Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

|  |
| --- |
| *К защите допустить*: |
| Заведующий кафедрой ИПиЭ |
| К. Д. Яшин |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к дипломному проекту

на тему

**ПРОГРАММНО-Аппаратный комплекс исследования динамики заучивания**

БГУИР ДП 1-58 01 01 021 ПЗ

|  |  |  |
| --- | --- | --- |
| Студент |  | И. А. Черкас |
| Руководитель |  | И. Г. Шупейко |
| Консультанты: |  |  |
| *от кафедры ИПиЭ* |  | И. Г. Шупейко |
| *по экономической части* |  | Т.Л. Слюсарь |
| *по ресурсо- и энергосбережению* |  | М.М. Борисик |
| Нормоконтролер |  | О.В. Булынко |
| Рецензент |  |  |

Минск 2017

**Реферат**

УДК 539.216.2; 621.3.049.77

Черкас, И. А.

Разработка программы для исследования динамики заучивания.: Дипл. проект по спец. «Инженерно-психологическое обеспечение информационных технологий». − Мн. : БГУИР, 2017. – 73 л.

Целью выполнения дипломного проекта является создание компьютерной программы для проведения лабораторных работ в учебных заведениях, для исследования динамики психического процесса запоинания информации.

Для автоматизации процесса сбора данных по методикам для исследования процесса заучивания необходимо спроектировать базу данных для обрабатываемой информации, программу для организации обработки данных, графический интерфейс пользователя. При проектировании программы и базы данных следует провести анализ предметной области для выявления основных сущностей, связей между ними и способов их представления на машинных языках. Эргономическое проектирование должно поспособствовать получению продукта с высокими уровнями результативности, эффективности и удобства работы пользователя. Результатом эргономического проектирования является проект деятельности оператора. Построение графического интерфейса пользователя основывается на выполненном проекте деятельности и позволяет построенной системе приобрести приемлемые для человека эстетические свойства.

Полученный программный комплекс применим для получения статистических данных для научных исследований динамических процессов памяти. Применение спроектированного комплекса в процессе обучения позволяет уменьшить долю организационных мероприятий при проведении лабораторных работ и концентрировать внимание на исследуемых явлениях в большей мере.

***Ключевые слова:*** программно-аппаратный комплекс, память, заучивание, эргономическое проектирование, системы «Человек-машина», база данных, статистическая обработка, проект деятельности оператора

Ил. – 18 , табл. – 14 , список лит. – 25 назв.;

графическая часть – 6л. А1.

Министерство образования Республики Беларусь

|  |
| --- |
| Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ |

Факультет компьютерного проектирования Кафедра инженерной психологии и эргономики

Специальность 1-58 01 01 Инженерно-психологическое обеспечение информационных технологий

УТВЕРЖДАЮ

Заведующий кафедрой ИПиЭ

Яшин К.Д.

20 февраля 2017 г.

**ЗАДАНИЕ**

**по дипломному проекту студента**

Черкаса Ильи Анатольевича

1 Тема проекта «Программно-аппаратный комплекс исследования динамики заучивания» утверждена приказом по университету от 17 февраля 2017г. №326-с.

2 Срок сдачи студентом законченного проекта 26 мая 2017 г.

3 Исходные данные к проекту: операционная система Windows 7 и более поздние версии, язык программирования C#, программная платформа .NET WPF, среда разработки Microsoft Visual Studio, хранение данных с использованием базы данных Microsoft SQL Server.

Назначение разработки: структурирование данных и автоматизация процессов обучения.

4 Содержание расчетно-пояснительной записки

Введение

1 Исследование процесса заучивания в деятельности человека

1.1 Процесс заучивания как вид запоминания

1.2 Существующие средства для исследования заучивания

1.3 Постановка задачи для реализации программно-аппаратного комплекса исследования динамики заучивания

2 Эргономическое проектирование программно-аппаратного комплекса

2.1 Анализ и распределение функций

2.2 Проектирование деятельности пользователей

2.3 Разработка средств деятельности пользователей

3 Программное проектирование комплекса исследования динамики заучивания.

3.1 Разработка структуры программы

3.2 Проектирование базы данных

3.3 Тестирование программы

4 Экономическое обоснование проекта по разработке программно-аппаратного комплекса

5 Обеспечение пожарной безопасности при разработке программно-аппаратного комплекса

Заключение

Список использованных источников

Приложение А (обязательное). Листинг программы

5 Перечень графического материала

Диаграмма вариантов использования (ПД) – формат А1, лист 1

Структурная схема комплекса (ПД) – формат А1, лист 1

Структура базы данных и диаграмма классов (ПД) – формат А2, листов 2

Блок-схемы алгоритмов работы пользователей (ПД) – формат А2, листов 2

Сценарий взаимодействия администратора с системой (ПЛ) – формат А1, лист1

Сценарий взаимодействия испытуемого с системой (ПЛ) – формат А1, лист1.

6 Содержание задания по технико-экономическому обоснованию.

Технико-экономическое обоснование эффективности разработки и использования программно-аппаратного комплекса исследования динамики заучивания в учебном процессе

Задание выдал Т.Л. Слюсарь

7 Содержание задания по ресурсо- и энергосбережению.

Разработка рекомендаций по обеспечению пожарной безопасности при разработке программно-аппаратного комплекса.

Задание выдал М.М. Борисик

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов дипломного проекта | Объём этапа, % | Срок выполнения этапов проекта | Примечание |
| Получение задания на дипломный проект |  | 26.02.2017 |  |
| Обзор научно-технической литературы по теме дипломного проекта | 10 | 22.02.2017 |  |
| Разработка эргономического раздела | 20 | 02.03.2017 |  |
| Разработка экономического раздела | 10 | 10.03.2017 |  |
| Разработка раздела по ресурсо- и энергосбережению | 10 | 17.03.2017 |  |
| Разработка программного модуля системы | 30 | 17.04.2017 |  |
| Оформление пояснительной записки | 10 | 26.04.2017 |  |
| Разработка графических материалов дипломного проекта | 10 | 15.05.2017 |  |
| Сдача дипломного проекта |  | 26.05.2017 |  |

Дата выдачи задания 26 февраля 2017 г. Руководитель Шупейко И. Г.

Задание принял к исполнению 26 февраля 2017 г Черкас И. А.

**СОДЕРЖАНИЕ**

Введение 5

1 Исследование процесса заучивания в деятельности человека7

1.1 Процесс заучивания как вид запоминания7

1.2 Существующие средства для исследования заучивания12

1.3 Постановка задачи для реализации программно-аппаратного комплекса исследования динамики заучивания18

2 Эргономическое проектирование программно-аппаратного комплекса 23

2.1 Анализ и распределение функций23

2.2 Проектирование деятельности пользователей29

2.3 Разработка средств деятельности пользователей39

3 Программное комплекса исследования динамики заучивания. 45

3.1 Разработка структуры программы 45

3.1 Проектирование базы данных 45

3.2 Тестирование программы50

4 Экономическое обоснование проекта по разработке программно-аппаратного комплекса55

5 Обеспечение пожарной безопасности при разработке программно-аппаратного комплекса64

Заключение68

Список использованных источников70

**ВВЕДЕНИЕ**

Актуальность исследования психических процессов в деятельности человека-оператора подтверждается наиболее распространённостью этого вида деятельности человека в современном мире. Деятельность человека-оператора является основой функционирования большинства современных технических систем, поэтому в настоящее время изучению процессов взаимодействия человека и техники уделяется внимание во многих научных дисциплинах [1].

Изучению подвергаются характеристики всех взаимодействующих сторон: у человека оператора: память, сила, скорость, внимание и другие; у технических устройств: скорость и объем потоков информации, распределение информации по каналам различной модальности, эргономические характеристики и другие [2].

Целью дипломного проекта является создание программного продукта, обладающего высокой эргономичностью. Пользователю должен быть предоставлен удобный интерфейс, система должна демонстрировать гибкость для пользователей различной квалификации, программа должна подсказывать пользователю ход решения задачи [3].

Для достижения цели поставлены следующие задачи:

* составление проекта деятельности испытуемого и экспериментатора с системой;
* проектирование базы данных для хранения информации о исследовании;
* построение приложения

Данный дипломный проект представляет собой реализацию в виде компьютерного программного продукта методик исследования динамики процесса заучивания. Исследования в данной области проводятся студентами в ходе лабораторных практикумов по дисциплине «Психология » и научными работниками кафедры инженерной психологии С помощью этой методики можно как изучать основные закономерности процесса заучивания, так и получать характеристики заучивания конкретного испытуемого или группы испытуемых в различных условиях [2].

Основные задачи проектируемой системы:

* облегчить подготовку к выполнению опытов по методике;
* ускорить выполнение опыта;
* позволить провести анализ результатов выполненных опытов.

Из задач определяются режимы работы системы:

* заполнение стимульного материала;
* выполнение опыта;
* анализ результатов.

Экспериментальным материалом служат в разных опытах буквы, цифры, бессмысленные слоги, не связанные между собой по смыслу слова. Материал предъявляют визуально на экране дисплея. Предъявление элементов ряда может производиться как последовательно, так и одновременно [2].

В каждом опыте испытуемому предлагают ряд из нескольких стимулов с требованием заучить его до безошибочного воспроизведения в любом порядке. Стимулы могут иметь различный вид и смысловую нагрузку (символы, слоги, слова). Набор символов может содержать элементы, объединенные каким-либо признаком, или несвязанные каким-либо образом между собой стимулы [2].

После каждого предъявления ряда испытуемый воспроизводит его, вводя в ПК запомненные стимулы, используя клавиатуру. Если воспроизведенный ряд эквивалентен предъявленному, опыт считается завершенным. Иначе ряд повторяется спустя заданный промежуток времени после окончания воспроизведения. Цикл длится до безошибочного воспроизведения [2].

В процессе создания системы выполнены основные этапы ее эргономического проектирования: составлена спецификация функций системы, выполнен анализ их содержания, проведено распределение функций между человеком и техническим звеном и определена структура системы, разработаны алгоритмы работы пользователей, сформулированы эргономические требования и обеспечен их учет при разработке прототипа пользовательского интерфейса системы. На основе данных, полученных в результате эргономического проектирования, была определена архитектура компьютерной системы [1].

1. **ИССЛЕДОВАНИЕ ПРОЦЕССА ЗАУЧИВАНИЯ В ДЕЯТЕЛЬНОСТИ ЧЕЛОВЕКА**

**1.1 Процесс заучивания как вид запоминания**

Память – процесс запоминания, сохранения и воспроизведения человеком образов, мыслей, эмоций, движений, т.е. всего, что составляет индивидуальный опыт каждого. Память является необходимым условием научения, приобретения знаний, формирования умений и навыков, способностей человека. Без нее невозможно поведение, мышление, сознание и подсознание [3].

После приема информации оператор так или иначе ее обрабатывает, анализирует и преобразует. В процессах переработки информации решающая роль принадлежит памяти и мышлению. В деятельности человека-оператора, управляющего машинами по приборам, в том или ином виде участвуют все основные виды памяти: кратковременная (непосредственная и оперативная) и долговременная. Эти виды различаются по продолжительности хранения информации. В кратковременной памяти информация хранится в течение секунд и минут, а в долговременной – в течение дней, месяцев, лет [1].

Кратковременная память подразделяется на непосредственную (иконическую) и оперативную. В непосредственной памяти сохраняется практически вся информация, поступившая в какой-то момент на органы чувств, однако она хранится очень недолго (всего несколько секунд) [1].

Оперативная память связана с запоминанием, сохранением и воспроизведением динамических (изменяющихся) элементов ситуации, т.е. текущего состояния технического звена системы. Под оперативной памятью понимаются процессы запоминания, сохранения и воспроизведения информации, получаемой и передаваемой при выполнении отдельного действия и необходимые для достижения частной цели, после чего они утрачивают свою актуальность. Длительность процессов оперативной памяти ограничивается длительностью осуществления данного действия [3].

Оперативная память, обеспечивая решение текущих задач оператора, хранит текущий алгоритм действий, играет важную роль в его деятельности. Исследованиями установлено, что большая часть ошибок оператора связана с процессами оперативной памяти [1]. Наиболее важными характеристиками являются:

- объем,

- длительность сохранения информации,

- правильность (точность) воспроизведения информации,

- помехоустойчивость [1].

Объем оперативной памяти определяется тем количеством сигналов (стимулов), которые оператор способен запомнить после одного, как правило, кратковременного предъявления (восприятия).

Различают объем памяти на статические и динамические сигналы.

В первом случае оператор должен запомнить и воспроизвести неизменяемую последовательность сигналов. Ее объем составляет около 5 — 9 символов. Причем, чем меньше длина алфавита символов, тем больше объем памяти и наоборот. Во втором случае оператор должен не только хранить в памяти предъявляемую последовательность сигналов, но и следить за ее изменениями в соответствии с изменениями обстановки. Ее объем не превышает в этом случае 3 — 4 символов [1].

Оперативная память выполняет функцию буфера с ограниченной емкостью. При этом вновь поступивший сигнал вытесняет из буфера один из поступивших ранее, если он не перешел к этому времени в долговременную память. Поэтому сигналы, поступившие в буфер первыми и последними, закрепляются в нем прочнее нежели сигналы средней части предъявляемой последовательности (“эффект края”) [1].

Длительность сохранения информации определяется тем промежутком

времени, в течение которого оператор безошибочно воспроизводит полученную информацию. Физиологической основой процесса сохранения является способность нервных клеток мозга определенное время сохранять изменения, возникающие под влиянием внешних воздействий (“следы” памяти). Безошибочное воспроизведение информации возможно, пока затухание «следа» не достигнет некоторого критического значения. Соответствующий этому промежуток времени и определяет время сохранения информации [1].

Правильность (точность) воспроизведения информации может быть количественно определена как вероятность безошибочного воспроизведения предъявляемой информации:

**Рпам = n / N,** (1.1)

где **n** и **N** – соответственно число правильно воспроизведенных и общее число предъявленных последовательностей сигналов [1].

Помехоустойчивость памяти определяется правильностью воспроизведения информации в условиях помех.

Рассмотренные характеристики оперативной памяти не являются строго постоянными величинами и во многом зависят от характера запоминаемой информации и условий работы оператора [1].

Время обработки информации с участием долговременной памяти больше, чем с участием только оперативной памяти. Оно различается на величину, которая определяет время поиска информации в долговременной памяти. Общее время обработки информации в этом случае равно

**tои = tоп + tпд**, (1.2)

где **tоп** – время обработки информации в оперативной памяти;

**tпд** – время поиска информации в долговременной памяти [1].

Кратковременная память играет большую роль в жизни человека. Благодаря ей перерабатывается самый большой объем информации, сразу отсеивается ненужная и остается потенциально полезная. Вследствие этого не происходит информационной перегрузки долговременной памяти излишними сведениями, экономится время человека [1].

Изучение памяти было одним из первых разделов психологической науки, где был применен экспериментальный метод: были сделаны попытки измерить доступный человеку объем памяти, быстроту, с которой он может запомнить материал, и время, в течение которого он может удерживать этот материал [2].



Рисунок 1.1.1 – Параметры сигналов, определяющие продуктивность оперативной памяти

Продуктивность оперативной памяти во многом зависит от параметров предъявляемых сигналов [1]. При этом значение имеют следующие 4 группы параметров:

- информационные,

- структурно-пространственные,

- временные,

- по признаку модальности (рис. 1.2.1).

Среди информационных параметров наибольшее значение имеет количество и ценность информации. Первое определяется степенью организации, упорядоченностью и объемом запоминаемого материала. При этом избыточность информации в разной форме (вероятностная, логическая) увеличивает объем памяти, а также скорость запоминания. Вместе с тем следует отличать избыточную информацию от иррелевантной (не относящейся к ситуации). Последняя ухудшает продуктивность памяти [3].

Вопросы влияния модальности сигналов на продуктивность памяти изучены пока еще недостаточно, однако считается, что при равных других условиях продуктивность зрительной памяти выше, чем слуховой. Это обусловлено тем, что по зрительному каналу сигналы поступают параллельно, а по слуховому – последовательно. Данное обстоятельство ограничивает возможности слуховой памяти [1].

Участвующие в деятельности оператора различные виды памяти взаимодействуют друг с другом. Вновь поступающая информация направляется в оперативную память, где она сохраняется в течение времени хранения, определяемого длительностью «следа» памяти. В случае, если объем поступившей информации превышает объем оперативной памяти или время хранения больше длительности «следа», то часть информации с некоторой вероятностью направляется в долговременную память, а остальная информация теряется [1].



Рисунок 1.1.2 — Зависимость вероятности правильного воспроизведения от длительности предъявления символа **tс** и длины последовательности **n**:

---- при одновременном предъявлении;

––– при последовательном предъявлении

Из временных параметров для продуктивности памяти наиболее значимы длительность экспозиции и характер предъявления (одновременное или последовательное). Закономерности влияния данных параметров на продуктивность запоминания представлены на рис. 1.1.2 [1].

**1.2 Существующие средства для исследования заучивания**

На данный момент в области психологического тестирования широко представлены программно-аппаратные комплексы, основной задачей которых является проведение инженерно-психологических исследований различных видов. Это и простые программы, реализующие какую-либо методику, так и интегрированные среды разработки тестовых опросов с поддержкой анализа данных [2]. Далее перечислены наиболее вероятные альтернативы для исследования динамики процесса заучивания.

**Базовый комплекс компьютерных психодиагностических программ** **"Психология в школе" (Effecton)** был создан в 1996 году (Свидетельство о регистрации в Российском фонде компьютерных программ учебного назначения Минобразования РФ N1717 от 10.12.96).



Рисунок 1.2.1 — Программа Effecton в режиме психологического тестирования

При создании комплекса разработчики руководствовались следующими принципами: включение в пакет только научно-обоснованных и информативных методик, позволяющих работать со всеми участниками образовательного процесса; методическое сопровождение методик; представление не только результатов тестирования, но и их интерпретации; простота работы с комплексом; удобство хранения и извлечения информации, возможность статистической обработки данных (рисунок 1.2.2).

В дальнейшем комплекс пополнялся новыми программами, появились модификации комплекса для средних специальных и высших учебных заведений – "Психология в образовании", для центров занятости, служб персонала – "Психология в управлении персоналом", для индивидуальных пользователей – "Персональная психология".

Преимуществом такого комплекса является возможность широкого применения для тестирования и психофизиологической экспертизы персонала. Программа достаточно универсальна и не ограничивает тематики исследований. Программа является коммерческой, лицензирование производится для каждой рабочей машины.

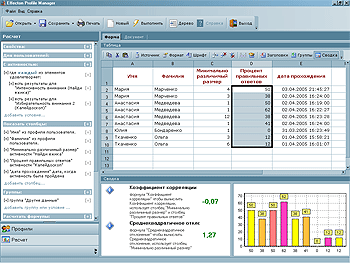


Рисунок 1.2.2 — Программа Effecton в режиме анализа результатов психологического тестирования.

Широкое применение и развитый функционал обуславливают относительно высокую стоимость лицензии. Еще одним недостатком является необходимость создание специальных тематических тестов. Методика для исследования динамики заучивания недостаточно адаптирована для программ такого типа.

Еще одним вариантом является **программа для лабораторная работы «Исследование динамики процесса заучивания»** курса «Психология восприятия и переработки информации», разработанная на кафедреинженерной психологии и эргономики [2].

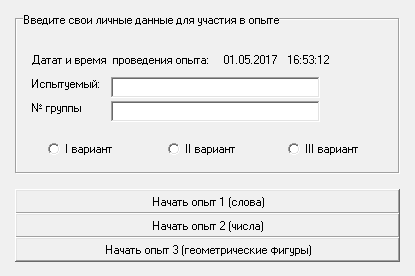


Рисунок 1.2.3 — Лабораторная работа «Исследование динамики процесса заучивания» кафедрыинженерной психологии и эргономики (выбор опыта).

Программа представляет собой графическое приложение для операционной системы Windows на основе библиотеки Visual Component Library. Visual Component Library (VCL) — библиотека визуальных компонентов для построения графических интерфейсов пользователя, используема в средах разработки Borland и RAD Studio. Программа предоставляет опыт со словами, опыт с числами и опыт с геометрическими фигурами. Каждый опыт содержит по три варианта (рисунок 1.2.3).

В режиме ввода запомненных значений программа предлагает ввести слова или слоги с использованием клавиатуры. В опыте со стимулами-изображениями испытуемому предлагается выбрать увиденные фигуры из множества вариантов, представляющих собой иллюстрации с текстовым описанием. Продолжительность предъявления стимулов составляет 300 миллисекунд. Программа поддерживает последовательный режим предъявления стимулов.

Программа адаптирована для методики исследования динамики процесса заучивания, позволяет работать в режиме. Для выполнения лабораторного практикума не обязательны дополнительные работы по созданию стимульного материала. Программа содержит теоретические сведения и инструкции для выполнения опытов [2].

**AVELife TestGold Studio 5** ‒ профессиональный инструмент тестирования, обеспечивающий создание тестов, комплексное тестирование знаний и психологическое тестирование для решения таких типовых задач как оценка персонала при приеме на работу, периодическая оценка и аттестации сотрудников, тестирование знаний студентов и учащихся в процессе обучения (рисунок 1.1.4).

Программа поддерживает базовый анализ результатов: сравнительное графическое представление результатов тестирования в виде столбиковой диаграммы, сводный отчет по тестируемому в виде совокупной характеристики, анализ ответов, тестируемых на определенный вопрос теста. Результаты тестирования накапливаются и доступны для просмотра в виде списка (таблицы) с возможностью фильтрации по тесту, тестируемому и дате проведения тестирования.

Задачей программы является создание тестов знаний и психологических опросников, тестирование студентов, оценка и аттестации персонала. При создании поддерживается 12 типов вопросов. Поддерживается возможность работы с изображениями, аудио- и видеофайлами. Для работы с данными используется система управления базами данных Microsoft SQL-server.

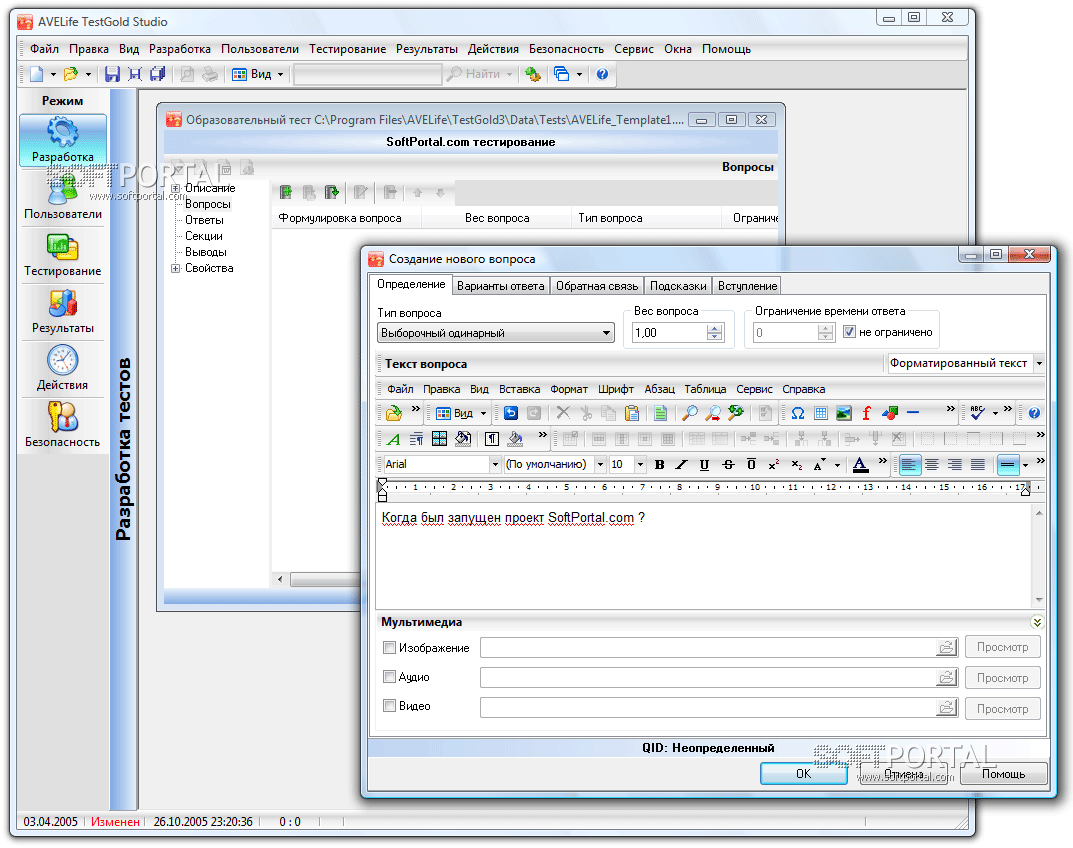


Рисунок 1.2.4 — Комплекс AVELife TestGold Studio (режим создания опросов).

Поддерживается расширенное форматирование текста вопросов и ответов c настройкой шрифтов и параграфов, использованием изображений, списков, таблиц в мощном встроенном текстовом редакторе (рисунок 1.2.5).

Администратору доступно гибкое формирование отчетов на основе настраиваемых шаблонов с использованием встроенного дизайнера отчетов для их редактирования. Поддерживается экспорт данных во внешние форматы такие как XML, HTML, XLS, CSV для интеграции с другим программным обеспечением или для использования данных в специализированных программах с дополнительными возможностями оформления, представления и анализа.

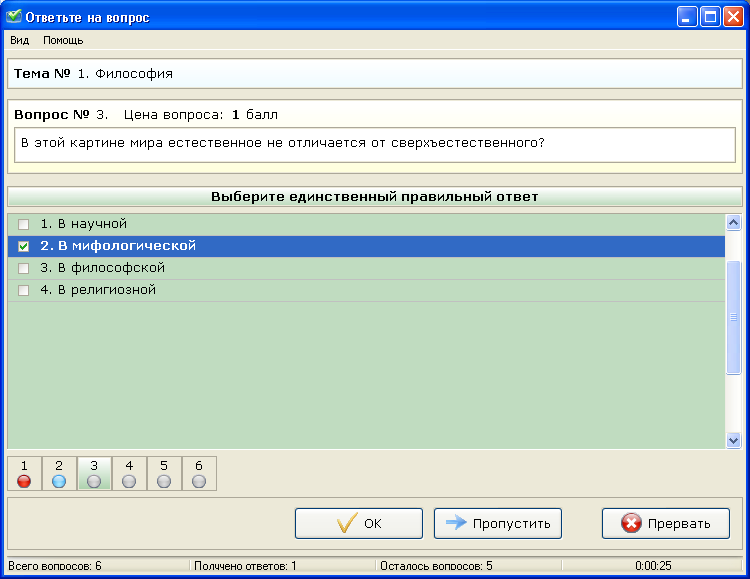


Рисунок 1.2.5 — Комплекс AVELife TestGold Studio (режим выполнения).

Программы Effecton и AVELife TestGold Studio, как указано ранее, являются многофункциональными интегрированными средствами разработки различного рода опросов и тестов. Это достаточно универсальные продукты, поэтому в случае их использования требуется создать опыты для исследования динамики процесса заучивания.

Необходимые трудозатраты на адаптацию методики к возможностям программы не позволяют эффективно использовать данный продукт для изучения динамики заучивания. Перед покупкой необходимого количества лицензий представляется рациональным рассмотреть другие продукты или вариант создания нового комплекса под заданные нужды.

Существующая лабораторная работа не имеет режима администратора и обладает фиксированным набором тестов, что затрудняет проведение исследований более широкой тематики чем лабораторная работа. С учетом большого объема работы, необходимой для модернизации программы рационально разработать новый программный продукт.

**1.3 Постановка задачи для реализации программно-аппаратного комплекса исследования динамики заучивания**

Методика исследования динамики процесса заучивания. Применяется в лабораторном исследовании студентами по психологическим дисциплинам. Экспериментальным материалом служат в разных опытах буквы, цифры, бессмысленные слоги, не связанные между собой по смыслу слова. Материал предъявляют визуально на экране дисплея. Предъявление элементов ряда может производиться как последовательно, так и одновременно [2].

В каждом опыте испытуемому предлагают ряд из нескольких стимулов с требованием заучить его до безошибочного воспроизведения в любом порядке. Стимулы могут иметь различный вид и смысловую нагрузку (символы, слоги, слова). Набор символов может содержать элементы, объединенные каким-либо признаком, или несвязанные каким-либо образом между собой стимулы [2].

После каждого предъявления ряда испытуемый воспроизводит его. Если воспроизведенный ряд эквивалентен предъявленному, опыт считается завершенным. Иначе ряд повторяется спустя заданный промежуток времени после окончания воспроизведения. Цикл длится до безошибочного воспроизведения [2].

Приведенная методика может применяться в бланковой форме. В этом случае экспериментатор взаимодействует с испытуемым напрямую, контролирует временные интервалы предъявления и интервалы между предъявлениями, регистрирует заученные стимулы [2].

Приведенный выше подход требует от экспериментатора дополнительных трудозатрат на взаимодействие с испытуемыми. Для проведения исследований на представительной и адекватной выборке из исследуемого контингента необходимо привлекать дополнительные ресурсы для сбора данных [2].

Применение возможностей персонального компьютера позволяет освободить экспериментатора от значительной доли трудозатрат по организации и проведению экспериментов. Исследователь может сфокусироваться на интерпретации полученных результатов в большей степени [2].

В компьютерные системах можно выделить несколько функциональных слоев в соответствии с уровнем абстракции решаемых задач (рис. 1.3.1). Так как при объектом исследования является память человека, то отсутствует необходимость разработки специфических аппаратных средств и системных программ. Необходимо создавать прикладную программу для существующих компьютерных платформ.

**Программное обеспечение** — все или часть программ, процедур, правил и соответствующей документации системы обработки информации (согласно ISO/IEC 2382-1:1993). Типичная схема компьютерной программы изображена на рис. 1.1.1. Современные программные продукты также могут содержать разделенные модули, что совокупностью связанных между собой модулей [4].



Рисунок 1.3.1 – Схема работы простейшей компьютерной программы

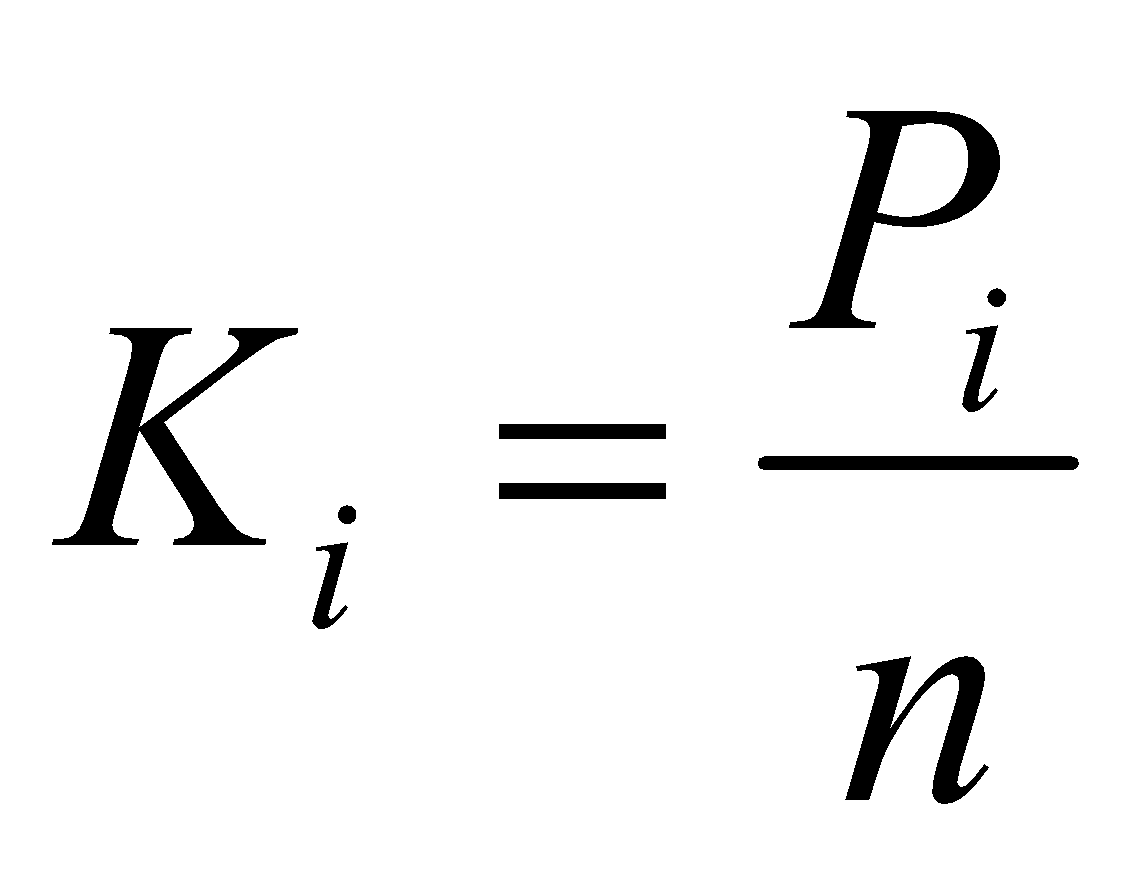
Пользователь – субъект взаимодействия компьютерной системы. В настоящее время это может быть не только человек, но и другая компьютерная программа, как это происходит в разделенных системах. Пользователь инициирует взаимодействие, использует программно-аппаратный комплекс для решения своих задач [1].

Программно-аппаратный комплекс (ПАК) — это набор технических и программных средств, работающих совместно для выполнения одной или нескольких сходных задач. На рис. 1.3.1 ПАК представляют три нижних уровня: прикладное программное обеспечение, системное программное обеспечение и аппаратное обеспечение [4].

Комплекс должен решать следующие задачи:

1. Предъявлять на экране компьютера справку о программе (ФИО разработчика, ФИО научного руководителя).
2. Ограничивать допуск испытуемого к некоторым функциям, которые должен выполнять только преподаватель;
3. Позволять администратору создавать и сохранять в памяти компьютера варианты (ряды) предъявляемых стимулов разных видов (слова, слоги, буквы).
4. Позволять администратору создавать и сохранять в памяти компьютера опыты с предъявлением стимулов разных видов (слова, слоги, буквы).
5. Позволять администратору редактировать вводные теоретические сведения.
6. Обеспечивать возможность просмотра преподавателем результатов выполненных студентами экспериментальных исследований.
7. Позволять администратору редактировать базу, сохраняемых результатов работы студентов (удалять файлы, потерявшие актуальность).
8. Проводить регистрацию студента (испытуемого).
9. Предъявлять на экране компьютера вводные теоретические сведения (по запросу испытуемого.
10. Инструктировать испытуемого о предстоящем опыте и его задачах.
11. Проводить перед началом экспериментов тренировочные серии с возможностью выбора студентом момента ее завершения.
12. Предъявлять на экране дисплея ряды стимулов.
13. Ввод в компьютер запомненных стимулов.
14. Выполнение на ПК необходимую математическую обработку полученных в исследованиях результатов.
15. Сохранение в памяти компьютера результаты работы испытуемого.
16. Предъявление на экране ПК результаты выполненного эксперимента.
17. Сохранение результатов эксперимента на переносном носителе информации.
18. Прекращение работы на любом ее этапе.

Одним из аспектов, в которых компьютер наиболее полезен в деятельности человека, является выполнение статистических и математических расчетов. Программно-аппаратный комплекс должен подсчитывать общее количество правильно воспроизведенных слов при каждом повторении, а также частоту воспроизведения каждого слова за все повторения (формула 1.3.1).

, (1.3.1)

где *К* – частота воспроизведения *i*-го слова;

*Рi* – количество его правильных воспроизведений;

*n* – количество повторений.

Важным компонентом прикладной программы является интерфейс взаимодействия ее с пользователем. Интерфейс — коммуникационное устройство, которое позволяет одному устройству взаимодействовать с другим и устанавливать соответствие между выходами одного устройства и входами [4].

Пользовательский интерфейс — интерфейс, обеспечивающий взаимодействие пользователя с компьютером. Понятие пользовательского интерфейса может означать общение между человеком и компьютером. Во многих определениях, интерфейс отождествляется с диалогом, который подобен диалогу или взаимодействию между двумя людьми. Подобно как наука и культура нуждается в правилах общения людей и взаимодействия их друг с другом в диалоге, также и человеко-машинный диалог также нуждается в правилах.

Основные принципы создания интерфейса:

1. Естественность (интуитивность). Работа с системой не должна вызывать у пользователя сложностей в поиске необходимых директив (элементов интерфейса) для управления процессом решения поставленной задачи [4].
2. Непротиворечивость. Если в процессе работы с системой пользователем были использованы некоторые приемы работы с некоторой частью системы, то в другой части системы приемы работы должны быть идентичны. Также работа с системой через интерфейс должна соответствовать установленным, привычным нормам (например, использование клавиши Enter) [4].
3. Неизбыточность. Это означает, что пользователь должен вводить только минимальную информацию для работы или управления системой. Например, пользователь не должен вводить незначимые цифры (00010 вместо 10). Аналогично, нельзя требовать от пользователя ввести информацию, которая была предварительно введена или которая может быть автоматически получена из системы. Желательно использовать значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации [4].
4. Непосредственный доступ к системе помощи. В процессе работы необходимо, чтобы система обеспечивала пользователя необходимыми инструкциями. Система помощи отвечает трем основным аспектам — качество и количество обеспечиваемых команд; характер сообщений об ошибках и подтверждения того, что система делает. Сообщения об ошибках должны быть полезны и понятны пользователю [4].
5. Гибкость. Интерфейс системы должен обслуживать пользователя с различными уровнями подготовки. Для неопытных пользователей интерфейс может быть организован как иерархическая структура меню, а для опытных пользователей как команды, комбинации нажатий клавиш и параметры [4].

Интерфейс не всегда может быть спроектирован в полном соответствии с перечисленными требованиями. Однако игнорирование данных требований и недостаточное внимание вопросам проектирования интерфейса программы могут значительно осложнить работу пользователя. В таком случае преимущества технической стороны продукта не всегда смогут компенсировать проблемы взаимодействия.

Для того чтобы избежать проблем при взаимодействии испытуемых и экспериментаторов с программно-аппаратным комплексом следует провести эргономическое проектирование продукта (раздел 2), которое включает анализ функций комплекса, формулирование эргономических требований, анализ алгоритмов работы пользователей и эргономическую оценку.

Создание комплекса для исследования динамики процесса заучивания согласно обозначенным требованиям призвано обеспечить процесс обучения на кафедре инженерной психологии и эргономики методическим материалом, уменьшить трудозатраты на выполнение лабораторного практикума студентами. Также созданный комплекс позволяет проводить научные исследования в области процессов запоминания [1].

1. **ЭРГОНОМИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА «ИССЛЕДОВАНИЕ ДИНАМИКИ ЗАУЧИВАНИЯ»**

**2.1 Анализ и распределение функций**

Предметом эргономического проектирования являются процесс, средства и условия деятельности человека-оператора или группы операторов. Результатом эргономического проектирования является эргономическое решение, которое представляет собой проект деятельности человека-оператора, выполненный при последовательной реализации эргономических требований с учетом специфики объекта проектирования [1].

Распределение функций между человеком и компьютером в проектируемом комплексе представлено в таблице 2.1.1

Таблица 2.1.1 – Распределение функций между человеком и компьютером в проектируемой СЧКС

|  |  |  |
| --- | --- | --- |
| Название функции | Что делает в системе человек | Что выполняет в системе компьютер |
| 1 | 2 | 3 |
| 1. Предъявлять на экране компьютера справку о программе (ФИО разработчика, ФИО научного руководителя); | Администратор или испытуемый нажимает соответствующую кнопку для просмотра справки; | Выводит на экран монитора диалоговое окно со справкой; |
| 1. Ограничивать допуск испытуемого к некоторым функциям, которые должен выполнять только преподаватель; | Администратор использует пароль для входа в систему; | Предоставляет доступ к защищенным функциям только при введенном правильно пароле; |
| 1. Позволять администратору создавать и сохранять в памяти компьютера опыты с предъявлением стимулов разных видов (слова, слоги, буквы); | Администратор создает последовательности стимулов для дальнейшей их демонстрации в ходе опытов; | Сохраняет последовательности в каком-либо электронном формате; |

Продолжение таблицы 2.1.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1. Позволять администратору выбирать для опыта вариант экспозиции и его настройки; | Администратор назначает, будут стимулы предъявляться последовательно или же одновременно, а также временные интервалы между предъявлениями; | Сохраняет настройки и обеспечивает предъявление в соответствии с ними; |
| 1. Позволять администратору редактировать вводные теоретические сведения; | Администратор редактирует текст сведений, импортирует или экспортирует его; | Сохраняет изменения в теоретических сведениях; |
| 1. Обеспечивать возможность просмотра преподавателем результатов выполненных студентами экспериментальных исследований; | Администратор выбирает результат для просмотра; | Отображает сохраненный результат; |
| 1. Позволять администратору редактировать базу, сохраняемых результатов работы студентов (удалять файлы, потерявшие актуальность); | Администратор выбирает данные для удаления; | Удаляет выбранные данные; |
| 1. Проводить регистрацию студента (испытуемого); | Испытуемый вводит своё ФИО, номер своей группы; | Сохраняет введенные данные для последующего использования. |
| 1. Предъявлять на экране компьютера вводные теоретические сведения (по запросу испытуемого); | Испытуемый нажимает соответствующую кнопку для показа вводных теоретических сведений; | Предъявляет вводные теоретические сведения; |

Продолжение таблицы 2.1.1

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1. Инструктировать испытуемого о предстоящем опыте и его задачах; | - | Система предъявляет инструкции перед началом выполнения; |
| 1. Проводить перед началом экспериментов тренировочные серии с возможностью выбора студентом момента ее завершения; | Испытуемый нажимает кнопку для тренировочной серии; | Программа проводит опыт не фиксируя результаты; |
| 1. Предъявлять на экране дисплея ряды стимулов; | Испытуемый выполняет опыт; | Предъявляет стимулы в соответствии с настройками; |
| 1. Ввод в компьютер запомненных стимулов; | Испытуемый вводит запомненные стимулы; | Программа проверяет полноту воспроизведения ряда стимулов и принимает решение об окончании опыта; |
| 1. Выполнение на ПК необходимую математическую обработку полученных в исследованиях результатов; | - | Система обрабатывает по заложенному в ней алгоритму результаты; |
| 1. Сохранение в памяти компьютера результаты работы испытуемого; | - | Программа сохраняет результаты в электронном виде; |
| 1. Предъявление на экране ПК результаты выполненного эксперимента; | - | ПАК показывает результаты на экране; |
| 1. Сохранение результатов эксперимента на переносном носителе информации; | Испытуемый вызывает функцию сохранения и указывает место назначения; | Программа сохраняет результаты в указанное место назначения; |
| 1. Прекращение работы на любом ее этапе. | Испытуемый или администратор закрывает программу; | Программа незамедлительно завершает работу. |

В результате проведенного анализа и распределения выполнения действий функций между человеком-оператором и проектируемым программно-аппаратным комплексом можно сделать следующие выводы:

* проектируемая система должна поддерживать работу с двумя типами пользователей: администратором и испытуемым;
* система должна ограничивать доступ пользователя к возможностям, предназначенным для другой пользовательской роли;
* на программно-аппаратный комплекс должны быть возложены функции, для которых функциональные возможности и степень эффективности выполнения действий компьютером выше, чем человеком.

Деятельность оператора в системе зависит от выполняемой им в ходе эксперимента роли, определяемой посредством выбора типа пользователя (студент, преподаватель) и вводом пароля для преподавателя. Преподаватель перед проведением исследования формирует задание для опытов, а после выполнения работы изучает результаты испытуемого. Студент вводит свои данные в форму регистрации, выбирает опыт и его настройки, читает инструкцию и проходит тренировочные серии до тех пор, пока ему не станет ясен смысл исследования, выполняет задание опыта, узнает итоги своей работы [2].

**2.2 Проектирование деятельности пользователей**

Алгоритмы пользователей – основа проекта деятельности в эргономическом проектировании. На их основе выполняется проработка сценариев взаимодействия человека и системы. В разрабатываемой системе два типа пользователей: испытуемый и администратор. Администратор имеет возможность просматривать результаты всех испытуемых, создавать и модифицировать тесты, удалять ненужные записи. Алгоритмы администратора представлены в таблицах 2.2.1-2.2.4.

Таблица 2.2.1 – Алгоритм работы человека в подсистеме «Администратор – ПК – среда» в режиме заполнения базы стимульного материала

|  |  |  |  |
| --- | --- | --- | --- |
| Блоки | Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1 | 2 | 3 | 4 |
| 1 | 1. Запуск программы | Ярлык на экране дисплея | Щелчок мышкой по ярлыку программы |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Сообщение об успешной аутентификации | Клавиатура |
| 2 | 1. Запуск модуля редактирования исходных наборов стимулов (если модуль уже запущен, необязательно) | Пункт меню «Ряды» | Щелчок мышкой |
| 1. Создание нового ряда | Кнопка «Создать новый». | Щелчок мышкой |
| 1. Ввод названия ряда | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод элементов нового ряда | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Сохранение ряда | Кнопка «Сохранить» | Щелчок мышкой |
| 1. Пункты 2-5 повторять пока не будет получено нужное количество рядов | | |
| 1. Выход из модуля редактирования исходных наборов стимулов (необязательно) | Кнопка «Закрыть» в углу окна «Ряды» | Щелчок мышкой |

Продолжение таблицы 2.2.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 3 | 1. Запуск модуля редактирования исходных наборов стимулов (если модуль уже запущен, необязательно) | Пункт меню «Ряды» | Щелчок мышкой |
| 1. Выбор ряда для редактирования | Элемент в списке доступных рядов | Щелчок мышкой |
| 1. Запуск редактирования выбранного ряда | Кнопка «Редактировать» | Щелчок мышкой |
| 1. Ввод названия ряда | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод элементов нового ряда | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Сохранение ряда | Кнопка «Сохранить» | Щелчок мышкой |
| 1. Пункты 2-6 повторять пока не будет получен желаемый результат редактирования | | |
| 1. Выход из модуля редактирования исходных наборов стимулов (необязательно) | Кнопка «Закрыть» в углу окна «Ряды» | Щелчок мышкой |
| 4 | 1. Запуск модуля редактирования исходных наборов стимулов (если модуль уже запущен, необязательно) | Пункт меню «Ряды» | Щелчок мышкой |
| 1. Выбор ряда для удаления | Элемент в списке доступных рядов | Щелчок мышкой |
| 1. Запуск удаления выбранного ряда | Кнопка «Удалить» | Щелчок мышкой |
| 1. Подтверждение удаления | Кнопка «Удалить» в окне подтверждения | Щелчок мышкой |
| Пункты 2-4 повторять пока не будут удалены все нежелательные ряды | | |
| 1. Выход из модуля редактирования (необязательно) | Кнопка «Закрыть» в углу окна «Ряды» | Щелчок мышкой |
| Примечание: Блоки операций 2, 3, 4 могут выполняться в различной последовательности 0 и более раз. | | | |

Таблица 2.2.2 – Алгоритм работы человека в подсистеме «Администратор – ПК – среда» в режиме задания настроек опытов

|  |  |  |  |
| --- | --- | --- | --- |
| Блоки | Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1 | 2 | 3 | 4 |
| 1 | 1. Запуск программы | Ярлык на экране дисплея | Щелчок мышкой по ярлыку программы |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Сообщение об успешной аутентификации | Клавиатура |
| 5 | 1. Запуск модуля редактирования возможных опытов (если модуль уже запущен, необязательно) | Пункт меню «Опыты» | Щелчок мышкой |
| 1. Создание нового опыта | Кнопка «Создать новый». | Щелчок мышкой |
| 1. Ввод названия опыта | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Выбор ряда | Элемент в списковом поле выбора | Клавиатура, мышь |
| 1. Выбор режима предъявления | Переключатель с пунктами «Последовательно» и «Одновременно» | Клавиатура, мышь |
| 1. Выбор времени экспозиции | Числовое поле | Клавиатура, мышь |
| 1. Выбор времени паузы | Числовое поле | Клавиатура, мышь |
| 1. Сохранение настроек опыта | Кнопка «Сохранить» | Щелчок мышкой |
| 1. Пункты 2-8 повторять пока не будет получено нужное количество опытов | | |
| 1. Выход из модуля редактирования возможных опытов (необязательно) | Кнопка «Закрыть» в углу окна «Опыты» | Щелчок мышкой |

Продолжение таблицы 2.2.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 6 | 1. Запуск модуля редактирования возможных опытов (если модуль уже запущен, необязательно) | Пункт меню «Опыты» | Щелчок мышкой |
| 1. Выбор опыта для редактирования настроек | Элемент в списке доступных опытов | Щелчок мышкой |
| 1. Запуск редактирования выбранного опыта | Кнопка «Редактировать» | Щелчок мышкой |
| 1. Ввод названия опыта | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Выбор ряда | Элемент в списковом поле выбора | Клавиатура, мышь |
| 1. Выбор режима предъявления | Групповой переключатель с пунктами «Последовательно» и «Одновременно» | Клавиатура, мышь |
| 1. Выбор времени экспозиции | Числовое поле | Клавиатура, мышь |
| 1. Выбор времени паузы | Числовое поле | Клавиатура, мышь |
| 1. Сохранение настроек опыта | Кнопка «Сохранить» | Щелчок мышкой |
| 1. Пункты 2-9 повторять пока не будет получен желаемый результат редактирования | | |
| 1. Выход из модуля редактирования возможных опытов (необязательно) | Кнопка «Закрыть» в углу окна «Опыты» | Щелчок мышкой |

Продолжение таблицы 2.2.2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 7 | 1. Запуск модуля редактирования возможных опытов (если модуль уже запущен, необязательно) | Пункт меню «Опыты» | Щелчок мышкой |
| 1. Выбор опыта для удаления | Элемент в списке доступных опытов | Щелчок мышкой |
| 1. Запуск удаления выбранного опыта | Кнопка «Удалить» | Щелчок мышкой |
| 1. Подтверждение удаления | Кнопка «Удалить» в окне подтверждения | Щелчок мышкой |
| Пункты 2-4 повторять пока не будут удалены все нежелательные опыты | | |
| 1. Выход из модуля редактирования (необязательно) | Кнопка «Закрыть» в углу окна «Опыты» | Щелчок мышкой |
| Примечание: Блоки операций 2, 3, 4 могут выполнятся в различной последовательности 0 и более раз. | | | |

Таблица 2.2.4 – Алгоритм работы человека в подсистеме «Администратор – ПК – среда» в режиме редактирования вводных теоретических сведений

|  |  |  |  |
| --- | --- | --- | --- |
| Блоки | Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1 | 1. Запуск программы | Ярлык на экране дисплея | Щелчок мышкой по ярлыку программы |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Сообщение об успешной аутентификации | Клавиатура |
| 8 | 1. Вставка текста из буфера обмена | Текстовое поле | Клавиатура, мышь |
| 1. Правка текста теоритических сведений | Текстовое поле | Клавиатура, мышь |
| 1. Сохранение сделанных правок | Кнопка «Сохранить» | Щелчок мышкой |
| 1. Выход из модуля теоретических сведений | Кнопка «Закрыть» в углу окна «Теория» | Щелчок мышкой |

Таблица 2.2.3 – Алгоритм работы человека в подсистеме «Администратор – ПК – среда» в режиме просмотра результатов

|  |  |  |  |
| --- | --- | --- | --- |
| Блоки | Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1 | 2 | 3 | 4 |
| 1 | 1. Запуск программы | Ярлык на экране дисплея | Щелчок мышкой по ярлыку программы |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Сообщение об успешной аутентификации | Клавиатура |
| 9 | 1. Запуск модуля просмотра результатов (если модуль уже запущен, необязательно) | Пункт меню «Результаты» | Щелчок мышкой |
| 1. Выбор результата (если элемент уже выбран, необязательно) | Элемент списка результатов | Щелчок мышкой |
| 1. Запуск модуля просмотра | Кнопка «Просмотр» | Щелчок мышкой |
| 1. Выход из модуля просмотра результатов (необязательно) | Кнопка «Закрыть» в углу окна «Результаты» | Щелчок мышкой |
| 10 | 1. Запуск модуля просмотра результатов (если модуль уже запущен, необязательно) | Пункт меню «Результаты» | Щелчок мышкой |
| 1. Выбор результата (если элемент уже выбран, необязательно) | Элемент списка результатов | Щелчок мышкой |
| 1. Запуск диалога сохранения | Кнопка «Сохранить» | Щелчок мышкой |
| 1. Выбор места сохранения и подтверждение | Диалог сохранения файла | Клавиатура, мышь |
| 1. Выход из модуля просмотра результатов (необязательно) | Кнопка «Закрыть» в углу окна «Результаты» | Щелчок мышкой |

Продолжение таблицы 2.2.4

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 11 | 1. Запуск модуля просмотра результатов (если модуль уже запущен, необязательно) | Пункт меню «Результаты» | Щелчок мышкой |
| 1. Выбор результата (если элемент уже выбран, необязательно) | Элемент списка результатов | Щелчок мышкой |
| 1. Запуск удаления | Кнопка «Удалить» | Щелчок мышкой |
| 1. Подтверждение удаления | Кнопка «Удалить» в окне подтверждения | Щелчок мышкой |
| 1. Выход из модуля просмотра результатов (необязательно) | Кнопка «Закрыть» в углу окна «Результаты» | Щелчок мышкой |
| Примечание: Блоки операций 2, 3, 4 могут выполнятся в различной последовательности 0 и более раз. | | | |

Испытуемый – менее привилегированная роль, она назначается выполняющим опыты. Испытуемые могут изучать краткие теоритические сведения, просматривать справку о программе, выбрать и провести доступные опыты, просмотреть и сохранить результаты. Перед проведением опыта испытуемый должен иметь возможность попробовать взаимодействовать с программно-аппаратным комплексом без сохранения результата. Этот режим называется тренировочной серией. Тренировочная серия должна завершаться по требованию испытуемого. После прохождения тренировочной серии испытуемый может приступить к выполнению опыта в основном режиме – в рабочей серии [1]. Алгоритмы испытуемого описаны в таблицах 2.2.5 – 2.2.8

.

Таблица 2.2.5 – Алгоритм работы человека в подсистеме «Испытуемый – ПК – среда» (в случае прохождения тренировочной серии)

|  |  |  |  |
| --- | --- | --- | --- |
| Блоки | Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1 | 1. Запуск программы | Ярлык на экране дисплея | Щелчок мышкой по ярлыку программы |
| 1. Выбор режима «Испытуемый» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод фамилии | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод имени | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод отчества | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод номера группы | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Нажатие кнопки «Вход» | Сообщение об успешной аутентификации | Мышь |
| 12 | 1. Запуск модуля просмотра кратких теоретических сведений | Пункт меню «Теория» или кнопка «Теория» | Щелчок мышкой |
| 1. Выбор опыта | Элемент в списке доступных опытов | Щелчок мышкой |
| 1. Запуск модуля опытов | Кнопка «Начать» | Щелчок мышкой |
| 1. Получение инструктажа | Текстовый блок в окне программы | - |
| 13 | Прохождение тренировочной серии | | |
| Завершение тренировочной серии | | |
| Примечания:  Выход из приложения доступен в любой момент выполнения опыта. | | | |

Таблица 2.2.6 – Алгоритм работы человека в подсистеме «Испытуемый – ПК – среда» (в случае прохождения рабочей серии)

|  |  |  |  |
| --- | --- | --- | --- |
| Блоки | Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1 | 1. Запуск программы | Ярлык на экране дисплея | Щелчок мышкой по ярлыку программы |
| 1. Выбор режима «Испытуемый» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод фамилии | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод имени | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод отчества | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Ввод номера группы | Текстовое поле для ввода | Клавиатура, мышь |
| 1. Нажатие кнопки «Вход» | Сообщение об успешной аутентификации | Мышь |
| 12 | 1. Запуск модуля просмотра кратких теоретических сведений | Пункт меню «Теория» или кнопка «Теория» | Щелчок мышкой |
| 1. Выбор опыта | Элемент в списке доступных опытов | Щелчок мышкой |
| 1. Запуск модуля опытов | Кнопка «Начать» | Щелчок мышкой |
| 1. Получение инструктажа | Текстовый блок в окне программы | - |
| 13 | Прохождение рабочей серии | | |
| Завершение рабочей серии | | |
| 14 | Сохранение результатов | | |
| Примечания:  Выход из приложения доступен в любой момент выполнения опыта. | | | |

Таблица 2.2.7 – Алгоритм работы человека в подсистеме «Испытуемый – ПК – среда» (блок прохождения тренировочной или рабочей серии)

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1. Запуск модуля тренировочной или рабочей серии | Кнопка «Рабочая серия» или кнопка «Тренировочная серия» | Щелчок мышкой |
| 1. Ознакомление с инструктажем | Текстовый блок в окне программы | - |
| 1. Запуск серии | Кнопка «Начать» | Щелчок мышкой |
| 1. Предъявление ряда стимулов | Блок в окне программы | - |
| 1. Ввод запомненных стимулов | Поля для ввода стимулов | Клавиатура, мышь |
| Пункты 4-5 повторять пока не будет получено точное совпадение со стимулами | | |

Таблица 2.2.8 – Алгоритм работы человека в подсистеме «Испытуемый – ПК – среда» (блок сохранения результатов)

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к средствам отображения информации | Обращение к органам управления |
| 1. Запуск диалога сохранения | Кнопка «Сохранить как…» | Щелчок мышкой |
| 1. Выбор места сохранения | Окно файлового диалога | Клавиатура, мышь |
| 1. Подтверждение | Кнопка «Сохранить» диалога | Щелчок мышкой |
| 1. Выход из приложения | Кнопка «Закрыть» окна программы | Щелчок мышкой |

Приготовленные алгоритмы отражены в прилжении В и приложении Г в виде блок-схем. Данные алгоритмы положены в основу построения сценария взаимодействия с пользователем.

**2.3 Разработка средств деятельности пользователей**

Эргономические требования к СЧКС – это требования к системе в целом, ее отдельным подсистемам, оборудованию, рабочей среде, определяемые свойствами человека и устанавливаемые для обеспечения его эффективной и безопасной деятельности. Они формируются на основании экспериментальных исследований и опыта эксплуатации СЧКС, требований эргономических стандартов [1]. Для программно-аппаратного комплекса «Исследование динамики процесса заучивания» можно предъявить требования, представленные в таблице 2.2.1.

Таблица 2.1.2 – Эргономические требования к проектируемой системе

|  |  |
| --- | --- |
| Группа эргономических требований | Номенклатура эргономических требований |
| Психологические ЭТ | * Соответствие цветов надписей и знаков стереотипам восприятия; * соответствие объемов информации, требующей запоминания, возможностям памяти человека; * отсутствие неоднозначного толкования требований, инструкций и команд; * соответствие компоновки ОУ и СОИ стереотипам восприятия; * соответствие индикации срабатывания ОУ сформированным навыкам, наличие индикации хода выполнения функции; * один и тот же характер команд на протяжении всего периода работы в системе в схожих ситуациях; * наличие указаний на проблемы, возникающие в процессе обслуживания системы;   + наличие предупреждений о нежелательных последствиях некоторых действий соответствие сложности инструкций, времени, отводимому на их усвоение;   + возможность убедиться в понимании процесса предстоящего опыта * возможность завершить программу на любом этапе взаимодействия пользователя и ПАК. |

Продолжеие таблицы 2.1.2

|  |  |
| --- | --- |
| Психофизиологические ЭТ | * Cоответствие размеров знаков на экране дисплея оперативному порогу зрения человека; * соответствие контраста знаков и фона оптимальным условиям восприятия; * соответствие вида контраста знаков и фона уровню освещенности рабочего места;   + соответствие расположения надписей условиям их оптимального считывания; * отсутствие требований, связанных с обслуживанием системы, несоответствующих возможностям органов чувств человека. |
| Физиологические ЭТ | * Соответствие компоновки ОУ принципам экономии рабочих движений; * Соответствие требований к скорости движений возможностям человека; * Соответствие времени экспозиции списков, меню, кнопок скоростным возможностям человека * Использование крутилок для ввода числовых значений * Использование значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации. * Отсутствие требований к пользователям вводить информацию, которая была предварительно введена или которая может быть автоматически получена из системы |
| Антропометрические ЭТ | * Соответствие размеров зон управления и обслуживания антропометрическим характеристикам человека соответствие размеров рабочего стола антропометрическим характеристикам человека; * соответствие размеров рабочего кресла антропометрическим характеристикам человека ‒ соответствие размеров зон обслуживания антропометрическим характеристикам человека. |
| Социально-психологические ЭТ | * Наличие средств ограничения допуска к некоторым функциям пользователям, не имеющим требуемого статуса. |
| Гигиенические ЭТ | * Соответствие параметров рабочей среды гигиеническим нормативам |

На следующем этапе проектирования средств деятельности проводится моделирование сценариев взаимодействия пользователей и системы путем создания графических эскизов работы программы [1].

**Сценарий работы администратора** представлен на рисунках 1-11.

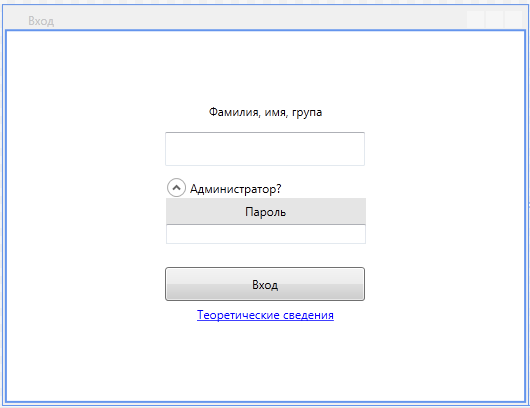


Рисунок 2.3.1 – Окно входа в систему

При входе в систему пользователь подтверждает свои права администратора путем ввода пароля.

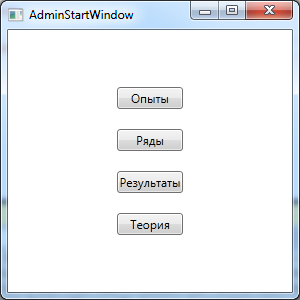


Рисунок 2.3.2 – Выбор действий для администратора

Система предъявляет доступные функции (рис. 2.3.2).

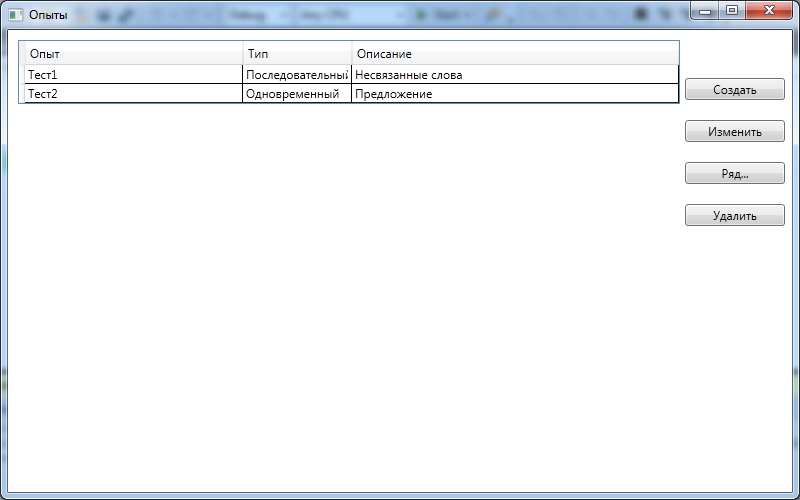


Рисунок 2.3.3 – Окно работы с опытами

В ходе работы с наборами опытов администратор имеет возможность добавлять, изменять, удалять опыты. Так же возможен переход к просмотру рядов стимулов (рис. 2.3.3).

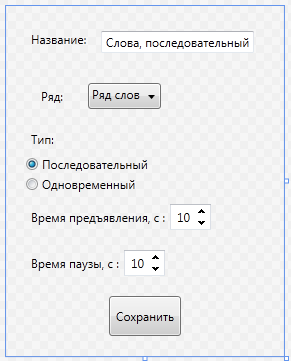


Рисунок 2.3.4 – Окно создания и редактирования опыта

Изменение или редактирование опыта изображено на рисунке 2.3.4.

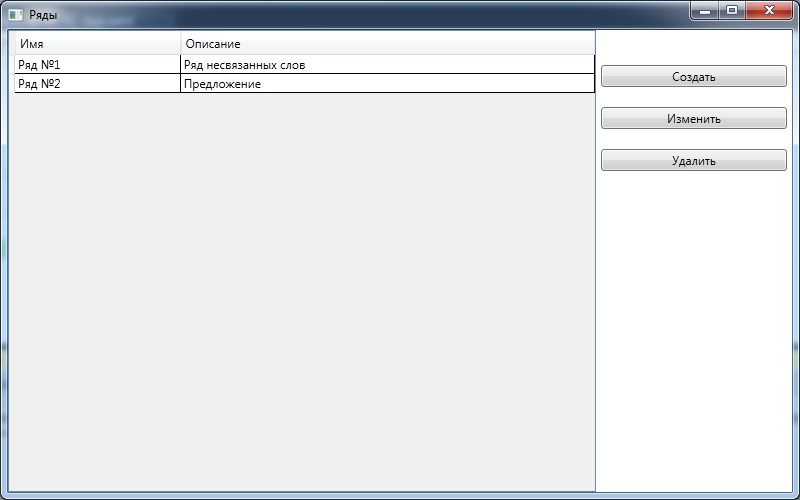


Рисунок 2.3.5 – Окно работы с рядами

В ходе работы с рядами стимулов администратор имеет возможность добавлять, изменять, удалять ряды (рис. 2.3.5).

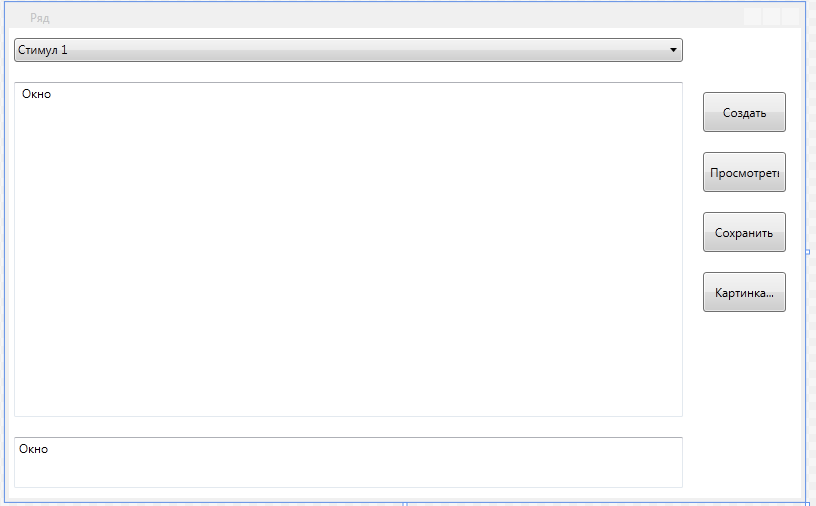


Рисунок 2.3.6 – Окно создания и редактирования ряда стимулов

Изменение или редактирование ряда изображено на рисунке 2.3.6.

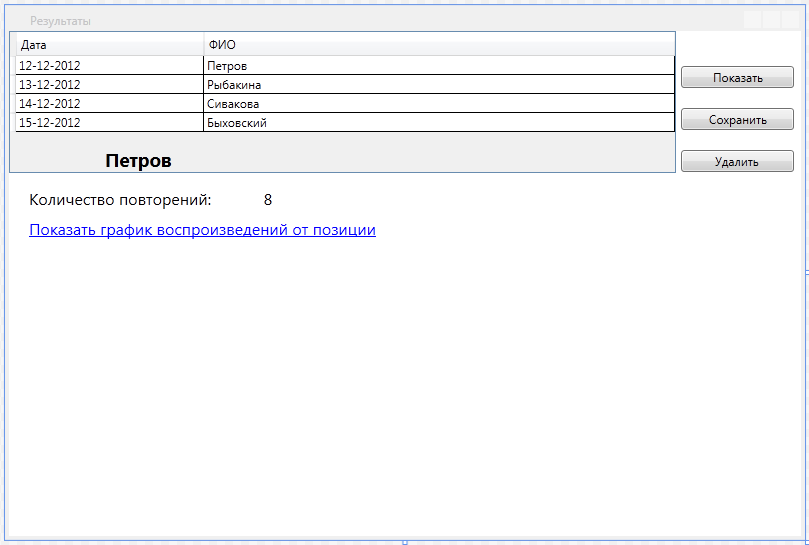


Рисунок 2.3.7 – Окно результатов

Администратор имеет доступ к результатам опытов всех испытуемых, сохраненным в системе (рис. 2.3.7)

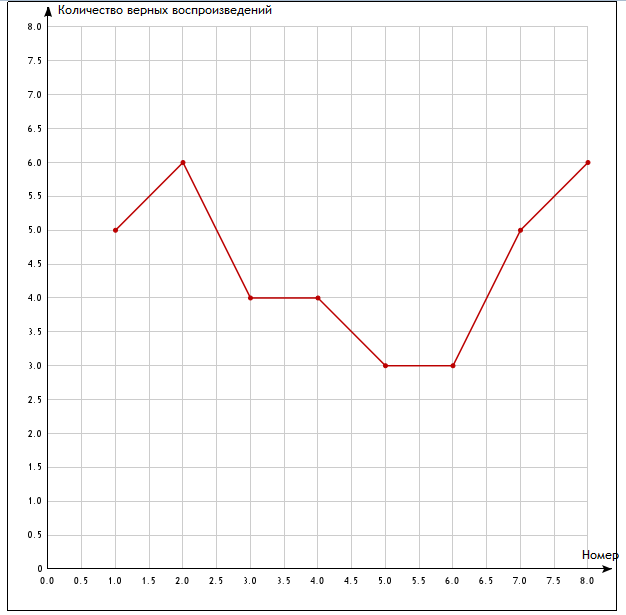


Рисунок 2.3.8 – Отображение результата в виде графика

На рисунке 2.3.8 изображено представление результата опыта в виде графика.

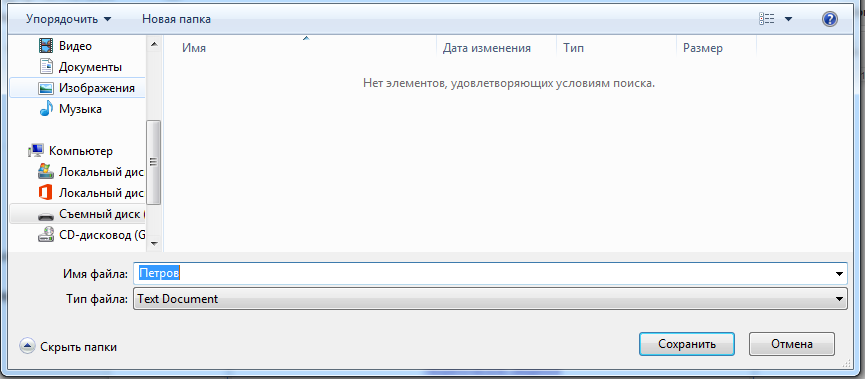


Рисунок 2.3.9 – Сохранение результатов

Результат в виде краткой сводки в текстовом файле может быть сохранен на внешний носитель (рис. 2.3.9).

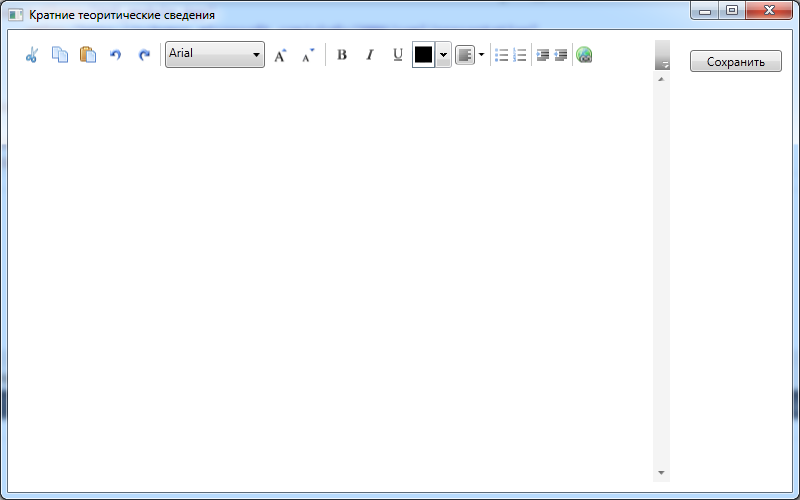


Рисунок 2.3.10 – Окно правки теоретических сведений

Краткие теоретические сведения могут быть изменены администратором (рис. 2.3.10).

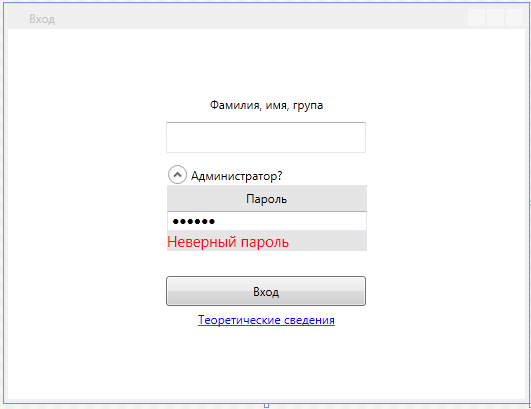


Рисунок 2.3.11 – Обработка ошибочных действий пользователя

Система должна предупреждать пользователя о возникающих ошибках в ходе работы. Так, при вводе неверного пароля для входа в систему как администратор доступ не должен быть предоставлен, а пользователь должен быть уведомлен о неверном пароле (рис. 2.3.11).

**Сценарий работы испытуемого** представлен на рисунках 2.3.12-2.3.24.

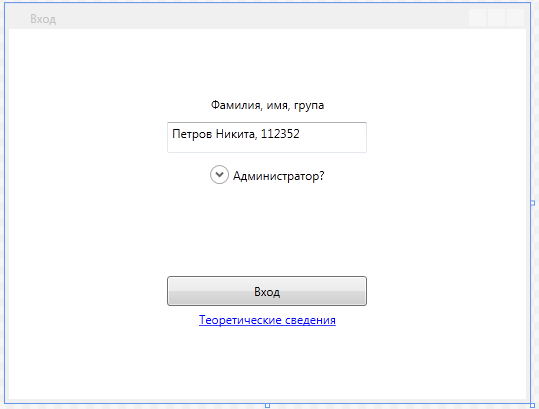


Рисунок 2.3.12 – Вход испытуемого

Испытуемый представляется системе (рис. 2.3.12). Для этого могут быть использованы его имя, фамилия, номер группы.

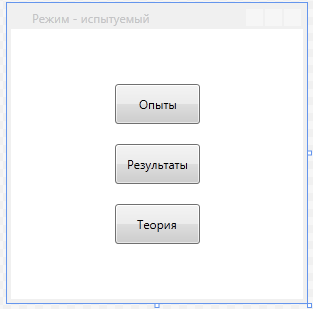


Рисунок 2.3.13 – Режим работы испытуемого

Система предъявляет доступные функции (рис. 2.3.13).

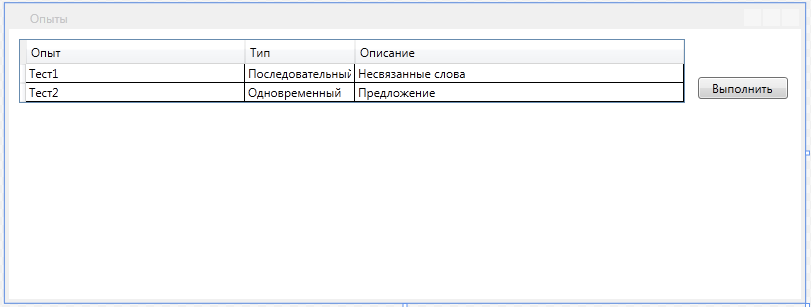


Рисунок 2.3.14 – Выбор опыта для выполнения

Испытуемый выбирает опыт для выполнения (рис. 2.3.14)



Рисунок 2.3.15 – Вводный инструктаж

Далее система предъявляет вводный инструктаж к опыту (рис. 2.3.15)



Рисунок 2.3.16 – Предъявление текстовых стимулов в последовательном режиме

Стимулы могут предъявляться последовательно по одному через заданный интервал времени или одновременно на одном экране приложения.  
Последовательный вариант предъявления представлен на рисунке 2.3.16.

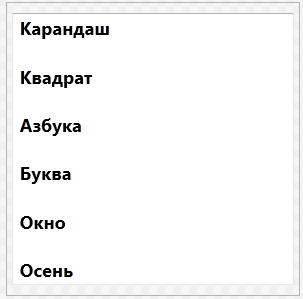


Рисунок 2.3.17 – Предъявление текстовых стимулов в одновременном режиме

Одновременный вариант предъявления представлен на рисунке 2.3.17.



Рисунок 2.3.18 – Предъявление графических стимулов в последовательном режиме

Стимулы могут предъявляться и в форме изображений (рис. 2.3.18).



Рисунок 2.3.19 – Предъявление графических стимулов в одновременном режиме

Пример одновременного предъявления стимулов-изображений представлен на рисунке 2.3.19.

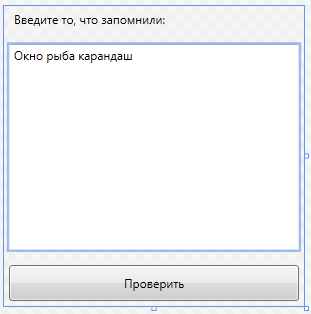


Рисунок 2.3.20 – Ввод стимулов, которые запомнил пользователь

Ввод запомненных стимулов осуществляется словам, разделенными знаком пробела (рис. 2.3.20).

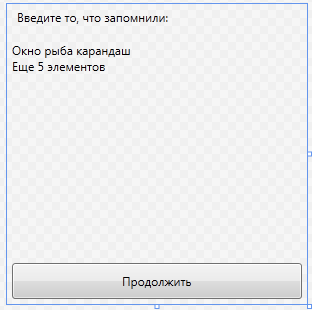


Рисунок 2.3.21 – Проверка стимулов, которые запомнил пользователь

После ввода запомненных стимулов система проверяет, все ли стимулы запомнены испытуемым. В случае успешного запоминания система переходит к окну вывода результатов. В противном случае предъявление стимулов повторяется (рис. 2.3.21).

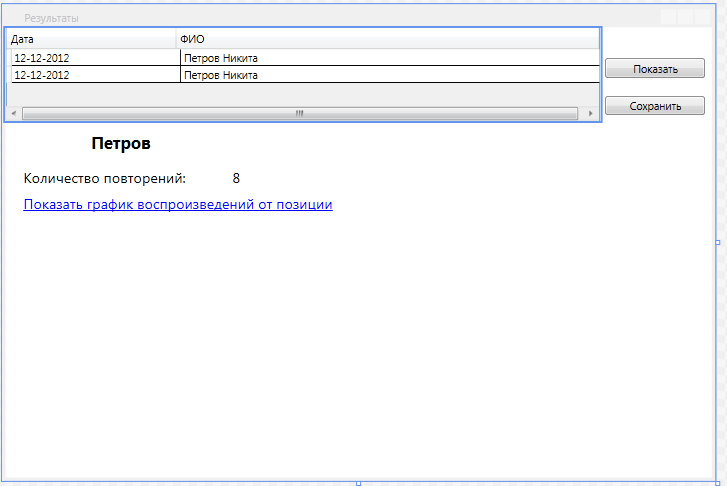


Рисунок 2.3.22 – Доступные пользователю результаты

После завершения опыта испытуемому доступны лишь результаты его опытов (рис. 2.3.22).

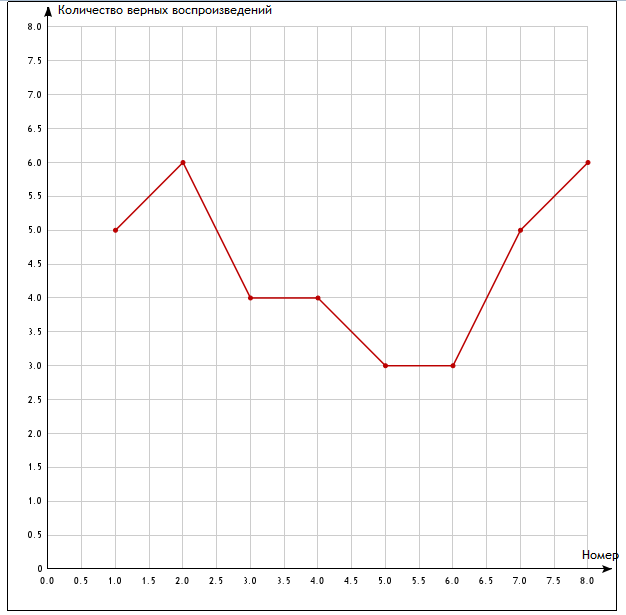


Рисунок 2.3.22 – Отображение результата в виде графика

На рисунке 2.3.22 изображено представление результата опыта в виде графика.

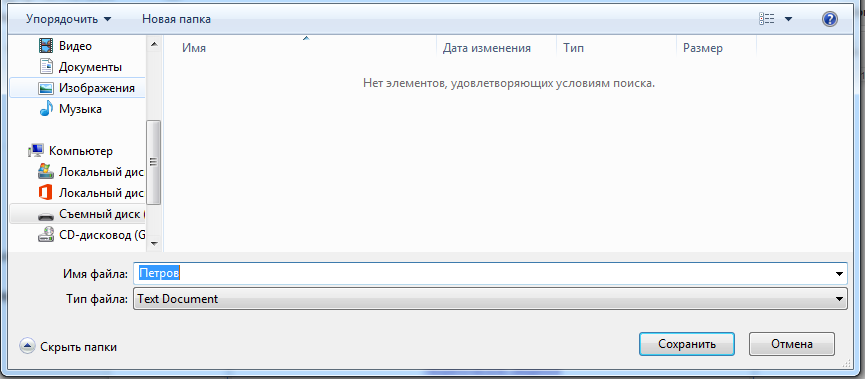


Рисунок 2.3.23 – Сохранение результатов

Результат в виде краткой сводки в текстовом файле может быть сохранен на внешний носитель (рис. 2.3.23).

Групповой эргономический показатель **ЭПгр** рассчитывается как общая оценка по группе единичных показателей:

**ЭПгр = ∑ 1 / (∑1 + ∑ 0)**, (2.3.1)

где **∑ 1** — суммарное число случаев, когда имеет место соответствие единичных показателей эргономическим требованиям;

**∑ 0**- суммарное число случаев, когда соответствия нет.

Сами единичные и групповые эргономических показатели и результаты оценки значений приведены в таблицах 2.3.2 и 2.3.3.

Поскольку для системы значимым является только одно свойство – «управляемость» именно это свойство будет определять эргономичность системы в целом. Эргономические свойства СЧМ определяются как некоторая совокупность групповых эргономических показателей, при этом чаще всего применяется аддитивная функция:

**ЭСВ = ∑ αнi \* ЭПгрj,** (2.3.2)

где αнi – нормированные весовые коэффициенты, сумма которых должна быть равна единице, т.е.

∑ αнi = 1.

Для оцениваемого эргономического свойства «управляемость» выбираем величины весовых коэффициентов (см. таблицу 2.3.4).

Таблица 2.3.2 – Общие эргономические требования к проектируемой системе и их единичные эргономические показатели

|  |  |  |
| --- | --- | --- |
| **Группа** | **Эргономические требования** | **Единичные эргономические показатели** |
| 1 | 2 | 3 |
| **Психо-физио-логи-ческие** | **ПФ-1 —** Cоответствие размеров знаков на экране дисплея оперативному порогу зрения человека | Размеры шрифта текста и знаков |
| **ПФ-2** – соответствие контраста знаков и фона оптимальным условиям восприятия | Величина контраста знаков и фона |
| **ПФ-3** – соответствие вида контраста знаков и фона уровню освещенности рабочего места | Вид контраста знаков и фона |
| **ПФ-4** – соответствие расположения надписей условиям их оптимального считывания | Расположение и ориентация надписей на экране дисплея |
| **ПФ-5 —** отсутствие требований, связанных с обслуживанием системы, несоответствующих возможностям органов чувств человека | Соответствие привычным нормам |
| **Психо-логи-ческие** | **П-1** – Соответствие цветов надписей и знаков стереотипам восприятия; | Цвета знаков, кнопок, надписей |
| **П-2** – соответствие объемов информации, требующей запоминания, возможностям памяти человека; | Длина инструкции и время ее экспозиции |
| **П-3** – отсутствие неоднозначного толкования требований, инструкций и команд; | Словарный состав текстовых инструкций |
| **П-4** – соответствие компоновки ОУ и СОИ стереотипам восприятия; | Компоновка ОУ и СОИ |
| **П-5** – соответствие индикации срабатывания ОУ сформированным навыкам, наличие индикации хода выполнения функции; | Обратная связь при взаимодействии с ОУ, наличие и вид индикатора выполнения |

Продолжение таблицы 2.3.2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| **Психо-логи-ческие** | **П-6** – один и тот же характер команд на протяжении всего периода работы в системе в схожих ситуациях; | Названия пунктов меню, содержание информационных сообщений и сообщений о ошибках |
| **П-7** – наличие указаний на проблемы, возникающие в процессе обслуживания системы; | Сообщения об ошибочных действиях пользователей |
| **П-8** – наличие предупреждений о нежелательных последствиях некоторых действий соответствие сложности инструкций, времени, отводимому на их усвоение; | Предупреждения о возможных нежелательных действиях |
| **П-9** – возможность убедиться в понимании процесса предстоящего опыта | Наличие возможности проведения тренировочной серии |
| **П-10** – возможность завершить программу на любом этапе взаимодействия пользователя и ПАК. | ОУ для завершения работы на любом этапе |
| **Физио-логичес-кие** | **Ф-1** – Соответствие компоновки ОУ принципам экономии рабочих движений; | Размеры меню, списков, кнопок на экране дисплея |
| **Ф-2 —** соответствие требований к скорости движений возможностям человека; | Соответствующие временные характеристики ОУ и СОИ |
| **Ф-3 —** Соответствие времени экспозиции списков, меню, кнопок скоростным возможностям человека | Длительность экспозиции средств взаимодействия |
| **Ф-4 —** Использование крутилок для ввода числовых значений | Наличие крутилок для ввода числовых значений |
| **Ф-5 —** Использование значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации. | Используемые значения по умолчанию |
| **Ф-6 —** Отсутствие требований к пользователям вводить информацию, которая была предварительно введена или которая может быть автоматически получена из системы | Отсутствие необходимости вводить информацию, которая была ранее введена или которая может быть автоматически получена из системы |
|  |  |  |

Продолжение таблицы 2.3.2

|  |  |  |
| --- | --- | --- |
| **Гигие-нические** | **Г-1** – Соответствие параметров рабочей среды гигиеническим нормативам | Энергетические и временные параметры изображения на экране дисплея |
| **Социально-психо-логические** | **СП-1** – Наличие средств ограничения допуска к некоторым функциям пользователям, не имеющим требуемого статуса. | Способ разграничения прав пользователей разных типов |

Таблица 2.3.3 – Значения единичных и групповых эргономических показателей проектируемой системы

|  |  |  |
| --- | --- | --- |
| **Группа ЭП** | **Значения единичных ЭП** | **Значения групповых ЭП** |
| Психофизиологические ЭП | ПФ-1, ПФ-2, ПФ-3, ПФ-4, ПФ-5 = 1 | 5 х 1 / 5 = 1 |
| Психологические ЭП | П-6, П-8 = 0  П-1, П-2, П-3, П-4, П-5, П-7, П-9, П-10 = 1 | 8 х 1 / 10 = 0,8 |
| Физиологические ЭП | Ф-5 = 0  Ф-1, Ф-2, Ф-3, Ф-4, Ф-6 = 1 | 5\* 1 / 6 = 0,83 |
| Гигиенические ЭП | Г-1 = 1 | 1 \* 1 / 1 = 1 |
| Социально-психологические ЭП | СП-1 =1 | 1 \* 1 / 1 = 1 |
| Антропометрические ЭП | Не актуальны для данной СЧМ |  |

Таблица 2.3.4 – Значения весовых коэффициентов для оценки эргономического свойства «управляемость»

|  |  |
| --- | --- |
| **Групповой ЭП** | **Значение весового коэффициента** |
| Психофизиологический | 0,25 |
| Психологический | 0,4 |
| Физиологический | 0,15 |
| Гигиенический | 0,1 |
| Социально-психологический | 0,1 |

**Эргономическое свойство «управляемость»** с учетом весовых коэффициентов равно 0,89. Это означает, что система имеет хорошую управляемость, а вместе с тем и эргономичность.

После такой общей оценки производится анализ единичных показателей, значения которых не соответствуют эргономическим требованиям (получили "нулевые" оценки) и намечаются мероприятия по рационализации оцениваемой системы. Результаты данного этапа представлены в таблице 2.3.5.

Таблица 2.3.5 – Рекомендации по улучшению эргономичности проектируемой системы

|  |  |
| --- | --- |
| **Невыполненное эргономическое требование** | **Предложение по улучшению эргономичности** |
| **П-6** – один и тот же характер команд на протяжении всего периода работы в системе в схожих ситуациях; | Провести согласование набора команд на протяжении всего периода работы в системе |
| **П-8** – наличие предупреждений о нежелательных последствиях некоторых действий соответствие сложности инструкций, времени, отводимому на их усвоение; | Предусмотреть и обнаружить возможные нежелательные действия пользователя и реализовать предупреждения |
| **Ф-5 —** Использование значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации. | Использовать значения по умолчанию где это необходимо |

**3 Разработка программно-аппаратного комплекса исследования динамики заучивания**

* 1. **Разработка структуры базы данных**

Реляционная база данных представляет собой набор таблиц для хранения абстрактных объектов (сущностей). Таблицы состоят из колонок и строк (кортежей). Внутри таблиц могут быть определены ограничения, между таблицами существуют отношения. Для описания модели хранимых данных применяются диаграммы баз данных [рис. 3.1.1]. При помощи языка запросов можно выполнять запросы, которые возвращают наборы данных, получаемых из одной или нескольких таблиц [9].

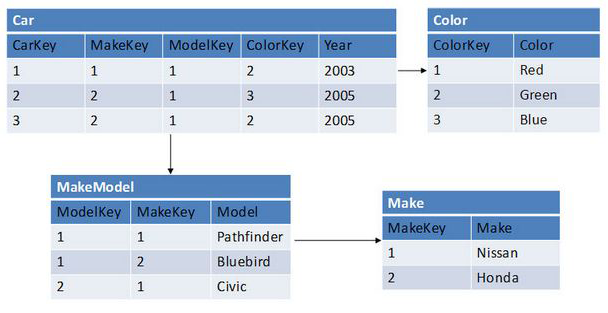


Рисунок 3.1.1 – Модель типичной реляционной базы данных

Структурированный Язык запросов (Structured Query Language — SQL) — это стандартный язык доступа к БД, таким как SQL Server, Oracle, MySQL, Sybase и Access. Знание SQL необходимо всем, кто хотел бы хранить и извлекать данные из БД [9].

SQL предназначен для выполнения запросов. Кроме того, в SQL входит синтаксис для обновления, вставки и уничтожения данных. Этот синтаксис вместе с командами обновления формирует язык управления данными (DML):

* SELECT — извлекает данные из таблицы БД
* UPDATE — обновляет данные в таблице БД
* DELETE — уничтожает данные в таблице БД
* INSERT INTO — вставляет новые данные в таблицу БД [11].

SQL Data Definition Language (DDL) — язык определения данных. DDL является частью SQL, которая управляет созданием и удалением таблиц в БД, Кроме того, с помощью DDL мы можем назначать индексы (ключевые слова), налаживать взаимосвязи между таблицами и накладывать ограничения на таблицы БД.

Важнейшими командами DDL являются следующие команды:

* CREATE TABLE — создание новой таблицы
* ALTER TABLE — изменение существующей таблицы
* DROP TABLE — удаление таблицы
* CREATE INDEX — создание индекса
* DROP INDEX — удаление индекса [11].

В рамках одного запроса данные получаются из нескольких таблиц путем их соединения (JOIN), чаще всего для соединения используются те же колонки, которые определяют отношения между таблицами. Нормализация — это процесс структурирования модели данных, обеспечивающий связность и отсутствие избыточности в данных [10].

Доступ к реляционным базам данных осуществляется через реляционные системы управления базами данных (РСУБД). Почти все системы баз данных, которые мы используем, являются реляционными, такие как Oracle, SQL Server, MySQL, Sybase, DB2, TeraData и так далее [11].

На протяжении всего существования реляционных БД они постоянно предлагали наилучшую смесь простоты, устойчивости, гибкости, производительности, масштабируемости и совместимости в сфере управлении данными [12].

Однако чтобы обеспечить все эти особенности, реляционные хранилища невероятно сложны внутри. Например, простой SELECT запрос может иметь сотни потенциальных путей выполнения, которые оптимизатор оценит непосредственно во время выполнения запроса. Все это скрыто от пользователей, однако внутри РСУБД создает план выполнения, основывающийся на вещах вроде алгоритмов оценки стоимости и наилучшим образом отвечающий запросу [11].

Одним из важнейших путей достижения высокой производительности SQL Server является использование индексов. Индекс ускоряет процесс запроса, предоставляя быстрый доступ к строкам данных в таблице, аналогично тому, как указатель в книге помогает вам быстро найти необходимую информацию [10].

Индексы создаются для столбцов таблиц и представлений. Индексы предоставляют путь для быстрого поиска данных на основе значений в этих столбцах. Например, если создаётся индекс по первичному ключу, а затем будет производится поиск строки с данными, используя значения первичного ключа, то SQL Server сначала найдет значение индекса, а затем использует индекс для быстрого нахождения всей строки с данными. Без индекса будет выполнен полный просмотр (сканирование) всех строк таблицы, что может оказать значительное влияние на производительность [10].

Создать индекс на большинстве столбцов таблицы или представления. Исключением, преимущественно, являются столбцы с типами данных для *хранения больших объектов (LOB), таких как image, text или varchar(max).* Можно создать индексы на столбцах, предназначенных для хранения данных в формате *XML*, но эти индексы устроены иначе чем стандартные [12].

Индекс состоит из набора страниц, узлов индекса, которые организованы в виде древовидной структуры — *сбалансированного дерева*. Эта структура является иерархической по своей природе и начинается с корневого узла на вершине иерархии и конечных узлов, листьев, в нижней части [рис. 3.1.2].

При формировании запроса на индексированный столбец, подсистема запросов начинает идти сверху от корневого узла и постепенно двигается вниз через промежуточные узлы, при этом каждый слой промежуточного уровня содержит более детальную информацию о данных. Подсистема запросов продолжает двигаться по узлам индекса до тех пор, пока не достигнет нижнего уровня с листьями индекса [12].

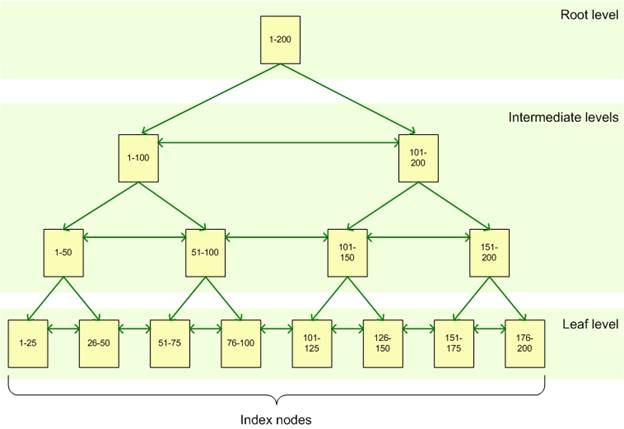


Рисунок 3.1.2 – Структура индекса базы данных

К примеру, если ищется значение 123 в индексированном столбе, то подсистема запросов сначала на корневом уровне определит страницу на первом промежуточном (intermediate) уровне. В данном случае первой страница указывает на значение от 1 до 100, а вторая от 101 до 200, таким образом подсистема запросов обратится ко второй странице этого промежуточного уровня. Далее будет выяснено, что следует обратиться к третьей странице следующего промежуточного уровня. Отсюда подсистема запросов прочитает на нижнем уровне значение самого индекса [рис. 3.1.2].

При анализе предметной области были выделены данные, подлежащие хранению в базе данных:

* Информация о пользователе
* Информация об опытах
* Информация о стимулах
* Результаты выполнения опытов [рис. 3.1.3].

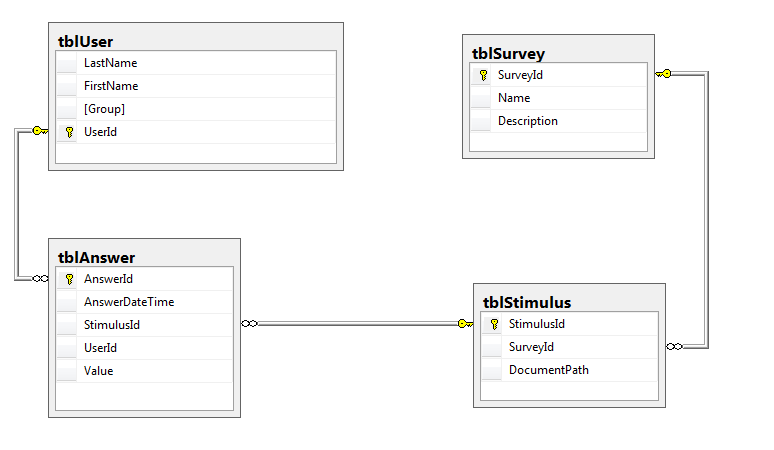


Рисунок 3.1.3 Основные сущности базы данных

Дополнительно было принято решение сохранить в базе данных настройки опытов. Настройки представляют собой абстрактную сущность, хранящую информацию о времени предъявления стимула, последовательности предъявления стимула, а также имеющую связь (ссылку) на набор стимулов. В базу данных был введен новая сущность [рис. 3.1.4]

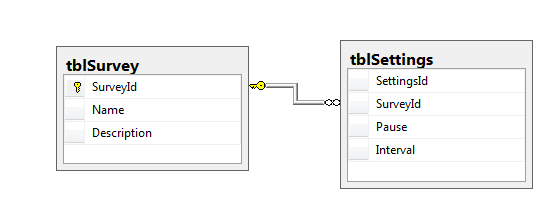


Рисунок 3.1.4 – Схема хранения настроек в базе данных.

Выполненный проект базы данных отражает основные свойства предметной области и при необходимости может быть модифицирован на этапе разработки программного модуля.

**3.2 Разработка структуры программы**

Модель-представление-контроллер (MVC — Model-view-controller) — это фундаментальный паттерн, который нашел применение во многих технологиях, дал развитие новым технологиям и каждый день облегчает жизнь разработчикам. Впервые паттерн MVC появился в языке smalltalk [12]. Разработчики должны были придумать архитектурное решение, которое позволяло бы отделить графический интерфейс от бизнес логики, а бизнес логику от данных. Таким образом, в классическом варианте, MVC состоит из трех частей, которые и дали ему название [13].

Под моделью, обычно понимается часть, содержащая в себе функциональную бизнес-логику приложения. Модель должна быть полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Достигается результат, позволяющий менять представление данных, то как они отображаются, не трогая саму модель [12].

Модель обладает следующими признаками:

* Модель — это бизнес-логика приложения;
* Модель обладает знаниями о себе самой и не знает о контроллерах и представлениях;
* Для некоторых проектов модель — это просто слой данных (dao, база данных, xml-файл);
* Для других проектов модель — это менеджер базы данных, набор объектов или просто логика приложения[13].

В обязанности представления входит отображение данных полученных от модели. Однако, представление не может напрямую влиять на модель. Можно говорить, что представление обладает доступом «только на чтение» к данным [14].

Примеры представления: html-страница, wpf форма, windows form.  
Представление обладает следующими признаками:

* В представлении реализуется отображение данных, которые получаются от модели любым способом;
* В некоторых случаях, представление может иметь код, который реализует некоторую бизнес-логику [15][16].

Наиболее распространенные виды mvc-паттерна, это:

* Model-view-controller [рис. 3.2.1]
* Model-view-presenter [рис. 3.2.2]
* Model-view-view model [рис. 3.2.3]

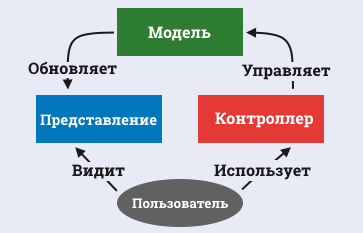


Рисунок 3.2.1 – Шаблон проектирования «Модель-представление-контроллер»

Основная идея этого паттерна в том, что и контроллер, и представление зависят от модели, но модель никак не зависит от этих двух компонент.  
**Признаки контроллера:**

* Контроллер определяет, какие представление должно быть отображено в данный момент;
* События представления могут повлиять только на контроллер. Контроллер может повлиять на модель и определить другое представление.
* Возможно несколько представлений только для одного контроллера [17].

При реализации контроллер перехватывает событие извне и в соответствии с заложенной в него логикой, реагирует на это событие изменяя модель, посредством вызова соответствующего метода. После изменения модель использует событие о том, что она изменилась, и все подписанные на это события представления, получив его, обращаются к модели за обновленными данными, после чего их и отображают. Пример использования: **mvc asp.net** [17].

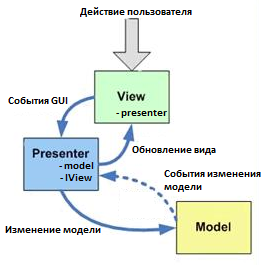


Рисунок 3.2.2 – Шаблон проектирования «Модель-представление-представитель»

Подход «Модель-представление-представитель» позволяет создавать абстракцию представления. Для этого необходимо выделить интерфейс представления с определенным набором свойств и методов. Презентер, в свою очередь, получает ссылку на реализацию интерфейса, подписывается на события представления и по запросу изменяет модель [16].

**Признаки презентера:**

* Двухсторонняя коммуникация с представлением;
* Представление взаимодействует напрямую с презентером, путем вызова соответствующих функций или событий экземпляра презентера;
* Презентер взаимодействует с view путем использования специального интерфейса, реализованного представлением;
* Один экземпляр презентера связан с одним отображением [17][18].

Каждое представление должно реализовывать соответствующий интерфейс. Интерфейс представления определяет набор функций и событий, необходимых для взаимодействия с пользователем (например, **iview**.showerrormessage(string msg)). Презентер должен иметь ссылку на реализацию соответствующего интерфейса, которую обычно передают в конструкторе [18].

Логика представления должна иметь ссылку на экземпляр презентера. Все события представления передаются для обработки в презентер и практически никогда не обрабатываются логикой представления (в т.ч. cоздания других представлений). Пример использования шаблона: технология**windows forms** [17].

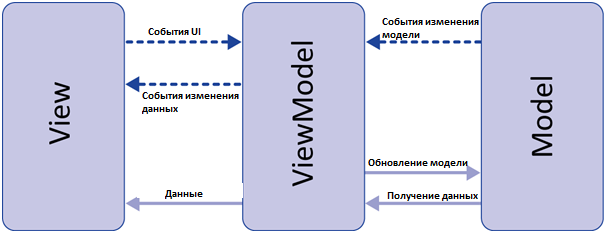


Рисунок 3.2.3 – Шаблон проектирования «Модель-представление-представление»

Подход позволяет связывать элементы представления со свойствами и событиями view-модели. Можно утверждать, что каждый слой этого паттерна не знает о существовании другого слоя[18].

**Признаки view-модели:**

* Двухсторонняя коммуникация с представлением;
* View-модель — это абстракция представления. Обычно означает, что свойства представления совпадают со свойствами view-модели / модели
* View-модель не имеет ссылки на интерфейс представления (iview). Изменение состояния view-модели автоматически изменяет представление и наоборот, поскольку используется механизм связывания данных (bindings)
* Один экземпляр view-модели связан с одним отображением [18].

При использовании этого паттерна, представление не реализует соответствующий интерфейс (iview). Представление должно иметь ссылку на источник данных (datacontex), которым в данном случае является view-модель. Элементы представления связаны (bind) с соответствующими свойствами и событиями view-модели. В свою очередь, view-модель реализует специальный интерфейс, который используется для автоматического обновления элементов представления. Примером такого интерфейса в wpf может быть inotifypropertychanged [18].

Таки образом в основу системы была положена динамически подгружаемая библиотека, реализующая бизнес-модели. С помощью классов слоя доступа к данным устанавливается связь между бизнес-логикой программы и данными. Бизнес модели представляют собой слой Model паттерна MVP [14].

Бизнес модели должны быть трансформированы в модели для представлений. Это позволит разъединить слой представлений от слоя бизнес-логики и слоя доступа к данным. Так же в эти модели можно добавить функционал, не нужный для бизнес-слоя, как например валидация. Данный слой будет «представлять» бизнес-модели для графического интерфейса (presenter) [16].

Слой представлений представляет собой набор форм на платформе Windows Presentation Forms. Данная технология позволит стилизовать пользовательский интерфейс в последствии. Представления будут частью программно-аппаратного комплекса, непосредственно взаимодействующими с пользователем и реализующими свойства дружественности, гибкости и простоты интерфейса пользователя.

Разделение программы по слоям в соответствии с подзадачами, выполняемыми программно-аппаратным комплексом в ходе работы (слой доступа к данным, бизнес-логика, представление) позволяет тестировать программу по частям, что значительно облегчит поиск и устранение дефектов. Так же наличие отдельного слоя доступа к данным позволяет при необходимости разработать другой модуль и использовать данный комплекс с другими источниками данных.

Возможность конфигурирования соединения с базой данных позволяет устанавливать соединение как с локальной, так и с удаленной базой данных. Это способствует развертыванию приложения на нескольких компьютерах и создавать распределенную систему.

В ходе дальнейшего развития системы можно использовать возможности фреймворка Windows Presentation Forms для изменения внешнего вида графического интерфейса пользователя, его интернационализации, а так же развить систему для улучшения возможности выполнять опыты на различных языках.

**3.3 Программа испытаний**

Модель-представление-контроллер (MVC — Model-view-controller) — это фундаментальный паттерн, который нашел применение во многих технологиях, дал развитие новым технологиям и каждый день облегчает жизнь разработчикам. Впервые паттерн MVC появился в языке smalltalk [12]. Разработчики должны были придумать архитектурное решение, которое позволяло бы отделить графический интерфейс от бизнес логики, а бизнес логику от данных. Таким образом, в классическом варианте, MVC состоит из трех частей, которые и дали ему название [13].

Под моделью, обычно понимается часть, содержащая в себе функциональную бизнес-логику приложения. Модель должна быть полностью независима от остальных частей продукта. Модельный слой ничего не должен знать об элементах дизайна, и каким образом он будет отображаться. Достигается результат, позволяющий менять представление данных, то как они отображаются, не трогая саму модель [12].

Модель обладает следующими признаками:

* Модель — это бизнес-логика приложения;
* Модель обладает знаниями о себе самой и не знает о контроллерах и представлениях;
* Для некоторых проектов модель — это просто слой данных (dao, база данных, xml-файл);
* Для других проектов модель — это менеджер базы данных, набор объектов или просто логика приложения[13].

В обязанности представления входит отображение данных полученных от модели. Однако, представление не может напрямую влиять на модель. Можно говорить, что представление обладает доступом «только на чтение» к данным [14].

Примеры представления: html-страница, wpf форма, windows form.  
Представление обладает следующими признаками:

* В представлении реализуется отображение данных, которые получаются от модели любым способом;
* В некоторых случаях, представление может иметь код, который реализует некоторую бизнес-логику [15][16].

Наиболее распространенные виды mvc-паттерна, это:

* Model-view-controller [рис. 3.2.1]
* Model-view-presenter [рис. 3.2.2]
* Model-view-view model [рис. 3.2.3]

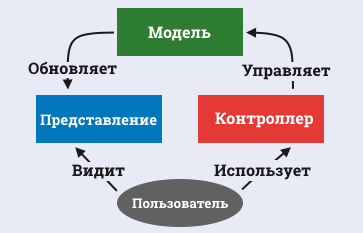


Рисунок 3.2.1 – Шаблон проектирования «Модель-представление-контроллер»

Основная идея этого паттерна в том, что и контроллер, и представление зависят от модели, но модель никак не зависит от этих двух компонент.  
**Признаки контроллера:**

* Контроллер определяет, какие представление должно быть отображено в данный момент;
* События представления могут повлиять только на контроллер. Контроллер может повлиять на модель и определить другое представление.
* Возможно несколько представлений только для одного контроллера [17].

При реализации контроллер перехватывает событие извне и в соответствии с заложенной в него логикой, реагирует на это событие изменяя модель, посредством вызова соответствующего метода. После изменения модель использует событие о том, что она изменилась, и все подписанные на это события представления, получив его, обращаются к модели за обновленными данными, после чего их и отображают. Пример использования: **mvc asp.net** [17].

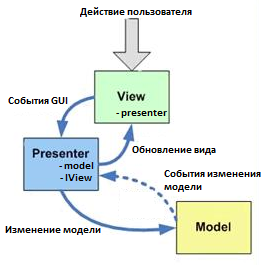


Рисунок 3.2.2 – Шаблон проектирования «Модель-представление-представитель»

Подход «Модель-представление-представитель» позволяет создавать абстракцию представления. Для этого необходимо выделить интерфейс представления с определенным набором свойств и методов. Презентер, в свою очередь, получает ссылку на реализацию интерфейса, подписывается на события представления и по запросу изменяет модель [16].

**Признаки презентера:**

* Двухсторонняя коммуникация с представлением;
* Представление взаимодействует напрямую с презентером, путем вызова соответствующих функций или событий экземпляра презентера;
* Презентер взаимодействует с view путем использования специального интерфейса, реализованного представлением;
* Один экземпляр презентера связан с одним отображением [17][18].

Каждое представление должно реализовывать соответствующий интерфейс. Интерфейс представления определяет набор функций и событий, необходимых для взаимодействия с пользователем (например, **iview**.showerrormessage(string msg)). Презентер должен иметь ссылку на реализацию соответствующего интерфейса, которую обычно передают в конструкторе [18].

Логика представления должна иметь ссылку на экземпляр презентера. Все события представления передаются для обработки в презентер и практически никогда не обрабатываются логикой представления (в т.ч. cоздания других представлений). Пример использования шаблона: технология**windows forms** [17].

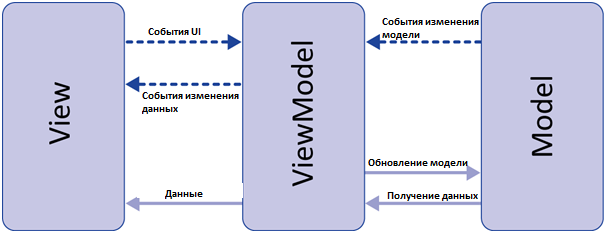


Рисунок 3.2.3 – Шаблон проектирования «Модель-представление-представление»

Подход позволяет связывать элементы представления со свойствами и событиями view-модели. Можно утверждать, что каждый слой этого паттерна не знает о существовании другого слоя[18].

**Признаки view-модели:**

* Двухсторонняя коммуникация с представлением;
* View-модель — это абстракция представления. Обычно означает, что свойства представления совпадают со свойствами view-модели / модели
* View-модель не имеет ссылки на интерфейс представления (iview). Изменение состояния view-модели автоматически изменяет представление и наоборот, поскольку используется механизм связывания данных (bindings)
* Один экземпляр view-модели связан с одним отображением [18].

При использовании этого паттерна, представление не реализует соответствующий интерфейс (iview). Представление должно иметь ссылку на источник данных (datacontex), которым в данном случае является view-модель. Элементы представления связаны (bind) с соответствующими свойствами и событиями view-модели. В свою очередь, view-модель реализует специальный интерфейс, который используется для автоматического обновления элементов представления. Примером такого интерфейса в wpf может быть inotifypropertychanged [18].

Таки образом в основу системы была положена динамически подгружаемая библиотека, реализующая бизнес-модели. С помощью классов слоя доступа к данным устанавливается связь между бизнес-логикой программы и данными. Бизнес модели представляют собой слой Model паттерна MVP [14].

Бизнес модели должны быть трансформированы в модели для представлений. Это позволит разъединить слой представлений от слоя бизнес-логики и слоя доступа к данным. Так же в эти модели можно добавить функционал, не нужный для бизнес-слоя, как например валидация. Данный слой будет «представлять» бизнес-модели для графического интерфейса (presenter) [16].

Слой представлений представляет собой набор форм на платформе Windows Presentation Forms. Данная технология позволит стилизовать пользовательский интерфейс в последствии. Представления будут частью программно-аппаратного комплекса, непосредственно взаимодействующими с пользователем и реализующими свойства дружественности, гибкости и простоты интерфейса пользователя.

Разделение программы по слоям в соответствии с подзадачами, выполняемыми программно-аппаратным комплексом в ходе работы (слой доступа к данным, бизнес-логика, представление) позволяет тестировать программу по частям, что значительно облегчит поиск и устранение дефектов. Так же наличие отдельного слоя доступа к данным позволяет при необходимости разработать другой модуль и использовать данный комплекс с другими источниками данных.

Возможность конфигурирования соединения с базой данных позволяет устанавливать соединение как с локальной, так и с удаленной базой данных. Это способствует развертыванию приложения на нескольких компьютерах и создавать распределенную систему.

В ходе дальнейшего развития системы можно использовать возможности фреймворка Windows Presentation Forms для изменения внешнего вида графического интерфейса пользователя, его интернационализации, а так же развить систему для улучшения возможности выполнять опыты на различных языках.

**4 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА ПО РАЗРАБОТКЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА**

**4.1 Функции, назначение и стоимостная оценка затрат на разработку комплекса исследования динамики заучивания**

Данный проект представляет собой реализацию в виде компьютерного программного продукта методик исследования динамики процесса заучивания. С помощью этой методики можно как изучать основные закономерности процесса заучивания, так и получать характеристики заучивания конкретного испытуемого или группы испытуемых в различных условиях [1].

Экспериментальным материалом служат в разных опытах буквы, цифры, бессмысленные слоги, не связанные между собой по смыслу слова. Материал предъявляют визуально на экране дисплея. Предъявление элементов ряда может производиться как последовательно, так и одновременно [2].

Целью дипломного проекта является создание программного продукта, обладающего высокой эргономичностью: пользователю должен быть предоставлен удобный интерфейс, система должна демонстрировать гибкость для пользователей различной квалификации, программа должна подсказывать пользователю ход решения задачи [3].

Основные задачи проектируемой системы:

1. облегчить подготовку к выполнению опытов по методике;
2. ускорить выполнение опыта;
3. позволить провести анализ результатов выполненных опытов.

Из задач определяются режимы работы системы:

1. заполнение стимульного материала;
2. выполнение опыта;
3. анализ результатов.

Таким образом следует произвести экономическое обоснование проекта по разработке программно-аппаратного комплекса для собственных нужд кафедры инженерной психологии и эргономики, включающее в себя расчеты стоимостной оценки затрат, стоимостной оценки результата внедрения и показателей эффективности использования программно-аппаратного комплекса.

Упрощенный расчет затрат на разработку программно-аппаратного комплекса исследования динамики заучивания включает в себя расчёты:

– затрат на основную заработную плату разработчиков;

– затрат на дополнительную заработную плату разработчиков;

– отчислений на социальные нужды;

– прочих затрат (амортизация оборудования, расходы на электроэнергию, командировочные расходы, накладные расходы и т.п.) [20].

Затраты на основную заработную плату команды разработчиков определяются исходя из состава и численности команды, размеров месячной

заработной платы каждого из участников команды, а также общей трудоемкости разработки программного обеспечения.

Расчет величины основной заработной платы участников команды осуществляется по формуле 4.1.1.

(4.1.1)

где *n* – количество исполнителей, занятых разработкой комплекса;

*TЧi* – часовая заработная плата i-го исполнителя (руб.);

*ti* – трудоемкость работ, выполняемых i-м исполнителем (ч) [20].

Часовая заработная плата определяется путем деления месячной заработной платы на количество рабочих часов в месяце. Количество рабочих часов в месяце принимается равным 168 [20]. Разработка рассчитана на работу одного разработчика интерфейса, одного разработчика серверных решений и одного инженера по качеству (тестировщика). Исходя из сложности разработки программно-аппаратного комплекса, объема выполняемых им функций была оценена и согласована трудоемкость разработки (табл. 4.1.1). Данные о заработной плате предоставлены Национальным статистическим комитетом Республики Беларусь [22]. Премиальный фонд был установлен в 75% от заработной платы согласно рекомендациям [20].

Таблица 4.1.1 – Расчет затрат на основную заработную плату команды

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Работник | Выполняемые работы | Месячная  заработная  плата, р. | Часовая  заработная  плата, р. | Трудоемкость работ, часов | Основная  заработная  плата, р. |
| 1. Разработчик интерфейса | Бизнес-анализ | 3453,8 | 20,56 | 8 | 164,48 |
| Прототипирование | 16 | 328,96 |
| Разработка | 40 | 822,40 |
| Устранение недостатков | 8 | 164,48 |
| 2. Разработчик серверных решений | Бизнес-анализ | 3453,8 | 20,56 | 8 | 164,48 |
| Проектирование связей с интерфейсом | 8 | 164,48 |
| Проектирование структур хранения данных | 40 | 822,40 |
| Устранение недостатков | 8 | 164,48 |
| 3. Инженер по качеству | Функциональное тестирование | 3453,8 | 20,56 | 30 | 616,80 |
| Нагрузочное тестирование | 8 | 164,48 |
| Повторное тестирование устраненных недостатков | 10 | 205,60 |
| Итог: | | | | | 3783,04 |
| ПРЕМИЯ (75%) | | | | | 2837,28 |
| Итого затраты на основную заработную плату разработчиков | | | | | 6620,32 |

Затраты на дополнительную заработную плату команды разработчиков включают выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью). Выполняется по формуле 4.1.2 [20].

(4.1.2)

где *Зо* – затраты на основную заработную плату с учетом премии (руб.);

*Нд* – норматив дополнительной заработной платы [20].

При нормативе дополнительной заработной платы на уровне 13% [20] и затратах на основную заработную плату с учетом премии равных 6620,32 руб. затраты на дополнительную заработную плату составляют:

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле 4.1.3 [20].

(4.1.3)

где Нсоц – норматив отчислений на социальные нужды [20].

При нормативе отчислений на социальные нужды на уровне 4,6 % по состоянию на 2017 год [21] и затратах на основную заработную плату с учетом премии равных 6620,32 рублей, затратах на дополнительную заработную плату равных 33.89 руб. отчисления на социальные нужды составляют:

Расчет прочих затрат осуществляется в процентах от затрат на основную заработную плату команды разработчиков с учетом премии (табл. 4.1.1) по формуле 4.1.4:

(4.1.4)

где Нпз – норматив прочих затрат [20].

При нормативе прочих затрат равном 100% от основной заработной платы команды разработчиков с учетом премии, расчет прочих затрат принимает вид:

Результат стоимостной оценки затрат представлен в таблице 4.1.2

Таблица 4.1.2 – Затраты на разработку программного обеспечения

|  |  |
| --- | --- |
| Статья затрат | Сумма, руб. |
| Основная заработная плата команды разработчиков | 6620,32 |
| Дополнительная заработная плата команды разработчиков | 860,64 |
| Отчисления на социальные нужды | 344,12 |
| Прочие затраты | 6620,32 |
| Общая сумма затрат на разработку | 14445,40 |

В общем виде эффект от использования программных продуктов может быть 2 видов:

1. Экономический эффект.

2. Неэкономический эффект [20].

Если использование компьютерной системы напрямую влияет на экономические показатели деятельности пользователя (например, автоматизированная система контроля качества значительно снижает потери от брака, вследствие чего снижаются затраты на производство продукции, а следовательно увеличивается прибыль), имеет место экономический эффект. Данный эффект легко поддается стоимостной оценке [20].

Неэкономический эффект напрямую не связан с экономическими результатами деятельности предприятия. Его влияние ощутима в социальной, экологической, политической, технической сферах. В данном случае использование программного обеспечения оказывает косвенное (опосредованное) влияние на экономические показатели деятельности пользователя, либо предоставляет ему дополнительные выгоды иного характера (например, сокращается время на подготовку разнообразной финансовой отчетности, облегчается бухгалтерский учет, увеличивается оперативность получения управленческой информации и т.п.) [20].

Использование программно-аппаратного комплекса исследования динамики заучивания не несет экономического эффекта, следовательно, нужно оценить неэкономический эффект от использования программного продукта в ходе учебного процесса [20].

**4.2 Стоимостная оценка результата внедрения разработанного комплекса исследования динамики заучивания**

Использование программно-аппаратного комплекса исследования динамики заучивания не несет экономического эффекта для процесса обучения студентов кафедры инженерной психологии и эргономики. В ходе выполнения студентам лабораторных работ по теме заучивания использование программных средств не является обязательным [2].

Не представляется возможным оценить эффект применения разрабатываемого продукта и для научной работы кафедры, поскольку характеристики памяти человека-оператора являются лишь частью исследований в области человеческого фактора [1].

Несмотря на отсутствие явного экономического эффекта неэкономический эффект применение программно-аппаратного комплекса значителен. Помимо использования современных технологий в процессе обучения использование данного продукта позволяет значительно облегчить процесс лабораторного практикума.

Лабораторное исследование подразумевает наличие испытуемого (или испытуемых) и экспериментатора. Если отказаться от применения программно-аппаратного комплекса, выполнение лабораторного исследования возможно лишь группой студентов, один из которых будет испытуемым, а другой экспериментатором [2].

Если исследование проводится группой студентов, то это влечет собой либо неполное участие студентов в исследовании, либо дублирование работы экспериментатора (в случае выполнения нескольких однотипных опытов со сменой ролей в группе) [2].

В случае выполнения исследования как экспериментатор всеми студентами затрачивается больше времени, которое могло быть использовано для выполнения большего количества опытов с различными типами запоминаемых объектов (буквы, знаки, слова, картинки), с различными свойствами запоминаемых объектов и с различными условиями запоминания [2].

Проектируемый комплекс представляет сторону исследователя. Исследователь имеет возможность заранее заготовить с помощью программы материал для опытов и предложить его испытуемым, задать настройки опыта.

Исследование можно проводить индивидуально. Подготовка испытуемого к процессу сводится к минимуму. Отсутствие необходимости фиксировать ход опыта самостоятельно облегчает выполнение лабораторной работы и позволяет глубже сконцентрироваться при выполнении опыта для получения более точных результатов.

Программно-аппаратный комплекс позволяет производить сбор результатов исследования для одних и тех же наборов стимулов для всех испытуемых, не требуя работы с каждым из них индивидуально. Значительно облегчается подготовка материала для запоминания, особенно рисунков.

Основные стороны неэкономического эффекта:

* возможность проводить исследование индивидуально;
* возможность использования рисунков;
* настройка временных параметров опыта;
* автоматический процесс фиксирования хода опыта;
* облегчение анализа результатов.

Реализованная с помощью прикладной программы методика позволяет испытуемому взаимодействовать в ходе опыта не с самим экспериментатором, а с компьютерной программой, проводящей предъявление заранее заготовленных наборов стимулов разного рода.

На комплекс также возлагаются задачи по регистрацию запоминаемых стимулов, определения окончания опыта, построение математических зависимостей. Экспериментатор, пользуясь данным программным средством имеет возможность фокусироваться на интерпретации полученных результатов в большей степени.

Качество и удобство проведения исследований обеспечивают также дополнительные функции комплекса: возможность использования графической информации в качестве стимулов, встроенные функции инструктирования и просмотра теоретических сведений. Эти функции существенно расширяют инструментарий исследователя, облегчая его труд по сравнению с реализацией методики без прикладной компьютерной программы [2].

Возможность проведения новых многоаспектных экспериментов для исследований динамики заучивания свидетельствуют о том, что спроектированный программно-аппаратный комплекс способен оптимизировать трудовой и учебный процесс в области психологических дисциплин.

**4.3 Показатели эффективности инвестирования в разработку комплекса исследования динамики заучивания**

Для оценки показателей эффективности при отсутствии экономического эффекта от использования разрабатываемого продукта производится сравнение затрат на разработку программного обеспечения с расходами, сопутствующими внедрению аналогичного продукта [20].

Внедрение аналога «AVELife TestGold Studio 5», являющегося интегрированной средой разработки тестов и опросов, требует затрат по следующим пунктам:

– основная заработная плата преподавателя по адаптации методики исследования и настройке продукта «AVELife TestGold Studio 5» для проведения исследований;

– дополнительная заработная плата преподавателя;

– отчисления на социальные нужды;

– прочие затраты (лицензирование программы) [20].

Расчет величины основной заработной платы преподавателя осуществляется по формуле 4.1.1. Часовая заработная плата определяется путем деления месячной заработной платы на количество рабочих часов в месяце. Количество рабочих часов в месяце принимается равным 168 [20].

Заработная плата преподавателя принята в размере 521,10 рублей согласно данным Национального статистического комитета Республики Беларусь [22].

Исходя из сложности разработки программно-аппаратного комплекса, объема выполняемых им функций была оценена и согласована трудоемкость адаптации методики и настройки «AVELife TestGold Studio 5» для исследований: она составила 20 часов. Премиальный фонд был установлен в 75% от заработной платы согласно рекомендациям [20]. Расчет представлен в таблице 4.3.3.

Таблица 4.3 – Расчет затрат на основную заработную плату команды

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Работник | Выполняемые работы | Месячная  заработная  плата, р. | Часовая  заработная  плата, р. | Трудоемкость работ, часов | Основная  заработная  плата, р. |
| 1. Преподаватель | Адаптация методики | 521,10 | 3,10 | 20 | 62 |
| ПРЕМИЯ (75%) | | | | | 46,5 |
| Итого затраты на основную заработную плату разработчиков | | | | | 108,5 |

Затраты на дополнительную заработную плату включают выплаты, предусмотренные законодательством о труде (оплата отпусков, льготных часов, времени выполнения государственных обязанностей и других выплат, не связанных с основной деятельностью). Выполняется по формуле 4.2 [20].

При нормативе дополнительной заработной платы на уровне 13% [20] и затратах на основную заработную плату с учетом премии равных 108,5 руб. затраты на дополнительную заработную плату составляют:

Отчисления на социальные нужды (в фонд социальной защиты населения и на обязательное страхование) определяются в соответствии с действующими законодательными актами по формуле 4.1.3 [20].

При нормативе отчислений на социальные нужды на уровне 4,6% по состоянию на 2017 год [21] и затратах на основную заработную плату с учетом премии равных 108,5 рублей, затратах на дополнительную заработную плату равных 14,11 руб. отчисления на социальные нужды составляют:

Прочие затраты принимаются равными цене лицензирования программы «AVELife TestGold Studio 5» (цена лицензии на двадцать рабочих мест составляет 18198 руб. [23]) и составляют:

Результат стоимостной оценки затрат представлен в таблице 4.3.4

Таблица 4.3.4 – Затраты на внедрение аналога

|  |  |
| --- | --- |
| **Статья затрат** | **Сумма, руб.** |
| Основная заработная плата преподавателя | 108,5 |
| Дополнительная заработная плата преподавателя | 14,11 |
| Отчисления на социальные нужды | 5,64 |
| Прочие затраты | 18198,00 |
| **Общая сумма затрат на внедрение аналога** | 18326,25 |
| **Общая сумма затрат на разработку комплекса** | 14445,40 |
| **Разница** | 3880,85 |

В случае покупки лицензии на двадцать рабочих мест аналога разработанного комплекса «AVELife TestGold Studio 5» расходы на внедрение превысят расходы на разработку собственного продукта на 17704 рубля (таблица 4.3.4). Неэкономический эффект от использования комплекса и значительная разница между расходами на внедрение аналогов и расходами на разработку комплекса исследования динамики заучивания позволяют сделать вывод об целесообразности инвестирования в разработку программно-аппаратного комплекса.

**5 ОБЕСПЕЧЕНИЕ ПОЖАРНОЙ БЕЗОПАСНОСТИ ПРИ РАЗРАБОТКЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА**

Данный проект представляет собой реализацию в виде компьютерного программного продукта методик исследования динамики процесса заучивания. С помощью этой методики можно как изучать основные закономерности процесса заучивания, так и получать характеристики заучивания конкретного испытуемого или группы испытуемых в различных условиях [1].

Экспериментальным материалом служат в разных опытах буквы, цифры, бессмысленные слоги, не связанные между собой по смыслу слова. Материал предъявляют визуально на экране дисплея. Предъявление элементов ряда может производиться как последовательно, так и одновременно [2].

Первоначальные стадии разработки дипломного проекта выполнялись на предприятии ООО «АйтиРекс Групп Бел» во время прохождения преддипломной практики. В настоящем разделе рассматриваются вопросы, связанные с обеспечением пожарной безопасности на предприятии.

Предприятие «АйтиРекс Групп Бел» занимается предоставлением услуг по разработке информационных систем для иностранных предприятий. В минском офисе компании на данный момент работает более 200 человек. Большое количество конкурирующих компаний, разрабатывающих программное обеспечение в Минске, способствует повышению общего уровня условий труда. Это, в частности, сказывается на комфортабельности рабочих мест. Работникам предоставляются светлые, проветриваемые, тихие кабинеты, гибкий график рабочего времени, специальные комнаты отдыха и т.д. Современные компании негласно ориентируются на соответствие лучшим мировым практикам в области охраны труда и, в частности, пожарной безопасности [24].

Основной род деятельности на предприятии — разработка информационных систем — не предусматривает непосредственный контакт с горючими или легковоспламеняющимися веществами, что сильно снижает риски возникновения пожара на предприятии.

На предприятии «АйтиРекс Групп Бел» за пожарную безопасность отвечает директор компании. Для каждого нового сотрудника производится инструктаж по пожарной безопасности и технике безопасности, знакомство с планом эвакуации при возникновении черезвычайных ситуаций [25]. За проведение инструктажа отвечает специальный человек из отдела материально-технического снабжения предприятия. В компании действует набор правил, обязательных для исполнения сотрудниками. В целях повышения пожарной безопасности курение в здании офиса запрещено. Все сотрудники обязаны в конце рабочего дня выключить свои персональные компьютеры и обесточить их. В конце рабочего дня специальный сотрудник проверяет соблюдение данного правила в каждом рабочем кабинете, чтобы там были выключены все электрические приборы: компьютеры, электрические чайники, кондиционеры, освещение и т.д. Все рабочие компьютеры подключены к источникам бесперебойного питания, которые подключены к сетевыми фильтрам, защищающим от скачков напряжения в электросети [24].

Офис компании расположен в центре города. Здание офиса представляет собой монолитную железобетонную конструкцию высотой шесть этажей, офис компании находится на третьем и четвертом этажах здания. Конструкция здания предусматривает три способа эвакуации с этажа: выход в паркинг, лестничная клетка с выходом на улицу, лестничная клетка с выходом на первый этаж паркинга [24].

В случае недоступности основных эвакуационных выходов из каждого кабинета можно через окно попасть на лоджию [24]. Схемы эвакуации выдаются в виде электронного документа каждому новому сотруднику, а также находятся на специальном стенде в рабочих кабинетах. Все кабинеты офиса расположены вдоль длинного коридора, который оборудован специальными аварийными светильниками и знаками, указывающими направление эвакуации. На случай отключения электроэнергии компания имеет два дизельных-генератора, обеспечивающих нужды предприятия на случай отключения электроэнергии [24].

Офис компании оборудован необходимыми средствами сигнализации о пожаре [25]. Каждый кабинет оборудован пожарным дымовым оптико-электрическим точечным извещателем ИП212-02М1 (рисунок 5.1). На предприятии производиться регулярный контроль и проверка работоспособности пожарных извещателей специальным человеком из отдела материально-технического снабжения предприятия.

В коридорных помещениях дополнительно установлены ручные пожарные извещатели ИП 5-2Р (рис. 5.1). Для извещения о пожаре также может быть использована корпоративная электронная почта, а также другие современные способы обмена информацией [25].

На случай возникновения пожара в каждом кабинете находиться ручной порошковый огнетушитель ОП-10 (з) МИГ М (рисунок 5.2), пригодный для тушения пожаров различного типа, в том числе для тушения электрических приборов [24].

Каждый этаж здания офиса оборудован двумя пожарными кранами для тушения пожара. Пожарные краны расположены в противоположных частях коридора, недалеко от эвакуационных выходов (рисунок 5.2).



(а) (б)

Рисунок 5.1 – а — автономный пожарный извещатель; б — ручной пожарный извещатель.

На случай воспламенения электрической проводки или другого электрического оборудования в каждом кабинете установлены электрические щитки, необходимые для отключения подачи электроэнергии в пределах кабинета. Во всех помещениях офиса предприятия установлена оросительная система пожаротушения для ликвидации возгорания до приезда пожарной службы [25]. При расследовании возможных причин возникновения пожара может быть задействована система видеонаблюдения, установленная во всех помещениях предприятия.

Наиболее вероятными причинами возникновения пожара, с учетом специфики предприятия, могут являться нарушение правил внутреннего распорядка — курение на рабочем месте, и неисправность электрического оборудования, которого в офисе компании достаточно [24].



(а) (б)

Рисунок 5.2 – а — порошковый огнетушитель ОП-10 (з) МИГ М; б — пожарный кран.

С целью снижения риска возникновения пожара по причине неисправности электрического оборудования в компании запрещено пользоваться неисправным оборудованием, а все исправное оборудование подключается в сеть через специальные сетевые фильтры и источники бесперебойного питания. В целом правила распорядка на предприятии и высокая культура работы с электрическим оборудованием снижают риски возникновения пожара до минимума [25].

Большой проблемой в достижении максимальной пожарной безопасности предприятия является доступность подъезда пожарной техники к зданию офиса. В будние дни прилегающие улицы, стоянки, пешеходные переходы заняты неправильно припаркованным личным транспортом. Большую часть светлого времени суток движение по прилегающим улицам затруднено [25].

Неукоснительное выполнение правил пожарной безопасности сотрудниками предприятия «АйтиРекс Групп Бел», соблюдение мер предосторожности лицами, находящимися на территории предприятия, а так же вышеперечисленные технические и организационные меры обеспечивают пожарную безопасность в процессе разработки программно-аппаратного комплекса [24].

**ЗАКЛЮЧЕНИЕ**

В данном дипломном проекте был разработан и испытан программно-аппаратный комплекс для исследования динамики заучивания. Данный комплекс позволяет испытуемым выполнять опыты по заучиванию наборов стимулов и проводить исследования в качестве экспериментатора [2].

Решение о проектировании комплекса было произведено на основе исследования предменой области и анализа существующих аналогов. На практике методику исследования динамики заучивания реализуют чаще всего без использования компьютерных средств. Такой вариант предусматривает непосредственное взаимодействие экспериментатора и испытуемого [1].

Специфика и узкая направленность исследования ограничивают использование существующих средств тестирования для реализации методики. Существующие средства как правило представляют собой многофункциональные интегрированные средства разработки, имеющие довольно высокую стоимость лицензии. Для адаптации методики к этим средствам требуются дополнительные трудовые затраты [1].

Существующие реализации методики, представленные в методическом обеспечении кафедры инженерной психологии и эргономики требовали значительных доработок. Решение построения новой реализации обосновано устаревшими технологиями, работа с которыми сопряжена с проблемами совместимости с современными программными средствами [15].

Для максимального удовлетворения потребностей сотрудников и студентов кафедры инженерной психологии и эргономики программно-аппаратным комплексом был составлен проект деятельности человека-оператора в данной системе. После создания была дана оценка соответствия разрабатываемого комплекса эргономическим требованиям и были выработаны рекомендации по устранению недочетов [1].

Разработанный комплекс представляет собой графическое приложение для операционных систем семейства Windows. Хранение информации осуществляется через систему управления базами данных Microsoft SQL Server. Данные могут храниться как на локальном компьютере, так и на выделенном сервере. Возможность подключения к удаленной базе данных предоставляет возможности к совместному использованию информации приложениями на разных компьютерах и к масштабированию приложения в будущем [10].

Приложение построено согласно с трехслойной архитектурой, Имея отдельные слои для доступа к данным, для операций с бизнес-моделями и для конечного представления информации пользователю. При разработке этих аспектов программы максимально независимо друг от друга сокращается риск ошибок в приложении и облегчается поддержка продукта.

Для построения графического интерфейса приложения была использована технология Windows Presentation Forms. Используя эту технологию был получен современный интерфейс, легко поддающийся модификации и стилизации [13].

Для минимизации трудовых затрат и проблем при сопряжении приложения с источником данных была применена технология ADO.Net Entity Framework. Этот инструментарий позволил избежать объемного непосредственного сообщения разрабатываемого приложения с базой данных. Задачей Entity Framework является перевод данных из реляцонной формы хранения в графовую, используемую в объектно-ориентированном программировании [13].

Реализованная с помощью прикладной программы методика позволяет испытуемому взаимодействовать в ходе опыта не с самим экспериментатором, а с компьютерной программой, проводящей предъявление заранее заготовленных наборов стимулов разного рода.

На комплекс также возлагаются задачи по регистрацию запоминаемых стимулов, определения окончания опыта, построение математических зависимостей. Экспериментатор, пользуясь данным программным средством имеет возможность фокусироваться на интерпретации полученных результатов в большей степени.

Качество и удобство проведения исследований обеспечивают также дополнительные функции комплекса: возможность использования графической информации в качестве стимулов, встроенные функции инструктирования и просмотра теоретических сведений. Эти функции существенно расширяют инструментарий исследователя, облегчая его труд по сравнению с реализацией методики без прикладной компьютерной программы [2].

Возможность проведения новых многоаспектных экспериментов для исследований динамики заучивания свидетельствуют о том, что спроектированный программно-аппаратный комплекс соответствует заданным требованиям и способен оптимизировать трудовой и учебный процесс в области психологических дисциплин.

**Список использованных источников**

1. Шупейко И.Г. КОНСПЕКТ ЛЕКЦИЙ по дисциплине «Теория и практика инженерно-психологического проектирования и экспертизы» в двух частях / И.Г. Шупейко — Минск: БГУИР, 2013 -301 с.
2. Шупейко И.Г. "Психология восприятия и переработки информации: лабораторный практикум"/ И.Г. Шупейко — Минск: БГУИР, 2008 — 79 с.
3. Лурия А. Р. Высшие корковые функции человека / А. Р. Лурия // Высшие корковые функции человека и их нарушения при локальных поражениях мозга второе дополненное издание, Москва, 1962 — 431 с.
4. Мунипов В. М. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учеб. для студентов вузов / В. М. Мунипов — Москва: Логос, 2001. — 356 с.
5. Len Bass, Paul Clements, Rick Kazman. Software Architecture in Practice (3rd Edition) // SEI Series in Software Engineering: Pittsburgh, 2012 — 640 с. — ISBN 978-0321815736.
6. Кушнир, А.Н. Новейшая энциклопедия Windows / А.Н. Кушнир. — Москва: Эксмо, 2009. — 960 c.
7. О важности эргономики [Электронный ресурс] // Interaction / С. Дрэй, [1995]. URL: http://www.usability.ru/Articles/software-ergonomics.htm (дата обращения: 1.04.2017).
8. Лурия А.Р. Альманах психологических тестов / А. Р. Лурия — Москва, 1995, 177 с.
9. Дейт К. Дж. Введение в системы баз данных — 8-е изд. / Дж. К. Дейт — Москва: Вильямс, 2005. — 1328 с. — ISBN 5-8459-0788-8.
10. Кузнецов С. Д. Основы баз данных. — 2-е изд. / С. Д. Кузнецов — Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с. — ISBN 978-5-94774-736-2.
11. Когаловский М.Р. Энциклопедия технологий баз данных / М. Р. Когаловский — Москва: Финансы и статистика, 2002. — 800 с. — ISBN 5-279-02276-4.
12. Гукин Оптимизация Windows. Руководство фирмы Microsoft / Гукин, Дэн. — Москва: СПб: Питер, 1995. — 362 c.
13. Джейсон Visual C# .NET. Полное руководство / Джейсон, Майк Прайс; , Гандэрлой. — Москва: Корона Принт, 2004. — 960 c.
14. Джонс Средства автоматизации в Microsoft Windows (+ CD-ROM) / Джонс, Дон. — Москва: Эком, 2006. — 416 c.
15. Нейгел, К. C# 2005 для профессионалов / К. Нейгел. — Москва: Вильямс, 2006. — 886 c.
16. Харт Системное программирование в среде Windows / Харт, М. Джонсон. — Москва: Вильямс, 2005. — 592 c.
17. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер, Джефри. — Москва: Питер, 2007. — 656 c.
18. Робинсон, С. C# для профессионалов / С. Робинсон, О. Корнес, Д. Глинн, и др.. — Москва: ЛОРИ, 2005. — 473 c.
19. Cистемы, Открытые Журнал сетевых решений/

LAN №07-08/2015 / Открытые системы. — Москва: Открытые Системы, 2015. — 536 c.

1. Носенко А.А. Методические рекомендации по технико-экономическому обоснованию дипломных проектов инженерного профиля / А.А. Носенко — Минск : БГУИР, 2010. — 27 с.
2. ЗАКОН РЕСПУБЛИКИ БЕЛАРУСЬ 29 февраля 1996 г. № 138-XІІІ Об обязательных страховых взносах в бюджет государственного внебюджетного фонда социальной защиты населения Республики Беларусь // Фонд социальной защиты населения [Электронный ресурс] URL: http://www.ssf.gov.by/priside/statesocialinsurance /norm\_doc/law2/~page\_\_m17=1~news\_\_m17=1509 (дата обращения: 11.05.2017)
3. БелСтат [Электронный ресурс] URL:

http://www.belstat.gov.by/ofitsialnaya-statistika/solialnaya-sfera/trud/ operativnaya-informatsiya\_8/nom\_zrab\_plata-po-kvartalam/ nachislennaya-zarabotnaya-plata-po-vidam-ekonomicheskoy-deyatelnosti-po-kvartalam-2016-g/ (дата обращения: 11.05.2017)

1. Компания АВЕЛайф [Электронный ресурс] URL: http://avelife.ru/order.htm (дата обращения: 11.05.2017)
2. Девисилов, В. А. Охрана труда : учебник. / В. А. Девисилов. 2-е изд. испр. и доп. — М. : Форум, ИНФРА — М., 2006. — 447 c.
3. Михнюк, Т. Ф. Охрана труда и основы экологии : учеб. пособие / Т. Ф. Михнюк — Минск : Выш. шк., 2007. — 356 c. Смирнов А.А. Психология памяти. — Москва: Просвещение, 1965.