_\_\_

Кафедра \_\_\_\_\_ программных систем \_\_\_\_\_

ЗАДАНИЕ

на курсовую работу по дисциплине  
«Программная инженерия»  
обучающимся в группе № 6401-020302D

С.А. Атякшеву

Д.О. Колбанову

Р.К. Лукашевичу

- 1 Тема проекта: «Клавиатурный тренажёр с функциями администратора» \_\_\_\_\_
- 2 Исходные данные к проекту: см. приложение к заданию \_\_\_\_\_
- 3 Перечень вопросов, подлежащих разработке:
  - 3.1 Произвести анализ предметной области: изучить основные принципы составления упражнений для клавиатурного тренажёра \_\_\_\_\_
  - 3.2 Выполнить обзор существующих систем-аналогов \_\_\_\_\_
  - 3.3 Разработать информационно-логический проект системы по методологии UML \_\_\_\_\_
  - 3.4 Разработать и реализовать программное и информационное обеспечение, провести его тестирование и отладку \_\_\_\_\_
  - 3.5 Оформить документацию курсовой работы \_\_\_\_\_
  - 3.6 Подготовить презентацию по разработанной системе \_\_\_\_\_
- 4 Перечень графических разработок:
  - 4.1 Структурная схема системы \_\_\_\_\_
  - 4.2 Канонические диаграммы UML \_\_\_\_\_
  - 4.3 Схемы основных алгоритмов \_\_\_\_\_

## 5 Календарный план выполнения работ

№ п/ п	Содержание работы по этапам	Объем этапа в % к общему объему проекта	Срок окончания	Фактическое выполнение
1	Оформление технического задания и его утверждение	5	17.09.2024	
2	Описание и анализ предметной области	10	24.09.2024	
3	Проектирование системы	40	13.12.2024	
3.1	Разработка структурной схемы системы	5	8.10.2024	
3.2	Разработка функциональной спецификации системы и прототипа интерфейса пользователя	10	25.10.2024	
3.3	Разработка информационно-логического проекта системы и его предъявление руководителю	25	13.12.2024	
4	Реализация проекта, разработка контрольных примеров. Предъявление реализации руководителю	40	13.12.2024	
5	Корректировка проекта и оформление документации проекта. Защита проекта с представлением презентации.	5	27.12.2024	

Задание принял  
к исполнению

\_\_\_\_\_ С.А. Атякшев  
 \_\_\_\_\_ Д.О. Колбанов  
 \_\_\_\_\_ Р.К. Лукашевич

## ПРИЛОЖЕНИЕ

к заданию на курсовую работу  
обучающимся в группе № 6401-020302D

С.А. Атякшеву

Д.О. Колбанову

Р.К. Лукашевичу

Тема проекта: «Клавиатурный тренажёр с функциями администратора»

Исходные данные к проекту:

1 Характеристика объекта автоматизации:

1) объект автоматизации: клавиатурный тренажёр;

2) виды автоматизируемой деятельности:

- процесс авторизации/регистрации пользователей;
- процесс ручного составления упражнения;
- процесс генерации упражнения;
- процесс выполнения упражнения;
- процесс настройки уровня сложности;
- процесс визуализации работы клавиатуры;

3) количество ролей пользователей – 2;

4) минимальная длина логина – 4 символа;

5) максимальная длина логина – 8 символов;

6) минимальная длина пароля – 4 символа;

7) максимальная длина пароля – 10 символов;

8) минимальное количество знаков в упражнении – 25;

9) максимальное количество знаков в упражнении – 200;

10) минимальное время нажатия на клавишу – 0,3 с;

11) максимальное время нажатия на клавишу – 1,5 с;

12) максимальное количество допустимых ошибок – 5;

13) количество уровней сложности – 5;

14) количество зон клавиатуры – 5;

15) количество способов создания упражнения – 2;

- 16) количество видов статистики – 2;
- 17) количество видов отображения статистики – 3.
- 2 Требования к информационному обеспечению:
- 1) информационное обеспечение разрабатывается на основе следующих источников:
    - клавиатурный тренажёр [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/Keyboard\\_trainer](https://ru.wikipedia.org/wiki/Keyboard_trainer) (дата обращения: 14.09.2024);
  - 2) звуковые эффекты хранятся в файлах формата wav;
  - 3) структура базы данных (БД) разрабатывается на основании следующих сведений:
    - об игроке (логин, пароль, роль, статистика);
    - об упражнении (номер, текст упражнения, уровень сложности);
    - о статистике (среднее время нажатий, количество пройденных уровней);
    - об уровне сложности (номер уровня, зоны клавиатуры, время нажатия на клавишу, количество допустимых ошибок, минимальное количество знаков, максимальное количество знаков).
  - 4) должна быть обеспечена целостность базы данных и защита от несанкционированного доступа.
- 3 Требования к техническому обеспечению:
- 3.1 Требования к техническому обеспечению серверной части:
- 1) тип ЭВМ – IBM PC совместимый;
  - 2) объем ОЗУ – не менее 2 Гб;
  - 3) объем свободного пространства на внешнем диске – не менее 50 Гб;
  - 4) наличие подключения к сети Интернет;
  - 5) манипулятор – мышь;
  - 6) технические характеристики определяются в процессе выполнения проекта.

### 3.2 Требования к техническому обеспечению клиентской части:

- 1) тип ЭВМ – IBM PC совместимый;
- 1) монитор с разрешающей способностью не ниже 800 x 600;
- 2) манипуляторы – клавиатура и мышь;
- 3) технические характеристики определяются в процессе выполнения проекта.

## 4 Требования к программному обеспечению:

### 4.1 Требования к программному обеспечению серверной части:

- 1) тип операционной системы – Windows 7 и выше;
- 2) СУБД – PostgreSQL 17.

### 4.2 Требования к программному обеспечению клиентской части:

- 1) тип операционной системы – Windows 7 и выше;
- 2) браузер – Google Chrome 86.0.4240.183 (64-битный) и выше, Firefox 83.0 (64-битный) и выше.

### 4.3 Требования к программному обеспечению рабочего места разработчика:

- 1) тип операционной системы – Windows 7 и выше;
- 2) язык программирования – TypeScript, Rust HTML, CSS;
- 3) среда программирования – Visual Studio Code;
- 4) СУБД – PostgreSQL 17;
- 5) фреймворк – React, Axum;
- 6) среда проектирования – StarUML 6.2.2.

## 5 Общие требования к проектируемой системе:

### 5.1 Функции, реализуемые системой:

- 1) функции системы:
  - аутентификация пользователя в системе, настройка интерфейса пользователя на заданную роль;
  - автоматическое составление текста упражнения по заданным параметрам;
  - формирование списка упражнений для заданного уровня

сложности;

- генерация длины упражнения;
- вычисление средней скорости набора;
- формирование статистики;
- контроль набора введённых символов;
- визуализация процесса работы с клавиатурой;
- контроль количества допущенных ошибок;
- выдача сообщений пользователю;
- выдача справочной информации о системе;

2) функции администратора:

- авторизация пользователя в системе (ввод логина и пароля);
- настройка уровня сложности:
  - 1) ввод номера уровня сложности;
  - 2) задание минимального количества символов;
  - 3) задание максимального количества символов;
  - 4) выбор зон клавиатуры;
  - 5) задание количества допустимых ошибок;
- создание/редактирование упражнения:
  - 1) выбор уровня сложности;
  - 2) выбор способа создания упражнения;
  - 3) задание длины упражнения (для автоматического режима создания упражнения);
  - 4) ручной ввод текста упражнения;
- сохранение упражнения в БД;
- загрузка упражнения из БД;
- просмотр справочной информации;
- работа со статистикой:
  - 1) по пользователям;
  - 2) по упражнениям;

3) функции игрока:

- регистрация пользователя в системе (ввод логина, пароля, имени);
- авторизация пользователя в системе (ввод логина и пароля);
- выбор уровня сложности;
- выбор упражнения из списка;
- выполнение упражнения;
- просмотр собственной статистики;
- включение/выключение виртуальной клавиатуры;
- включение/выключение звуковых эффектов;
- просмотр справочной информации.

5.2 Технические требования к системе:

- 1) режим работы – диалоговый;
- 2) время автоматической генерации упражнения – не более 10 с;
- 3) система должна удовлетворять санитарным правилам и нормам СанПин 2.2.2./2.4.2198-07;
- 4) условия работы средств вычислительной техники (содержание вредных веществ, пыли и подвижность воздуха) должны соответствовать ГОСТ 12.1.005, 12.01.007;
- 5) температура окружающего воздуха – 15-35°C;
- 6) влажность воздуха – 45-75%.

Руководитель

проекта \_\_\_\_\_ Л.С. Зеленко

Задание принял

к исполнению \_\_\_\_\_ С.А. Атякшев

\_\_\_\_\_ Д.О. Колбанов

\_\_\_\_\_ Р.К. Лукашевич



## РЕФЕРАТ

Пояснительная записка 40 с, 14 рисунков, 5 таблиц<sup>1</sup>, 12 источников, 2 приложения.

Графическая часть: ??? слайдов презентации PowerPoint.

ДЕРЕВО ПОИСКА, ГЕНЕРАТОР КРОССВОРДОВ, ГОЛОВОЛОМКА, СЛОВАРЬ ТЕРМИНОВ, ВАРИАНТ ОТОБРАЖЕНИЯ, РАЗГАДЫВАНИЕ

Объектом автоматизации является клавиатурный тренажёр.

Во время курсового проектирования разработаны алгоритмы и соответствующая им программа, позволяющая выполнять автоматическую генерацию линейного кроссворда по заданной теме. Задания (понятие и его расшифровка) хранятся в текстовом файле и могут дополняться вручную внутри программы, при этом ограничений на длину словаря не существует. Тема кроссворда выбирается пользователем в соответствии с содержанием словаря заданий. Программа позволяет сформировать кроссворд, учитывая ограничения на параметры. В системе имеется возможность сохранения кроссвордов в файл с целью последующего их разгадывания.

Программа написана на языке C# в среде Visual Studio 2015 и функционирует под управлением операционной системы Windows 7 и выше. Доступ к данным осуществляется с помощью СУБД PostgreSQL 17.

---

<sup>1</sup> Количество страниц, рисунков, таблиц указывается с учетом приложений

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	12
1 Описание и анализ предметной области.....	14
1.1 Описание предметной области .....	14
1.2 Описание систем-аналогов.....	19
1.2.1 Stamina-online .....	19
1.2.2 Klavarog .....	21
1.3 Диаграмма объектов предметной области .....	23
1.4 Постановка задачи .....	24
2 Проектирование системы .....	28
2.1 Выбор и обоснование архитектуры системы .....	28
2.1.1 Виды архитектур .....	28
2.1.2 Клиент-серверные архитектуры .....	30
2.1.3 Типы клиентов.....	33
2.1.4 Протоколы передачи данных .....	34
2.2 Структурная схема системы .....	35
2.3 Разработка спецификации требований .....	38
2.3.5 Функциональная спецификация .....	38
2.3.6 Перечень исключительных ситуаций .....	38
2.4 Разработка прототипа интерфейса пользователя системы .....	42
2.5 Разработка информационно-логического проекта системы.....	43
2.5.7 Язык UML .....	45
2.5.8 Диаграмма вариантов использования .....	45
2.5.9 Сценарии .....	45
2.5.10 Диаграмма классов.....	47
2.5.11 Диаграмма состояний .....	47
2.5.12 Диаграмма деятельности .....	47
2.5.13 Диаграмма последовательности .....	48
2.6 Логическая модель данных (при необходимости).....	49

2.7	Выбор и обоснование алгоритмов обработки данных /Разработка и описание алгоритмов обработки данных .....	50
2.8	Выбор и обоснование комплекса программных средств.....	53
2.8.1	Выбор языка программирования .....	53
2.8.2	Выбор среды программирования .....	54
2.8.3	Выбор операционной системы .....	54
2.8.4	Выбор системы управления базами данных (при необходимости)	54
3	Реализация системы .....	55
3.1	Разработка и описание интерфейса пользователя .....	55
3.2	Диаграммы реализации .....	56
3.2.1	Диаграмма компонентов .....	56
3.2.2	Диаграмма развертывания.....	57
3.2.3	Диаграмма классов.....	57
3.3	Физическая модель данных (при необходимости) .....	58
3.4	Выбор и обоснование комплекса технических средств.....	59
3.4.1	Расчет объема занимаемой памяти.....	59
3.4.2	Минимальные требования, предъявляемые к системе.....	61
	ЗАКЛЮЧЕНИЕ .....	62
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	63
	ПРИЛОЖЕНИЕ А Руководство пользователя.....	66
A.1	Назначение системы .....	66
A.2	Условия работы системы .....	66
A.3	Установка системы .....	66
A.4	Работа с системой .....	67
A.4.1	Работа с системой в режиме администратора (если необходимо). ..	67
A.4.2	Работа с системой в режиме пользователя .....	67
	ПРИЛОЖЕНИЕ Б Листинг модулей программы .....	68

## ВВЕДЕНИЕ

Клавиатурный тренажер – это инструмент, который помогает пользователям улучшить скорость и точность печати на клавиатуре. Несмотря на то, что сегодня клавиатура стала неотъемлемой частью нашей жизни, её история насчитывает не один десяток лет.

Первые попытки создать системы обучения печати относятся к концу 19-го века. В 1888 году Франк Эдвард МакГуррин издал книгу "Touch Typewriting", которая стала первым учебником по слепой печати.

В начале 20-го века появились механические клавиатурные тренажеры, которые использовали систему с "закрепленными" клавишами, чтобы научить пользователей правильно позиционировать пальцы.

С развитием компьютеров появились электронные клавиатурные тренажеры, которые могли отслеживать скорость и точность печати, а также предлагать упражнения и тесты. Mavis Beacon Teaches Typing – одна из самых известных ранних программ для обучения набору текста. Она предлагала пользователям интерактивные уроки, отслеживала прогресс и давала рекомендации по улучшению техники. Это была одна из первых программ, которая использовала игровой подход, превращая обучение в интересный процесс. В конце 1980-х и 1990-х годах стали популярны обучающие программы, которые использовались в образовательных учреждениях для тренировки печати у школьников и студентов [1].

Во время курсового проектирования необходимо разработать клавиатурный тренажёр, с помощью которого можно будет выполнять упражнения для тренировки слепой печати, также с помощью данной системы можно будет конструировать упражнения в ручном или автоматическом режиме в соответствии с заданным уровнем сложности.

Разработка системы будет производиться по технологии быстрой разработки приложений RAD (Rapid Application Development), которая поддерживается методологией структурного проектирования и включает

элементы объектно-ориентированного проектирования и анализа предметной области [2].

При проектировании системы будут использоваться методология ООАП (Object-Oriented Analysis/Design), в основу которой положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов, и язык моделирования UML (Unified Modeling Language), который является стандартным инструментом для разработки «чертежей» программного обеспечения [3].

## 1 Описание и анализ предметной области

Под предметной областью (application domain) принято понимать ту часть реального мира, которая имеет существенное значение или непосредственное отношение к процессу функционирования программы. Другими словами, предметная область включает в себя только те объекты и взаимосвязи между ними, которые необходимы для описания требований и условий решения некоторой задачи [4].

### 1.1 Описание предметной области

Клавиатурный тренажёр – вид компьютерных программ или онлайн-сервисов, предназначенных для обучения набору на компьютерной клавиатуре [?].

Компьютерная клавиатура – устройство ввода. Представляет собой набор клавиш (кнопок), расположенных в определённом порядке. Бывает совмещённой в одно устройство с тачпадом, тензометрическим джойстиком или трекболом. На рисунке 1 представлен пример компьютерной клавиатуры.



Рисунок 1 – Пример компьютерной клавиатуры

Обычно целями тренажёров являются:



кликовые, каждый из которых имеет свои уникальные характеристики и ощущения при нажатии клавиш. Также от них зависит громкость нажатия [5]. На рисунке 4 представлен пример механической клавиатуры.



Рисунок 4 – Пример механической клавиатуры

Также существуют несколько раскладок клавиатуры. Раскладка клавиатуры – соглашение о соответствии типографических символов (букв, цифр, знаков препинания и т. д.) письменного языка клавишам клавиатуры компьютера, пишущей машинки или другого устройства, с помощью которого вводится текст [5]. Среди раскладок для английского языка выделяют QWERTY и Colemak. При создании раскладки QWERTY требовалось, чтобы буквы, образующие в английском языке устойчивые комбинации, располагались как можно дальше друг от друга по разные стороны клавиатуры и были разбросаны по разным рядам, что уменьшало вероятность «перепутывания» рычажков пишущей машинки. В 2006 году Шаем Коулманом (Shai Coleman) была разработана раскладка Colemak. Название происходит от Coleman+Dvorak. Раскладка приспособлена к современным компьютерным реалиям. Её принцип — эффективный и эргономичный набор текстов на английском языке на компьютерной клавиатуре. Примеры раскладок QWERTY и Colemak представлены на рисунках 5 и 6 соответственно.









~ 1	! 2	@ 3	# 4	\$ 5	% 6	^ 7	& 8	* 9	( 0	) -	+ =	 Backspace	
Tab 	Q	W	E	R	T	Y	U	I	O	P	{ [	} ]	 \
Caps Lock 	A	S	D	F	G	H	J	K	L	:	" '	Enter 	
Shift 	Z	X	C	V	B	N	M	< ,	> .	? /	Shift 		
Ctrl	Win Key	Alt							Alt	Win Key	Menu	Ctrl	

Рисунок 5 – Пример раскладки QWERTY



Рисунок 6 – Пример раскладки Colemak

Среди русских раскладок клавиатуры выделяют ЙЦУКЕН и «фонетическая раскладку». Более распространённой из них является раскладка ЙЦУКЕН, название которой происходит от шести левых символов верхнего ряда раскладки. Считается, что раскладка ЙЦУКЕН далека от оптимальной при печати слепым десятипальцевым методом [6]:

- нагрузка по пальцам распределяется неравномерно, из-за чего одни пальцы «пробегают» по клавиатуре большие расстояния, чем другие;
- приходится часто гнуть пальцы, так как основной «домашний» ряд клавиатуры мало задействован;
- часто два и более символа нажимаются подряд в одной зоне, одним пальцем.

Пример раскладки ЙЦУКЕН представлен на рисунке 7.

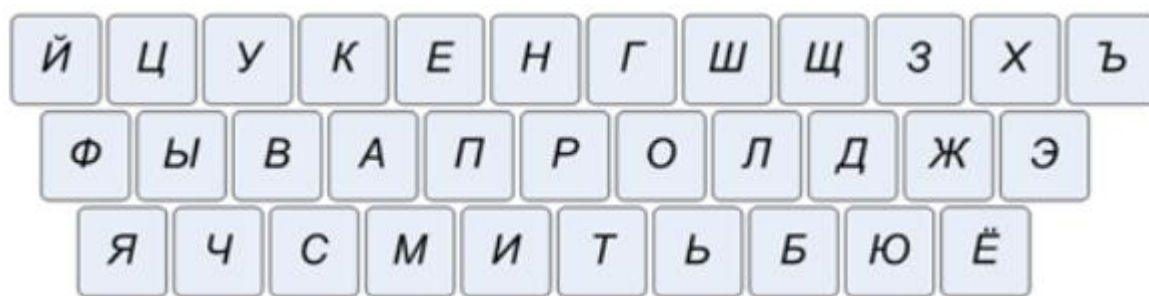


Рисунок 7 – Пример раскладки ЙЦУКЕН

Чтобы быстро выполнить запись текстового материала следует овладеть слепым методом печати, когда каждой клавише с буквой или цифрой соответствует свой палец. На рисунке 8 представлены зоны клавиатуры для каждого пальца.

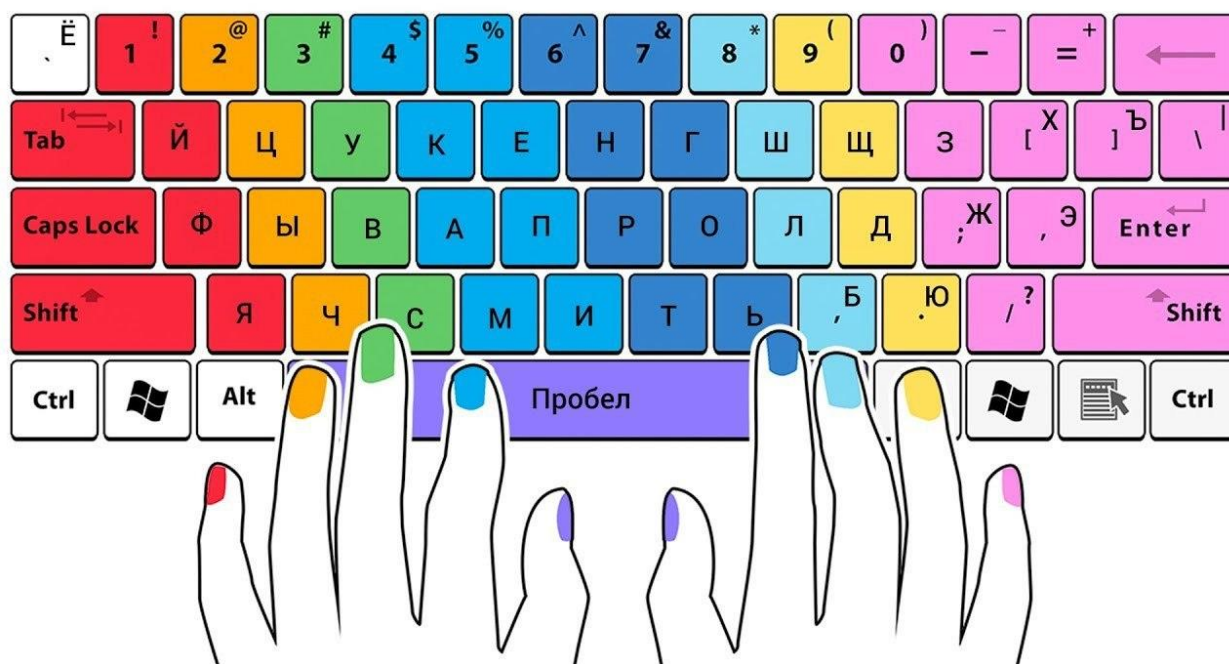


Рисунок 8 – Зоны клавиатуры

Такое расположение букв на клавиатуре закономерно. Те буквы, которые используются чаще всего в текстах, сосредоточены в центре, и за них отвечают указательные пальцы – самые подвижные и натренированные. Редкие буквы и клавиши, находящиеся рядом с буквенной клавиатурой, отводятся мизинцам [7].

В таблице 1 представлены символы разных зон клавиатуры.

Таблица 1 – Зоны клавиатуры

Зоны клавиатуры	Набор символов
Синяя, темно-синяя	4, 5, 6, 7, к, е, н, г, а, п, р, о, м, и, т, ь
Зеленая, голубая	3, 8, у, ш, в, л, с, б
Оранжевая, желтая	2, 9, ц, щ, ы, д, ч, ю
Красная, розовая	1, 0, й, з, х, ь, ф, ж, э, я
Фиолетовая	Пробел

Методика составления упражнений соответствует принципам методики В.В. Шахиджаняна. Методика Шахиджаняна является одной из самых эффективных методик обучения слепой печати. Она помогает научиться печатать быстро и точно, а также развивает координацию пальцев и увеличивает скорость реакции. Основные принципы методики Шахиджаняна [8]:

- постепенное усложнение: упражнения становятся все более сложными по мере освоения навыков;
- повторение: упражнения повторяются несколько раз, чтобы закрепить навыки;
- индивидуальный подход: методика предусматривает возможность изменения упражнений в зависимости от индивидуальных особенностей обучающегося.

## 1.2 Описание систем-аналогов

В интернете существуют большое количество версий клавиатурного тренажёра. Все они разные имеют свои особенности и функционал. Рассмотрим основные характеристики нескольких систем-аналогов.

### 1.2.1 Stamina-online

Тренажер Stamina-online помогает освоить десятипальцевую слепую машинопись не только традиционным методом расположения пальцев на клавишах ASDF, но и альтернативным – когда пальцы лежат на нижнем ряду

клавиатуры SDFV [8]. По мнению разработчика, в таком положении кисти рук меньше устают. На сегодняшний день это единственный тренажер, который позволяет отработать ввод из двух разных исходных позиций. Из практических занятий доступны уроки для запоминания расположения клавиш и работа с текстами/фразами, предоставляемыми как программой, так и из любого внешнего файла. Интересным образом выполнена виртуальная клавиатура: она не только подсвечивает нужные буквы, но и примерную область работы каждого пальца. Своеобразной «фишкой» программы является легкий и юморной язык, которым написаны FAQ и разделы помощи для пользователя: полезная информация разбавляется анекдотами и интересными логическими задачами. На рисунке 4 приведена страница сайта «Stamina-online».



Рисунок 9 – Страница сайта «Stamina-online»

К достоинствам данной системы относятся:

- бесплатный доступ;
- понятный интерфейс;
- наличие обучения из альтернативного варианта расположения пальцев;
- отображение используемых пальцев;
- наличие теории;
- возможность тренироваться на языках программирования;
- визуализация клавиатуры.

К недостаткам системы относятся:

- большая нагрузка на руки с первых занятий;
- большое количество рекламы;
- отсутствие визуализации статистики.

### 1.2.2 Klavarog

Клавиатурный тренажёр «Klavarog» удобен тем, что предлагает несколько уровней сложности и отдельную категорию упражнений на скорость: на «начальном» можно освоить первичные навыки десятипальцевой печати классическим методом и только потом продвигаться дальше в удобном для себя темпе [9]. Из словаря подбираются слова, которые нужно набрать без ошибок, но в случае неоднократных опечаток тренажер предлагает ввести проблемные слова заново. На сайте есть таймер, но его основная задача – это организация отдыха и перерывов, а не «наказание» пользователя за несоблюдение минимума набранных знаков в минуту. В разделе статистики собирается информация о всех сессиях, скорости (по количеству символов в минуту и реальному числу введенных слов) и ошибках. А еще у сайта есть приятная опция «Дзен», которую можно включить нажатием на логотип – в этом режиме можно просто расслабиться во время печати 10.

К достоинствам данной системы относятся:

- бесплатный доступ;
- визуализация клавиатуры;
- возможность тренироваться на языках программирования.

К недостаткам системы относятся:

- отсутствие обновлений;
- отсутствие звукового сопровождения;
- отсутствие уровней сложности;
- устаревший интерфейс;
- отсутствие статистики.



Рисунок 10 – Страница сайта «Klavarog»

На основании анализа возможностей систем-аналогов были сформулированы требования к разрабатываемой системе (см. таблицу 2).

Таблица 2 – Сравнительные характеристики систем-аналогов

Название системы Название показателя	Система «Stamina- online»	Система «Klavarog»	Разрабатываемая система
Бесплатный доступ	+	+	+
Удобство использования	+	-	+
Уровень сложности	+	-	+
Звуковое сопровождение	+	-	+
Визуализация клавиатуры	+	+	+
Визуализация статистики	-	-	+
Упражнения на иностраннные языки	+	+	-
Подсчёт времени	+	+	+

### 1.3 Диаграмма объектов предметной области

В объектном или объектно-ориентированном подходе в первую очередь выделяется множество основных объектов системы и впоследствии определяется множество операций над объектами. Такой подход базируется на абстрактных типах, и решение задачи выражается в терминах выделенных объектов [10].

Объектно-ориентированный анализ и проектирование (ООАП, Object-Oriented Analysis/Design) - технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов [11].

На рисунке 12 приведена диаграмма объектов предметной области.

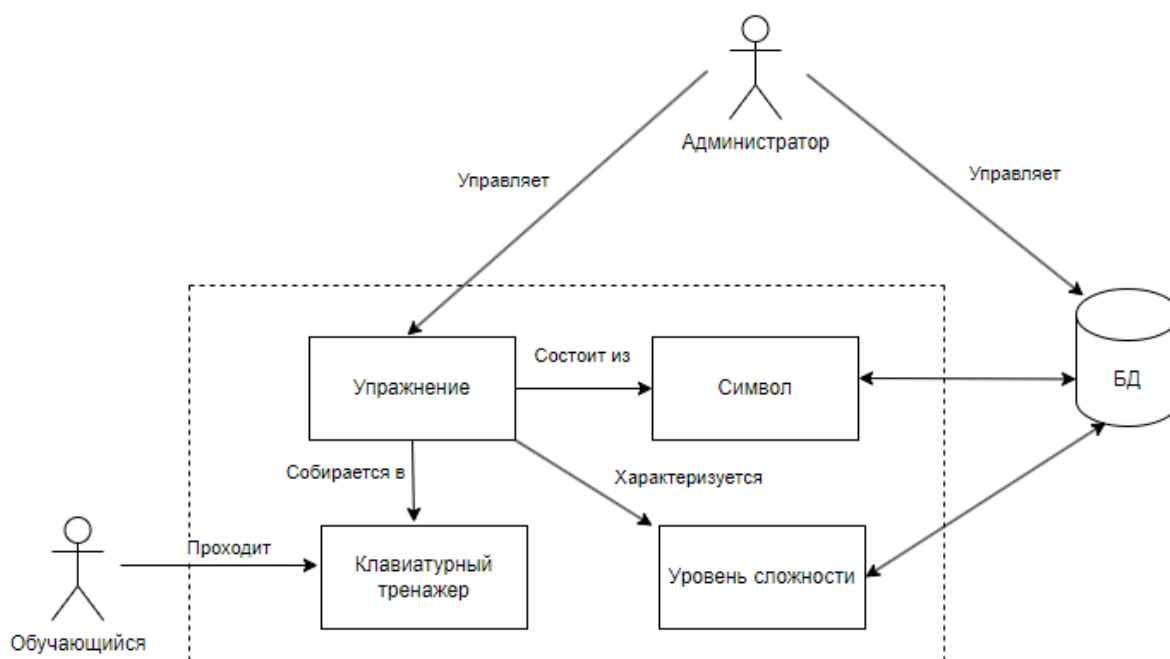


Рисунок 12 – Диаграмма объектов предметной области

Как видно из диаграммы, клавиатурный тренажер представляет собой совокупность упражнений, которые в свою очередь состоят из символов и характеризуются уровнем сложности, который определяет количество

допустимых ошибок, максимальную и минимальную скорость нажатия и количество символов. Упражнения создаются и редактируются администратором, который также управляет базой данных.

#### 1.4 Постановка задачи

Во время курсового проектирования необходимо разработать автоматизированную систему составления и прохождения упражнений для тренировки слепой печати, с помощью которой администратор может конструировать упражнение в ручном или автоматическом режиме в соответствии с заданным уровнем сложности, а обучающийся просматривать статистику по упражнениям. Система должна быть реализована в виде веб-приложения.

В системе должно быть реализовано две роли пользователей: администратор и игрок, поэтому для того, чтобы работать с системой, при первом входе в систему пользователи должны пройти процедуру регистрации: ввести логин и пароль (длина логина должна быть от 4 до 8 символов, длина пароля – от 4 до 10 символов). При повторном входе, пользователю необходимо авторизоваться в системе и затем, после аутентификации, система должна настроить интерфейс пользователя на заданную роль.

##### 1.4.1 Режим администратора

Администратору в первую очередь должна быть доступна функция настройка уровня сложности, в системе должно быть реализовано 5 уровней сложности. При настройке уровня сложности администратор должен задать следующие параметры: минимальное (от 25) и максимальное (до 200) количество символов, выбрать зоны клавиатуры (от 1 до 5) и максимальное количество допустимых ошибок (до 5).

Также администратору должна быть доступна функция создания и редактирования упражнения. Администратор должен иметь возможность создавать и редактировать упражнения, а именно: выбирать уровень



сложности, выбирать способ создания упражнения (ручной или автоматический), вручную вводить текст упражнения (от 25 до 200 символов), задавать длину упражнения (для автоматического режима создания упражнения). В автоматическом режиме система должна автоматически сгенерировать текст упражнения в соответствии с заданными параметрами. При ручном вводе система должна контролировать набор символов и длину упражнения.

Администратор должен иметь возможность работать со статистикой (по пользователям и по упражнениям). Статистика по пользователям должна отображаться в виде таблицы, по упражнениям – в виде столбиковой диаграммы.

#### 1.4.2 Режим обучающегося

Основная функция обучающегося – это выполнение упражнений. Обучающийся сначала должен выбрать уровень сложности, система должна сформировать список доступных упражнений, из которых пользователь должен выбрать интересные ему.

Перед началом выполнения упражнения пользователь должен иметь возможность включить или отключить виртуальную клавиатуру, а также настроить звуковые эффекты. При выполнении упражнений система должна контролировать количество допущенных ошибок, время нажатия клавиши и в случае превышения заданных параметров останавливать выполнение упражнения с выдачей соответствующего уведомления. Во время набора система должна рассчитывать среднюю скорость набора.

У обучающегося должна быть возможность посмотреть собственную статистику по всем упражнениям.

Вся информация, необходимая для работы системы будет храниться в базе данных, структура базы данных определяется на основании следующих сведений:

- об игроке (логин, пароль, роль, статистика);

- об упражнении (номер, текст упражнения, уровень сложности);
- о статистике (среднее время нажатий, количество пройденных уровней);
- об уровне сложности (номер уровня, зоны клавиатуры, время нажатия на клавишу, количество допустимых ошибок, минимальное количество знаков, максимальное количество знаков).

В системе должна быть обеспечена целостность базы данных и защита от несанкционированного доступа.

В системе также должна быть обеспечена возможность получения справочной информации как о самой системе, так и предоставляемых ею возможностях.

Таким образом, система должна решать следующие задачи:

1) функции системы:

- аутентификация пользователя в системе, настройка интерфейса пользователя на заданную роль;
- автоматическое составление текста упражнения по заданным параметрам;
- формирование списка упражнений для заданного уровня сложности;
- генерация длины упражнения;
- вычисление средней скорости набора;
- формирование статистики;
- контроль набора введённых символов;
- визуализация процесса работы с клавиатурой;
- контроль количества допущенных ошибок;
- выдача сообщений пользователю;
- выдача справочной информации о системе;

2) функции администратора:

- авторизация пользователя в системе (ввод логина и пароля);

- настройка уровня сложности:
  - 1) ввод номера уровня сложности;
  - 2) задание минимального количества символов;
  - 3) задание максимального количества символов;
  - 4) выбор зон клавиатуры;
  - 5) задание количества допустимых ошибок;
- создание/редактирование упражнения:
  - 1) выбор уровня сложности;
  - 2) выбор способа создания упражнения;
  - 3) задание длины упражнения (для автоматического режима создания упражнения);
  - 4) ручной ввод текста упражнения;
- сохранение упражнения в БД;
- загрузка упражнения из БД;
- просмотр справочной информации;
- работа со статистикой:
  - 1) по пользователям;
  - 2) по упражнениям;
- 3) функции обучающегося:
  - регистрация пользователя в системе (ввод логина, пароля, имени);
  - авторизация пользователя в системе (ввод логина и пароля);
  - выбор уровня сложности;
  - выбор упражнения из списка;
  - выполнение упражнения;
  - просмотр собственной статистики;
  - включение/выключение виртуальной клавиатуры;
  - включение/выключение звуковых эффектов;
  - просмотр справочной информации.

## 2 Проектирование системы

Проектирование – это глубокое погружение в суть будущей системы. Оно позволяет увидеть целостную картину, рассмотреть возможные риски и заранее планировать решения. Проектирование не только делает систему более функциональной, но и делает ее развитие более плавным и предсказуемым. Это вложение времени и сил оправдывает себя на всех этапах жизненного цикла системы, от разработки до поддержки.

Проектирование основано на глубоком понимании требований и целей заказчика. Это позволяет создать систему, которая будет идеально отвечать его потребностям. Проектирование также учитывает текущие и будущие технологические тренды, чтобы система оставалась актуальной и конкурентоспособной. Проектирование упрощает коммуникацию между разработчиками, заказчиками и другими участниками проекта. Это позволяет всем участникам работать в едином векторе, уменьшая количество недоразумений и ошибок.

Проектирование системы — это инвестиция в успех проекта. Хорошо продуманная архитектура системы делает ее более надежной, эффективной и масштабируемой. Это упрощает дальнейшую разработку, обслуживание и поддержку системы. В итоге, проектирование системы — это ключ к созданию успешного и долговечного продукта.

### 2.1 Выбор и обоснование архитектуры системы

Архитектура – принципиальная организация системы, воплощенная в её элементах, их взаимоотношениях друг с другом и со средой, а также принципы, направляющие её проектирование и эволюцию [12].

#### 2.1.1 Виды архитектур

За годы развития программного обеспечения (ПО) разработчикам удалось придумать надёжные подходы, чтобы устранить недостатки проектирования без архитектуры [13].

Рассмотрим следующие виды архитектур:

– многослойная архитектура. Многослойная архитектура (n-tier client-server architecture) – архитектура, где процессы представления, обработки и управления данными являются логически отделенными друг от друга процессами (см. рисунок 13). Модель многослойной архитектуры помогает создать гибкое и многократно используемое программное обеспечение. В случае изменений надо их делать лишь в отдельных слоях, а не сразу во всем приложении. Это сулит меньше работы, меньших затрат времени и меньше потенциальных ошибок;

– облачная архитектура. Облачная архитектура — это процесс проектирования, создания, развертывания и управления приложениями специально для облачных сред. Он использует возможности облака, такие как масштабируемость, отказоустойчивость и ресурсы по требованию, для оптимизации производительности, скорости и эффективности приложений [14]. При традиционной разработке приложений ПО часто проектируется для работы на определенном оборудовании или виртуальных машинах



Рисунок 13 – Многослойная архитектура

Однако облачная архитектура делает упор на создание платформо-независимых приложений, способных беспрепятственно работать и масштабироваться на нескольких облачных платформах. Используя специализированные облачные сервисы, облачные приложения могут эффективно адаптироваться к изменениям рабочих нагрузок и упростить процесс развертывания.

Пример схемы облачной архитектуры приведен на рисунке 14.

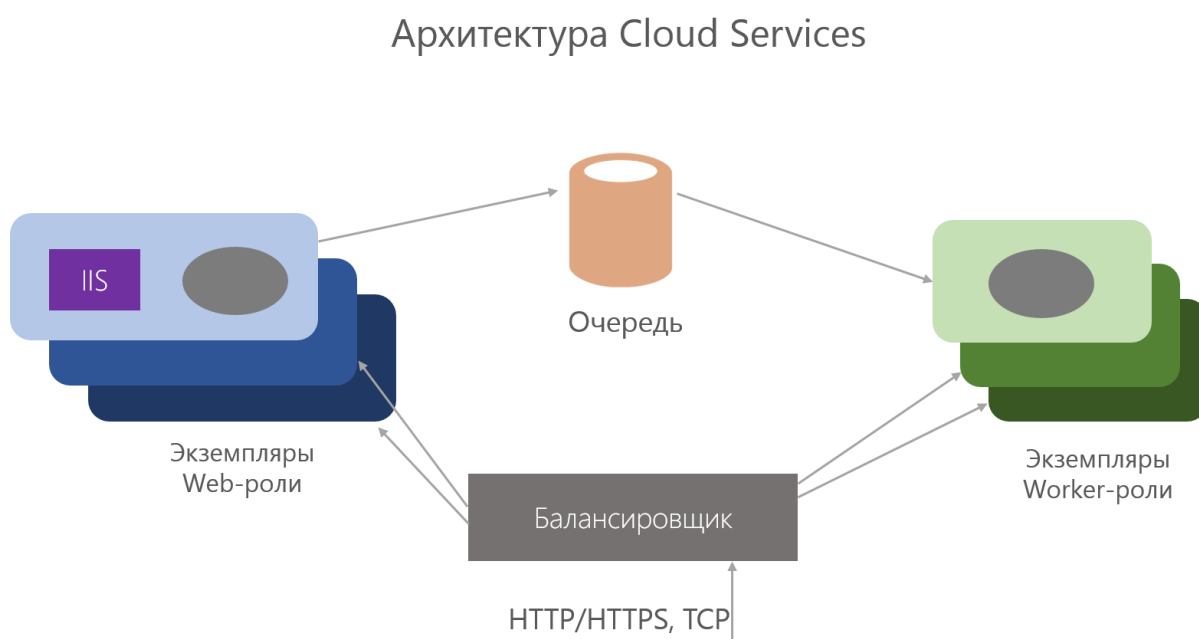


Рисунок 14 – Облачная архитектура

### 2.1.2 Клиент-серверные архитектуры

Клиент-серверная архитектура – это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами [15].

Схема клиент-серверной архитектуры представлена на рисунке 15.

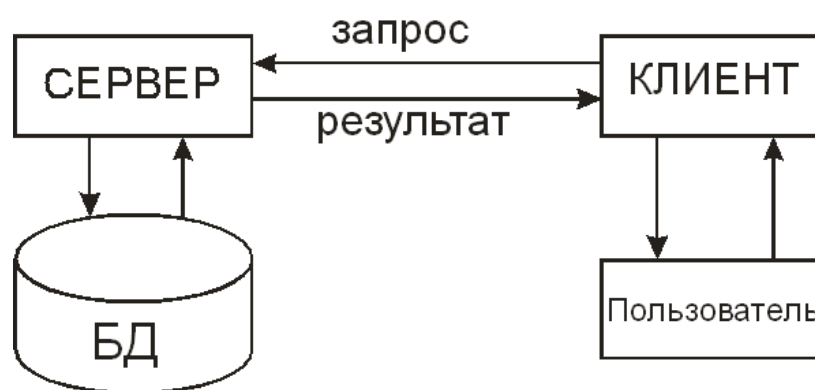


Рисунок 15 – Схема клиент-серверной архитектуры

Самыми используемыми являются следующие типы клиент-серверных архитектур [16]:

- одноуровневая архитектура «клиент-сервер» (см. рисунок 16). В этом типе архитектуры все компоненты, включая представление, бизнес-логику и работу с данными, находятся внутри одной программы или на одном уровне. Это означает, что все операции и логика происходят в пределах одного программного пакета;

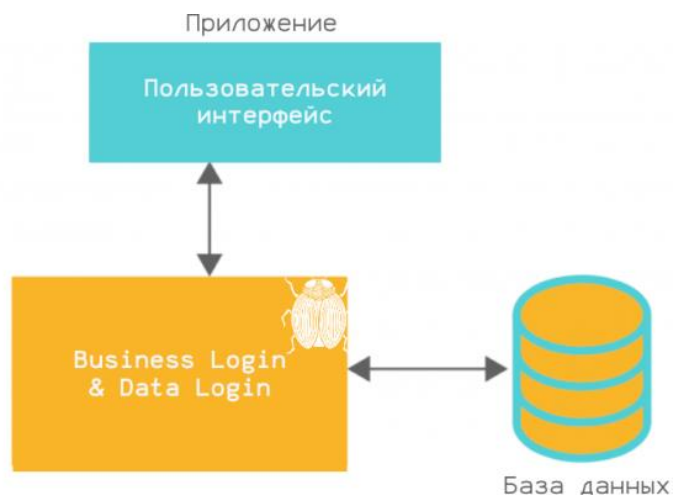


Рисунок 16 – Одноуровневая архитектура

- двухуровневая архитектура (см. рисунок 17). В клиент-серверной архитектуре данного типа логика приложения разделена на два основных слоя. База данных действует как отдельная часть системы. Сначала создаются и настраиваются базы данных. Затем основное приложение включает в себя всю логику, связанную с пользовательским интерфейсом, бизнес-процессами и

взаимодействием с базой данных. Такая архитектура позволяет более удобно организовать и управлять приложением;

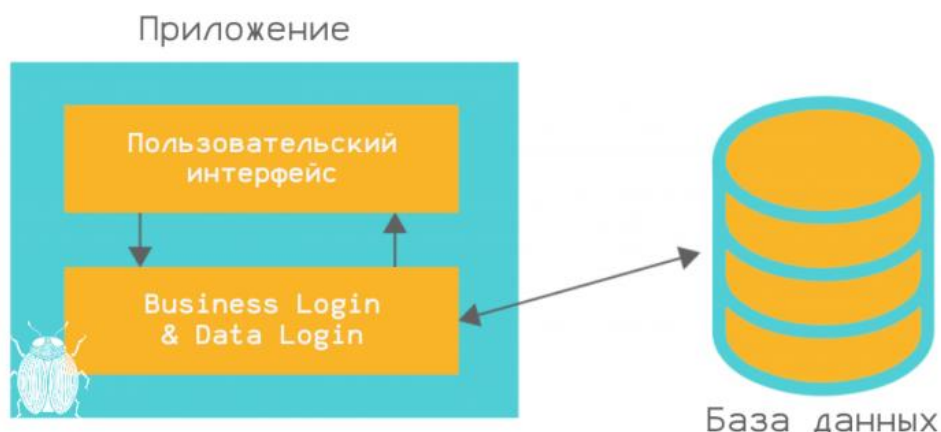


Рисунок 17 – Двухуровневая архитектура

– трёхуровневая архитектура (см. рисунок 18). В отличие от архитектуры двухуровневой системы, в которой отсутствует промежуточное ПО, в трехуровневой системе между клиентом и сервером имеется промежуточное ПО. Когда клиент запрашивает информацию у сервера, запрос сначала поступает на промежуточное ПО. Затем запрос отправляется на сервер для обработки. Аналогичным образом сервер отправляет клиенту ответ.



Рисунок 18 – Трёхуровневая архитектура

– в многоуровневой архитектуре несколько серверов приложений используют результаты работы друг друга, а также данные от различных серверов баз данных, файловых серверов и других видов серверов.

Web-приложение – программное обеспечение, которое работает на веб-сервере. Доступ к нему можно получить по сети, в браузере. Веб-приложение



позволяет пользователям взаимодействовать с данными и функционалом приложения, хранящимися на сервере. Оно создается для того, чтобы решать задачи клиента и задачи бизнеса: например, для управления проектами или онлайн-покупок. Web-приложения интерактивны и используют специальные технологии для обновления данных без перезагрузки страницы [17].

Разрабатываемое приложение будет реализовано на основе двухуровневой клиент-серверной архитектуры.

Преимущества двухуровневой архитектуры:

- легко конфигурировать и модифицировать приложения;
- пользователю обычно легко работать в такой среде;
- хорошая производительность и масштабируемость;

Однако, у двухуровневой архитектуры есть и ограничения:

- производительность может падать при увеличении числа пользователей;
- потенциальные проблемы с безопасностью, поскольку все данные и программы находятся на центральном сервере;
- все клиенты зависимы от базы данных одного производителя.

### 2.1.3 Типы клиентов

В клиент-серверной архитектуре выделяют следующие виды клиентов [18]:

– «толстый» клиент (Fat Client). Приложение, которое выполняет значительную часть обработки данных на стороне клиента. Примером толстого клиента может быть офисное программное обеспечение, такое как Microsoft Word. Толстые клиенты обычно требуют установки на компьютере пользователя и могут работать автономно, без постоянного подключения к серверу;

– «тонкий» клиент (Thin Client). Приложение, которое выполняет минимальную обработку данных на стороне клиента, передавая основную часть задач на сервер. Примером тонкого клиента может быть веб-браузер.

Тонкие клиенты обычно не требуют установки и работают через интернет, что делает их удобными для использования на различных устройствах.

В приложении будет использоваться «тонкий» клиента. Все вычисления будут производиться на стороне сервера.

#### 2.1.4 Протоколы передачи данных

Протокол передачи данных – набор правил, определяющих и регулирующих процедуру обмена данными между подключенными к сети компьютерами. Другими словами, протокол – это язык, на котором компьютеры «общаются» между собой. Только в этом «языке» все должно быть точно определено, а все данные сгруппированы и структурированы. Все сетевые протоколы стандартизованы специализированной международной организацией и одинаковы во всем мире. Рассмотрим несколько самых популярных сегодня протоколов, работающих на разных уровнях [19]:

1) MAC – Media Access Control, низкоуровневый протокол, который используется для идентификации сетевых устройств.

2) IP – Internet Protocol, протокол который объединил различные устройства в единую сеть. Основная его задача – маршрутизация датаграмм. Данный протокол не устанавливает соединение, не подтверждает доставку пакетов получателю, как и не контролирует целостность данных.

3) TCP – Transmission Control Protocol, обеспечивает и контролирует передачу данных и следит за надёжностью и целостностью.

4) UDP – User Datagram Protocol, обеспечивает передачу данных, не создавая соединения. Этот протокол считается недостаточно надёжным, но имеет более высокую скорость доставки данных по сравнению с TCP

5) HTTP – Hyper Text Transfer protocol, обеспечивает передачу гипертекста, на его основе работают веб-сайты. Интернет-протокол HTTP — это шаблон, по которому формируется запрос на передачу данных, а затем передаются интернет-страницы, видео, аудио и текст. Чаще всего с помощью

HTTP передают веб-страницы, то есть контент сайтов, которые отображаются в интернете.

6) FTP – File Transfer Protocol, служит для передачи файлов.

7) SSH – Secure Shell. Протокол, отвечающий за создание удалённого канала для удаленного управления другой операционной системой.

Для реализации взаимодействия между сервером и клиентом будет устанавливаться соединение по протоколу HTTP. Это самый популярный протокол в интернете, он считается основным и универсальным, на нём работают практически все сайты в мире. Также HTTP подробно описан на разных языках, и в документации есть ответы на большинство вопросов.

## 2.2 Структурная схема системы

Система (греч. «составленное из частей», «соединение» от «соединяю») – множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство [20].

Как следует из определения, отличительным (главным свойством) системы является её целостность: комплекс объектов, рассматриваемых в качестве системы, должен обладать общими свойствами и поведением. Очевидно, необходимо рассматривать и связи системы с внешней средой. В самом общем случае понятие «система» характеризуется:

- наличием множества элементов;
- наличием связей между ними;
- целостным характером данного устройства или процесса.

Структурная схема — это совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели [21].

Сущность структурного подхода к разработке системы заключается в её декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая

система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны [22].

На рисунке 19 приведена структурная схема разрабатываемой системы, которая разделяется на клиентскую и серверную части. Взаимодействие между ними осуществляется по HTTP протоколу:

В состав клиентской части входит:

- 1) подсистема ввода, которая включает в себя:
  - подсистему «Администратор», которая состоит из:
    - 1) подсистемы настройки уровня сложности;
    - 2) подсистемы работы с упражнениями;
  - подсистему «Обучающийся», которая состоит из:
    - 1) подсистемы настройки параметров упражнения;
    - 2) подсистемы выполнения упражнений;

2) подсистема вывода, которая отвечает за отображение информации в виде изображения и звуков, в ее состав входит подсистема визуализации, которая отображает текст упражнения, клавиатуру и результаты действия пользователя на экран;

3) справочная подсистема, которая содержит сведения о системе (руководство пользователю) и об её разработчиках;

4) подсистема взаимодействия с сервером, которая осуществляет установку соединения с сервером, формирование и отправку запросов.

В состав серверной части входит:

1) подсистема взаимодействия с клиентской частью системы, которая осуществляет приём данных с клиента и передачу их на обработку;

2) подсистема аутентификации, которая осуществляет проверку введённых пользователем идентификационных данных для подтверждения входа;

3) подсистема генерации упражнения, которая отвечает за создание упражнения в автоматическом режиме;

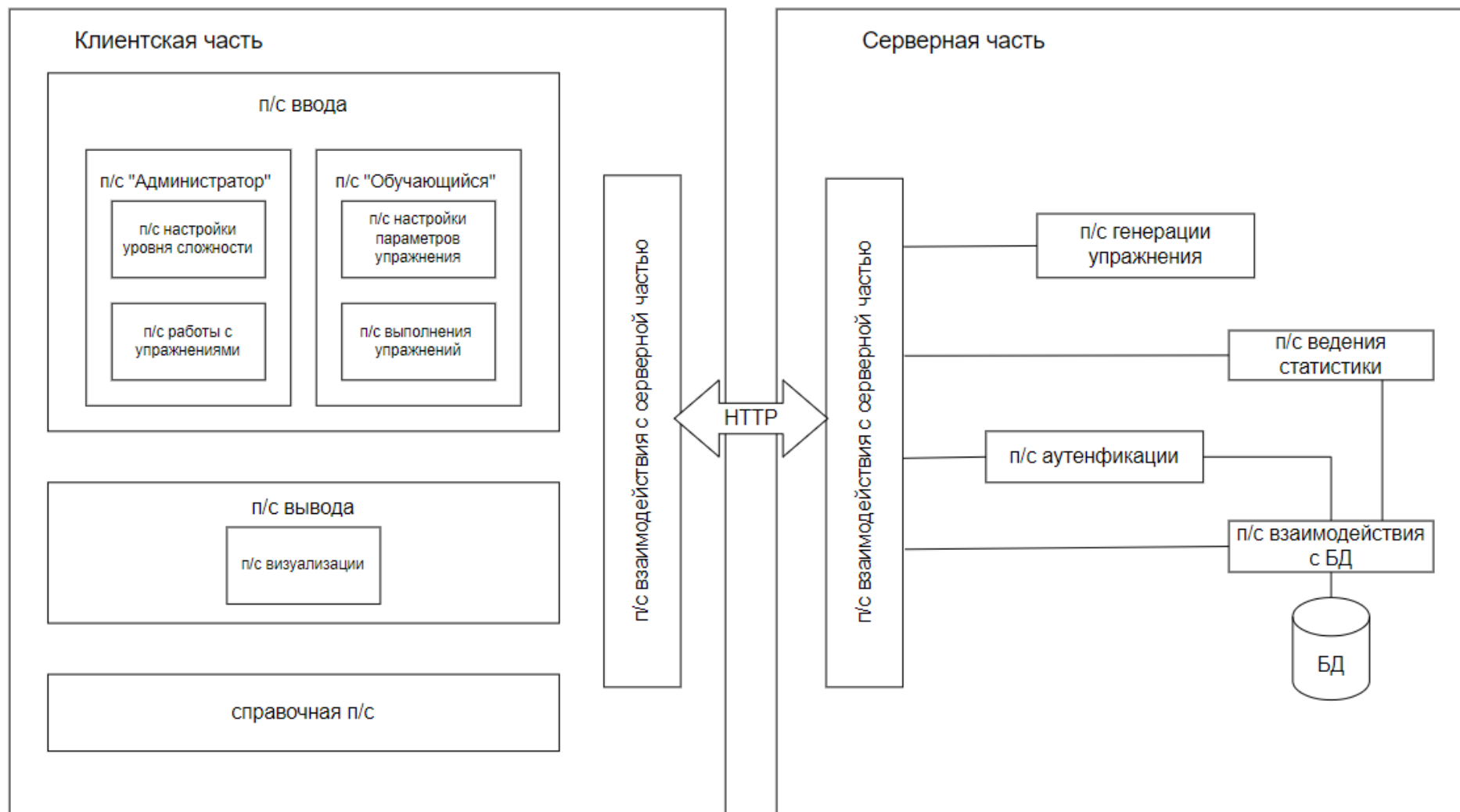


Рисунок 19 – Структурная схема системы

4) подсистема ведения статистики, осуществляющая сбор, обработку и выдачу статистики по упражнениям и пользователям;

5) подсистема взаимодействия с базой данных, которая представляет собой СУБД, позволяющую управлять БД;

6) база данных, которая осуществляет хранение данных о совершённых играх, рейтинге игроков, картинках в галерее.

## 2.3 Разработка спецификации требований

Разработка спецификации программного обеспечения является одним из фундаментальных процессов технологии разработки ПО. Этот процесс анализа, формирования, документирования и проверки функциональных возможностей и ограничений системы называется «разработка требований» (спецификация требований). Он является критическим этапом в создании всех видов программных систем, что обусловлено тем, что ошибки, допущенные на этой стадии, ведут к возникновению серьёзных проблем на этапах проектирования и разработки [23].

Требования – это свойства, которыми должно обладать ПО для адекватного определения функций, условий и ограничений выполнения ПО, а также объёмов данных, технического обеспечения и среды функционирования [24].

### 2.3.1 Функциональная спецификация

Функциональные требования задают «что» система должна делать; нефункциональные – с соблюдением «каких условий» (например, скорость отклика при выполнении заданной операции). При разработке этих требований в первую очередь необходимо учитывать потребности пользователя (заказчика). Пользовательские требования (User Requirements) – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы. Часто пользовательские требования представляют в

виде сценариев (вариантов использования) Use Case [32]. Функциональная спецификация системы приведена в таблице 3.

### 2.3.2 Перечень исключительных ситуаций

Исключительная ситуация – это ситуация, при которой система не может выполнить возложенных на нее функций или которая может привести к денормализации работы системы.

В таблице 4 приведен перечень исключительных ситуаций для разрабатываемой системы и описаны реакции системы на их возникновение.

Таблица 3 – Перечень функций, выполняемых системой

Название подсистемы	Название функции	Информационная среда			
		Входные данные		Выходные данные	
		Назначение (наименование)	Тип, ограничения	Назначение (наименование)	Тип, ограничения
1	2	3	4	5	6
Справочная	Выдать сведения о разработчиках	Сведения о разработчиках системы (ФИО, номер группы)	Текст	Визуальное отображение информации	–
	Выдать сведения о системе	Файл справки	Текстовый (*.HTML)	Код ошибки	Целое
Настройки уровня сложности	Создать и редактировать уровень сложности	Указание номера уровня сложности	Целое	Уровень сложности	Объект «Уровень сложности»
		Зоны клавиатуры	Массив целых чисел		
		Минимальная длина упражнения	Целое		
		Максимальная длина упражнения	Целое		



Продолжение таблица 3

1	2	3	4	5	6
Настройки уровня сложности		Количество допустимых ошибок	Целое	Код ошибки	Целое
	Выбрать уровень сложности	Номер уровня сложности	Целое	Уровень сложности	Объект «Уровень сложности»
				Код ошибки	Целое
Работы с упражнениями	Ручное создание упражнения	Номер уровня сложности	Целое	Упражнение	Объект «Упражнение»
		Текст упражнения	Строка	Код ошибки	Целое
	Автоматическая генерация упражнения	Номер уровня сложности	Целое	Упражнение	Объект «Упражнение»
		Длина упражнения	Целое	Код ошибки	Целое

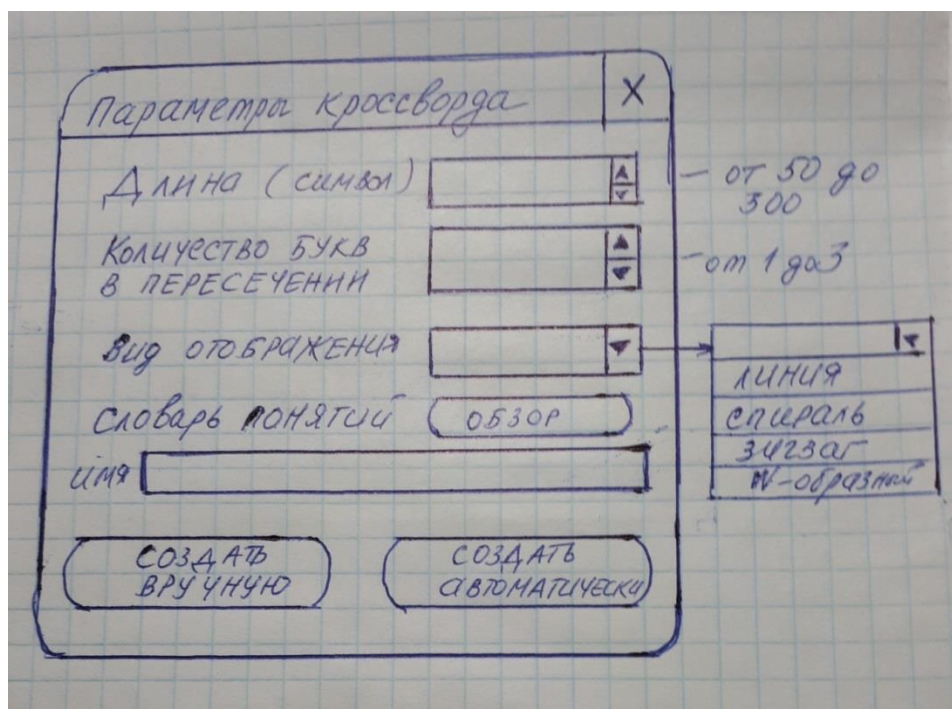
Таблица 4 – Перечень исключительных ситуаций

Название подсистемы	Название исключительной ситуации	Реакция системы
1 Справочная	1.1 Не возможно открыть файл справки	Выдача сообщения «Файл справки поврежден»
	1.2 Не возможно найти файл справки	Выдача сообщения «Отсутствует файл справки»
2 Файловая	2.1 Попытка открытия файла с несобственным форматом	Выдача сообщения «Файл поврежден или недопустимого формата»
...	...	...

#### 2.4 Разработка прототипа интерфейса пользователя системы

Дать определение интерфейса, отметить основные особенности разработки интерфейса.

Здесь должны быть разработаны прототипы **всех** основных форм приложения с описанием привязанной к ней функциональности, например:



### ***Пример.***

На рисунке xxx приведен прототип экранной формы начальной настройки приложения. Здесь пользователь должен выбрать язык программирования, на котором написан алгоритм, категорию (поиск или сортировка) и нажать кнопку «Далее» для перехода к следующему экрану (форме).

Начальная настройка приложения

Выберите язык программирования

Java

Выберите категорию алгоритма

☒ Сортировка

☐ Поиск

Далее

Рисунок xxx – Прототип экранной формы начальной настройки приложения

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

На Рисунок **XXX** приведена навигационная модель разрабатываемого приложения.

## **2.5 Разработка информационно-логического проекта системы**

Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова  
Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова  
Вводные слова Вводные слова Вводные слова Вводные слова Вводные слова  
Вводные слова.

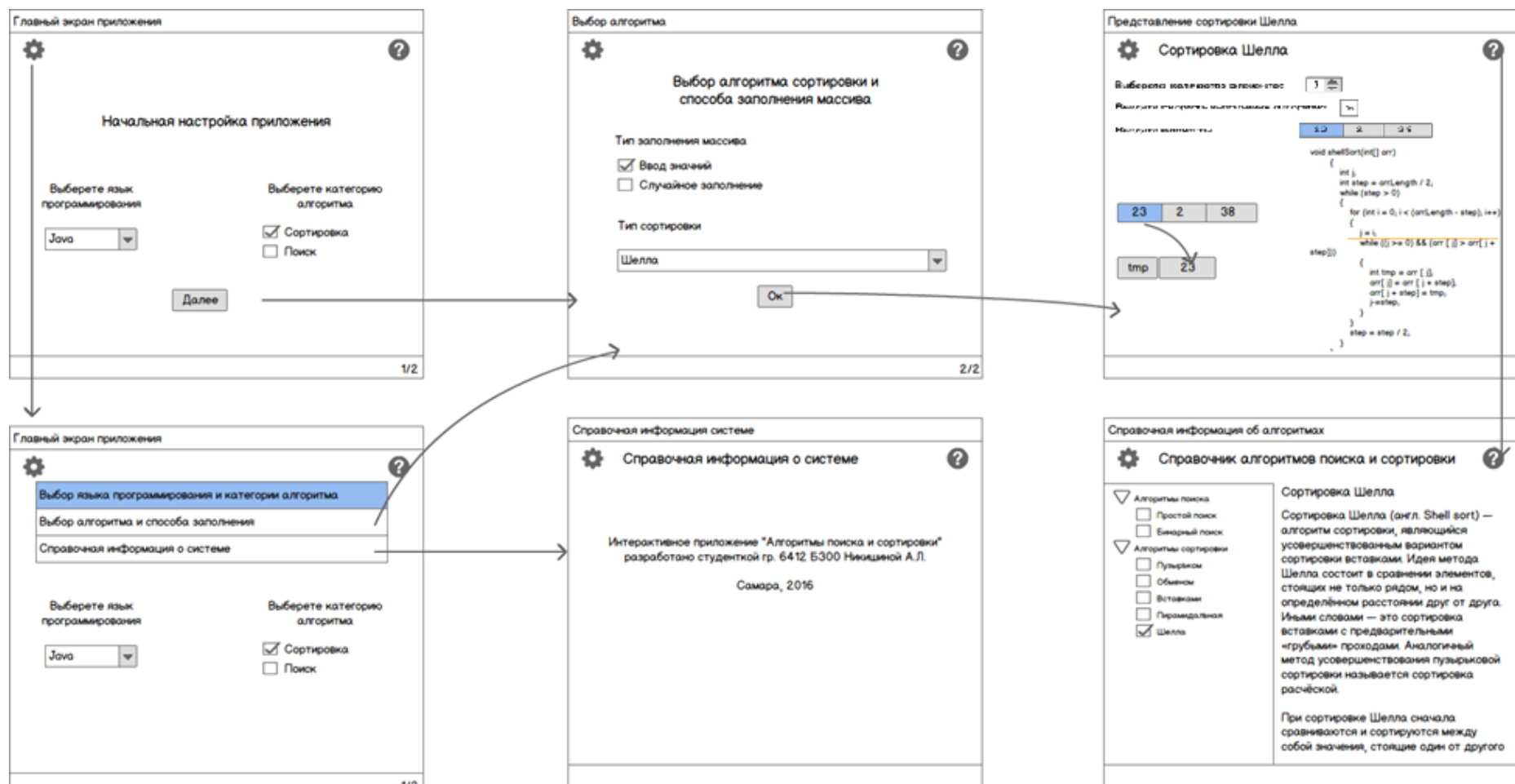


Рисунок XXX – Навигационная модель приложения

### 2.5.3 ЯЗЫК UML

Для специфицирования (построения точных, недвусмысленных и полных моделей) системы и ее документирования используется унифицированный язык моделирования UML.

Вводные слова Вводные слова Вводные слова Вводные слова Вводные  
слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные  
слова Вводные слова Вводные слова Вводные слова Вводные слова Вводные  
слова.

#### 2.5.4 Диаграмма вариантов использования

Диаграмма вариантов использования представляет собой наиболее общую концептуальную модель сложной системы, которая является исходной для построения всех остальных диаграмм. На ней изображаются отношения между актерами и вариантами использования.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке XXX приведена диаграмма вариантов использования (пользователя). Здесь должно быть описание диаграммы.

### 2.5.5 Сценарии

Сценарий (scenario) - определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме обычного текста [27].

В контексте языка UML сценарий используется для дополнительной иллюстрации взаимодействия актеров и вариантов использования.

Рассмотрим несколько сценариев.

Сценарии определяются преподавателем.

Рисунок XXX – Диаграмма вариантов использования системы

### 2.5.6 Диаграмма классов

Диаграммы классов – это наиболее часто используемый тип диаграмм, которые создаются при моделировании объектно-ориентированных систем, они показывают набор классов, интерфейсов и коопераций, а также их связи. На практике диаграммы классов применяют для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения [XXX].

На рисунке XX приведена диаграмма классов системы (этап проектирования). В таблице XX приведено описание классов.

Таблица XX – Описание классов системы

Название класса	Назначение
1	2

### 2.5.7 Диаграмма состояний

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке XXX приведена диаграмма состояний системы. Здесь должно быть описание диаграммы (диаграмм).

### 2.5.8 Диаграмма деятельности

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке XXX приведена диаграмма деятельности системы. **Здесь должно быть описание диаграммы (диаграмм).**

#### 2.5.9 Диаграмма последовательности

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке XXX приведена диаграммы последовательности системы для варианта использования «???». **Диаграммы построены на основании сценариев, приведенных в п.2.4.3.**



## 2.6 Логическая модель данных (при необходимости)

Логическая информационная модель – модель данных, в которой учитывается способ логического хранения данных в памяти ЭВМ. При построении модели базы данных (БД) используются следующие понятия.

Сущность – объект предметной области, который можно отличить от других понятий по некоторым признакам. Сущность состоит из множества своих экземпляров. Каждая сущность обладает свойствами – атрибутами **[Ошибка! Источник ссылки не найден.]**.

Атрибут – определенное свойство сущности. Именно набор атрибутов, в общем случае уникальный для каждой сущности, позволяет выделить ее среди других объектов и назвать уникальным именем.

Атрибут или набор атрибутов, используемый для идентификации экземпляра сущности, называется ключом сущности. В случае если для идентификации экземпляра используется один атрибут, ключ называется простым; в противном случае ключ составной. Каждый экземпляр сущности однозначно определяется ключом **[Ошибка! Источник ссылки не найден.Ошибка! Источник ссылки не найден.]**.

Логическая модель БД разрабатываемой системы приведена на рисунке XXX.

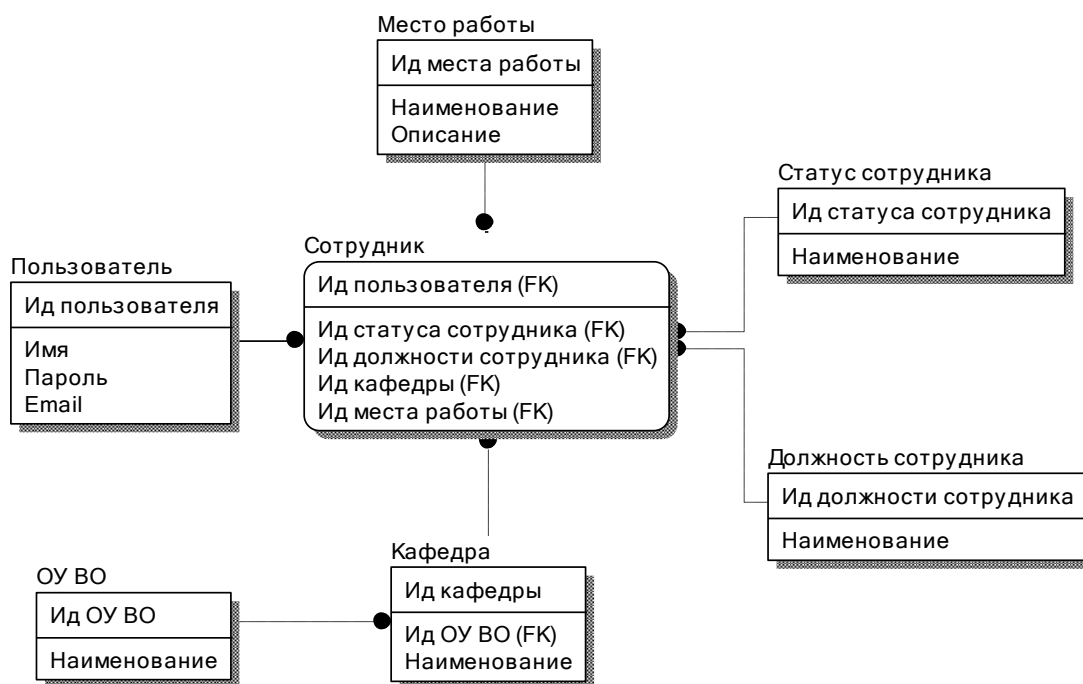


Рисунок 1 – Логическая модель данных

Описание объектов рассматриваемой предметной области, которые хранятся в базе данных, приведено в таблицах 2-???

Таблица 2 – Сущность «Пользователь»

Идентификатор	Тип данных	Описание
Ид пользователя	Целый	Уникальный идентификатор пользователя
Имя	Символьный[30]	Имя, используемое при идентификации пользователя и его взаимодействии с системой
Пароль	Символьный[10]	Пароль пользователя, преобразованный в закодированную строку
Email	Символьный[50]	Электронная почта, указанная пользователем при регистрации

## 2.7 Выбор и обоснование алгоритмов обработки данных /Разработка и описание алгоритмов обработки данных

Вводные слова про необходимость разработки алгоритмов.

На рисунке XXX приведена схема алгоритма обработки элементов массива. Здесь должно быть краткое описание алгоритма.

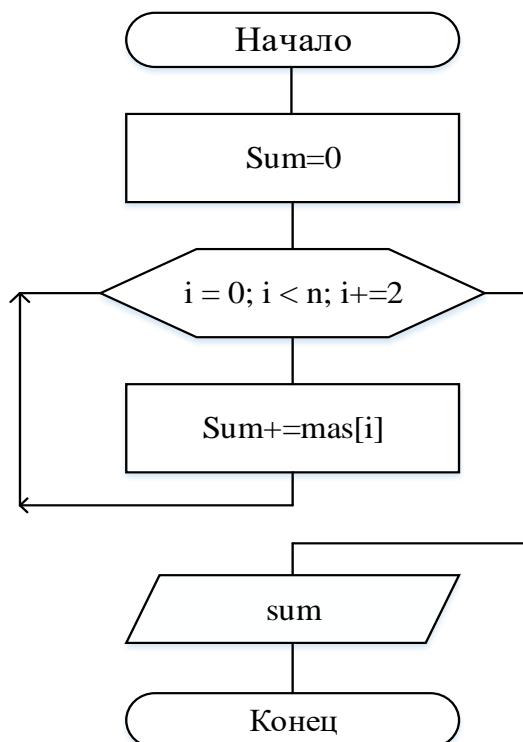


Рисунок XXX – Схема алгоритма обработки элементов массива

На рисунке XXXX приведена схема алгоритма вычисления исходного выражения.

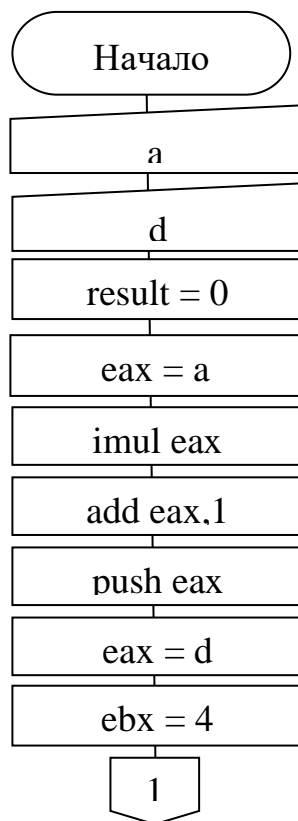
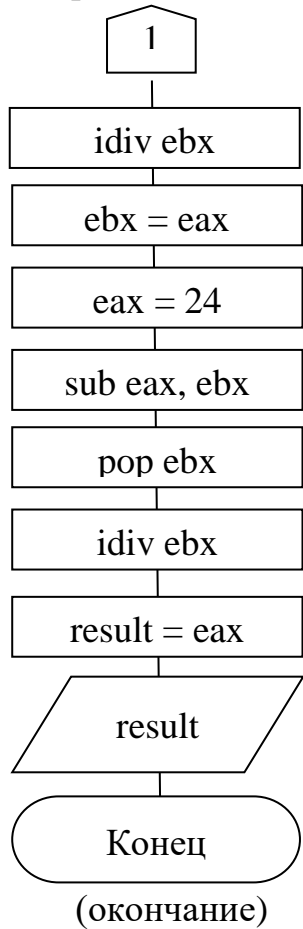


Рисунок XXX – Схема алгоритма вычисления исходного выражения (начало)

Рисунок XXXX– Схема алгоритма вычисления исходного выражения



## 2.8 Выбор и обоснование комплекса программных средств

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

### 2.8.1 Выбор языка программирования

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

## 2.8.2 Выбор среды программирования

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

## 2.8.3 Выбор операционной системы

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

## 2.8.4 Выбор системы управления базами данных (при необходимости)

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

### 3 Реализация системы

#### 3.1 Разработка и описание интерфейса пользователя

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунках приведены примеры того, как нужно оформить сведения о разработчиках.

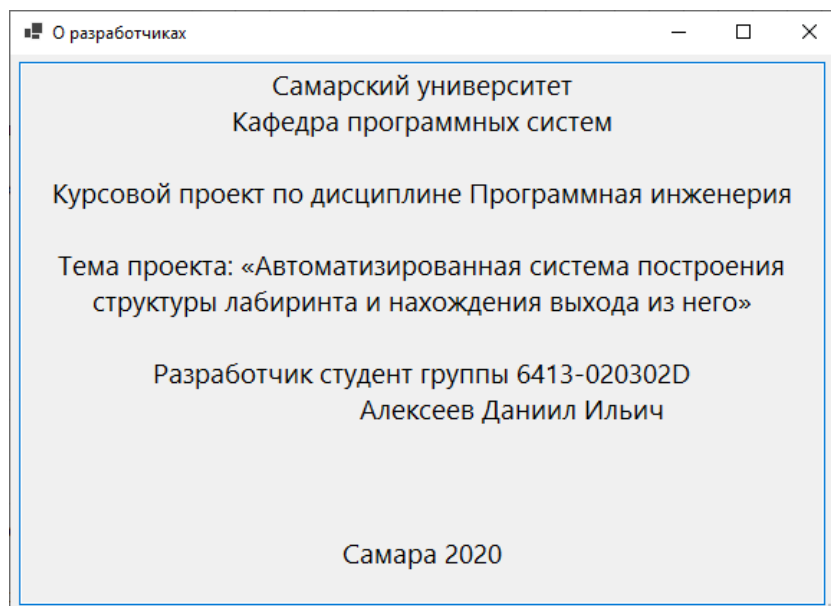


Рисунок XXX – Сведения о разработчиках

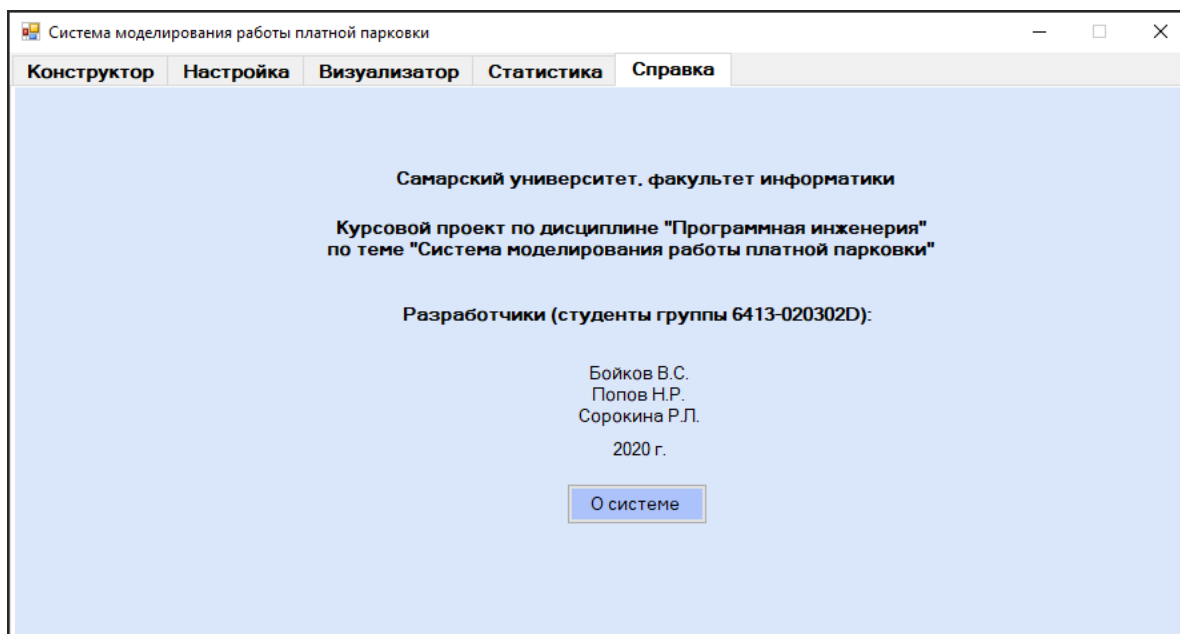


Рисунок XXX – Сведения о разработчиках

## 3.2 Диаграммы реализации

Диаграммы реализации предназначены для отображения состава компилируемых и выполняемых модулей системы, а также связей между ними. Диаграммы реализации разделяются на два конкретных вида: диаграммы компонентов (component diagrams) и диаграммы развертывания (deployment diagrams) [XXX].

### 3.2.1 Диаграмма компонентов

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке XXX приведена диаграмма компонентов, их описание приведено в таблице XXX.

Рисунок XXX – Диаграмма компонентов системы



Таблица XXX – Описание компонентов системы

Название компонента	Назначение компонента	Подсистема

3.2.2 Диаграмма развертывания

Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст  
Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст Текст.

На рисунке XXX приведена диаграмма развертывания системы. **Здесь должно быть описание** тех компонентов, которые развернуты на узлах ЭВМ.

Рисунок XXX – Диаграмма развертывания системы

3.2.3 Диаграмма классов

В соответствии со спецификацией, приведенной в п. 2.5.6, и с учетом выбранного языка программирования (см. п. 2.8.1) разработана диаграмма классов системы (этап реализации), приведенная на рисунке XXX.

Рисунок XXX – Описание классов системы (этап реализации)

### 3.3 Физическая модель данных (при необходимости)

Физическое проектирование является последним этапом проектирования базы данных, при выполнении которого принимается решение о способах реализации разрабатываемой базы данных. Во время логического проектирования была определена логическая структура базы данных (которая описывает отношения и ограничения в рассматриваемой прикладной области).

Физическая модель базы данных содержит все детали, необходимые конкретной СУБД для создания базы: наименования таблиц и столбцов, типы данных, определения первичных и внешних ключей [??].

На рисунке **Ошибка! Источник ссылки не найден.** представлена физическая модель данных системы.

Рисунок XXX – Физическая модель данных системы

В таблицах ??-?? приведено описание сущностей БД. Первичные ключи выделены жирным шрифтом, а внешние – курсивом.

Таблица XXX – Сущность « User »

Имя поля	Имя атрибута	Тип	Размер (байт)
<b>user Id</b>	<b>uniqueidentifier</b>	<b>int</b>	<b>4</b>
Name	Имя пользователя	varchar(30)	30
Password	Пароль	varchar(10)	10
e-mail	Адрес электронной почты	varchar(50)	50

Размер записи	94
---------------	----

### 3.4 Выбор и обоснование комплекса технических средств

#### 3.4.1 Расчет объема занимаемой памяти

##### Расчет объема внешней памяти

Для расчета необходимого объема свободной внешней памяти, необходимой для функционирования системы, воспользуемся следующей формулой:

$$V_{\text{ЖД}} = V_{\text{ОС}} + V_{\text{ПР}} + V_{\text{СПО}} + V_{\text{БД}} + V_{\text{справки}},$$

где  $V_{\text{ОС}}$  – объем памяти, занимаемый операционной системой (операционная система Windows 7 Professional 64 бит с пакетом обновлений SP1,  $V_{\text{ОС}} = 20$  Гб);

$V_{\text{ПР}}$  – объем памяти, занимаемый непосредственно файлами приложения ( $V_{\text{ПР}} = 2$  Мб);

$V_{\text{СПО}}$  – объем памяти, занимаемый сопутствующим программным обеспечением (библиотеки cryptopp.dll, simplexlsx.dll, sqlite3.dll, sqlitcpp.dll, Qt Framework 5.11.1, Internet Explorer 9; дадим оценку сверху  $V_{\text{СПО}}$  в 3 Гб);

$V_{\text{БД}}$  – объем памяти, занимаемый базой данных (всеми таблицами) при ее максимальном заполнении. Расчет этой составляющей приведен в таблице XXX ( $V_{\text{БД}} = \text{???? байт} = \text{??? Кб} = \text{??? Мб} = \text{??? Гб}$ ).

$V_{\text{справки}}$  – объем памяти, необходимый для хранения файла справки ( $V_{\text{справки}} = 0,8$  Мб).

Таким образом, суммарный объем внешней памяти составит:

$$V_{\text{ЖД}} = 20 \text{ Гб} + 2 \text{ Мб} + 3 \text{ Гб} + \text{??? Мб} + 1 \text{ Мб} \sim \text{??? Гб}.$$

Таблица 1 – Расчет объема внешней памяти, необходимой для хранения БД

Таблица	Размер записи (байт)	Максимум записей	Всего (байт)
Пользователь	94	10	940
Сотрудник		30	
Статус сотрудника		10	
Должность сотрудника		10	
Место работы		10	
Кафедра		10	
ОУ ВО		10	
Итого			

#### Расчет объема ОЗУ

Для расчета необходимого объема ОЗУ воспользуемся следующей формулой:

$$V_{\text{ОЗУ}} = V_{\text{ОС}} + V_{\text{ПР}} + V_{\text{БД}} + V_{\text{браузера}},$$

где  $V_{\text{ОС}}$  – ОЗУ, занимаемое операционной системой (2 Гб);

$V_{\text{ПР}}$  – ОЗУ, которое займет само приложение (не превысит 80 Мб);

$V_{\text{БД}}$  – объем данных из базы, который может быть одновременно загружен в оперативную память (дадим ему оценку сверху в 10 Мб).

$V_{\text{браузера}}$  – ОЗУ, занимаемое браузером (оценим его сверху значением в 100 Мб).

Суммарные объемы ОЗУ составит:

$$V_{\text{ОЗУ}} = 2 \text{ Гб} + 80 \text{ Мб} + 10 \text{ Мб} + 100 \text{ Мб} \sim 2.2 \text{ Гб}.$$

Таким образом, 2.2 Гб оперативной памяти можно считать минимально необходимым для функционирования системы.

### 3.4.2 Минимальные требования, предъявляемые к системе

Для корректного функционирования системы необходимо:

- тип ЭВМ: x86-64 совместимый;
- объем ОЗУ – не менее 3 Гб;
- объем свободного дискового пространства – не менее ??? Гб;
- клавиатура или иное устройство ввода;
- мышь или иное манипулирующее устройство;
- процессор – Intel Pentium не менее 1,5 ГГц;
- дисплей с разрешением не менее  $1024 \times 768$  пикселей;
- операционная система Windows 7 и выше;
- браузер Internet Explorer 9 и выше;
- Qt framework 5.11 и выше.

## ЗАКЛЮЧЕНИЕ

В процессе выполнения курсового проекта была разработана автоматизированная система ..., позволяющая ....

В первом разделе приведены основные понятия предметной области, описаны характеристики систем-аналогов, приведен их сравнительный анализ. На основе проведенного анализа выполнена объектная декомпозиция, отраженная в диаграмме объектов, и сформулирована постановка задачи.

Во втором разделе ...

В третьем разделе ...

Разработанная система может использоваться ...

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

### Книги

#### *Целиком*

1 Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. 546 с.

*Если нужно указать номера конкретных страниц*

2 Буч Г., Рамбо Д., Якобсон А. Язык UML. Руководство пользователя. Изд. 2-е. М.: ДМК Пресс, 2006. С. 21.

*Если повторная ссылка на тот же документ*

3 Буч Г., Рамбо Д., Якобсон А. Язык UML ... С. 31.

*Если больше 3 авторов*

4 Нестационарная аэродинамика баллистического полета/ Липницкий Ю.М. и [др.]. М.: Физматлит, 2003. 176 с.

### Журналы

5 Зеленко Л.С., Шумская Е.А. Комплекс программ для работы с учебным контентом в дистанционных обучающих системах// Известия СНЦ РАН. 2015. №2 (5). Т. 17. С. 992-1003.

### Руководящие материалы и ГОСТы

6 РД 34.20.571. Методические указания по расчету показателей готовности к работе электростанции и энергосистем. Введ. 1976-10-22. М., 1976. 25 с.

7 ГОСТ Р 7.0.4-2006. Издания. Выходные сведения. Общие требования и правила оформления. М., 2006. II. 43 с. (Система стандартов по информ., библиот. и изд. делу).

### Методические указания или учебные пособия

8 Зеленко Л.С. Методические указания к лабораторному практикуму по дисциплине «Программная инженерия». Самара: СГАУ, 2012. 67 с.

## Электронные ресурсы

- 9 Слепой метод печати [Электронный ресурс] // Википедия: [сайт]. URL: [https://ru.wikipedia.org/wiki/slepoi\\_metod\\_pechyati](https://ru.wikipedia.org/wiki/slepoi_metod_pechyati) (дата обращения: 23.09.2024)
- 10 Российская гидроэнергетика [Электронный ресурс] // Русгидро: [сайт]. URL: <http://www.rushydro.ru/industry/russianhydropower/> (дата обращения: 20.12.2024).
- 11 Гидроэлектростанция (гидроэлектрическая станция, ГЭС) // Энциклопедический словарь юного техника М.: Издательство «Педагогика», 1987 [Электронный ресурс] // Библиотекарь.Ру: электрон. библ. 2006-2024. URL: <http://www.bibliotekar.ru/enc-Tehnika/58.htm> (дата обращения: 20.12.2024).
- 12 Субботин А.С. Основы гидротехники [Электронный ресурс]. URL: <http://www.cawater-info.net/bk/dam-safety/files/subbotin.pdf> (дата обращения: 03.02.2024).
- 13 Филиальная структура компании [Электронный ресурс] // Системный оператор Единой энергетической системы: [сайт]. [2009-2017]. URL: <http://so-ups.ru/index.php?id=about> (дата обращения: 20.12.2024).
- 14 Автоматизированные системы управления технологическими процессами гидроэлектростанции [Электронный ресурс] // Микроника. Инжиниринговый центр: [сайт]. [1999-2016]. URL: <http://mikronika-energo.ru/products/asutp/ges-asu-tp/> (дата обращения: 24.12.2024).
- 15 Автоматизированная система управления производственными процессами [Электронный ресурс] // MEScontrol: [сайт]. [2003-2017]. URL: <http://mescontrol.ru/articles/systems> (дата обращения: 02.04.2024).
- 16 Пушкинов А.Ю. Введение в системы управления базами данных: учеб. пособие [Электронный ресурс] // CITForum: электрон. библиотека. 1997-2017. URL: <https://citforum.ru/database/dblearn/dblearn06.shtml> (дата обращения: 20.12.2024).



17 Пользовательский интерфейс [Электронный ресурс] // Википедия: электрон. энциклопедия. 2001-2017. URL: [https://ru.wikipedia.org/wiki/Пользовательский\\_интерфейс](https://ru.wikipedia.org/wiki/Пользовательский_интерфейс) (дата обращения: 17.03.2024).

***Если необходимо указать системные требования для доступа к документу (наличие специального ПО), то***

18 Белова С.В. Язык UML. Диаграмма вариантов использования. Систем. требования: PowerPoint. URL: [nkse.ru/component/k2/item/download/7\\_754f5a247edc6ec6be78218f187338a5.html](http://nkse.ru/component/k2/item/download/7_754f5a247edc6ec6be78218f187338a5.html) (дата обращения: 17.10.2024).

### **Сборники научных трудов или трудов конференций**

19 Философия культуры и философия науки: проблемы и гипотезы: межвуз. сб. науч. тр./ Саратов. гос. ун-т; [под ред. С.Ф. Мартыновича]. Саратов: изд-во Сарат. ун-та, 1999. 199 с.

20 Акимова А.Е., Трешников А.А., Зеленко Л.С. Информационная среда ГЭС. Подсистема расчета показателей эффективности работы оборудования // Перспективные информационные технологии (ПИТ-2017): сб. науч. тр. межд. научно-техн. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во СНИЦ РАН, 2017. С. 41-44.

### ***Если электронное издание***

21 Акимова А.Е., Трешников А.А., Зеленко Л.С. Подсистема расчета показателей эффективности работы оборудования // Математика. Компьютер. Образование: труды XXIV межд. конф., 23-28 января 2017 г., г. Пущино. URL: <http://www.mce.su/rus/presentations/p283063/> (дата обращения: 02.03.2017).

## ПРИЛОЖЕНИЕ А

### Руководство пользователя

#### А.1 Назначение системы

Приводится краткое описание возможностей системы.

#### А.2 Условия работы системы

##### ***Пример.***

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств.

##### 1 Требования к техническому обеспечению:

- ЭВМ типа IBM PC;
- процессор типа x86 или x64 тактовой частоты 1400 МГц и выше;
- ...

##### 2 Требования к программному обеспечению:

- Windows 7 Professional 64 бит с пакетом обновлений SP1 и выше;
- установленная платформа .Net версии 4.0 и выше;
- установленная СУБД ....

#### А.3 Установка системы

##### ***Пример.***

Система поставляется в **виде zip-архива**. Данный файл необходимо распаковать в любую директорию на жестком диске. Запускаемым файлом системы является файл **xxx.exe**.<sup>2</sup>

---

<sup>2</sup> Если необходимы дополнительные ресурсы для обеспечения работоспособности системы, то все для них также должны быть перечислены условия установки. *Если установка нестандартная, то она должна быть подробно описана (в объеме, достаточном для понимания пользователя).*

## А.4 Работа с системой

### А.4.1 Работа с системой в режиме администратора (если необходимо)

Вход в систему (авторизация)

...

### А.4.2 Работа с системой в режиме пользователя

Вход в систему (авторизация)

Вход в систему (регистрация)

Настройка параметров кроссворда

## ПРИЛОЖЕНИЕ Б

### Листинг модулей программы

7-10 страниц исходного кода шрифт Times New Roman 10 пт 1 интервал