Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

|  |
| --- |
|  |
|  |
|  |

Отчет по преддипломной практике на тему:  
«Программно-аппаратный комплекс сравнения процессов воспроизведения и узнавания»

|  |  |  |
| --- | --- | --- |
| Выполнила |  | студент гр. 210901  Михалёва К.С. |
| Проверил |  | И.Г. Шупейко |
|  |  |  |
|  |  |  |
|  |  |  |

Минск 2017

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc478423845)

[1 Анализ научно-технической литературы 6](#_Toc478423846)

[1.1 Компьютеризированные методики инженерно- психологических исследований 8](#_Toc478423847)

[1.2 Отличительные особенности компьютеризированных методик психологических исследований 17](#_Toc478423848)

[1.3 Выводы и постановка задачи 18](#_Toc478423849)

[2 Эргономическое проектирование информационной системы 21](#_Toc478423850)

[2.1 Анализ функций и их распределение в проектируемой системе 21](#_Toc478423851)

[2.2 Разработка алгоритмов работы пользователей программного   
комплекса 26](#_Toc478423852)

[2.3 Разработка эргономических требований и сценария информационного взаимодействия 32](#_Toc478423853)

[3 Разработка программно – аппаратного комплекса для исследования процессов памяти 51](#_Toc478423854)

[3.1 Обоснование выбора языка реализации и среды разработки 51](#_Toc478423855)

[3.2 Структура программного средства 53](#_Toc478423856)

[3.3 Тестирование программно-аппаратного комплекса 60](#_Toc478423857)

[4 Технико-экономическое обоснование эффективности разработки и использования программно-аппаратного комплекса для исследования процессов памяти 66](#_Toc478423858)

[4.1 Стоимостная оценка затрат 67](#_Toc478423859)

[4.2 Стоимостная оценка результата 70](#_Toc478423860)

[4.3 Показатели эффективности использования программного продукта. 71](#_Toc478423861)

[5 Проведение эргономической оценки рабочего места разработчика программно-аппаратного комплекса 74](#_Toc478423865)

Заключение……………………………………………………………………….68

[Список использованных источников 80](#_Toc478423866)

Приложение А (обязательное). Листинг программы ...…………….…………81

# **ВВЕДЕНИЕ**

Память – форма психического отражения, заключающаяся в закреплении, сохранении и последующем воспроизведении прошлого опыта, делающая, возможным его повторное использование в деятельности или возвращение в сферу сознания. Память связывает прошлое субъекта с его настоящим и будущим и является важнейшей познавательной функцией, лежащей в основе развития и обучения.

Память – основа важнейших проявлений психической деятельности. Без нее невозможно понять основы формирования поведения, мышления, сознания, подсознания. И поэтому для лучшего понимания человека необходимо как можно больше знать о нашей памяти. Основные процессы памяти: запоминание, сохранение, воспроизведение, узнавание, забывание.

Запоминание – это процесс памяти, посредством которого происходит запечатление следов, ввод новых элементов ощущений, восприятие, мышления или переживания в систему ассоциативных связей.

Сохранение – процесс накопления материала в структуре памяти, включающий его переработку и усвоение. Сохранение опыта дает возможность для обучения человека, развития его перцептивных процессов, мышления и речи.

Забывание выражается в невозможности вспомнить или в ошибочном узнавании и воспроизведении. Физиологической основой забывания являются некоторые виды коркового торможения, мешающего актуализации временных нервных связей [1].

Воспроизведение – процесс памяти, в результате которого происходит актуализация закрепленного ранее содержания путем извлечения его из долговременной памяти и перевода в оперативную. Реконструкция при воспроизведении проявляется в отборе главного и отсеве второстепенного материала, в обобщении и привнесении нового содержания, в изменении последовательности изложения, в различных заменах и искажениях воспроизводимого материала.

Процессы узнавания функционально отличаются от процессов воспроизведения. Оно предполагает наличие объекта, в то время как воспроизведение – его поиск. Узнавание - более простой и генетически более ранний процесс, чем воспроизведение. Узнавание – это и восприятие, но в отличие от первичного восприятия узнавание – всегда повторное восприятие [2].

Традиционные диагностические методики исследования предполагают использование напечатанных на бумажном носителе материалов (стимульного материала, форм регистрации ответов, листов подсчета и представления результатов). Использование современной компьютерной техники предоставляет качественно новые возможности для проведения диагностики личности и группы. Это можно отнести ко всем этапам процесса диагностики. Так с помощью компьютеров возможно формировании и предъявлении тестируемому гораздо большего количества стимулов, их контекстная коррекция в зависимости от сделанного выбора. Значительно упрощается фиксация и обработка ответов респондента при одновременном снижении вероятности ошибок на данном этапе диагностики. Существенным плюсом компьютерных средств психологической и профессиональной диагностики является быстрота перевода полученных первичных данных по тестам в стандартные значения и наличие базовых вариантов интерпретации показателей. Таким образом, компьютерные системы диагностики освобождают пользователя от трудоемких рутинных операций и позволяют сосредоточиться на решении содержательных профессиональных задач.

Предметом дипломного проекта является разработка, создание и внедрение программно-аппаратного комплекса для сравнения процессов воспроизведения и узнавания. В случае успешного проектирование получится система, эффективная с точки зрения затрат на её разработку, изготовление, функционирование; обеспечивающая условия рабочей среды, не наносящие вред оператору, отвечающие требованиям эстетики и удобства эксплуатации.

# **1 АНАЛИЗ НАУЧНО-ТЕХНИЧЕСКОЙ ЛИТЕРАТУРЫ**

Компьютерные психодиагностические методики – это методики, разработанные и адаптированные с учетом специфики и возможностей компьютера, предусматривающих постановку задачи тестов на экране компьютера и запись ответов, вводимых испытуемым.

Использование компьютерной техники дает потенциальную возможность не только протестировать участника, но и соотнести полученные результаты с несколькими выборками стандартизации теста [3].

Компьютерные версии психодиагностических методик приобретают все большее значение, становятся важным инструментарием психологов в самых различных областях. При их разработке создаются системы, с помощью которых делают диагностический вывод по результатам исследования конкретного человека в виде связного и непротиворечивого текста отражает измеряемые психологические параметры.

В психодиагностических исследованиях интерпретация результатов и написание психодиагностического заключения является творческим процессом, который выполняют психологи после обработки полученных данных. Точность и адекватность этого заключения о личности исследуемого обеспечивает опыт исследователя. Разработка автоматизированного психодиагностического заключения предусматривает моделирование рассуждений психолога при интерпретации результатов тестирования и переноса его знаний и опыта в структуры, которые воспринимаются компьютером [4].

Компьютерные версии психодиагностических методик повышают эффективность работы психолога за счет быстрого обработки данных и полученных результатов тестирования, освобождение от трудоемких рутинных операций, улучшения четкости и тщательности психологического исследования вследствие точности регистрации результатов и исключения ошибок при обработке исходных данных. Появилась возможность в сжатые сроки осуществлять массовые психодиагностические исследования путем одновременного тестирования многих испытуемых; повысился уровень стандартизации условий психодиагностического исследования за счет обеспечения одинаковых для всех исследуемых условий. Исследуемые стали откровеннее во время эксперимента благодаря конфиденциальности автоматизированного тестирования. Исследователь с помощью компьютера может не только устанавливать необходимый темп психодиагностического тестирования, но и отслеживать время как диагностический параметр; накапливаются и хранятся данные о испытуемых, результаты тестирования, базы данных испытуемых.

Таким образом, основными преимуществами современных компьютеризированных методик психодиагностики являются:

* неизменность реализованной программы, постоянство условий тестирования, точность и однозначность регистрации;
* возможность восстановить и проследить последовательность действий испытуемого;
* наличие единых баз психодиагностических данных;
* автоматизированное конструирование тестов, отсутствие рутинной, трудоемкой работы при их конструировании;
* возможности расширения практики группового тестирования и тиражирования методик применения математически-статистического аппарата анализа данных, упрощения разработки новых процедур анализа;
* обеспечение конфиденциальности результатов тестирования;
* хранение диагностических данных на переносных носителях, снижение себестоимости обследования;
* применение экспресс-методик, которые позволяют быстро получить результаты;
* минимизация негативных воздействий, возникающих в ситуации межличностного взаимодействия между экспериментатором и испытуемым;
* актуализация игровой мотивации у исследуемых, что повышает достоверность результатов;
* анализ поведения испытуемого в процессе обследования с учетом многих параметров, влияющих на ситуацию;
* организация диалога с испытуемым в режиме реального времени.

Программно-аппаратный комплекс (ПАК) — это набор технических и программных средств, работающих совместно для выполнения одной или нескольких сходных задач. Состоит, соответственно, из двух основных частей:

* аппаратная часть (Hardware) — устройство сбора и/или обработки информации, например компьютер, биометрический детектор, калибратор и т. д;
* программная часть (Software) — специализированное ПО (как правило, написано компанией — производителем аппаратной части), обрабатывающее и интерпретирующее данные, собранные аппаратной частью.

Современная методология системного проектирования требует, чтобы создаваемые программно-технические средства были не только надежными и эффективными, но и эргономичными, т.е. удобными в использовании и освоении.

## **1.1 Компьютеризированные методики инженерно- психологических исследований**

В данном подразделе произведен обзор аналогов компьютеризированных методик инженерно-психологических исследований.

Одним из примеров автоматизированной методологии исследования является веб-приложение для оценки кратковременной зрительной памяти [5].

Методика исследования кратковременной памяти заключается в следующем: испытуемому необходимо за определенный промежуток времени запомнить максимальное количество числовых значений, а затем воспроизвести их.

На рисунке 1.1 изображен первый этап опыта – испытуемому демонстрируется инструкция проведения эксперимента, таймер, показывающий время до окончания первой части эксперимента, а также числовой ряд, который необходимо запомнить.

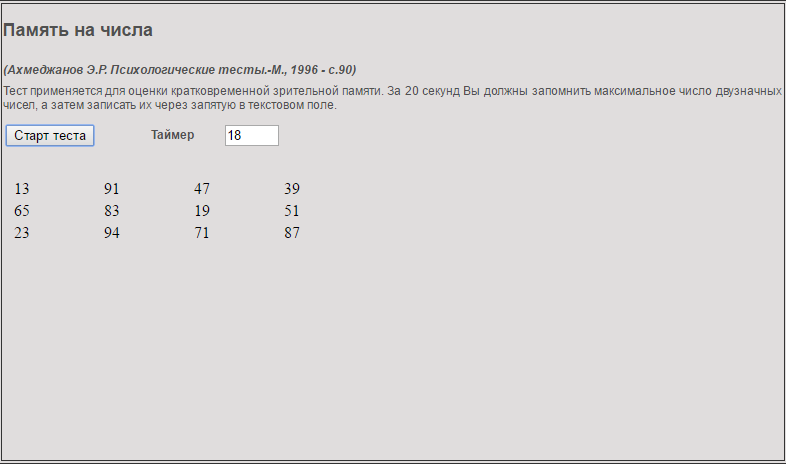


Рисунок 1.1 – Демонстрация стимульного ряда

После истечения 20 секунд, испытуемому демонстрируется текстовое поле, в которое необходимо ввести запомнившиеся числа (рисунок 1.2).

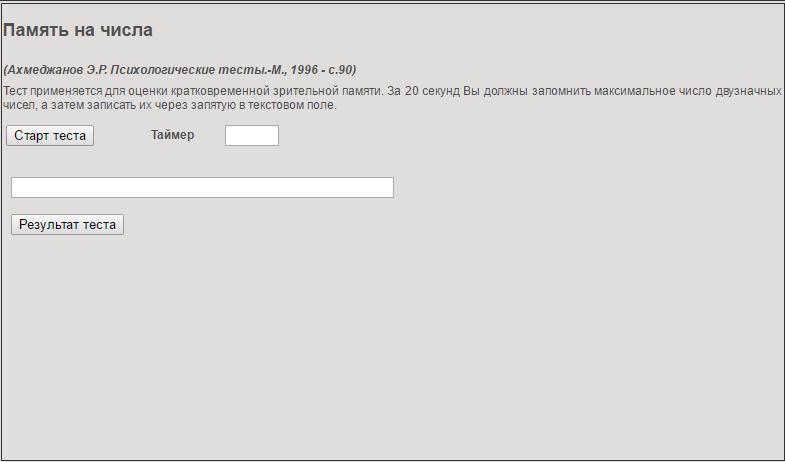


Рисунок 1.2 – Ввод запомнившегося материала

После ввода испытуемым числового ряда и нажатия кнопки «Результат теста», на экране монитора демонстрируется надпись, указывающая ниже, выше, либо равно значение объема и точности кратковременной зрительной памяти пользователя норме (рисунок 1.3).

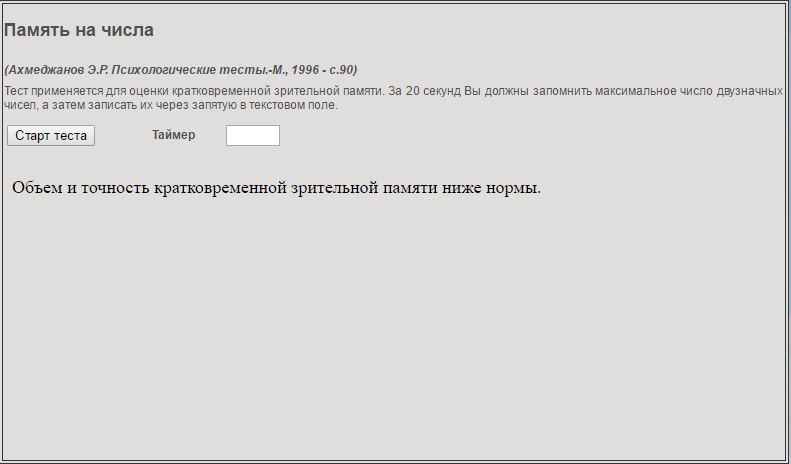


Рисунок 1.3 – Отображение результатов опыта

Преимуществами использования данного веб ресурса является доступный и понятный пользователю интерфейс, простота исследования. Однако присутствует целый ряд недостатков:

* отсутствие возможности регистрации пользователя;
* недостаточная точность формулировки результата;
* отсутствие идентификации пользователя в качестве испытуемого либо экспериментатора;
* отсутствие возможности сохранения, либо повторного просмотра результатов;
* обязательное наличие соединения с интернетом;
* база данных не обновляется, следовательно, испытуемый может пройти эксперимент только один раз.

Другим аналогом является программно-аппаратный комплекс «Исследование восприятия текстовой информации». Комплекс предназначен для проведения экспериментального исследования одного из основных свойств восприятия – избирательности. Данное свойство определяет такую особенность восприятия, которая проявляется в том, что не все объекты, находящиеся в поле зрения, человек воспринимает одинаково. Некоторые объекты воспринимаются лучше и на это влияют как объективные, так и субъективные факторы. В данном программно-аппаратном комплексе подобными особенностями является способ подачи текстовой информации на экран монитора. Такими способами являются все основные возможности выделения фрагментов текста, предоставляемые опцией «форматирование», а именно:

* выделение размером шрифта;
* выделение цветом шрифта;
* выделение жирностью шрифта;
* выделение типом шрифта;
* выделение курсивом.

Процедура проведения экспериментального исследования заключается в следующем:

1. испытуемый заполняет регистрационную форму, знакомится с инструкцией по проведению эксперимента и выполняет несколько тренировочных заданий (рисунок 1.4);

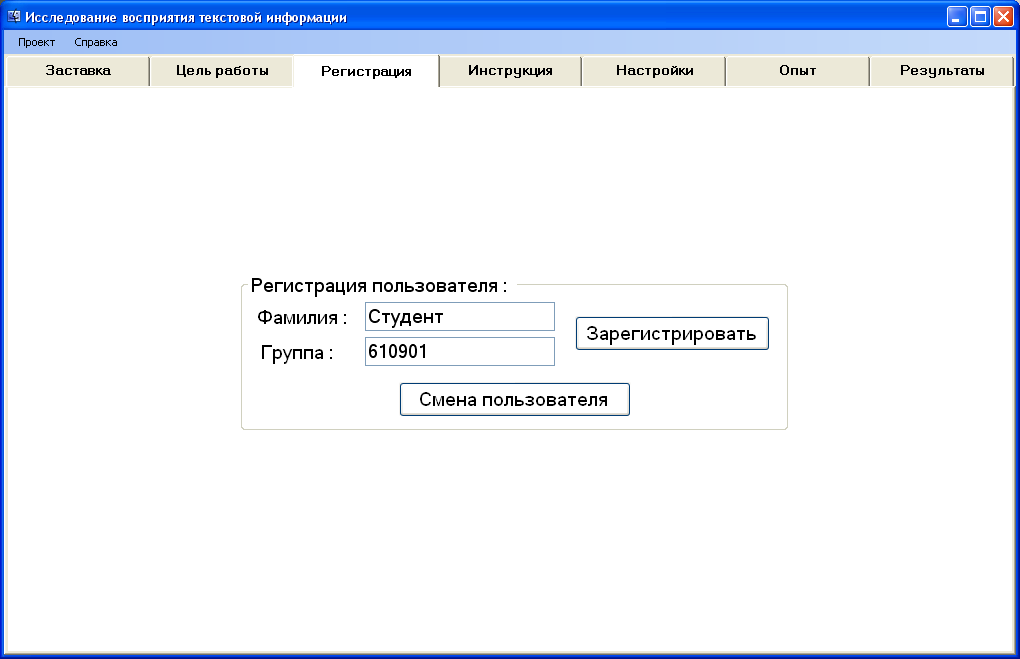


Рисунок 1.4 – Регистрация испытуемого

1. в каждом опыте на экране дисплея предъявляется стимул – набор слов, время экспозиции которого фиксировано и ограничено (рисунок 1.5). После прекращения экспозиции испытуемый вводит слова, которые он успел считать, используя клавиатуру компьютера (рисунок 1.6);

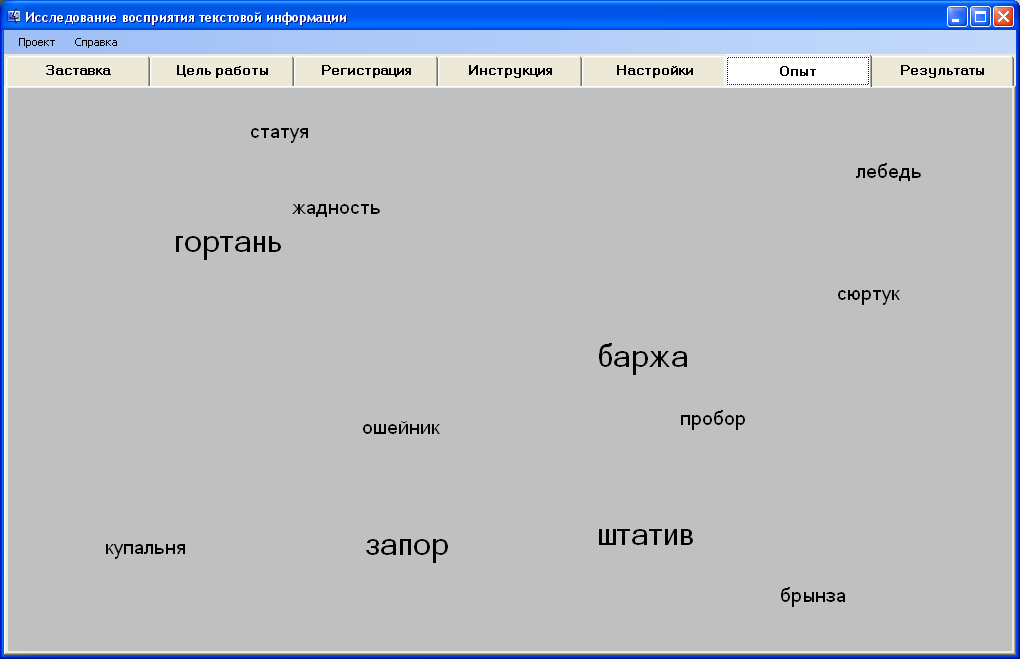
****

Рисунок 1.5 – Проведение опыта

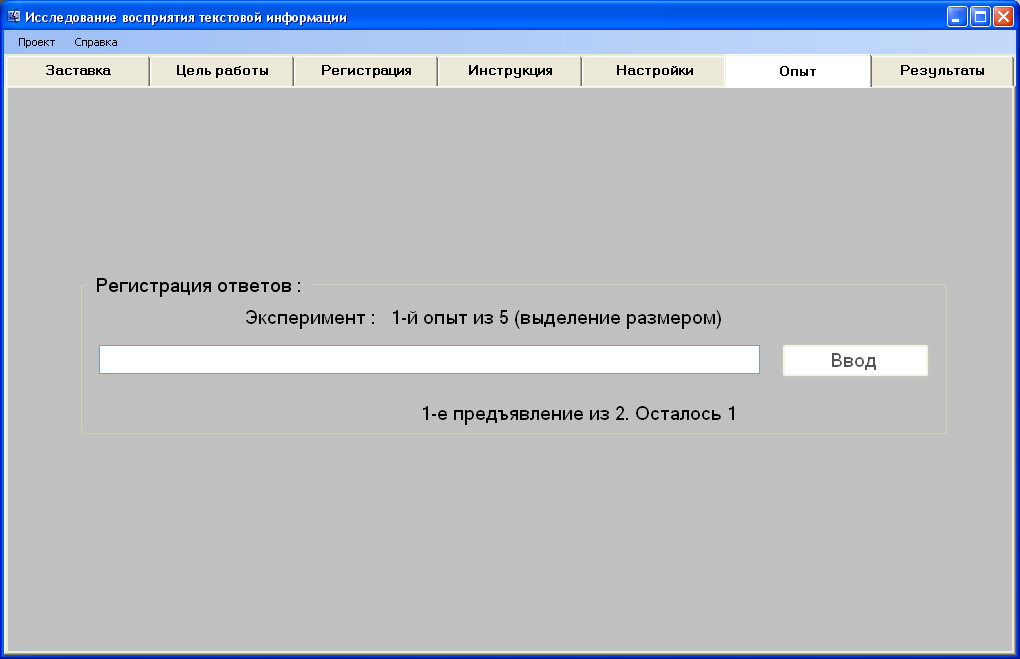
****

Рисунок 1.6 – Форма ввода слов

1. испытуемый выполняет экспериментальное задание, количество опытов в котором и параметры предъявляемых стимулов задаются заранее экспериментатором в настройках программно-аппаратного комплекса (рисунок 1.7);

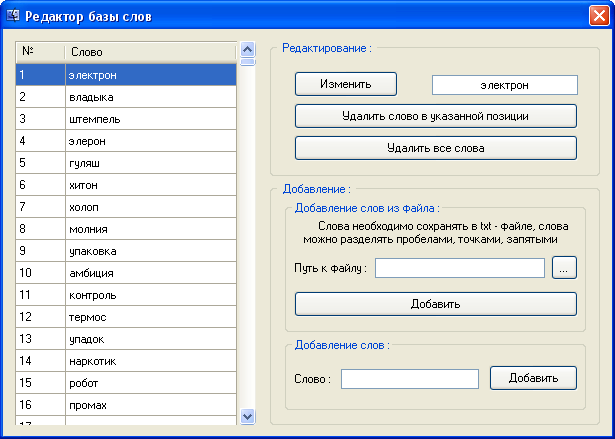


Рисунок 1.7 – Редактор базы слов

1. результаты работы каждого испытуемого сохраняются в специальном файле, с возможностью ограниченного доступа к ним для анализа и редактирования (рисунок 1.8).

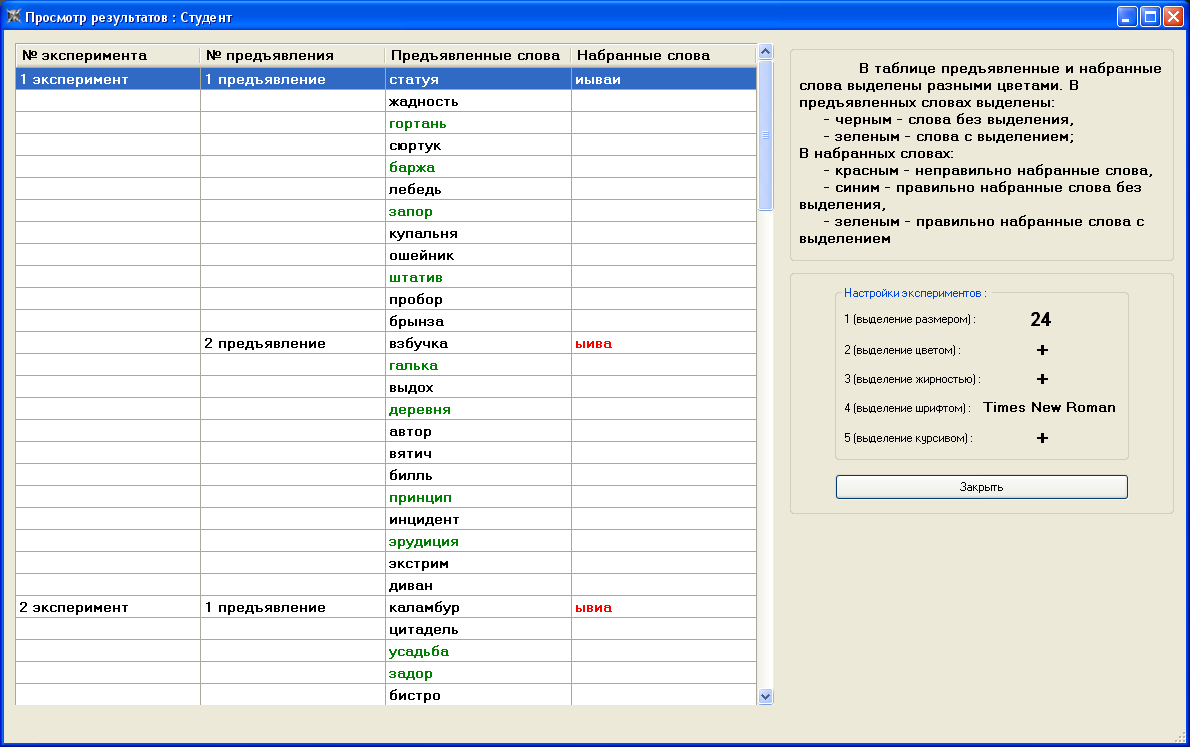
****

Рисунок 1.8 – Предъявленные и набранные слова

Еще одним аналогом является программно-аппаратный комплекс исследования восприятия знаковой информации, предназначенный для проведения экспериментального исследования процессов восприятия знаковой информации в зависимости от яркости, контраста и размеров знаков. Для проведения экспериментального исследования разработана следующая методика.

На экране дисплея испытуемому предъявляются наборы знаков, состоящие из 2-5 знаков (рисунок 1.9). Параметры набора остаются неизменными в каждом опыте, а предъявляемые наборы не повторяются. Время экспозиции в одних опытах неограниченное, в других – фиксированное и задаётся в настройках опыта. Изменяемыми параметрами в каждом опыте являются контраст знаков и фона или размер знаков.

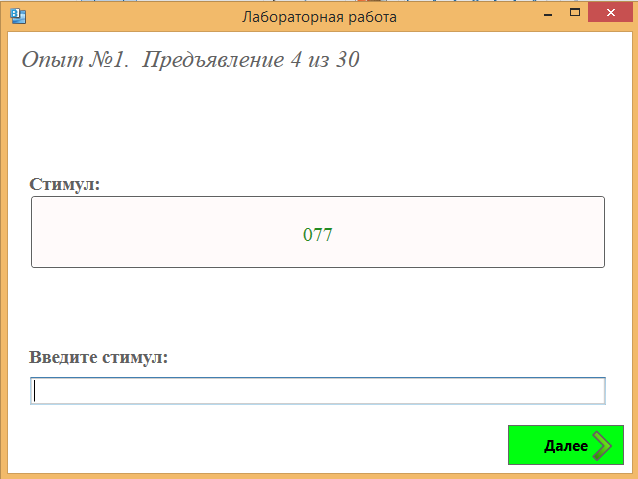


Рисунок 1.9 – Проведение опыта

Задача испытуемого в каждом случае: считать с экрана предъявленный набор и ввести его в ПК. При этом измеряется и регистрируется в протоколе опыта время, затраченное на считывание информации.

Работа включает четыре опыта, в каждом из которых решается своя исследовательская задача. Исследование может выполняться с различными сочетаниями опытов, т.е. каждый опыт является относительно независимым.

Так как испытуемый работает с изменяемыми параметрами, в данной системе разработана функция редактирования основных настроек проведения опыта, продемонстрированная на рисунке 1.10.

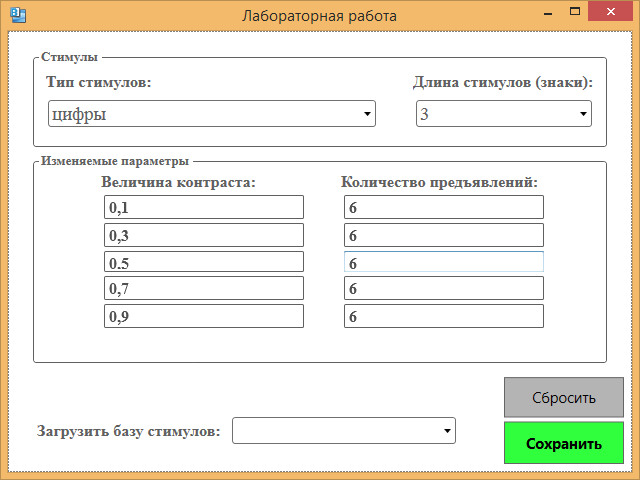


Рисунок 1.10 – Окно настроек опыта

По результатам выполнения каждого опыта создаётся протокол, который содержит: ФИО студента, дату, номер опыта, предъявленные стимулы, воспроизведённые стимулы, время считывания. Кроме этого в каждой серии определяется среднее время считывания и количество ошибок (рисунок 1.11).

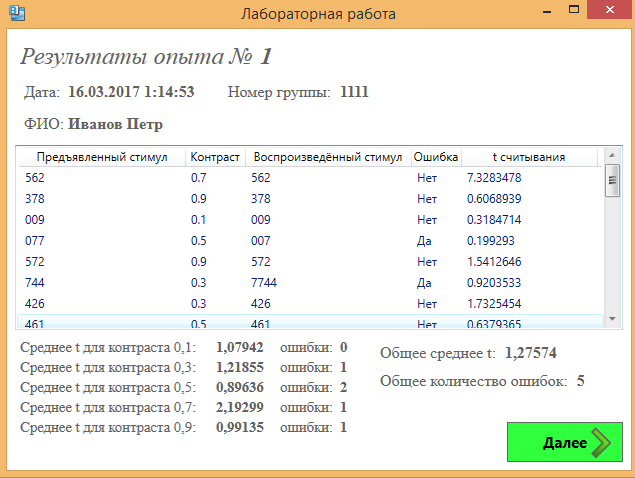


Рисунок 1.11 – Результаты проведения опыта

Функция возможности идентификации пользователей позволяет давать либо ограничивать доступ к определенному функционалу системы (рисунок 1.12).

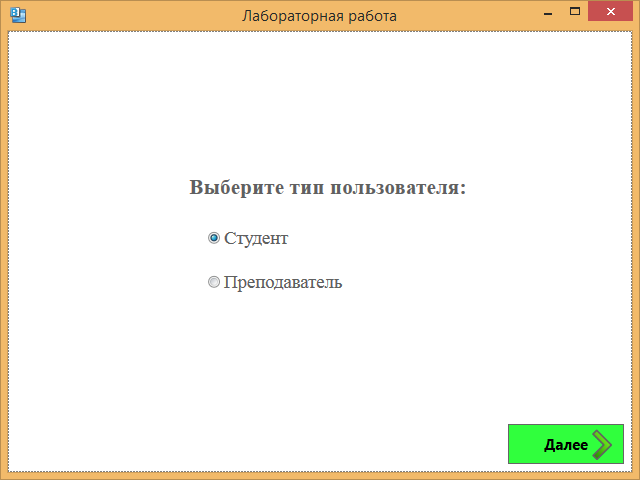


Рисунок 1.12 – Идентификация пользователя

Так при входе в систему в качестве преподавателя, у пользователя появляется возможность создания собственной базы символов, путем заполнения текстовых полей необходимым стимульным материалом (рисунок 1.13). Так же при создании базы необходимо указать ее название, определиться с типом стимулов и их длинной.

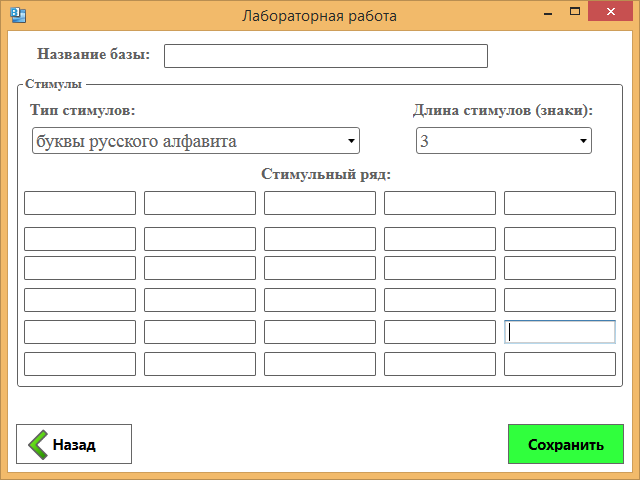


Рисунок 1.13 – Создание базы стимулов

## **1.2 Отличительные особенности компьютеризированных методик психологических исследований**

Рассмотренные выше компьютерные системы предназначены для различных инженерно-психологических или психологических исследований. Программные комплексы для инженерно-психологических исследований обладают большой функциональностью и способны решать большое количество задач.

В результате анализа аналогов, был выделен основной функционал, позволяющие создать наиболее удобный для эксплуатации ПАК. Функции, реализующие данный функционал являются:

1. идентификация пользователя в качестве испытуемого и экспериментатора;
2. регистрация испытуемого;
3. ограничение доступа к некоторым функциям, которые может использовать только экспериментатор;
4. создание и сохранение экспериментатором в памяти компьютера базы стимулов;
5. изменение настроек экспериментов экспериментатором (выбор предъявляемого в эксперименте стимульного материала, количества предъявлений в опыте, продолжительности экспозиции);
6. инструктирование испытуемого о предстоящем эксперименте и его задачах;
7. проведение тренировочной серии;
8. последовательное предъявление на экране дисплея элементов заданного набора стимулов;
9. фиксация реакции испытуемого на каждый предъявляемый стимул;
10. сохранение в памяти компьютера результатов работы студента;
11. возможность сохранения результатов на переносной носитель;
12. прекращение выполнения программы при необходимости.

## **1.3 Выводы и постановка задачи**

Разработанный ПК предназначен для проведения экспериментального исследования процессов воспроизведения и узнавания методом удержанных членов ряда и методом тождественных рядов.

Методика исследования заключается в следующем: на экране дисплея испытуемому предъявляются наборы стимулов. Способ предъявления задается настройками эксперимента. В каждом опыте выполняется N предъявлений. При этом величина N имеет определенное значение в зависимости от опыта. Предъявляемые наборы не повторяются. Время экспозиции предъявляемых стимулов задается настройками эксперимента.

Процедура всех опытов одна и та же, опыты различаются только видом предъявляемых стимулов. Каждый опыт состоит из двух частей.

В первой части испытуемому предлагается на экране дисплея набор стимулов А, состоящий из N элементов. Задача испытуемого – запомнить предъявленные стимулы и затем воспроизвести их спустя определенное время после окончания экспозиции. Результат воспроизведения регистрируется в протоколе опыта.

Во второй части опыта испытуемому предъявляют такой же экспозицией набор стимулов В, тоже состоящий из N элементов. Спустя определенное время после окончания его экспозиции испытуемому предъявляют набор С, который содержит в 2 раза больше элементов, при этом в нем содержатся в случайном порядке элементы набора В. Задача испытуемого – узнать «старые» элементы и с помощью мыши указать их. Время узнавания ограничено, оно задается настройками эксперимента. Результаты узнавания регистрируются в протоколе опыта.

Во всех опытах при последовательном предъявлении каждый очередной стимул предъявляется через определенный временной интервал, величина которого задается настройками эксперимента, при этом его предъявлению предшествует команда «Внимание».

Работа включает 5 опытов, различающихся видом используемых стимулов.

Учитывая назначение проектируемого программно-аппаратного комплекса, а также методику и процедуру эксперимента, определим задачи (функции), которые она должна решать:

1. Предъявлять на экране ПК справку о программе (ФИО разработчика, ФИО научного руководителя);

2. Ограничивать допуск испытуемого к некоторым функциям, которые должен выполнять только преподаватель;

3. Позволять преподавателю создавать и сохранять в памяти компьютера базовые массивы, из которых формируются наборы предъявляемых стимулов (или наборы предъявляемых стимулов);

4. Позволять преподавателю задавать настройки опытов (вариант задания, количество элементов в наборе, способ предъявления, продолжительность экспозиции, интервал);

5. Позволять преподавателю редактировать вводные теоретические сведения;

6. Обеспечивать возможность просмотра преподавателем результатов, выполненных студентами экспериментальных исследований;

7. Позволять преподавателю редактировать базу, сохраняемых результатов работы студентов (удалять файлы, потерявшие актуальность);

8. Проводить регистрацию студента (испытуемого);

9. Предъявлять на экране ПК вводные теоретические сведения

10. Инструктировать испытуемого о предстоящем опыте и его задачах;

11. Проводить перед началом экспериментов тренировочные серии с возможностью выбора студентом момента ее завершения;

12. Последовательно предъявлять на экране дисплея заданные наборы стимулов;

13. Позволять испытуемому вводить в компьютер воспроизведенные стимулы;

14. Позволять испытуемому вводить в компьютер узнанные стимулы;

15. Сохранять в памяти компьютера результаты работы испытуемого;

16. Предъявлять на экране ПК результаты выполненного эксперимента;

17. Включать в предъявляемую на экране ПК и сохраняемую информацию о результатах работы студента данные регистрации (ФИО, группа, дата и время работы);

18. Предъявлять на экране дисплея требования к математической обработке экспериментальны данных, содержащие все необходимые формулы для расчетов.

19. Давать возможность просматривать на экране ПК все наборы предъявляемых в эксперименте стимулов в том виде, в каком они предъявлялись испытуемому;

20. Обеспечивать возможность сохранения результатов эксперимента на переносном носителе информации;

21. Предоставлять студенту возможность выполнять опыты в любой очередности.

22. Предоставлять пользователю возможность прекращать работу на любом ее этапе.

# **2 ЭРГОНОМИЧЕСКОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Основной целью проектирования дипломного проекта является эргономическое проектирование системы, позволяющей сравнить продуктивность процессов воспроизведения и узнавания одного и того же материала. При проектировании системы требуется учитывать множество факторов: её стоимость, быстродействие, удобство пользования ею и др. Эти факторы можно объединить в группы, каждая из которых относится к той или иной составляющей системы: оборудование, рабочая среда и др. При эргономическом проектировании эти составляющие системы рассматриваются как взаимосвязанные элементы одного функционирующего объекта.

## **2.1 Анализ функций и их распределение в проектируемой системе**

Целью проекта является разработка программно-аппаратного комплекса (ПАК) для сравнения процессов воспроизведения и узнавания.

Назначение ПАК:

1)Проведение экспериментального исследования процессов воспроизведения и узнавания методом удержанных членов ряда и методом тождественных рядов.

2) ПАК предназначен для использования в качестве лабораторной работы по дисциплине «Психология восприятия и переработки информации»

Учитывая назначение проектируемого программно-аппаратного комплекса, а также методику и процедуру эксперимента, определим функции, которые она должна решать, а так же проведем их анализ:

1. Предъявлять на экране ПК справку о программе (ФИО разработчика, ФИО научного руководителя).

Справка о программе представляет текстовое сообщение, предъявляемое на дисплее ПК после щелчка левой кнопкой мыши по кнопке «Справка о программе», находящейся на титульной странице, содержащее ФИО разработчика программы и научного руководителя. Длительность экспозиции справки устанавливается фиксированной.

2. Ограничивать допуск испытуемого к некоторым функциям, которые должен выполнять только преподаватель.

ПАК проектируется для двух основных групп пользователей: «Администратор» (преподаватель) и «Испытуемый» (студент).

В соответствии с данным распределение пользователей на группы, ПАК будет выполнять определенные функции для каждой из групп.

Авторизация осуществляется путем выбора пользователем предложенных на экране дисплея вариантов: «Преподаватель», «Студент». Для обеспечения доступа к функциональности «Преподаватель», пользователю необходимо ввести пароль в специальное поле. При правильности введенного пароля пользователь проходит в систему дальше, иначе происходит ограничение доступа.

При входе в систему в качестве «Студента» ПАК реализует такие функции как:

1. Регистрация пользователя.

2. Инструктирование испытуемого о предстоящем опыте и его задачах.

3. Возможность проведения тренировочные серии экспериментов с возможностью выбора студентом момента ее завершения.

4. Предоставлять студенту возможность выполнять опыты в любой очередности.

5. Ввод испытуемым в компьютер воспроизведенные стимулы.

6. Ввод испытуемым в компьютер узнанные стимулы.

7. Предъявление на экране ПК результаты выполненного эксперимента.

8. Предъявление на экране дисплея требования к математической обработке экспериментальны данных, содержащие все необходимые формулы для расчетов.

9. Обеспечивать возможность сохранения результатов эксперимента на переносном носителе информации.

При входе в систему в качестве «Преподаватель» ПАК реализует такие функции как:

1. Позволять преподавателю создавать и сохранять в памяти компьютера базовые массивы, из которых формируются наборы предъявляемых стимулов.

2. Позволять преподавателю задавать настройки опытов.

3. Позволять преподавателю редактировать вводные теоретические сведения.

4. Обеспечивать возможность просмотра преподавателем результатов, выполненных студентами экспериментальных исследований.

5. Позволять преподавателю редактировать базу, сохраняемых результатов работы студентов.

3. Позволять преподавателю создавать и сохранять в памяти компьютера базовые массивы, из которых формируются наборы предъявляемых стимулов.

При входе в систему в качестве «Преподавателя» пользователь выбирает из выпадающего списка меню вкладку «Данные». В открывшемся окне пользователю предоставляется возможность ввода данных, необходимых для проведения эксперимента, в специальные формы с использованием клавиатуры компьютера. Для сохранение введенного набора стимулов преподавателю необходимо нажать кнопку «Сохранить».

4. Позволять преподавателю задавать настройки опытов (вариант задания, количество элементов в наборе, способ предъявления, продолжительность экспозиции, интервал).

При входе в систему в качестве «Преподавателя» пользователь выбирает из выпадающего списка меню вкладку «Настройки». В открывшемся окне пользователю предоставляется возможность задавать настройки опыта путем выбора варианта из выпадающего списка, а также количества элементов в наборе, способа предъявления, продолжительности экспозиции и интервала.

5. Позволять преподавателю редактировать вводные теоретические сведения.

При входе в систему в качестве «Преподавателя» пользователь выбирает из выпадающего списка меню вкладку «Теория». В открывшемся окне пользователю предоставляется возможность корректировки вводных теоретических сведений в специальной форме.

6. Обеспечивать возможность просмотра преподавателем результатов, выполненных студентами экспериментальных исследований.

При входе в систему в качестве «Преподавателя» пользователь выбирает из выпадающего списка меню вкладку «Результаты». В открывшемся окне пользователю выводится на экран дисплея таблица, состоящая из результатов, выполненных студентами экспериментальных исследований

7. Позволять преподавателю редактировать базу, сохраняемых результатов работы студентов (удалять файлы, потерявшие актуальность).

При входе в систему в качестве «Преподавателя» пользователь выбирает из выпадающего списка меню вкладку «Результаты». В открывшемся окне пользователю предоставляется возможность удаления из памяти компьютера файлов, потерявших актуальность.

8. Проводить регистрацию студента (испытуемого).

При входе в систему в качестве «Студента» пользователю открывается специальная форма, предназначенная для регистрации, которая осуществляется путем внесение пользователем свое фамилии и номера группы в специальные поля.

9. Предъявлять на экране ПК вводные теоретические сведения

При идентификации пользователя как «Студента» и нажатии испытуемым на кнопку «Теория» в открывающемся окне осуществляется вывод на экран вводных теоретических сведений.

10. Предоставлять студенту возможность выполнять опыты в любой очередности.

При идентификации пользователя как «Студента», испытуемый переходит на форму с опытами, где ему предоставляется возможность выбора очередности эксперимента путем выбора соответствующего опыта из выпадающего списка

11. Инструктировать испытуемого о предстоящем опыте и его задачах.

При идентификации пользователя как «Студента» и выборе им из выпадающего списка соответствующего опыта, на вплывающей форме осуществляется вывод на экран текстового сообщения, содержащего информацию о задачах предстоящего опыта и инструкцию к выполнению.

12. Проводить перед началом экспериментов тренировочные серии с возможностью выбора студентом момента ее завершения.

При идентификации пользователя как «Студента» и выборе им опыта, испытуемому предоставляется возможность выбора варианта похождения опыта, а именно в качестве «Тренировочная серия» либо «Рабочая серия». При выборе пользователем кнопки «Тренировочная серия» осуществляется вывод на дисплей ПК подгруженных из базы компьютера данных, необходимых для проведения тренировочной серии эксперимента. В любой момент выполнения тренировочной серии испытуемый может завершить ее, нажав на кнопку «Завершить».

13. Последовательно предъявлять на экране дисплея заданные наборы стимулов.

При идентификации пользователя как «Студента» и выборе им опыта из выпадающего списка, испытуемый осуществляет выбор варианта прохождения эксперимента («Тренировочная серия» либо «Рабочая серия»). Затем в появившемся окне осуществляется вывод на экран дисплея заданных наборов стимулов, подгруженных из базы памяти компьютера в соответствии с выбором определенного опыта.

14. Позволять испытуемому вводить в компьютер воспроизведенные стимулы.

При идентификации пользователя как «Студента» и выборе им опыта из выпадающего списка, испытуемый осуществляет выбор варианта прохождения эксперимента («Тренировочная серия» либо «Рабочая серия»). Далее происходит демонстрация рядов символов; пользователю предоставляется возможность ввода воспроизведенных стимулов в специальные формы.

15. Позволять испытуемому вводить в компьютер узнанные стимулы.

При идентификации пользователя как «Студента» и выборе им опыта из выпадающего списка, испытуемый осуществляет выбор варианта прохождения эксперимента («Тренировочная серия» либо «Рабочая серия»). Далее происходит демонстрация рядов символов; пользователю предоставляется возможность ввода узнанных стимулов в специальные формы.

16. Сохранять в памяти компьютера результаты работы испытуемого.

Сохранение раннее введенных испытуемым данных происходит по умолчанию путём создания специального файла в определённой директории на жёстком диске компьютера

17. Предъявлять на экране ПК результаты выполненного эксперимента.

При идентификации пользователя как «Студента» и нахождении испытуемого на вкладке «Результаты» организуется вывод на экран текстового сообщения в виде таблицы, содержащей результаты выполненного эксперимента.

18. Включать в предъявляемую на экране ПК и сохраняемую информацию о результатах работы студента данные регистрации (ФИО, группа, дата и время работы).

При идентификации пользователя как «Студента» и нахождении испытуемого на вкладке «Результаты» организуется вывод на экран текстового сообщения, содержащего результаты выполненного эксперимента и данные регистрации (ФИО, группа, дата и время работы).

19. Предъявлять на экране дисплея требования к математической обработке экспериментальны данных, содержащие все необходимые формулы для расчетов.

В случае идентификации пользователя как «Студента» и нахождении испытуемого на вкладке «Результаты», при нажатии пользователем на кнопку «Расчетные формулы» выполняется вывод всплывающего окна, содержащего требования к математической обработке экспериментальны данных, все необходимые формулы для расчетов

20. Давать возможность просматривать на экране ПК все наборы предъявляемых в эксперименте стимулов в том виде, в каком они предъявлялись испытуемому.

При идентификации пользователя как «Студента» и нахождении испытуемого на вкладке «Результаты», вместе с результатами студента осуществляется предъявление на экране дисплея всех наборов предъявляемых в эксперименте стимулов в том виде, в каком они предъявлялись испытуемому. Благодаря такой демонстрации пользователь сможет сравнить свои результаты с предъявленными.

21. Обеспечивать возможность сохранения результатов эксперимента на переносном носителе информации.

Так как сохранение результатов эксперимента организуется в виде текстового файла, находящегося в определенной директории на жестком диске компьютера, пользователь может скопировать данный файл на переносной носитель.

22. Предоставлять пользователю возможность прекращать работу на любом ее этапе.

Завершение работы приложения осуществляется при нажатии пользователем на кнопку «Закрыть», находящуюся в верхнем правом углу приложения.

## **2.2 Разработка алгоритмов работы пользователей программного комплекса**

Специфической и одной из наиболее важных задач эргономического проектирования, принципиально отличающей его от других видов системного проектирования, является проектирование деятельности оператора.

В результате ее решения, в частности, определяются структура и алгоритмы деятельности оператора в различных режимах работы СЧМ, способы выполнения этой деятельности, требования к психофизиологическим характеристикам человека производится проверка выполнения предельно допустимых норм деятельности оператора.

Таблица 2.1 - Алгоритм работы человека в подсистеме «преподаватель – ПК- среда» в режиме создание/удаление наборов предъявляемых стимулов

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к СОИ | Обращение к ОУ |
| 1 | 2 | 3 |
| 1. Включение ПК | Индикатор на системном блоке | Кнопка включения на системном блоке |
| 1. Включение дисплея | Индикатор включения дисплея | Кнопка включения дисплея |
| 1. Загрузка программы | Ярлык на экране дисплея | Щелчок мышкой |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Знакоместо на экране дисплея.  Сообщение об успешной авторизации | Клавиатура ПК, мышь |
| 1. Выбор режима работы «База стимулов» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Выбор номера эксперимента | Выпадающий список | Щелчок мышкой |
| **Создание базы стимулов** | | |
| 1.Запуск модуля создание массива стимулов | Кнопка «Создать» на экране дисплея | Щелчок мышкой |
| 2.Заполнение форм массива стимулов | Знакоместо на экране дисплея | Клавиатура ПК |
| 3.Заполнение форм массива изображениями | Кнопка «Выбрать файл» | Щелчок мышкой |
| 4.Выбор изображения | Форма окна выбора файлов | Щелчок мышкой |
| 5. Проверка корректности введенных данных | Форма приложения на экране дисплея |  |
| **Удаление базы стимулов** | | |
| 1. Запуск модуля удаления массива стимулов | Кнопка «Удалить» на экране дисплея | Щелчок мышкой |
| 2.Выбор удаляемого файла | Выделение удаляемого файла | Щелчок мышкой |
| 3.Подтверждение удаления | Сообщение на экране дисплея.Нажатие кнопки «Удалить» | Щелчок мышкой |
|  |  |  |
| 1. Сохранение внесенных изменений | Кнопка «Сохранить» на экране дисплея  Сообщение о внесении изменений | Щелчок мышкой |
| 1. Завершение работы с приложением | Кнопка «Закрытия приложения» на экране дисплея | Щелчок мышкой |
| 1. Выключение ПК | Кнопка «Пуск-> Завершение работы» на экране дисплея | Щелчок мышкой |
| 1. Выключение дисплея | Индикатор выключения дисплея | Кнопка выключения  дисплея |

Таблица 2.2 - Алгоритм работы человека в подсистеме «преподаватель – ПК- среда» в режиме задания настроек опыта.

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к СОИ | Обращение к ОУ |
| 1 | 2 | 3 |
| 1. Включение ПК | Индикатор на системном блоке | Кнопка включения на системном блоке |
| 1. Включение дисплея | Индикатор включения дисплея | Кнопка включения дисплея |
| 1. Загрузка программы | Ярлык на экране дисплея | Щелчок мышкой |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Знакоместо на экране дисплея | Клавиатура ПК |
| 1. Выбор режима работы «Настройки» | Кнопка «Настройки» на экране дисплея | Щелчок мышкой |
| **Задание настроек опыта** | | |
| 1.Запуск модуля «Задания настроек опыта» | Кнопка «Настройки опыта» на экране дисплея | Щелчок мышкой |
| 2.Ввод варианта задания | Знакоместо на экране дисплея | Клавиатура ПК |
| 3.Ввод количества элементов в наборе | Знакоместо на экране дисплея | Клавиатура ПК |
| 4.Ввод способа предъявления | Выпадающий список | Щелчок мышкой |
| 5.Ввод продолжительности экспозиции | Знакоместо на экране дисплея | Клавиатура ПК |
| 6.Задание интервала | Знакоместо на экране дисплея | Клавиатура ПК |
| 7.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
| **Ввод теоретических сведений** | | |
| 1.Запуск модуля «Теория» | Кнопка «Теория» на экране дисплея | Щелчок мышкой |
| 2. Ввод теоретических сведений | Знакоместо на экране дисплея | Клавиатура ПК |
| 3.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
| **Редактирование теоретических сведений** | | |
| 1.Запуск модуля «Теория» | Кнопка «Теория» на экране дисплея | Щелчок мышкой |
| 2.Переход в режим редактирования | Кнопка «Редактировать» на экране дисплея | Щелчок мышкой |
| 3.Ввод теоретических сведений | Знакоместо на экране дисплея | Клавиатура ПК |
| 4.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
| **Ввод инструкции по выполнению эксперимента** | | |
| 1.Запуск модуля «Инструкции» | Кнопка «Инструкции» на экране дисплея | Щелчок мышкой |
| 2.Ввод теоретических сведений | Знакоместо на экране дисплея | Клавиатура ПК |
| 3.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
| **Редактирование инструкции по выполнению эксперимента** | | |
| 1.Запуск модуля «Инструкции» | Кнопка «Инструкции» на экране дисплея | Щелчок мышкой |
| 2.Переход в режим редактирования | Кнопка «Редактировать» на экране дисплея | Щелчок мышкой |
| 3.Ввод теоретических сведений | Знакоместо на экране дисплея | Клавиатура ПК |
| 4.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
| **Ввод требований к математической обработке** | | |
| 1.Запуск модуля «Расчеты» | Кнопка «Расчеты» на экране дисплея | Щелчок мышкой |
| 2.Ввод теоретических сведений | Знакоместо на экране дисплея | Клавиатура ПК |
| 3.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
| **Редактирование требований к математической обработке** | | |
| 1.Запуск модуля «Расчеты» | Кнопка «Расчеты» на экране дисплея | Щелчок мышкой |
| 2.Переход в режим редактирования | Кнопка «Редактировать» на экране дисплея | Щелчок мышкой |
| 3.Ввод теоретических сведений | Знакоместо на экране дисплея | Клавиатура ПК |
| 4.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
|  |  |  |
| 1. Сохранение внесенных изменений | Кнопка «Сохранить» на экране дисплея  Сообщение о внесении изменений | Щелчок мышкой |
| 1. Завершение работы с приложением | Кнопка «Закрытия приложения» на экране дисплея | Щелчок мышкой |
| 1. Выключение ПК | Кнопка «Пуск-> Завершение работы» на экране дисплея | Щелчок мышкой |
| 1. Выключение дисплея | Индикатор выключения дисплея | Кнопка выключения  дисплея |

Таблица 2.3 - Алгоритм работы человека в подсистеме «преподаватель – ПК- среда» в режиме работы (просмотр/удаление) с базой результатов студентов

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к СОИ | Обращение к ОУ |
| 1 | 2 | 3 |
| 1. Включение ПК | Индикатор на системном блоке | Кнопка включения на системном блоке |
| 1. Включение дисплея | Индикатор включения дисплея | Кнопка включения дисплея |
| 1. Загрузка программы | Ярлык на экране дисплея | Щелчок мышкой |
| 1. Выбор режима «Преподаватель» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод пароля | Знакоместо на экране дисплея | Клавиатура ПК |
| 1. Выбор режима работы «Результаты» | Кнопка на экране дисплея | Щелчок мышкой |
| **Удаление результатов** | | |
| 1.Выбор результатов, потерявших актуальность | Выделение результата в списке | Щелчок мышкой |
| 2.Удаление результатов | Кнопка на экране дисплея | Щелчок мышкой |
| 3.Подтверждение удаления | Сообщение на экране дисплея  Кнопка «Удалить» | Щелчок мышкой |
| 4.Сохранение внесенного изменения | Кнопка на экране дисплея | Щелчок мышкой |
|  |  |  |
| 1. Завершение работы с приложением | Кнопка «Закрытия приложения» на экране дисплея | Щелчок мышкой |
| 1. Выключение ПК | Кнопка «Пуск-> Завершение работы» на экране дисплея | Щелчок мышкой |
| 1. Выключение дисплея | Индикатор выключения дисплея | Кнопка выключения  дисплея |

Таблица 2.4 - Алгоритм работы человека в подсистеме «студент – ПК- среда» в режиме выполнения эксперимента

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к СОИ | Обращение к ОУ |
| 1 | 2 | 3 |
| 1. Включение ПК | Индикатор на системном блоке | Кнопка включения на системном блоке |
| 1. Включение дисплея | Индикатор включения дисплея | Кнопка включения дисплея |
| 1. Загрузка программы | Ярлык на экране дисплея | Щелчок мышкой |
| 1. Выбор режима «Студент» | Кнопка на экране дисплея | Щелчок мышкой |
| 1. Ввод регистрационных данных (ФИО, номер группы) | Знакоместо на экране дисплея | Клавиатура ПК |
| 1. Сохранение пользователя в базе компьютера | Кнопка «Регистрация» на экране дисплея;  Сообщение об успешной аутентификации | Щелчок мышкой |
| 1. Ознакомление с теоретическими сведениями | Пункт меню «Теория» или кнопка «Теория» | Щелчок мышкой |
| 1. Выбор эксперимента | Элемент из выпадающего списка | Щелчок мышкой |
| 1. Запуск модуля опытов | Кнопка «Начать» | Щелчок мышкой |
| 1. Ознакомление с инструкцией по выполнению эксперимента | Форма приложения на экране дисплея |  |
| 1. Прохождение тренировочной серии | Кнопка «Начать» на экране дисплея | Щелчок мышкой |
| Прохождение тренировочной серии (см. табл. 2.5) | | |
| 1. Прохождение основной части эксперимента | Кнопка «Начать» на экране дисплея | Щелчок мышкой |
| Прохождение основной части эксперимента (см. табл. 2.5) | | |
| 1. Просмотр результатов эксперимента | Форма приложения на экране дисплея |  |
| 1. Просмотр требований к математической обработке | Пункт меню «Расчеты» или кнопка «Расчеты» | Щелчок мышкой |
| 1. Завершение работы с приложением | Кнопка «Закрытия приложения» на экране дисплея | Щелчок мышкой |
| 1. Выключение ПК | Кнопка «Пуск-> Завершение работы» на экране дисплея | Щелчок мышкой |
| 1. Выключение дисплея | Индикатор выключения дисплея | Кнопка выключения  дисплея |
| Примечания:  Выход из приложения доступен в любой момент выполнения опыта. | | |

Таблица 2.5 - Алгоритм работы человека в подсистеме «студент – ПК- среда» в режиме прохождения тренировочной или рабочей серии

|  |  |  |
| --- | --- | --- |
| Содержание операции | Обращение к СОИ | Обращение к ОУ |
| 1 | 2 | 3 |
| 1. Просмотр заданных наборов стимулов | Форма приложения на экране дисплея |  |
| 1. Ввод воспроизведенных стимулов | Знакоместо на экране дисплея | Клавиатура ПК, мышь |
| 1. Просмотр заданных наборов стимулов | Форма приложения на экране дисплея |  |
| 1. Ввод узнанных стимулов | Кнопки «Да» и «Нет» на экране дисплея | Щелчок мышкой |
| 1. Просмотр заданных наборов графических стимулов | Форма приложения на экране дисплея |  |
| 1. Ввод воспроизведенных графических стимулов | Панель для отрисовки | Перемещение мышки с нажатой клавишей |
| 1. Просмотр заданных наборов графических стимулов | Форма приложения на экране дисплея |  |
| 1. Ввод узнанных графических стимулов | Форма приложения на экране дисплея | Щелчок мышкой |

## **2.3 Разработка эргономических требований и сценария информационного взаимодействия**

В результате эргономического проектирования был разработан пользовательский интерфейс, который удовлетворяет условиям технического задания. Полный сценарий продемонстрирован в приложении Ж.

Эргономическая оценка инженерных решений − это комплекс научно-технических и организационно-методических мероприятий по оценке выполнения в проектных документах и в образцах СЧМ эргономических требований технического задания, нормативно-технических и руководящих документов, а также разработка рекомендаций для устранения отступлений от этих требований. Указанная оценка проводится при обосновании выполнения каждого этапа опытно-конструкторской разработки: технического предложения, эскизного проекта, рабочего проекта.

Исходными материалами для эргономической оценки служат техническое задание на разработку систем, техническая документация, показывающая результаты эргономического проектирования, конструкторские документы, образцы системы «человек-машина – среда» и их составные части.

Далее будет произведен расчет эргономической оценки пользовательского интерфейса проектируемой системы при использовании экспертного метода. Общие эргономические требования к проектируемой системе приведены в таблице 2.6.

Таблица 2.6 – Общие эргономические требования к проектируемой системе и соответствующие им единичные эргономические показатели

|  |  |  |
| --- | --- | --- |
| **Группа** | **Эргономические требования** | **Единичные эргономические показатели** |
| 1 | 2 | 3 |
| **Психофизиологические** | **ПФ-1**. Соответствие 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 размеров графических изображений на экране дисплея оперативному порогу зрения человека; | Размеры графических изображений |
| **ПФ-5.** Соответствие расположения надписей условиям их оптимального считывания | Расположение и ориентация надписей на экране дисплея |
| **Психологические** | **П-1.** Соответствие сложности инструкций, времени, отводимому на их восприятие | Длина инструкции и время ее экспозиции |
| **П-2**. Один и тот же характер команд на протяжении всего периода работы в системе в схожих ситуациях | Тип ОУ и их обозначение |
| **П-3.** Наличие указаний на проблемы, возникающие в процессе обслуживания системы | Сообщения об ошибочных действиях пользователей |
| **П-4.** Наличие подсказок о следующих шагах работы в системе | Сообщения о следующих действиях пользователей |
| **П-5.** Наличие предупреждений о нежелательных последствиях некоторых действий | Предупреждения о возможных нежелательных действиях |
| **П-6.** Соответствие цветов знаков и надписей сформированным стереотипам восприятия цвета | Цвета знаков, кнопок, надписей |
| **П-7.** Соответствие формы и расположения знаков сформированным стереотипам восприятия | Форма и ориентация знаков |
| **П-8.** Отсутствие нестандартных сокращений и аббревиатур | Словарный состав текстовых инструкций |
| **П-9.** Использование необходимых средств привлечения внимания | Используемые средства привлечения внимания пользователя (цвет, мигание, звуковые сигналы) |
| **П-10.** Наличие индикатора степени выполнения функций | Наличие и вид индикатора выполнения |
| **П-11.** Наличие кратких и понятных заголовков окон | Наличие и вид заголовков окон |
| **П-12.** Использование для названий пунктов меню одного слова (глагола для действий, существительного для объектов) | Названия пунктов меню |
| **П-13.** Применение в названиях пунктов меню норм использования заглавных букв, принятых в языке. | Названия пунктов меню |
|  | **П-14.** Отсутствие у пользователей сложностей в поиске необходимых директив (элементов интерфейса) для управления процессом решения поставленной задачи | Естественность взаимодействия |
|  | **П-15.** Сообщение об ошибке должно отвечать всего на три вопроса:  - в чем заключается проблема?  - как исправить эту проблему сейчас?  - как сделать так, чтобы проблема не повторилась? | Содержание сообщений об ошибках |
|  | **П-16.** Вежливое и понятное пользователю сообщение об ошибках | Содержание сообщений об ошибках |
|  | **П-17.** Целесообразно использовать в рамках одного приложения окна, построенные по одному шаблону, в которых одинаковые элементы расположены одинаково. | Окна интерфейса в программы |
|  | **П-18.** Интерфейсные элементы должны иметь не только согласованные изображения, но и согласованное управление. Например, активизация всех пиктограмм - двойным щелчком мыши. | Средства управления элементами интерфейса |
|  | **П-19.** Следует учитывать при проектировании меню и диалоговых окон стереотипную логическую последовательность чтения текста справа налево и сверху вниз. В левом верхнем углу следует располагать элемент, с которым пользователь должен работать в первую очередь, а в правом нижнем углу - тот, который используется в конце. Не следует первым элементом меню ставить опцию "Выход". | Компоновка опций меню и диалоговых окон |
| **Физиологические** | **Ф-1.** Соответствие размеров зон установки курсора физиологическим возможностям движений | Размеры меню, списков, кнопок на экране дисплея |
| **Ф-2**. Использование значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации. | Используемые значения по умолчанию |
| **Ф-3.** Использование командных кнопок для ввода явных действий | Наличие командных кнопок для ввода явных действий |
| **Ф-4.** Отсутствие необходимости устанавливать фокус ввода в открывающихся текстовых полях | Наличие фокуса ввода в текстовых полях по умолчанию |
| **Ф-5.** Соответствие времени экспозиции списков, меню, кнопок скоростным возможностям человека | Длительность экспозиции средств взаимодействия |
| **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 |
| **Гигиенические** | **Г-1.** Соответствие параметров изображения на экране дисплея условиям комфорта зрительной работы пользователей (отсутствие мельканий т тд.) | Энергетические и временные параметры изображения на экране дисплея |
| **Г-2.** Соответствие уровней шума и вибрации на рабочем месте гигиеническим нормам | Отсутствие раздражающих шумовых факторов |
| **Г-3** Соответствие параметров микроклимата и газового состава воздуха рабочей зоны гигиеническим нормам | Поддержание микроклимата в соответствие с нормами, отсутствие вредных веществ в составе воздуха. |
| **Социальнопсихологи-ческие** | **СП-1.**Наличие средств ограничения допуска к некоторым функциям пользователям, не имеющим требуемого статуса. | Способ разграничения прав пользователей разных типов |

Далее проводим оценку значений единичных эргономических показателей. Единичные эргономические показатели оцениваются по бинарной шкале, они принимают значение, равное "1", если фактическое значение показателя соответствует рекомендуемому, и равное "0", если оно ему не соответствует.

Групповой эргономический показатель (ЭПгр) рассчитывается как общая оценка по группе единичных показателей по формуле 2.1

ЭПгр = ∑ 1 / ∑1 + ∑ 0, ( 2.1)

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

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

Результаты оценки значений единичных и групповых эргономических показателей приведены в таблице 2.7.

Таблица 2.7 – 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-5, П-8, П-15 = 0  П-1, П-2, П-3, П-4, П-6, П-7, П-9,  П-10, П-11, П-12, П-14, П-16, П-17, П-18,П-19 = 1 | 16 х 1 / 19 = 0,84 |
| Физиологические ЭП | Ф-2 = 0  Ф-1, Ф-3, Ф-4, Ф-6= 1 | 5 \* 1 / 6 = 0,83 |
| Гигиенические ЭП | Г-1, Г-2, Г-3 = 1 | 3 \* 1 / 3 = 1 |
| Социально-психологические ЭП | СП-1 =1 | 1 \* 1 / 1 = 1 |
| Антропометрические ЭП | Не актуальны для данной СЧМ |  |

Далее оцениваются эргономические свойства СЧМ. Однако поскольку для нашей 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, ( 2.2 )

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 2.8).

Таблица 2.8 – 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 |

С учетом данных таблицы 2.8 и таблицы 2.9 по формуле (2.2) определяем количественное значение эргономического свойства «управляемость»

ЭСВ управляемость = (0,25 \* 1) + (0,4 \* 0,84) + (0,15 \* 0,83) + (0,1 \* 1) + (0,1 \* 1) = 0,91

Поскольку в нашей системе значимым с точки зрения формирования интегральной оценки – эргономичности - является только одно эргономическое свойство – «управляемость» принимаем за оценку эргономичности полученное значение.

Следовательно, эргономичность нашей системы равна 0,91, что соответствует оценке "отлично" - эргономические характеристики изделия соответствуют базовым значениям.

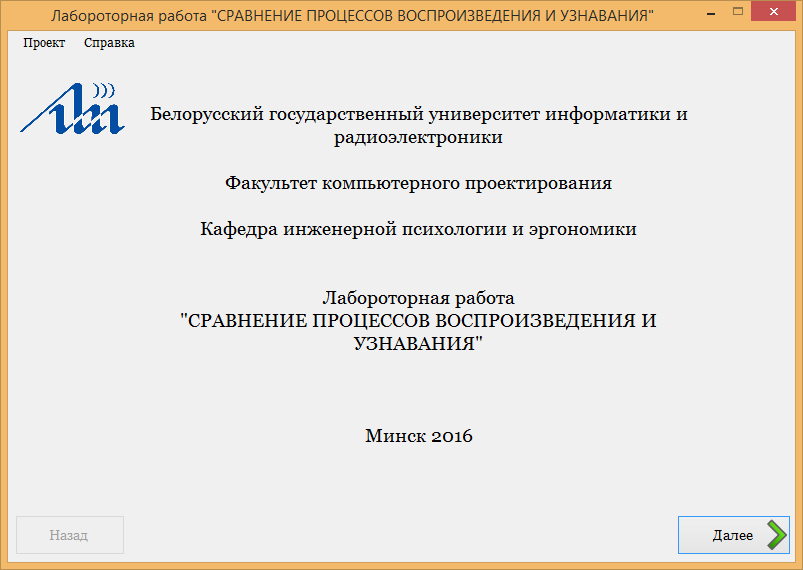
После такой общей оценки производится анализ единичных показателей, значения которых не соответствуют эргономическим требованиям и намечаются мероприятия по рационализации оцениваемой системы. Результаты данного этапа представлены в таблице 2.10.

Таблица 2.10 – Рекомендации по улучшению эргономичности проектируемой системы

|  |  |
| --- | --- |
| Невыполненное эргономическое требование | Предложение по улучшению эргономичности |
| **П-5.** Наличие предупреждений о нежелательных последствиях некоторых действий | Выводить сообщения о возможности не сохранения полученных результатов |
| **П-8.** Выделение в текстовых инструкциях смысловых фрагментов | В инструкциях испытуемому выделить абзацами смысловые фрагменты |
| **П-15.** Сообщение об ошибке должно отвечать всего на три вопроса:  - в чем заключается проблема?  - как исправить эту проблему сейчас?  - как сделать так, чтобы проблема не повторилась? | Сформулировать текст сообщения об ошибке, соответствующий поставленным требованиям |
| **Ф-2**. Использование значения по умолчанию где только возможно, чтобы минимизировать процесс ввода информации. | Разработать алгоритм проведения эксперимента, позволяющий использовать значения по умолчанию |

В результате эргономического проектирования был разработан пользовательский интерфейс, который удовлетворяет условиям технического задания.

Запуск программы осуществляется нажатием значка на рабочем столе, после чего пользователь видит главное окно компьютерной системы (см. рисунок 2.1). На информационном поле, кроме общей информации, находятся кнопка изменения размера экрана, кнопка закрытия программы и кнопка перехода на следующую страницу.

  
  
Рисунок 2.1 – Заставка программы

Затем осуществляется выбор типа пользователя: студент или преподаватель (см. рисунок 2.2).

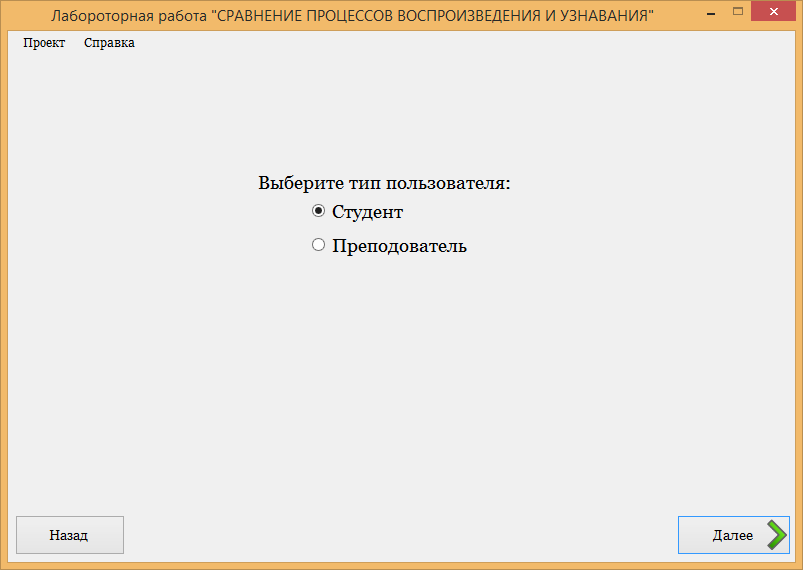


Рисунок 2.2 - Выбор типа пользователя

Поле «Студент» отмечено по умолчанию, так как ожидается более частое использование этого типа пользователя.

При нажатии кнопки "Далее" осуществляется переход к следующему информационному полю: регистрация испытуемого (см. рисунок 2.3).

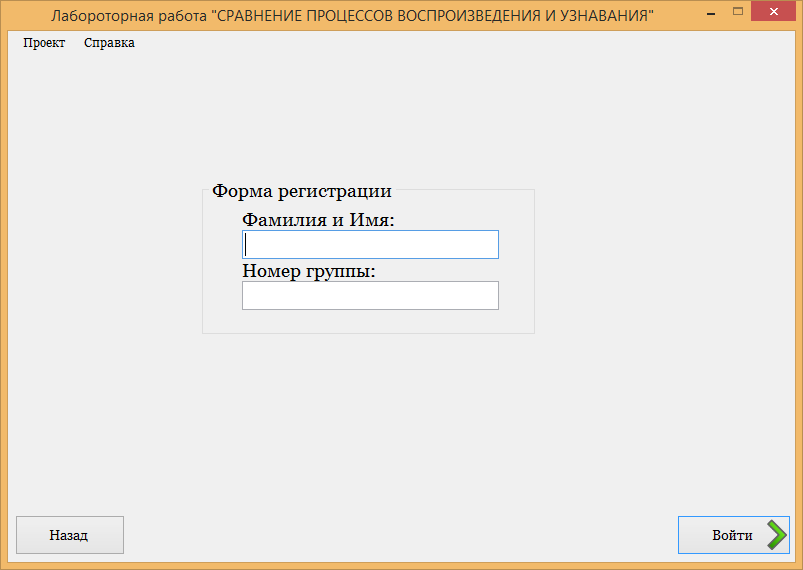


Рисунок 2.3 – Форма регистрации испытуемого

Форма регистрации представляет из себя поля для ввода информации. Поле для ввода фамилии и имени является активным по умолчанию, с целью сокращения лишнего действия по активации данного поля пользователем. Во второе поле пользователю необходимо ввести номер группы. Оба эти поля являются обязательными для заполнения.

После заполнения формы регистрации и нажатия кнопки «Войти» испытуемый переходит на форму выбора опыта, где осуществляется выбор требуемого эксперимента с последующим предъявлением инструкций к нему. На рисунке 2.4 представлена общая инструкция к первому опыту. При выборе другого опыта инструкция изменяется в соответствии с условиями эксперимента.

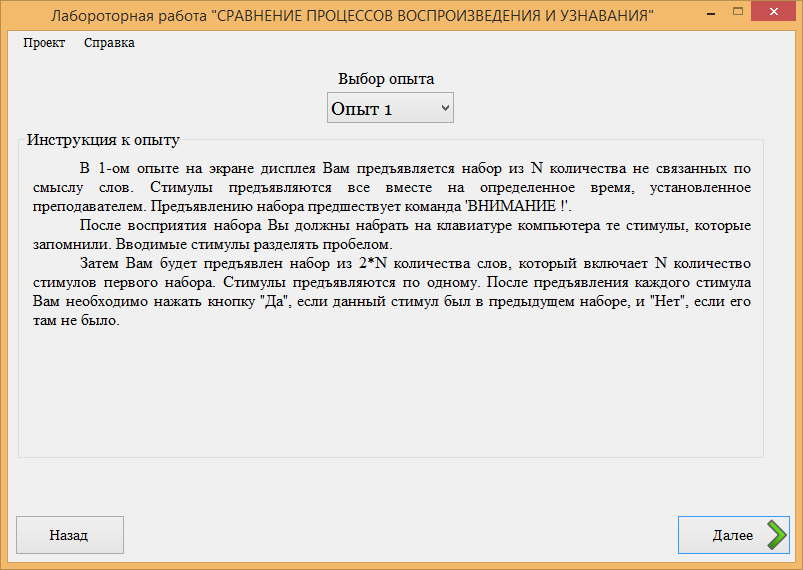


Рисунок 2.4 - Выбор опыта и предъявление инструкций

По умолчанию выбран первый опыт, так как ожидается последовательное выполнение опытов, начиная с первого.

Следующим этапом является выбор: проведение тренировочной серии или выполнение опыта (см. рисунок 2.5). Каждый из опытов состоит из двух частей.

При переходе в режим тренировочной серии испытуемому в течение нескольких секунд демонстрируется надпись «ВНИМАНИЕ!», а затем подается стимульный ряд (см. рисунок 2.6, а). При выборе тренировочной серии проводится предъявление стимульного материала в точности демонстрирующего стимульный материал для опытов.

После прекращения предъявления стимульного ряда, испытуемому необходимо воспроизвести запомнившиеся ему стимулы и нажать кнопку «Далее».

Затем начинается вторая часть опыта (см. рисунок 2.6, б). Испытуемому по одному в произвольном порядке демонстрируются стимулы присутствующие в первой части опыта, объединенные со стимулами, ранее не предъявляемыми в эксперименте. Студенту необходимо опознать имел ли место данный стимул в предыдущей части опыта и нажать кнопку «Да», если он считает, что стимул присутствовал, и – «Нет», если отсутствовал.

Испытуемый может закончить прохождение тренировочной серии в любой момент при нажатии на кнопку «Назад».

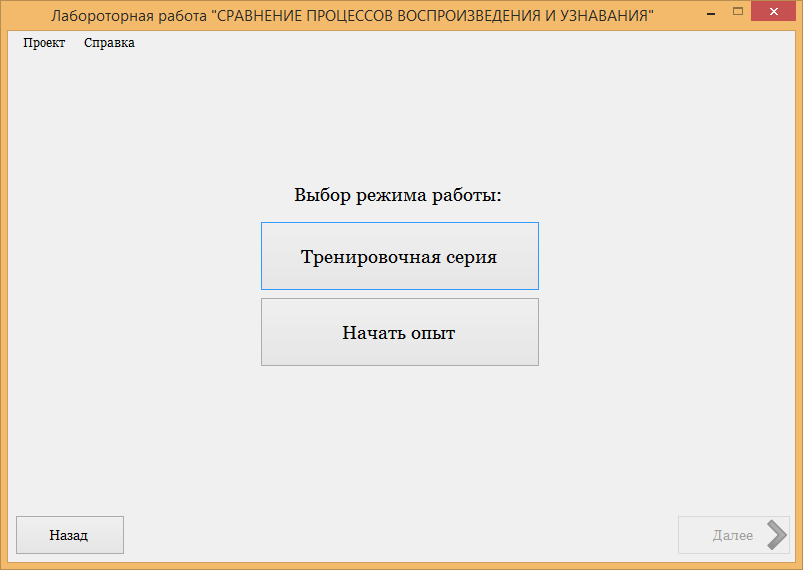
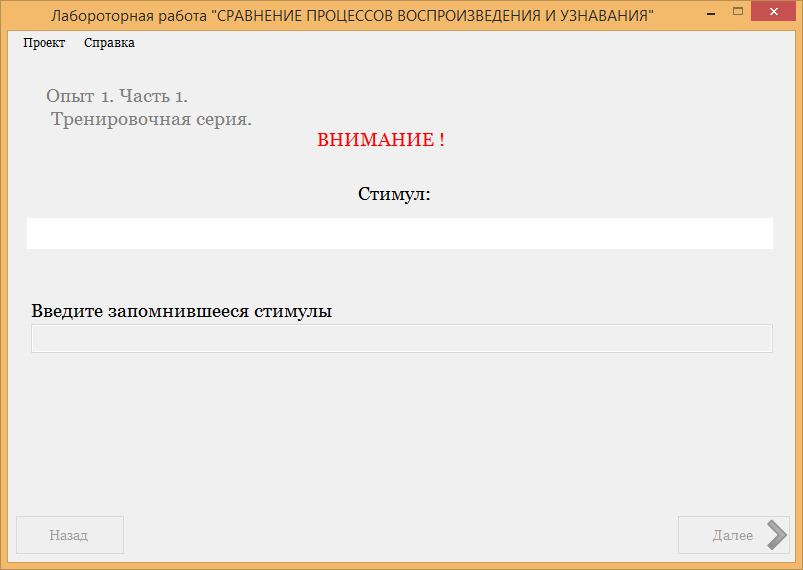
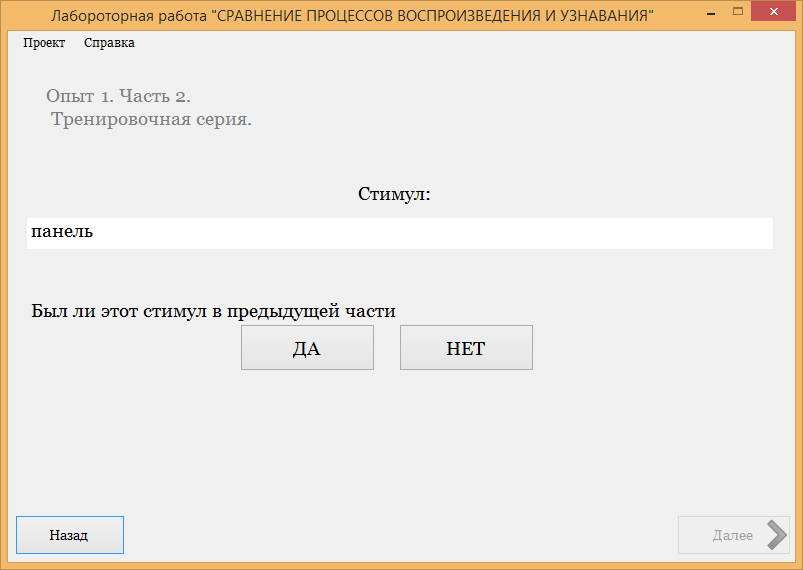


Рисунок 2.5 - Выбор режима работы испытуемого



а)



б)

Рисунок 2.6 – Тренировочная серия а) воспроизведение запомнившихся стимулов, б) узнавание стимулов.

Форма прохожнения основного эксперимента имеет тот же интерфейс, что и тренировочная серия.

После завершения прохождения основного эксперимента, испытуемому демонстрируется его результаты (см. рисунок 2.7). При нажатии кнопки «Назад» испытуемый попадает на главную страницу опытов (см. рисунок 2.4)

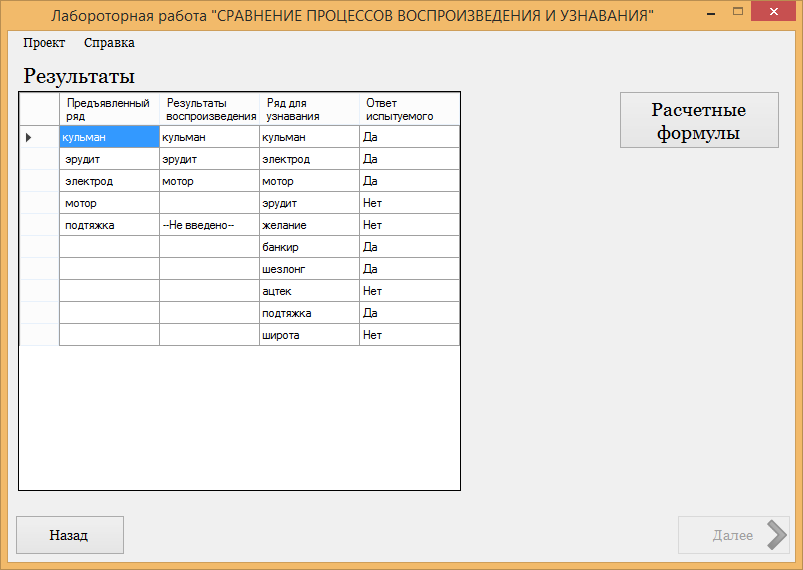


Рисунок 2.7 – Отображение результатов эксперимента испытуемому

Опыты 1 – 4 являются однотипными, и отличаются лишь предъявляемым стимульным материалов. В связи с этим они имеют одинаковый интерфейс пользователя.

При выборе испытуемым опыта 5 испытуемому в течение определенного времени, демонстрируется матрица графических изображений.

На рисунке 2.8 представлена форма для воспроизведения графических изображений. Используя мышку в качестве кисти испытуемому необходимо изобразить на матрице запомнившиеся стимулы. При переключении курсора в режим «Ластик», студент может стирать нарисованные изображения. При нажатии кнопки «Очистить» поле для зарисовки будет полностью очищаться.

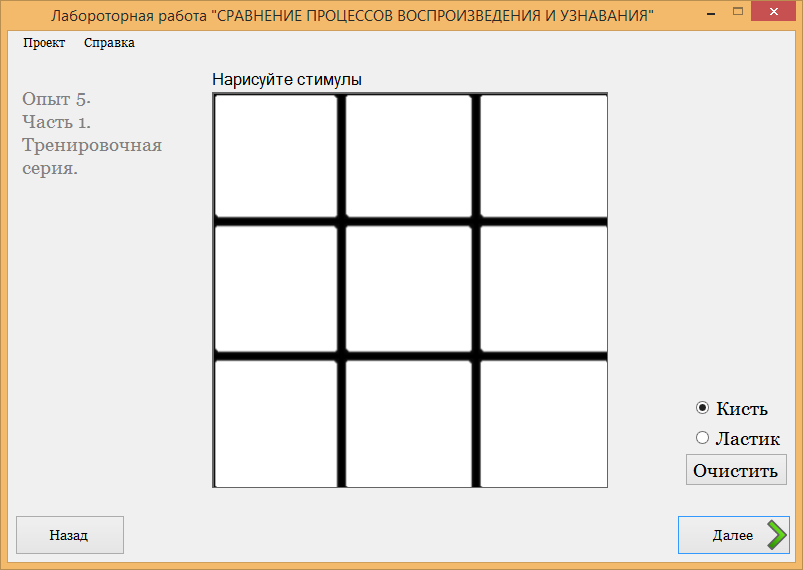


Рисунок 2.8 – Воспроизведение запомнившихся графических изображений

Во второй части опыта испытуемому, демонстрируется новая матрица графических изображений.

Далее студенту необходимо выбрать из представленных 20 графических изображений те, которые были продемонстрированы ему ранее (см. рисунок 2.9).

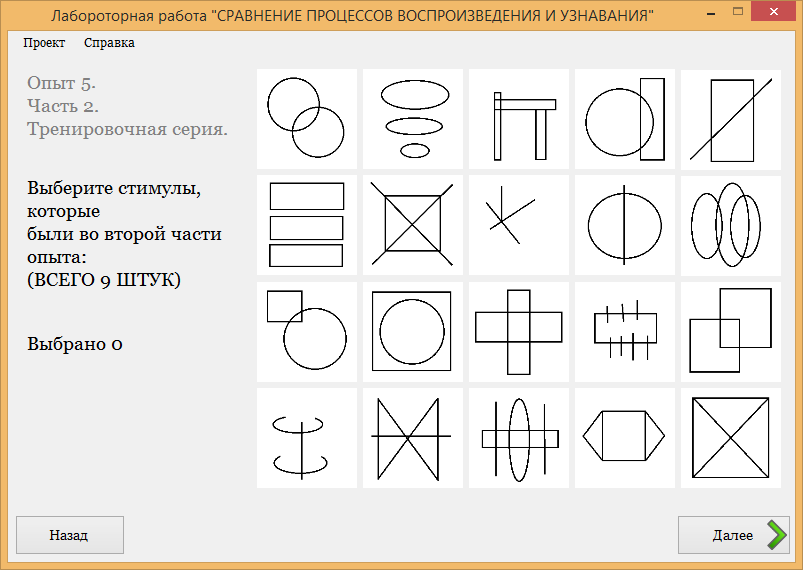


Рисунок 2.9 –Узнавание графических изображений

Основная часть эксперимента полностью соответстует опытам тренировочной серии. После завершения опыта испытуемому демонстрируется форма результатов эксперимента (рисунок 2.10).

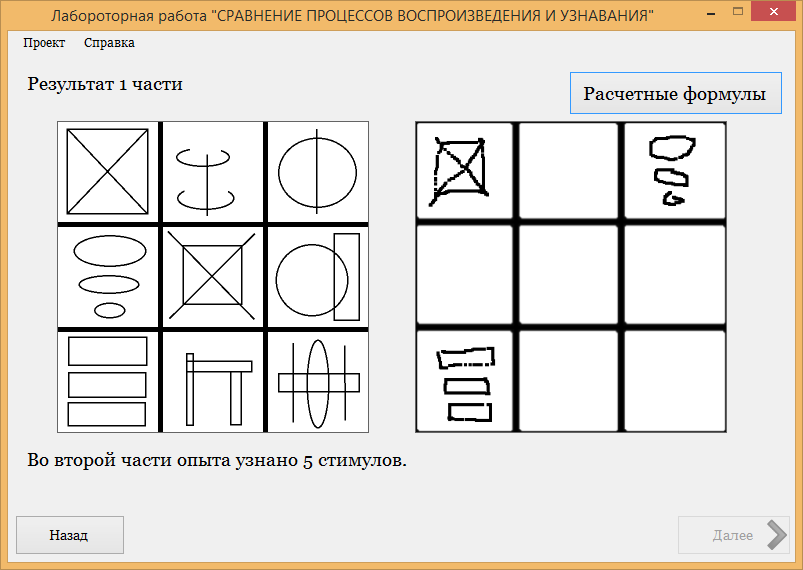


Рисунок 2.10 – Результаты опыта с графическими изображениями

При выборе режима «Преподаватель» пользователю необходимо пройти аутентификацию (см. рисунок 2.11). Для этого пользователю необходимо ввести пароль.

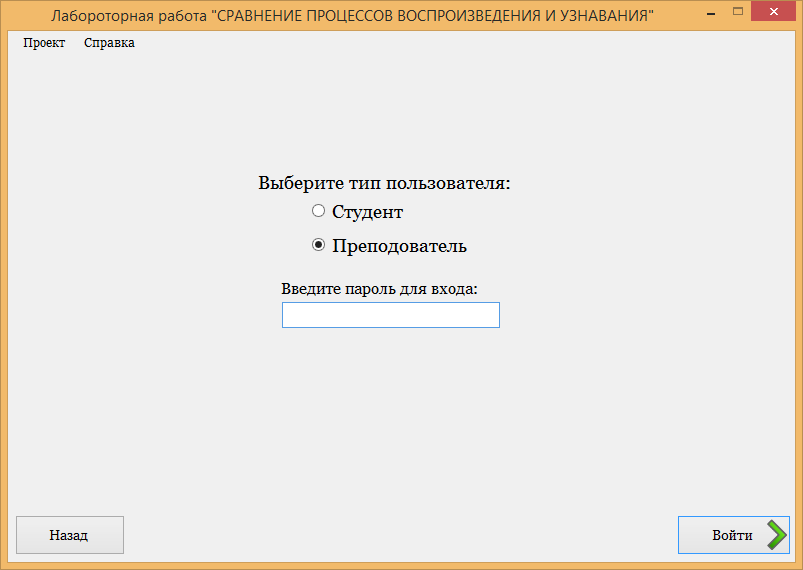


Рисунок 2.11 – Регистрация преподавателя.

Далее преподаватель может выбрать режим работы (см. рисунок 2.12): создание базы стимулов, задание настроек опытов и просмотр результатов. Преподаватель имеет возможность создавать и сохранять в памяти компьютера базовые массивы, из которых формируются наборы предъявляемых стимулов. Преподаватель так же может редактировать базу, сохраняемых результатов работ студентов (удалять файлы, потерявшие актуальность).

В случае выбора режима «Создание базы стимулов», появится форма, представленная на рисунке 2.13. Преподаватель имеет возможность создать новую базу стимулов для опытов с различным стимульным материалом.

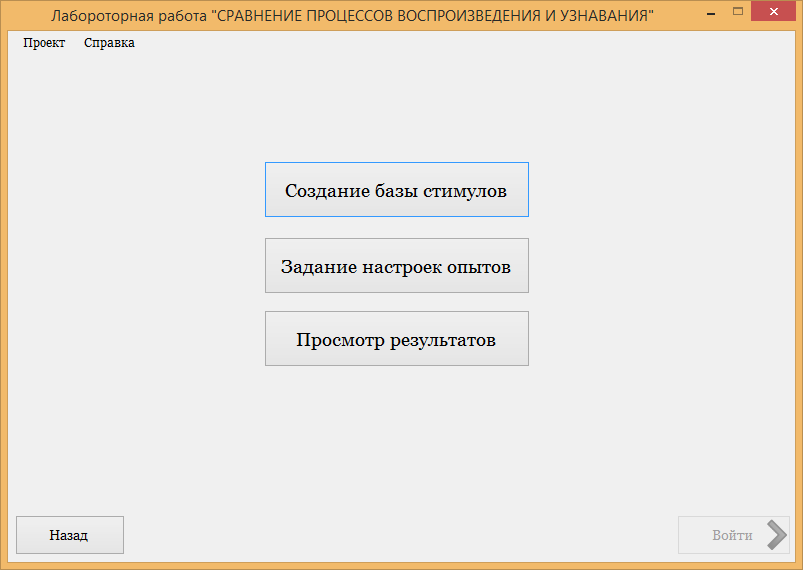


Рисунок 2.12 – Выбор режима работы преподавателя

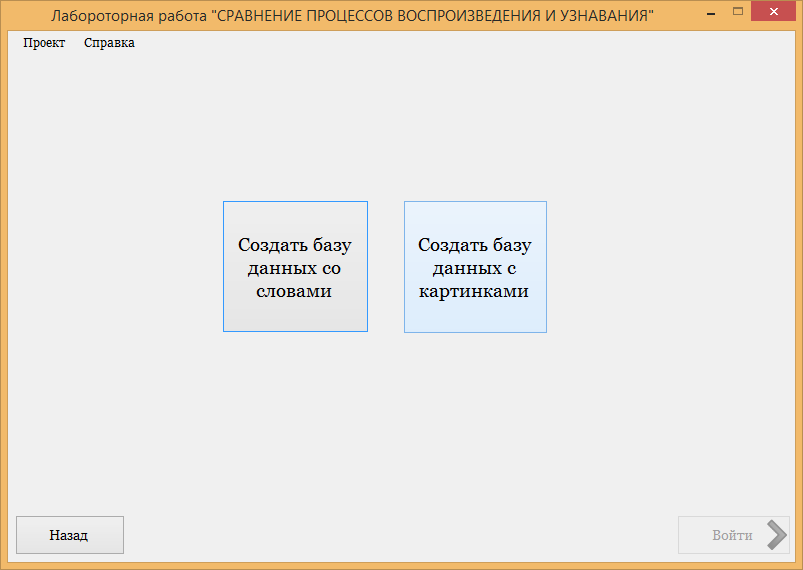


Рисунок 2.13 - Выбор режима работы с массивом стимулов

Если выбран режим «Создать базу данных со словами» то появится форма, представленная на рисунке 2.14. На данной форме преподаватель вводит название новой базы стимулов. Сами стимулы заносятся в поле ввода, разделяясь при этом пробелами. Для финального создания базы необходимо нажать кнопку «Сохранить».

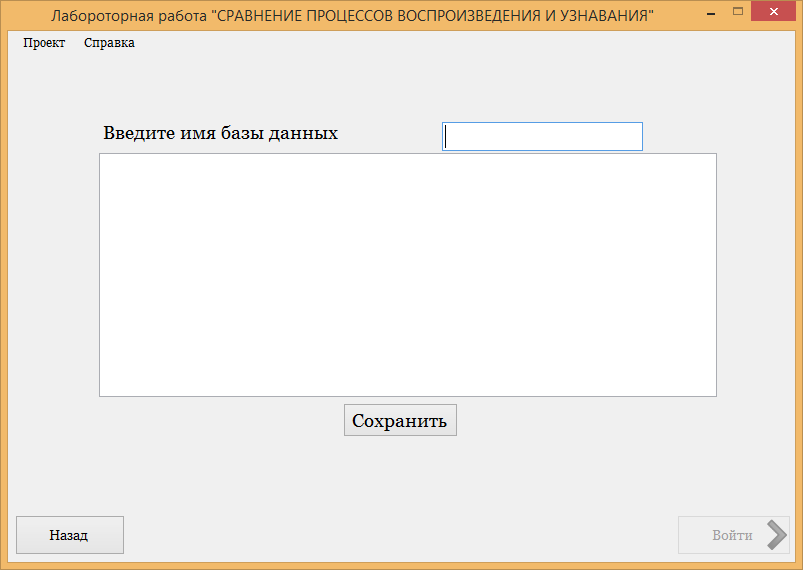


Рисунок 2.14 – Создание базы стимулов со словами

При выборе преподавателем на рисунке 21 режима «Создание базы данных с картинками», пользователь попадает на форму, представленную на рисунке 2.15. В ней ему необходимо заполнить поле с названием создаваемой базы, а также выбрать необходимые графические изображения. Исходя из условий опыта, количество картинок должно быть больше либо равно 20. Для финального создания базы необходимо нажать кнопку «Сохранить».

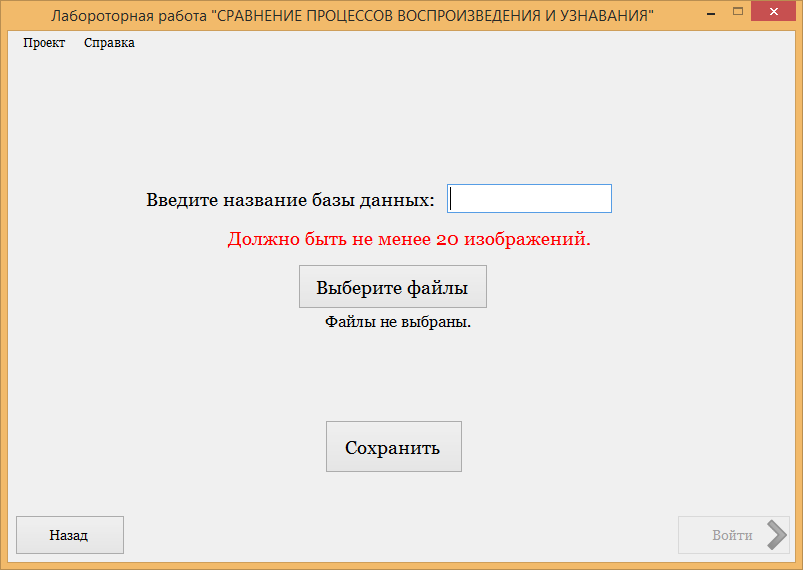


Рисунок 2.15 – Создание базы данных с графическими изображениями

Если преподаватель выбирает режим «Задание настроек опытов», то открывается форма, продемонстрированная на рисунке 2.16. В данной форме присутствует выпадающий список, который позволяет пользователю выбрать определенный опыт, в котором буду задаваться настройки.

При изменении настроек пользователь может работать с такими данными как: время экспозиции стимулов и количество предъявляемых в одном опыте стимулов. Так же существует возможность работать с базой данных, созданной администраторов, путем выбора ее из выпадающего списка, а также возможность ее удаления. При необходимости преподаватель также может внести изменения в описания опыта, путем нажатия кнопки «Редактировать описание опыта», и заполнения формы, представленной на рисунке 2.17.

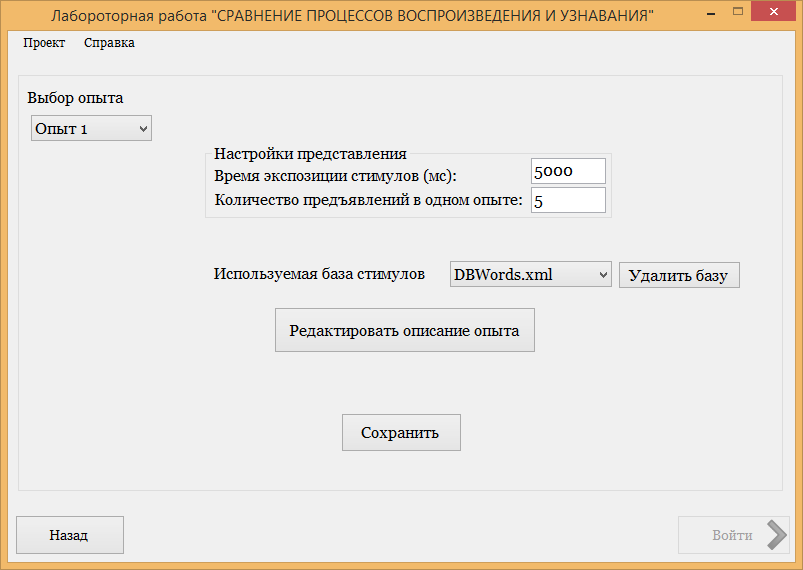


Рисунок 2.16 – Режим задания настроек опытов

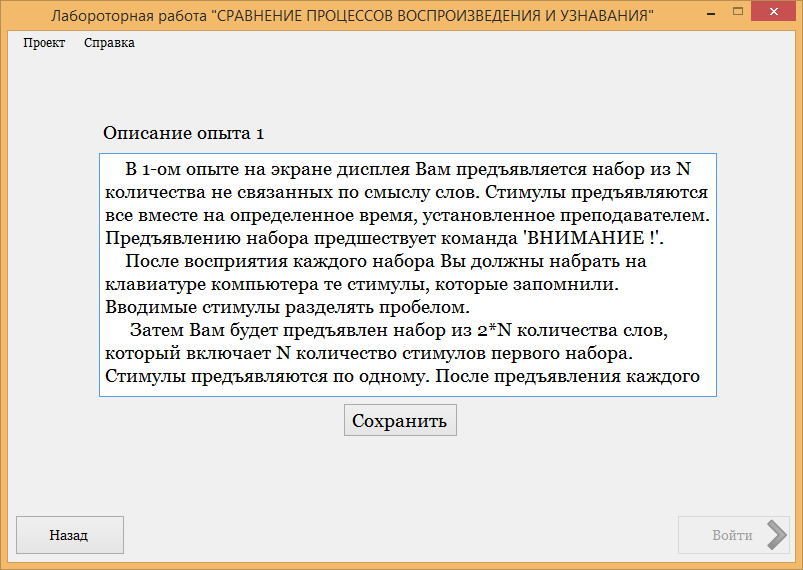


Рисунок 2.17 – Режим редактирования описания опыта

Если преподаватель выбрал режим «Просмотр результатов», то появится форма, представленная на рисунке 2.18. Здесь преподаватель может осуществить удаление сразу нескольких результатов путём множественного выделения и нажатия кнопки удаления. Также каждый результат можно посмотреть полностью, путем двойного нажатия на фамилию студента курсором мыши.

Из выпадающего списка преподаватель может выбрать номера опытов, результаты которых ему необходимо просмотреть.

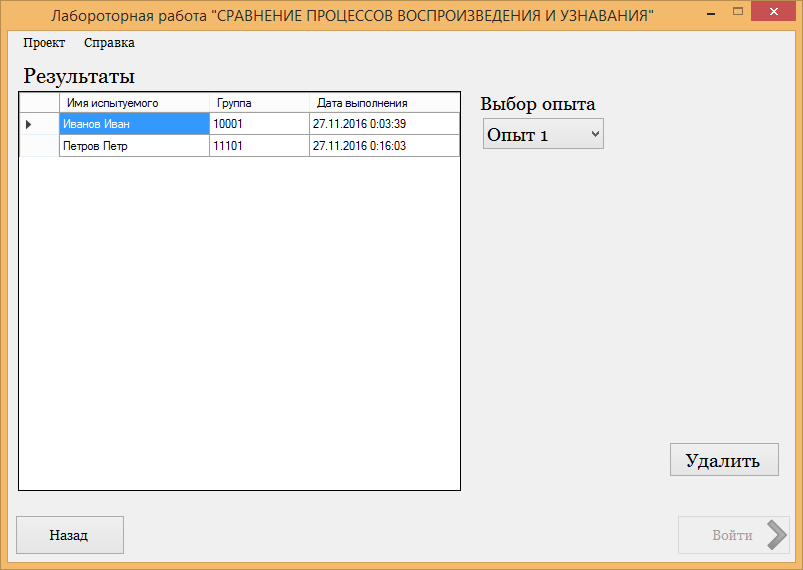


Рисунок 2.18 – Режим просмотра результатов преподавателем.

При нажатии кнопки удаления результата будет произведено уточнение о действительности намерения.

Аналогично, при нажатии кнопки закрытия в любой из моментов работы программы будет произведено уточнение о действительности намерений выхода из программы. При подтверждении выход будет осуществлен.

В результате эргономического проектирования было составлено целевое назначение СЧКС; проведен анализ содержания основных функций и определение структуры СЧКС; составлено распределение функций в системе между человеком и компьютером, а также разработка алгоритмов работы пользователей СЧКС; разработаны эргономические требования к системе «человек – компьютер – среда» и проведена оценка эргономичности СЧКС. Также была осуществлена разработка сценария информационного взаимодействия человека – пользователя и ПК.

# **3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРОГРАММНО – АППАРАТНОГО КОМПЛЕКСА ДЛЯ ИССЛЕДОВАНИЯ ПРОЦЕССОВ ПАМЯТИ**

## **3.1 Обоснование выбора языка реализации и среды разработки**

Программный комплекс разрабатывался в среде программирования Microsoft Visual Studio 2015 на языке программирования C# с использованием XML.

Средой написания приложение является Microsoft Visual Studio – линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств, позволяющих разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Microsoft Silverlight..

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и как отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения. [6]

Microsoft .Net Framework является так называемой программной платформой. Архитектура платформы состоит из двух частей. Основой является исполняющая среда Common Language Runtime (CLR), которая может выполнять как обычные программы, так и серверные приложения. Вторая, не менее важная часть, это библиотека классов Framework Class Library (FCL), содержащая в себе множество компонентов для работы с базами данных, сетью, вводом/выводом, файлами, пользовательским интерфейсом и т.д. Это позволяет разработчику не заниматься низкоуровневым программированием, а использовать уже готовые классы.

Важные части библиотеки классов:

* Windows Forms — отвечает за разработку графического интерфейса. Фактически является обёрткой над Win32 API.
* ADO.NET — предоставляет доступ данным. В основном используется для работы с базами данных.
* ASP.NET — технология разработки веб-сайтов, веб-приложений и веб-сервисов.
* Language Integrated Query (LINQ) — реализация языка запросов, напоминающего по синтаксису SQL в программах на .Net.
* Windows Presentation Foundation (WPF) — система создания графических интерфейсов, использующая язык разметки XAML. В отличие от Windows Forms использует графическую технологию DirectX, что обеспечивает более быструю работу за счет аппаратного ускорения графики.
* Windows Communication Foundation (WCF) — система обмена данными между приложениями .Net. Используется для создания распределённых приложений.

Одной из основных идей, заложенной в .Net, является совместимость различных частей приложения, которые могут быть разработаны на разных языках. Например программа, написанная на C# может обратиться к методу из библиотеки, написанной на Visual Basic .NET, или класс на Managed C++ может быть унаследован от класса на Delphi .Net. Языки, включённые в Visual Studio: C#, J#, Visual Basic .NET, JScript .NET, C++/CLI, F#.

Также существуют независимые проекты, позволяющие разрабатывать программы под .Net Framework на других языках [7].

На сегодняшний день XML является одним из наиболее распространенных стандартов документов, который позволяет в удобной форме сохранять сложные по структуре данные. Поэтому разработчики платформы .NET включили в фреймворк широкие возможности для работы с XML. XML — расширяемый язык разметки. Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка. Расширение XML — это конкретная грамматика, созданная на базе XML и представленная словарём тегов и их атрибутов, а также набором правил, определяющих какие атрибуты и элементы могут входить в состав других элементов. Сочетание простого формального синтаксиса, удобства для человека, расширяемости, а также базирование на кодировках Юникод для представления содержания документов привело к широкому использованию как собственно XML, так и множества производных специализированных языков на базе XML в самых разнообразных программных средствах.

Язык написания приложения – C#. C# – объекто-ориентированный язык программирования. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, исключения, комментарии в формате XML. Основными преимуществами данного языка являются:

– компонентно-ориентированное программирование;

– безопасный (по сравнению с языками C и C++) код;

– унифицированная система типизации;

– поддержка событийно-ориентированного программирования;

– объединение лучших современных языков программирования: Java,C++, Visual Basic и др.

## **3.2 Структура программного средства**

При разработке было принято решение разбить приложение на следующие модули:

1. Модуль проведения эксперимента – модуль отвечает за проведение всех опытов в программе и вывод результатов на экран.
2. Модуль работы с файлами – осуществляет чтение, редактирование, удаление и создание файлов с базой слов, результатов и настроек.
3. Модуль администрирования – доступ и изменение настроек опыта, а также создание новых баз стимулов; просмотр и удаление результатов проведенных опытов.
4. Модуль регистрации – проводит идентификацию пользователя (студент/преподаватель), а также осуществляет проверку введенного пользователем пароля при входе в систему в качестве преподавателя.

Описание модулей продемонстрировано на рисунке 3.1.



Рисунок 3.1 – Описание модулей

Описанные ранее модули были распределены между соответствующими классами. Класс — это конструктор, который позволяет создавать свои собственные пользовательские типы путем группирования переменных других типов, методов и событий. Он определяет данные и поведение типа. Классы в С# могут содержать следующие члены:

- поля;

- константы;

- свойства;

- конструкторы;

- методы;

- события;

- операторы;

- индексаторы;

- вложенные типы.

На рисунке 3.2 представлена диаграмма классов разрабатываемого ПАК.

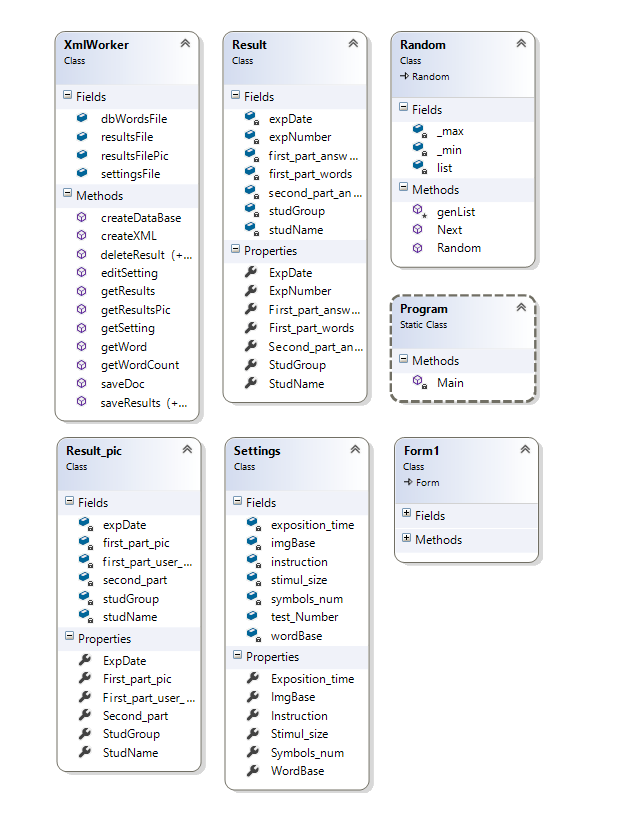


Рисунок 3.2 – Диаграмма классов

Описание основных классов:

1. XmlWorker.cs – класс, предназначенный для работы с XML файлами. Основными методами, прописанными в данном классе, являются методы, связанные с созданием, сохранением, изменением и удалением данных. В ПАК они реализуются как в профиле студента: создание новой учетной записи пользователя, демонстрация условий эксперимента, подгрузка знакового стимульного материала, подгрузка графического стимульного материала; так и в профиле администратора: создание новой базы знаковых стимулов, создание новой базы графических стимулов, удаление базы стимулов, задание настроек эксперимента, изменение настроек эксперимента, загрузка стимульной базы, формирование теоретических сведений, формирование описания к опыту, подгрузка результатов пользователей о пройденных экспериментах, удаление результатов пользователей, формирование результатов пользователей в формат .doc и открытие их в Microsoft Word.
2. Result.cs - класс, объявляющий основные переменные, необходимые для формирования отчетов об экспериментах, в которых в качестве стимульного материала выступают знаковые стимулы. Данными переменными являются: ФИО студента, номер группы, номер опыта, дата проведения, результаты первой части эксперимента, а также результаты второй части эксперимента.
3. Random.cs – так как основным условием к опытам была организация подачи стимульного материала в случайном порядке, была реализована функция рандома. Класс Random содержит основной метод, позволяющий обеспечить демонстрацию стимулов в качесстве случайной выборки.
4. Settings.cs - класс, объявляющий основные переменные, задаваемые экспериментатором: задание настроек к опыту, продолжительность экспозиции, длина стимульного ряда, подаваемые знаковые (буквы, цифры, слова, слоги) либо графические стимулы.
5. Program.cs – класс, содержащий основной метод Main(), осуществляющий компиляцию всех форм и запуск приложения.
6. Result\_pic.cs - класс, объявляющий основные переменные, необходимые для формирования отчетов об экспериментах, в которых в качестве стимульного материала выступают графические стимулы. Данными переменными являются: ФИО студента, номер группы, дата проведения, номер опыта, результаты первой части эксперимента, а также результаты второй части эксперимента.
7. Form1.cs –данный класс имеет подкласс Form1.Designer.cs содержащий в себе перечень элементов формы. Form1.cs предназначен для формирования окон формы приложения, а также описания методов, конструирующих основную логику работы приложения.

Основной составляющей модулей являются методы. Метод — это блок кода, содержащий ряд инструкций. Программа инициирует выполнение инструкций, вызывая метод и указывая все аргументы, необходимые для этого метода. Далее будут описаны основные методы, содержащиеся в программных модулях (таблицы 3.1 – 3.4):

Таблица 3.1 – Описание методов модуля проведение эксперимента.

|  |  |
| --- | --- |
| Название метода | Описание метода |
| 1 | 2 |
| public void beginTestWords() | метод анализирует введенную информацию пользователем, и в зависимости от этого начинает выбранный тест. При выборе, первого опыта начинается эксперимент со словами, второго – опыт со слогами, третьего – с числовыми значениями, при выборе четвертого опыта, в качестве стимульного материала выступают буквы. Также данный метод предъявляет стимульный ряд испытуемому, в соответвии с выбранным экспериментом, и проводит первую часть опыта - осуществляет запись введенных испытуемых данных; |
| public void endTestWords() | метод, осуществляющий основную часть теста, а именно проводит вторую часть эксперимента по узнаванию стимулов из первой части опыта. По окончанию сохраняет результаты; |
| public void beginTestpic() | осуществляется начало эксперимента с графическими изображениями. Метод отвечает за тренировочную серию эксперимента, а также за проведение первой части опыта: происходит предъявление стимула и его воспроизведение. Формируются набор стимулов из случайных изображений, расположенных в директории. Затем следует отрисовка запомнившихся картинок в специальной форме; |
| public void endTestPic() | метод, описывающий заключительную часть эксперимента с изображениями. В нем пользователь пытается узнать стимулы, предъявленные в предыдущей части опыта; |
| public void res\_pic() | подведение итогов эксперимента с изображениями и получение результата. |

Таблица 3.2 – Описание методов модуля работы с файлами

|  |  |
| --- | --- |
| Название метода | Описание метода |
| 1 | 2 |
| public static Settings getSetting(string settingId) | метод считывает из XML файла данные и возвращает объект, содержащий настройки проведения определенного опыта; |
| public static void editSetting(Settings setting) | осуществляется изменение существующих настроек опыта; |
| public static int getWordCount() | считает количество стимулов в базе слов и возвращает это значение; |
| public static DBWords getWord(int wordId) | метод считывает из XML файла с базы слов по его Id и возвращает слово; |
| public static void saveResults(Result results) | создание структуры файла из переданного объекта результата и сохранение его в XML файл; |
| public static void saveResults(Result\_pic results) | перегрузка метода сохранения результатов в файл для эксперимента с изображениями; |
| public static List<Result> getResults() | считывание сохраненных результатов из XML файла и запись их в List; |
| public static List<Result\_pic> getResultsPic() | считывание сохраненных результатов для эксперимента с графическими стимулами из XML файла и запись их в List; |
| public static void deleteResult(Result result) | удаление результатов; |
| public static void deleteResult(Result\_pic result) | удаление результатов для эксперимента с графическими стимулами; |
| public static void saveDoc (Result res) | сохранение отчета о проведенных опытах в Word; |
| public static void createXML(string xmlName, string rootElement) | создание XML файла для сохранения результатов; |
| public static void createDataBase(List<string> DBword, string name) | создание новой базы стимулов при создании ее администратором. |

Таблица 3.3 – Описание методов модуля администрирования

|  |  |
| --- | --- |
| Название метода | Описание метода |
| 1 | 2 |
| private void button\_save\_admin(object sender, EventArgs e) | сохраняет все настройки, введенные администратором; |
| private void button\_deleteBase\_Click(object sender, EventArgs e) | удаление выбранной базы слов; |
| private void button25\_Click(object sender, EventArgs e) | открывает форму создания новой базы слов; |
| private void button\_save\_Click(object sender, EventArgs e) | в зависимости от выбора администратора сохраняет новую базу слов либо описание к опыту; |
| private void button\_showResults\_Click(object sender, EventArgs e) | открывает форму просмотра результатов выполненных испытуемыми опытов; |
| private void radioButton2\_CheckedChanged(object sender, EventArgs e) | идентификация пользователя: метод вызывается при входе в систему с учетной записи администратора и выполняет проверку пароля. |

Таблица 3.4 – Описание методов регистрации

|  |  |
| --- | --- |
| Название метода | Описание метода |
| 1 | 2 |
| private void buttonNext\_Click(object sender, EventArgs e) | метод осуществляющий контроль доступа к учетной записи администратора путем сверки введенного пользователем пароля, с паролем, прописанным администратором. |

Для осуществления перехода межу окнами формы были разработаны кнопки «Далее» и «Назад». Методами, описывающими данные кнопки приведены в таблице 3.5

Таблица 3.5 – Описание методов регистрации

|  |  |
| --- | --- |
| Название метода | Описание метода |
| 1 | 2 |
| private void buttonNext\_Click(object sender, EventArgs e) | метод для осуществления контроля доступа к элементам вкладок программы: переход на следующую вкладку; |
| private void buttonBack\_click(object sender, EventArgs e) | метод для осуществления контроля доступа к элементам вкладок программы: переход на предыдущую вкладку. |

Метод static void Main() – главная точка входа в программу и запуск основной формы приложения.

## **3.3 Тестирование программно-аппаратного комплекса**

Тестирование программного обеспечения (Software Testing) – проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом. В более широком смысле, **тестирование** – это одна из техник контроля качества, включающая в себя активности по планированию работ (Test Management), проектированию тестов (Test Design), выполнению тестирования (Test Execution) и анализу полученных результатов (Test Analysis). Основными целями проведения тестирования ПО являются демонстрация разработчикам и заказчикам, что программа соответствует требованиям и выявление ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.

Все виды тестирования программного обеспечения, в зависимости от преследуемых целей, можно условно разделить на следующие группы:

1. Функциональные виды тестирования, которые базируются на функциях и особенностях, а также взаимодействии с другими системами, и могут быть представлены на всех уровнях тестирования. Функциональные виды тестирования рассматривают внешнее поведение системы. Самые распространенные виды функциональных тестов:

* функциональное тестирование (Functional testing);
* тестирование безопасности (Security and Access Control Testing);
* тестирование взаимодействия (Interoperability Testing).

1. Нефункциональные виды тестирования, описывающие тесты, необходимые для определения характеристик программного обеспечения, которые могут быть измерены различными величинами. В целом, это тестирование того, "Как" система работает. Основные виды нефункциональных тестов:

* все виды тестирования производительности;
* тестирование установки (Installation testing);
* тестирование удобства пользования (Usability Testing);
* тестирование на отказ и восстановление (Failover Testing);
* конфигурационное тестирование (Configuration Testing).

1. Связанные с изменениями виды тестирования, работающие после проведения необходимых изменений, таких как исправление бага, когда программное обеспечение должно быть перетестировано для подтверждения, что проблема была действительно решена. Ниже перечислены виды тестирования, которые необходимо проводить после установки программного обеспечения, для подтверждения работоспособности приложения или правильности осуществленного исправления дефекта:

* дымовое тестирование (Smoke Testing);
* регрессионное тестирование (Regression Testing);
* тестирование сборки (Build Verification Test);
* санитарное тестирование или проверка согласованности/исправности (Sanity Testing).

По степени изолированности компонентов можно выделить следующие виды тестирования:

1. Модульное тестирование – процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы. Идея состоит в том, чтобы писать тесты для каждой функции или метода. Это позволяет проверить, не привело ли очередное изменение кода к регрессии, а также облегчает обнаружение и устранение таких ошибок.
2. Интеграционное тестирование – одна из фаз тестирования программного обеспечения, при которой отдельные программные модули объединяются и тестируются в группе. Целью интеграционного тестирования является проверка соответствия проектируемых единиц функциональным, приёмным и требованиям надежности.
3. Системное тестирование программного обеспечения – это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям. Основной задачей системного тестирования является проверка как функциональных, так и не функциональных требований к системе в целом[8].

Тестирование программного обеспечения в рамках дипломного проекта проводится в виде функционального ручного тестирования, так как оно имитирует фактическое использование системы и позволяет сразу же обнаружить дефекты в работе приложения. Также создание Test case обеспечивает возможность более детального рассмотрения работы основного функционала приложения. В таблице 3.6 приведены примеры составления Test case.

Таблица 3.6 – Пример Test case для разработанного ПАК

|  |  |  |  |
| --- | --- | --- | --- |
| № | Форма | Функциональность | Тестовое покрытие |
| 1 | 2 | 3 | 4 |
| 1 | Окно приветствия | Запуск приложения | да |
| 2 | Идентификация пользователя | Возможность выбора пользователя (студент/преподаватель) | да |
| 3 | При выборе «студента» осуществляется переход на форму ввода ФИО и номера группы | да |
| 4 | При выборе режима «преподаватель» -> проверка пароля (ввод корректного и некорректного значения) | да |
| 5 | Форма регистрации | Ввод данных в поле «Фамилия и Имя» (корректное и некорректное значение) | да |
| 6 | Ввод данных в поле «Номер группы» (корректное и некорректное значение) | да |
| 7 | Форма выбора опыта | Возможность студенту выбрать соответствующий опыт | да |
| 8 | Наличие инструкции к каждому опыту | да |
|  | Начало опыта при нажатии на кнопку «Далее» | да |
| 9 | Форма выбора режима работы | Наличие кнопок «Тренировочной серии» и «Начать опыт» | да |
| 10 | Отсутствие функции сохранения результатов при прохождении «Тренировочной серии» | да |
| 11 | Наличие функции сохранения результатов при прохождении основного эксперимента | да |
| 12 | Форма прохождения первой части опыта | Демонстрация стимульного ряда в соответствии с выбранным опытом | да |
| 13 | Возможность ввода стимульного материала | да |
| 14 | Осуществление перехода на вторю часть опыта | да |
| 15 | Форма прохождения второй части опыта | Демонстрация стимульного материала | да |
| 16 | Демонстрация стимулов, показывающихся в первой части опыта | да |
| 17 | Демонстрация стимулов, отображенных ранее со стимулами из БД в случайном порядке | да |
| 18 | Возможность идентификации стимула (Наличие кнопок «Да» и «Нет») | да |
| 19 | Возможность зарисовки запомнившихся стимулов (эксперимент с изображениями) | да |
| 20 | Форма администратора | Наличие возможности перехода в следующие режимы: - создание базы стимулов; - задание настроек опытов; - просмотр результатов | да |
| 21 | Форма создания базы стимулов | Создание базы данных со словами | да |
| 22 | Возможность ввода новых слов | да |
| 23 | Возможность сохранения новой базы | да |
| 24 | Создание базы данных с изображениями | да |
| 25 | Подгрузка изображений из файла | да |
| 26 | Задание названия базы | да |
| 27 | Форма задания настроек опыта | Возможность выбора номера опыта из выпадающего списка | да |
| 28 | Задание настройки предъявлений (корректные/некорректные значения) |  |
| 29 | Возможность подгрузки собственной базы стимулов (для опытов со словами и изображениями) | да |
| 30 | Возможность удаления базы стимулов | да |
| 31 | Редактирование описания к опыту | да |
| 32 | Сохранение изменений | да |
| 33 | Демонстрация результатов испытуемого в формате ФИО, группа, дата выполнения | да |
| 34 | При двойном нажатии определенного студента – подробное отображение результатов о прохождении опыта | да |
| 35 | Удаление неактуальных результатов | да |
| 36 |  | Перевод результатов в формат .doc | да |
| 37 | Форма приложения | Возможность завершения работы приложения в любой момент | да |

При анализе данных таблицы 3.6 можно сделать вывод, что разработанный программно-аппаратный комплекс прошел полное позитивное тестирование, не было выделено каких-либо критических ошибок, следовательно, приложение готово к эксплуатации пользователем.

4 Технико-экономическое обоснование ЭФФЕКТИВНОСТИ РАЗРАБОТКИ И ИСПОЛЬЗОВАНИЯ Программно-аппаратного комплекса для исследования процессов памяти

Специфической особенностью современного этапа развития общества является широкое внедрение информационных технологий во все сферы жизнедеятельности человека. Данная тенденция реализуется путем разработки программно-аппаратных комплексов различного назначения. В настоящее время повышение эффективности процесса обучения в образовательных учреждениях всех видов и профилей невозможно без создания и использования таких комплексов.

Целью данного проекта является разработка программно-аппаратного комплекса для исследования психологических закономерностей и механизмов памяти человека - сравнения процессов воспроизведения и узнавания. Основным его назначением является проведение экспериментального исследования.

Потенциальными пользователями программно-аппаратного комплекса являются студенты и преподаватели университета. Ввиду постоянно расширяющегося количества читаемых курсов и увеличения количества студентов на специальностях с предметом ПВиПИ, спрос на продукт является актуальным.

Разработка проектов программных средств связана со значительными затратами ресурсов (трудовых, материальных, финансовых). В связи с этим создание и реализация каждого проекта программного обеспечения нуждаются в соответствующем технико-экономическом обосновании (ТЭО).

Для оценки экономической эффективности инвестиционного проекта по разработке и внедрению программного продукта необходимо рассчитать:

1. Доход, получаемый от использования программного продукта;

2. Инвестиции, необходимые для разработки программного продукта;

3. Показатели эффективности инвестиционного проекта по производству ПП.

Экономическая целесообразность инвестиций в разработку и использование программного продукта осуществляется на основе расчета и оценки следующих показателей:

* чистая дисконтированная стоимость (ЧДД);
* срок окупаемости инвестиций (ТОК);
* рентабельность инвестиций (Ри)[9][10].

4.1 Стоимостная оценки затрат

Основная заработная плата исполнителей на наш программный продукт рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (4.1) |

где *n* − количество исполнителей, занятых разработкой наше программного продукта;

*Tчi* − часовая тарифная ставка i-го исполнителя (руб.);

*Фп* − плановый фонд рабочего времени i-го исполнителя (дн.);

*Tч* − количество часов работы в день (ч);

*К* − коэффициент премирования.

Коэффициент премирования 1,5. Для расчета заработной платы месячная тарифная ставка 1-го разряда на предприятии 80 рубль.

Таблица 1.1 - Расчет заработной платы

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Категория исполнителя | Разряд | | Тарифный коэффициент | Часовая тарифная ставка, руб. | Трудоемкость, дн. | Основная заработная плата, руб. |
| программист III категории | 12 | 2.84 | | 13 | 30 | 3120 |
| начальник, руководитель проекта | 16 | 3,72 | | 15 | 18 | 2160 |
| Итого с премией (50%), Зо | - | - | | - | - | 7920 |

Дополнительная заработная плата на наш программный продукт (*Зд*) включает выплаты, предусмотренные законодательством о труде и определяется по нормативу в процентах к основной заработной плате:

|  |  |
| --- | --- |
| , | (4.2) |

где *Зд* − дополнительная заработная плата исполнителей ( руб.);

*Нд* − норматив дополнительной заработной платы равный 10%.



Отчисления в фонд социальной защиты населения и обязательное страхование (*Зсз*) определяются по формуле:

|  |  |
| --- | --- |
| , | (4.3) |

где *Нсз* − норматив отчислений в фонд социальной защиты населения и на обязательное страхование (34 + 0,6%).



Расходы по статье «Машинное время» (*Рм*) определяются по формуле:

|  |  |
| --- | --- |
|  | (4.4) |

где *Цм* − цена одного машино-часа. Рыночная стоимость машино-часа компьютера со всеми необходимым оборудованием (1-1,4 руб. / ч);

Тпр – время работы над программным продуктом (240 ч).



Расходы по статье «Научные командировки» (*Рнк*) на програмнное средство определяются по формуле:

|  |  |
| --- | --- |
| , | (4.5) |

где *Нрнк* – норматив расходов на командировки в целом по организации (%). Норматив на командировки − 10 % от основной заработной платы.

,

Расходы по статье «Прочие затраты» (*Пз*) на программное средство расчитываются по формуле:

|  |  |
| --- | --- |
| , | (4.6) |

где *Hп­з* − норматив прочих затрат в целом по организации равен 20%



Затраты по статье «Накладные расходы» (*Рн*):

|  |  |
| --- | --- |
| , | (4.7) |

где *Pн* – накладные расходы на программный продукт (руб.);

*Нрн* – норматив накладных расходов в целом по организации,100%.



Общая сумма расходов по смете (*Ср*) на программный продукт рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (4.8) |



Кроме того, организация-разработчик осуществляет затраты на сопровождение и адаптацию программного продукта (Рса), которые определяются по формуле:

|  |  |
| --- | --- |
| , | (4.9) |

где Нрса – норматив расходов на сопровождение и адаптацию 10%.



Общая сумма расходов на разработку (с затратами на сопровождение и адаптацию) как полная себестоимость программно продукта (Сп) определяется по формуле:

|  |  |
| --- | --- |
| , | (4.10) |



Прибыль рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (4.11) |

где *По* − прибыль от реализации программного продукта заказчику (руб.);

*Урп* − уровень рентабельности программного продукта 25%;

*Сп* − себестоимость програмнного продукта ( руб.) .

,

Прогнозируемая цена нашего программного продукта без налогов (*Цп*):

|  |  |
| --- | --- |
| , | (4.12) |



## **4.2 Стоимостная оценка результата**

Результатом (Р) в сфере использования нашего программного продукта является прирост чистой прибыли и амортизационных отчислений.

Прирост чистой прибыли представляет составит:

|  |  |
| --- | --- |
| , | (4.13) |

где Nп – плановый объем работ по анализу и обработки результатов, сколько раз выполнялись в году ( 3 раз);

tc - трудоемкость выполнения работы до внедрения программного продукта 240 нормо.ч;

tn - трудеемкость выполнения работы после вднедрения програмнного продукта (30 нормо часов);

Tc - часовая тарифная ставка, соответсвующая разряду выполеняемых работ до внедрения программного продукта (13 руб/ч.);

Tn - часовая тарифная ставка, соответсвующая разряду выполеняемых работ после внедрения программного продукта (13 руб. /ч);

Кпр - коэффициент премий 1.5;

Нд - номратив дополнительной заработной платы 20%;

Нпо - ставка отчислений в ФСЗН и обязательное страхование 34+0,6%.



Прирост чистой прибыли рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (4.14) |

где n – виды затрат, по которым получена экономия;

Э – сумма экономии, полученная за счет снижения i-ых затрат, руб.

Нп - ставка налога на прибыль, 18%.



Расчет амортизационных отчислений осуществляется по формуле:

|  |  |
| --- | --- |
| , | (4.15) |

где На - норма амортизации программного продукта 20%;

Иоб - стоимость программного продукта, руб.



4.3 Показатели эффективности использования программного продукта

При оценке эффективности инвестиционных проектов необходимо осуществить приведение затрат и результатов, полученных в разные периоды времени, к расчетному году, путем умножения затрат и результатов на коэффициент дисконтирования , который определяется следующим образом:

|  |  |
| --- | --- |
| , | (4.16) |

где  - требуемая норма дисконта, 38%

 - порядковый номер года, затраты и результаты которого приводятся к расчетному году;

 - расчетный год, в качестве расчетного года принимается год вложения инвестиций, равный 1.









Расчет чистого дисконтированного дохода за четыре года реализации проекта и срока окупаемости инвестиций представлены в таблице 1.1.

# Таблица 1.2 Экономические результаты работы предприятия

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование показателей | Един. измер | Усл.  обоз. | По годам использования программного продукта | | | |
| 1-й | 2-й | 3-й | 4-й |
| **Результат** |  |  |  |  |  |  |
| 1. Прирост чистой прибыли | руб. | ∆ | 15019,1 | 30038,2 | 30038,2 | 30038,2 |
| 2. Прирост амортизационных отчислений | руб. | ∆А | 5701,3 | 5701,3 | 5701,3 | 5701,3 |
| 3. Прирост результата | руб. | ∆ | 20720,4 | 35739,4 | 35739,4 | 35739,4 |
| 4. Коэффициент дисконтирования |  |  | 1 | 0,72 | 0,52 | 0,38 |
| 5. Результат с учетом фактора времени | руб. |  | 20720,4 | 25732,36 | 18584,4 | 13580,9 |
| **Затраты (инвестиции)** |  |  |  |  |  |  |
| 6. Инвестиции | руб | Иоб | 28506,85 | - | - | - |
| 7. Инвестиции с учетом фактора времени | руб. |  | 28506,85 | - | - | - |
| 8. Чистый дисконтированный доход по годам | руб |  | -7786,45 | 25732,36 | 18584,4 | 13580,9 |
| 9. ЧДД нарастающим итогом | руб |  | -7786,45 | 17945,91 | 36530,31 | 55114,71 |

Рассчитаем рентабельность инвестиций (РИ) по формуле

|  |  |
| --- | --- |
| , | (4.17) |

где З - затраты на приобретения нашего программного продукта;  - среднегодовая величина чистой прибыли за расчетный период, руб., которая определяется по формуле:

|  |  |
| --- | --- |
| , | (4.18) |

где  - чистая прибыль, полученная в году t, руб.



В результате технико-экономического обоснования инвестиций по производству нового изделия были получены следующие значения показателей их эффективности:

1. Чистый дисконтированный доход за четыре года производства продукции составит 55114,71 руб.

2. Все инвестиции окупаются на 2 год;

3. Рентабельность инвестиций составляет 48%

Таким образом, внедрение программно-аппаратного комплекса для исследования процессов памяти является эффективным и инвестиции в его разработку целесообразны.

# **5 ПРОВЕДЕНИЕ ЭРГОНОМИЧЕСКОЙ ОЦЕНКИ РАБОЧЕГО МЕСТА РАЗРАБОТЧИКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА**

Целью дипломного проекта являлась разработка программно-аппаратного комплекса сравнения процессов воспроизведения и узнавания. Проведение экспериментального исследования процессов воспроизведения и узнавания было осуществлено методом удержанных членов ряда и методом тождественных рядов. С целью реализации возможности использования данного ПАК в лабораторных условиях было разработано приложения, написанное на языке С# с использованием XML.

В настоящем разделе рассмотрим вопросы, связанные с обеспечением проведения эргономической оценки рабочего места разработчика ПАК.

Эргономическое обеспечение при проектировании направлено на максимально возможное согласование технической части системы с возможностями и особенностями человека. Оно основывается на эргономических требованиях, которые определяются свойствами человека-оператора и устанавливаются с целью оптимизации его деятельности. При этом под свойствами человека-оператора понимают его физиологические, психофизиологические, социально психологические, гигиенические и психологические характеристики и возможности.

Эргономическая оценка - определение соответствия показателей объекта оценки эргономическим требованиям и установление эргономического уровня качества оцениваемого объекта, т. е. степени реализации эргономических требований. Цель этой оценки - повышение эффективности функционирования СЧМ и удобства работы в ней оператора путем контроля за выполнением в проектных, конструкторских и эксплуатационных документах основанных на учете характеристик технической части системы, человека-оператора и их взаимодействия. Указанная оценка проводится при обосновании выполнения каждого этапа опытно-конструкторской разработки: технического предложения, эскизного проекта, рабочего проекта[11].

Далее будет произведен расчет эргономической оценки рабочего места разработчика программно-аппаратного комплекса, при использовании экспертного метода. Его сущность заключается в проведении экспертами интуитивно-логического анализа с количественной оценкой суждений и обработкой результатов. При этом достоверность экспертизы зависит от количества экспертов и их квалификации.

Рабочее место - это зона пространства, оснащена необходимым оборудованием, где происходит трудовая деятельность одного работника или группы работников. Рациональная планировка рабочего места должна обеспечивать: наилучшее размещение орудий и предметов труда, не допускать общего дискомфорта, уменьшать утомляемость работника, повышать его продуктивность труда. Площадь рабочего места должна быть такой, чтобы работник не делал лишних движений и не чувствовал неудобства во время работы. Важно иметь также возможность изменить рабочую позу. Проведенные исследования показывают, что при рациональной организации рабочих мест производительность труда растет на 15-25%.

Основные эргономические требования к проектированию рабочего места изображены на рисунках 5.1 и 5.2.

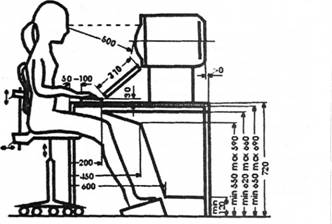


Рисунок 5.1 - Рабочий стол и размещения пользователя ПК



Рисунок 5.2 – Зоны досягаемости рук в горизонтальной плоскости

Далее приведены физиологические требования (Ф) к рабочему месту:

1. Обеспечение конструкции рабочего стула показателям:

* Ф1 - ширина и глубина поверхности сиденья не менее 400 мм;
* Ф2 - наличие поверхности сиденья с закругленным передним краем;
* Ф3 - регулировка высоты поверхности сиденья в пределах 400-550 мм и углам наклона вперед до 15 градусов и назад до 5 градусов;
* Ф4 - высоту опорной поверхности спинки 300±20 мм, ширину не менее 380 мм и радиус кривизны горизонтальной плоскости – 400 мм;
* Ф5 - угол наклона спинки в вертикальной плоскости в пределах 0±30 градусов;
* Ф6 - регулировка расстояния спинки от переднего края сиденья в пределах 260-400 мм;
* Ф7 - стационарные или съемные подлокотники длиной не менее 250 мм и шириной – 50-70 мм;
* Ф8 - регулировка подлокотников по высоте над сиденьем в пределах 230±30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм;
* Ф9 - поддержание рациональной рабочей позы при работе, позволяющее изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления.

1. Обеспечение конструкции рабочего стола показателям:

* Ф10 - рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм;
* Ф11 - высота рабочей поверхности стола равна 725 мм;
* Ф12 - модульные размеры рабочей поверхности: ширина 800, 1000, 1200 и 1400 мм, глубина 800 и 1000 мм.

1. Расположение клавиатуры и мыши, обеспечивающие поддержание рациональной рабочей позы:

* Ф13 - клавиатуру следует располагать на поверхности стола на расстоянии от 100 до 300 мм от переднего края;
* Ф14 - клавиатура на рабочем месте оператора должна располагаться так, чтобы обеспечивалась оптимальная видимость экрана.

1. Соответствие требованиям к дисплею:

* Ф15 - лицевые поверхности экрана расположены с отклонением не более 45 градусов от плоскости [12];
* Ф16 - экран видеомонитора должен находиться на расстоянии 600-700 мм от глаз пользователя, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов;
* Ф17 - уровень глаз должен приходиться на центр или 2/3 высоты экрана.

Далее проводим оценку значений единичных эргономических показателей.

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

ЭПгр = ∑ 1 / ∑1 + ∑ 0, ( 5.1)

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

∑ 0- суммарное число случаев, когда соответствия нет.

Исходя из физиологических показателей расставляем значения единичных эргономических показателей:

Ф-1, Ф-3, Ф-4, Ф-5, Ф-6, Ф-7, Ф-8, Ф-9, Ф-10, Ф-11, Ф-12, Ф-14, Ф-16, Ф-17 =1;

Ф-13, Ф-15 = 0.  
 Далее по формуле 5.1 находим значение группового показателя:

ЭПгр = 15 \* 2/17 = 1,76

Производим оценку эргономического свойства рабочего места:

ЭСВ = αнi \* ЭПгрj, ( 5.2 )

где αнi – нормированные весовой коэффициент αнi = 0,5

ЭСВ управляемость = 0,5 \* 1,76 = 0,865.

Данное значение показателя оценивается как "отлично"-эргономические характеристики изделия соответствуют базовым значениям.

После общей оценки производится анализ единичных показателей, значения которых не соответствуют эргономическим требованиям и намечаются мероприятия по рационализации оцениваемой системы. Далее приведены невыполненные физиологические требования, а также предложения по их улучшению:

* Ф13 - клавиатуру следует располагать на поверхности стола на расстоянии от 100 до 300 мм от переднего края, обращенного к оператору, или на специальной регулируемой по высоте рабочей поверхности, отделенной от основной столешницы. Предложение по улучшению: обеспечение возможности испытуемому располагать ОУ самостоятельно;
* Ф15 - лицевые поверхности экрана расположены с отклонением не более 45 градусов от плоскости. Предложение по улучшению: изменение угла наклона в соответствие с индивидуальными особенностями испытуемого.

Таким образом, изложенные выше предложения обеспечат проведение эргономической оценки рабочего места испытуемого.

# **ЗАКЛЮЧЕНИЕ**

В данном дипломном проекте был разработан программно-аппаратный комплекс для сравнения процессов воспроизведения и узнавания. С целью формирования основных требований к проектируемому комплексу был проведен анализ научно – технической литературы, обзор аналогов, выявление их основных положительных качеств, а также поставлены задачи к разрабатываемой системе.

При эргономическом проектировании системы были сформулированы основные функции системы, а также их распределение в системе; спроектированы алгоритмы работы пользователя в разрабатываемой системе, а также разработаны основные эргономические требования. Для наиболее детальной демонстрации интерфейса приложения были разработаны сценарии информационного взаимодействия.

В разделе разработки программного обеспечения программно-аппаратного комплекса была продемонстрирована структура разрабатываемой системы, а также проведено ее тестирование с целью выявления несоответствия требованиям.

В разделе технико-экономического обоснования проекта было доказано, что внедрение ПАК является эффективным и инвестиции в его разработку целесообразны.

Для обеспечения эргономичности рабочего места разрабатываемого ПАК были вывалены основные физиологические требования, а также проведена эргономическая оценка, сформулированы предложения по улучшению.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Процессы памяти. [Электронный ресурс]. – Режим доступа http://www.psychologos.ru/articles/view/processy\_pamyati

[2] Шупейко, И. Г. Психология восприятия и переработки информации: лабораторный практикум для студентов специальности I – 50 01 01 «Инженерно – психологическое обеспечение информационных технологий» дневной формы обучения / И. Г. Шупейко. – Минск : БГУИР, 2008. – 44 с.

[3] Компьютеризированные и компьютерные психодиагностические тесты [Электронный ресурс]. – Режим доступа http://studbooks.net/32930/psihologiya/kompyuterizirovannye\_kompyuternye\_psihodiagnosticheskie\_testy

[4] Современные компьютерные системы психологической диагностики [Электронный ресурс]. – Режим доступа https://www.psycho.ru/library/93

[5] Психологические тесты онлайн. Память на числа [Электронный ресурс]. – Режим доступа <http://www.psyq.ru/test/test07.html>

[6] Microsoft Visual Studio [Электронный ресурс]. – Режим доступа <https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>

[7] Microsoft .Net Framework [Электронный ресурс]. – Режим доступа <http://net-framework.ru/article/chto-takoe>

[8] Документация «Виды тестирования ПО» [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.protesting.ru/testing/testtypes.html>

[9] Палицын, В.А. Технико-экономическое обоснование дипломных проектов: Метод. пособие для студ. всех спец. БГУИР. В 4-х ч. Ч.4: Проекты программного обеспечения / В.А. Палицын. – Мн.: БГУИР, 2006. – 76 с.

[10] Горовой В.Г., Грицай А.В., Пархименко В.А. Экономическое обоснование проекта по разработке программного обеспечения: Метод. пособие для студ. всех спец. БГУИР. – Мн.: БГУИР, 2014.

[11] Михнюк, Т.Ф. Эргономическая экспертиза инженерных решений: Метод. пособие для студ. всех спец. и форм обучения БГУИР / Т.Ф.Михнюк - Мн.: БГУИР, 2007. - 21 с.

[12] СанНиП «Требования при работе с видеодисплейными терминалами и электронно-вычислительными машинами», утв. Постановлением Минздрава РБ 28 июня 2013 № 59

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Листинг программы**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace noname

{

public partial class Form1 : Form

{

Bitmap bmp = new Bitmap(global::noname.Properties.Resources.true\_empty);

int chosedTest;

bool pressed = false;

bool paint = false;

SolidBrush color;

string first\_part\_answer;

Dictionary<string, string> second\_part\_answers = new Dictionary<string, string>();

List<String> randPull = new List<string>();

string words = "";

bool RET = false;

bool ADMIN = false;

bool train\_mode;

public Form1()

{

InitializeComponent();

pictureBox1.Image = bmp;

}

private void buttonNext\_Click(object sender, EventArgs e)

{

if (tabConrol1.SelectedTab == tabConrol1.TabPages["do\_with\_words"])

{

first\_part\_answer = textBox\_stimuls.Text;

Thread myThread = new Thread(new ThreadStart(endTestWords));

myThread.Start();

next\_button.Enabled = false;

return;

}

if (tabConrol1.SelectedTab == tabConrol1.TabPages["do\_with\_pic"])

{

saveImage();

pressed = true;

radioButton3.Visible = false;

radioButton4.Visible = false;

button\_clear.Visible = false;

foreach (PictureBox p in pic\_boxes)

p.Image = new Bitmap(Properties.Resources.true\_empty);

tabConrol1.SelectTab("select\_mode");

label9.Text = "Стимул:";

five\_test();

//button3\_Click(new object(), new EventArgs());

return;

}

if (tabConrol1.SelectedTab == tabConrol1.TabPages["end\_with\_pic"])

{

if (train\_mode)

{

buttonBack\_click(new object(), new EventArgs());

return;

}

tabConrol1.SelectTab("result\_pic");

button\_formulsPic.Visible = true;

res\_pic();

pictureToDoc();

return;

}

if (tabConrol1.SelectedIndex == 1)

next\_button.Text = "Войти";

else

next\_button.Text = "Далее";

if (radioButton2.Checked == true)

{

if (textBox3.Text == "admin")

{

tabConrol1.SelectTab("admin\_mode");

textBox3.Text = "";

ADMIN = true;

return;

}

else

MessageBox.Show("Не верный пароль!", "Внимание!");

return;

}

else

{

if (textBox1.Text == "" && tabConrol1.SelectedTab == tabConrol1.TabPages["registration"])

{

toolTip1.Show("Это поле не должно быть пустым!", textBox1, 3000);

return;

}

if (textBox2.Text == "" && tabConrol1.SelectedTab == tabConrol1.TabPages["registration"])

{

toolTip1.Show("Это поле не должно быть пустым!", textBox2, 3000);

return;

}

}

if (tabConrol1.SelectedIndex < 3)

{

tabConrol1.SelectedIndex = tabConrol1.SelectedIndex + 1;

back\_button.Enabled = true;

}

else {

tabConrol1.SelectedIndex = tabConrol1.SelectedIndex + 1;

next\_button.Enabled = false;

}

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

if (paint)

{

Graphics formGraphics = Graphics.FromImage(bmp);

if (radioButton3.Checked)

{

color = new SolidBrush(Color.Black);

formGraphics.FillEllipse(color, e.X, e.Y, 5, 5);

}

else {

color = new SolidBrush(Color.White);

formGraphics.FillEllipse(color, e.X, e.Y, 20, 20);

}

formGraphics.Dispose();

pictureBox1.Image = bmp;

}

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

paint = false;

pictureBox1.Image = bmp;

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

paint = true;

}

//save

List<PictureBox> pic\_boxes = new List<PictureBox>();

private void saveImage()

{

//---merge 2 image (grid and user art)

Bitmap fimg = new Bitmap(global::noname.Properties.Resources.empty);

Bitmap bimg = bmp;

for (int i = 0; i < bmp.Width; i++)

for (int j = 0; j < bmp.Height; j++)

{

Color fm = fimg.GetPixel(i, j);

Color bm = bimg.GetPixel(i, j);

byte af = (byte)(fm.A \* 255 / byte.MaxValue);

byte a = bm.A;

byte r = (byte)((fm.R \* af + bm.R \* (byte.MaxValue - af)) / byte.MaxValue);

byte g = (byte)((fm.G \* af + bm.G \* (byte.MaxValue - af)) / byte.MaxValue);

byte b = (byte)((fm.B \* af + bm.B \* (byte.MaxValue - af)) / byte.MaxValue);

bmp.SetPixel(i, j, Color.FromArgb(a, r, g, b));

}

Image img = new Bitmap(620, 620);

Graphics gra = Graphics.FromImage(img);

Rectangle fillRect = new Rectangle(0, 0, 620, 620);

SolidBrush blueBrush = new SolidBrush(Color.Black);

Region fillRegion = new Region(fillRect);

gra.FillRegion(blueBrush, fillRegion);

gra.DrawImage(pic\_boxes[0].Image, 0, 0 , 200f, 200f);

gra.DrawImage(pic\_boxes[1].Image, 0, 210, 200f, 200f);

gra.DrawImage(pic\_boxes[2].Image, 0, 420, 200f, 200f);

gra.DrawImage(pic\_boxes[5].Image, 210, 0, 200f, 200f);

gra.DrawImage(pic\_boxes[4].Image, 210, 210, 200f, 200f);

gra.DrawImage(pic\_boxes[3].Image, 210, 420, 200f, 200f);

gra.DrawImage(pic\_boxes[8].Image, 420, 0, 200f, 200f);

gra.DrawImage(pic\_boxes[7].Image, 420, 210, 200f, 200f);

gra.DrawImage(pic\_boxes[6].Image, 420, 420, 200f, 200f);

pictureBox12.Image = new Bitmap(bmp);

button4\_Click(new object(), new EventArgs());

}

private void pictureToDoc()

{

Word.Application winWord = new Word.Application();

winWord.Visible = false;

object missing = System.Reflection.Missing.Value;

Word.Document doc = winWord.Documents.Add(ref missing, ref missing, ref missing, ref missing);

Word.Paragraph para1 = doc.Content.Paragraphs.Add(ref missing);

object styleHeading1 = "Заголовок 1";

para1.Range.set\_Style(styleHeading1);

para1.Range.Text = "Результаты опыта 5";

para1.Range.InsertParagraphAfter();

para1.Range.Text = "Первая часть:";

para1.Range.InsertParagraphAfter();

Clipboard.SetImage(new Bitmap(pictureBox13.Image,300,300));

winWord.ActiveDocument.Paragraphs[3].Range.Paste();

para1.Range.InsertParagraphAfter();

Clipboard.SetImage(new Bitmap(pictureBox12.Image, 300, 300));

winWord.ActiveDocument.Paragraphs[4].Range.Paste();

para1.Range.InsertParagraphAfter();

para1.Range.Text = label17.Text;

Directory.CreateDirectory(@"Results\" + textBox1.Text);

object filename = Application.StartupPath.ToString() + @"\Results\" + textBox1.Text+ @"\Опыт 5 " + DateTime.Now.ToString().Replace('.', '-').Replace(':', '\_') + ".docx";

doc.SaveAs(filename);

doc.Close(ref missing, ref missing, ref missing);

doc = null;

winWord.Quit(ref missing, ref missing, ref missing);

winWord = null;

//--------SAVE RESULTS------------

Result\_pic result = new Result\_pic();

result.StudName = textBox1.Text;

result.StudGroup = textBox2.Text;

result.ExpDate = DateTime.Now.ToString();

result.First\_part\_pic = textBox1.Text + DateTime.Now.ToString().Replace('.', '-').Replace(':', '\_') +"pic"+ ".jpeg";

result.First\_part\_user\_pic = textBox1.Text + DateTime.Now.ToString().Replace('.', '-').Replace(':', '\_') + "user" + ".jpeg";

result.Second\_part = label17.Text;

XmlWorker.saveResults(result);

Directory.CreateDirectory(@"Data\Cashe\");

pictureBox12.Image.Save(@"Data\Cashe\" + textBox1.Text + DateTime.Now.ToString().Replace('.', '-').Replace(':', '\_') + "pic" + ".jpeg", System.Drawing.Imaging.ImageFormat.Jpeg);

pictureBox13.Image.Save(@"Data\Cashe\" + textBox1.Text + DateTime.Now.ToString().Replace('.', '-').Replace(':', '\_') + "user" + ".jpeg", System.Drawing.Imaging.ImageFormat.Jpeg);

}

private void buttonBack\_click(object sender, EventArgs e)

{

textBox\_stimuls.Text = "";

if (tabConrol1.SelectedTab == tabConrol1.TabPages["create\_base"] || tabConrol1.SelectedTab == tabConrol1.TabPages["createBaseIMG"]

|| tabConrol1.SelectedTab == tabConrol1.TabPages["create\_change"])

{

if(!create)

tabConrol1.SelectTab("admin");

else

tabConrol1.SelectTab("admin\_mode");

return;

}

if (tabConrol1.SelectedTab == tabConrol1.TabPages["result\_pic"]) {

if (ADMIN && !dataGridView2.Visible)

{

dataGridView2.Visible = true;

button\_delete.Visible = true;

}

else

if (ADMIN)

{

tabConrol1.SelectTab("admin");

dataGridView2.Visible = false;

}

else {

tabConrol1.SelectTab("select\_mode");

dataGridView1.Rows.Clear();

next\_button.Enabled = true;

}

}

if (tabConrol1.SelectedTab == tabConrol1.TabPages["do\_with\_pic"] || tabConrol1.SelectedTab == tabConrol1.TabPages["end\_with\_pic"]) {

radioButton3.Visible = false;

radioButton4.Visible = false;

button\_clear.Visible = false;

pictureBox10.Visible = true;

pictureBox2.Visible = true;

pictureBox3.Visible = true;

pictureBox4.Visible = true;

pictureBox5.Visible = true;

pictureBox6.Visible = true;

pictureBox7.Visible = true;

pictureBox8.Visible = true;

pictureBox9.Visible = true;

tabConrol1.SelectTab("select\_mode");

label9.Text = "Стимул:";

button4\_Click(new object(), new EventArgs());

}

else

if (tabConrol1.SelectedTab == tabConrol1.TabPages["results"])

{

if (ADMIN && !dataGridView2.Visible)

{

dataGridView2.Visible = true;

button\_delete.Visible = true;

}

else

if (ADMIN)

{

tabConrol1.SelectTab("admin\_mode");

dataGridView2.Visible = false;

}

else {

tabConrol1.SelectTab("choseTest");

dataGridView1.Rows.Clear();

next\_button.Enabled = true;

}

}

else {

if (tabConrol1.SelectedTab == tabConrol1.TabPages["admin\_mode"])

{

tabConrol1.SelectTab("choseUser");

next\_button.Enabled = true;

return;

}

if (tabConrol1.SelectedIndex > 1)

{

if (tabConrol1.SelectedTab != tabConrol1.TabPages["admin"])

next\_button.Enabled = true;

tabConrol1.SelectedIndex = tabConrol1.SelectedIndex - 1;

}

else {

tabConrol1.SelectedIndex = tabConrol1.SelectedIndex - 1;

back\_button.Enabled = false;

}

}

next\_button.Text = "Далее";

RET = true;

words = "";

pressed = false;

randPull.Clear();

second\_part\_answers.Clear();

label6.Text = "Введите запомнившееся стимулы";

textBox\_stimuls.Enabled = false;

textBox\_stimuls.Visible = true;

Button\_Yes.Visible = false;

Button\_NO.Visible = false;

}

private void Wait(double seconds)

{

int ticks = System.Environment.TickCount + (int)Math.Round(seconds \* 1000.0);

while (System.Environment.TickCount < ticks)

{

Application.DoEvents();

}

}

private void showAlert()

{

if (chosedTest != 4)

{

label\_alert.Visible = true;

label\_alert.Refresh();

back\_button.Enabled = false;

back\_button.Refresh();

Wait(3.0);

label\_alert.Visible = false;

label\_alert.Refresh();

}

else

{

label\_alert\_img.Visible = true;

label\_alert\_img.Refresh();

back\_button.Enabled = false;

back\_button.Refresh();

label22.Refresh();

Wait(3.0);

label\_alert\_img.Visible = false;

label\_alert\_img.Refresh();

}

}

public void beginTestWords()

{

RET = false;

showAlert();

Settings settings = new Settings();

settings = XmlWorker.getSetting(chosedTest.ToString());

Random r = new Random(0, XmlWorker.getWordCount());

StringBuilder rs = new StringBuilder();

string letters = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя";

string numbers = "1234567890";

if (chosedTest == 0)

for (int i = 0; i < Int32.Parse(settings.Stimul\_size) \* 2; i++)

{

randPull.Add(XmlWorker.getWord(r.Next()).Word);

}

if (chosedTest == 1)

for (int i = 0; i < Int32.Parse(settings.Stimul\_size) \* 2; i++)

{

for (int j = 0; j < Int32.Parse(settings.Symbols\_num); j++) {

rs.Append(letters[r.Next(32)]);

}

randPull.Add(rs.ToString());

rs.Clear();

}

if (chosedTest == 2)

for (int i = 0; i < Int32.Parse(settings.Stimul\_size) \* 2; i++)

{

for (int j = 0; j < Int32.Parse(settings.Symbols\_num); j++)

{

rs.Append(numbers[r.Next(10)]);

}

randPull.Add(rs.ToString());

rs.Clear();

}

if (chosedTest == 3)

{

r = new Random(0, 32);

for (int i = 0; i < Int32.Parse(settings.Stimul\_size) \* 2; i++)

{

randPull.Add(letters[r.Next()].ToString());

}

}

r = new Random(0, Int32.Parse(settings.Stimul\_size) \* 2);

for (int i = 0; i < Int32.Parse(settings.Stimul\_size); i++)

words += randPull[r.Next()] + ", ";

words = words.Remove(words.Length - 2) + ".";

stim\_words\_label.Text = words;

}

List<string> truePaths = new List<string>();

public void beginTestpic() {

if (!pressed)

{

label\_alert\_img.Text = " ВНИМАНИЕ !";

label22.Text = "Опыт номер 5. \nЧасть 1.";

if (train\_mode)

label22.Text += "\nТренировочная \nсерия.";

showAlert();

}

else {

label\_alert\_img.Text = "ВНИМАНИЕ! НАЧИНАЕТСЯ ВТОРАЯ ЧАСТЬ ОПЫТА!";

label22.Text = "Опыт номер 5. \nЧасть 2.";

if (train\_mode)

label22.Text += "\nТренировочная \nсерия.";

showAlert();

}

DirectoryInfo directoryInfo = new DirectoryInfo(@"Library\Pictures\"+ XmlWorker.getSetting("4").ImgBase);

Random r = new Random(0, 20);

List<string> paths\_pic = new List<string>();

foreach (FileInfo file in directoryInfo.GetFiles())

{

paths\_pic.Add(file.FullName);

}

for (int i = 0; i < 20; i++)

randPull.Add(paths\_pic[r.Next()]);

if (pressed)

truePaths.Clear();

r = new Random(0, 20);

int ran = r.Next(0, 20);

pictureBox10.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox2.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox3.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox4.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox5.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox6.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox7.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox8.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

ran = r.Next();

pictureBox9.Image = new Bitmap(randPull[ran]);

truePaths.Add(randPull[ran]);

if (pressed)

{

Thread myThread = new Thread(new ThreadStart(sleep));

myThread.Start();

}

}

List<Button> btn = new List<Button>();

public void endTestPic() {

Random rand = new Random(0, 20);

label23.Invoke(new Action(() => label23.Text = "Опыт номер 5.\nЧасть 2."));

if (train\_mode)

label23.Invoke(new Action(() => label23.Text = "Опыт номер 5.\nЧасть 2.\nТренировочная серия."));

int ran;

for (int i = 0; i < btn.Count; i++) {

ran = rand.Next();

btn[i].BackgroundImage = new Bitmap(randPull[ran]);

btn[i].FlatAppearance.BorderColor = Color.White;

btn[i].Tag = randPull[ran];

}

}

public void res\_pic()

{

int count\_true\_answer = 0;

for (int i = 0; i < truePaths.Count; i++) {

for (int j = 0; j < 20; j++)

{

if (truePaths[i] == btn[j].Tag.ToString()) {

if (btn[j].FlatAppearance.BorderColor == Color.Lime)

count\_true\_answer++;

}

}

}

label17.Text = "Во второй части опыта узнано " + count\_true\_answer + " стимулов.";

}

public void sleep() {

Settings settings = new Settings();

settings = XmlWorker.getSetting(chosedTest.ToString());

back\_button.Invoke(new Action(() => back\_button.Enabled = false));

if (chosedTest != 4)

{

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Enabled = false));

Thread.Sleep(Int32.Parse(settings.Exposition\_time));

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Enabled = true));

stim\_words\_label.Invoke(new Action(() => stim\_words\_label.Text = ""));

}

else

{

Thread.Sleep(Int32.Parse(settings.Exposition\_time));

foreach (PictureBox p in pic\_boxes)

p.Invoke(new Action(() => p.Visible = false));

label9.Invoke(new Action(() => label9.Text = "Нарисуйте стимулы"));

radioButton3.Invoke(new Action(() => radioButton3.Visible = true));

radioButton4.Invoke(new Action(() => radioButton4.Visible = true));

button\_clear.Invoke(new Action(() => button\_clear.Visible = true));

if (pressed)

{

tabConrol1.Invoke(new Action(() => tabConrol1.SelectTab("end\_with\_pic")));

endTestPic();

foreach (PictureBox p in pic\_boxes)

p.Invoke(new Action(() => p.Visible = true));

back\_button.Invoke(new Action(() => back\_button.Enabled = true));

next\_button.Invoke(new Action(() => next\_button.Enabled = true));

}

}

back\_button.Invoke(new Action(() => back\_button.Enabled = true));

next\_button.Invoke(new Action(() => next\_button.Enabled = true));

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Focus()));

}

private void tabControl1\_Selected(object sender, TabControlEventArgs e)

{

if (e.TabPage.Name == "choseTest")

{

comboBox1.SelectedIndex = 0;

return;

}

if (e.TabPage.Name == "registration"){

textBox1.Text = "";

textBox2.Text = "";

}

if (e.TabPage.Name == "admin")

{

Settings settings = new Settings();

settings = XmlWorker.getSetting("0");

textBox\_exposition\_time.Text = settings.Exposition\_time;

chose\_test\_admin.SelectedIndex = 0;

next\_button.Enabled = false;

comboBox2.Items.Clear();

DirectoryInfo directoryInfo = new DirectoryInfo(@"Library\Words");

int count = 0;

foreach (FileInfo file in directoryInfo.GetFiles())

{

comboBox2.Items.Add(file.Name);

if (!file.Name.Equals(settings.WordBase))

count++;

}

comboBox2.SelectedIndex = count;

}

if(e.TabPage.Name == "result\_pic" || e.TabPage.Name == "results")

back\_button.Text = "В меню";

else

back\_button.Text = "Назад";

if (e.TabPage.Name == "results")

{

next\_button.Enabled = false;

if (ADMIN)

{

chose\_test\_result.SelectedIndex = 0;

label\_chose\_result.Visible = true;

chose\_test\_result.Visible = true;

return;

}

else

{

label\_chose\_result.Visible = false;

chose\_test\_result.Visible = false;

}

}

if (tabConrol1.SelectedTab == tabConrol1.TabPages["choseUser"])

radioButton1.Checked = true;

if (e.TabPage.Name == "select\_mode" || e.TabPage.Name == "admin\_mode" || e.TabPage.Name == "result\_pic" || e.TabPage.Name == "create\_change")

{

next\_button.Enabled = false;

return;

}

if (e.TabPage.Name == "createBaseIMG")

next\_button.Enabled = false;

back\_button.Enabled = true;

}

private void chose\_test\_admin\_SelectedIndexChanged(object sender, EventArgs e)

{

Settings settings = new Settings();

settings = XmlWorker.getSetting(chose\_test\_admin.SelectedIndex.ToString());

textBox\_exposition\_time.Text = settings.Exposition\_time;

textBox\_stimul\_size.Text = settings.Stimul\_size;

if (chose\_test\_admin.SelectedIndex == 1 || chose\_test\_admin.SelectedIndex == 2) {

label19.Visible = false;

comboBox2.Visible = false;

button\_deleteBase.Visible = false;

groupBox4.Size = new Size(407,104);

label15.Visible = true;

textBox\_symbols\_num.Visible = true;

textBox\_symbols\_num.Text = settings.Symbols\_num;

}

else

{

if (chose\_test\_admin.SelectedIndex == 0)

{

label19.Visible = true;

comboBox2.Visible = true;

button\_deleteBase.Visible = true;

}

groupBox4.Size = new Size(407, 74);

label15.Visible = false;

textBox\_symbols\_num.Visible = false;

}

chosedTest = chose\_test\_admin.SelectedIndex;

if (chosedTest == 4)

{

label19.Visible = true;

comboBox2.Visible = true;

button\_deleteBase.Visible = true;

textBox\_stimul\_size.Visible = false;

label14.Visible = false;

groupBox4.Size = new Size(407, 52);

DirectoryInfo directoryInfo = new DirectoryInfo(@"Library\Pictures\");

DirectoryInfo[] dirs = directoryInfo.GetDirectories();

comboBox2.Items.Clear();

for (int i = 0; i < dirs.Length; i++)

comboBox2.Items.Add(dirs[i]);

}

else

{

textBox\_stimul\_size.Visible = true;

label14.Visible = true;

comboBox2.Items.Clear();

DirectoryInfo directoryInfo = new DirectoryInfo(@"Library\Words");

foreach (FileInfo file in directoryInfo.GetFiles())

comboBox2.Items.Add(file.Name);

}

comboBox2.SelectedIndex = 0;

}

private void button\_save\_admin(object sender, EventArgs e)

{

Settings settings = new Settings();

settings.test\_Number = chose\_test\_admin.SelectedIndex;

settings.Exposition\_time = textBox\_exposition\_time.Text;

settings.Stimul\_size = textBox\_stimul\_size.Text;

settings.Symbols\_num = textBox\_symbols\_num.Text;

settings.Instruction = XmlWorker.getSetting(chosedTest.ToString()).Instruction;

if (chose\_test\_admin.SelectedIndex == 0)

settings.WordBase = comboBox2.Items[comboBox2.SelectedIndex].ToString();

else

settings.WordBase = XmlWorker.getSetting(chosedTest.ToString()).WordBase;

if(chose\_test\_admin.SelectedIndex == 4)

settings.ImgBase = comboBox2.Items[comboBox2.SelectedIndex].ToString();

else

settings.ImgBase = XmlWorker.getSetting(chosedTest.ToString()).ImgBase;

XmlWorker.editSetting(settings);

MessageBox.Show("Сохранено.", "", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);

}

public void endTestWords()

{

label6.Invoke(new Action(() => label6.Text = "Был ли этот стимул в предыдущей части опыта?"));

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Visible = false));

Button\_Yes.Invoke(new Action(() => Button\_Yes.Visible = true));

Button\_NO.Invoke(new Action(() => Button\_NO.Visible = true));

if(!train\_mode)

label\_view\_mode.Invoke(new Action(() => label\_view\_mode.Text = "Опыт номер " +(chosedTest + 1)+". Часть 2."));

else

label\_view\_mode.Invoke(new Action(() => label\_view\_mode.Text = "Опыт номер " + (chosedTest + 1) + ". Часть 2.\n Тренировочная серия."));

Settings settings = new Settings();

settings = XmlWorker.getSetting(chosedTest.ToString());

for (int i = 0; i < Int32.Parse(settings.Stimul\_size) \* 2; i++)

{

stim\_words\_label.Invoke(new Action(() => stim\_words\_label.Text = randPull[i]));

settings = XmlWorker.getSetting(chosedTest.ToString());

while (!pressed)

{

if (RET)

return;

Thread.Sleep(100);

}

pressed = false;

}

if (train\_mode)

{

tabConrol1.Invoke(new Action(() => tabConrol1.SelectTab("select\_mode")));

RET = true;

words = "";

pressed = false;

randPull.Clear();

second\_part\_answers.Clear();

stim\_words\_label.Invoke(new Action(() => stim\_words\_label.Text = ""));

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Text = ""));

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Enabled = false));

label6.Invoke(new Action(() => label6.Text = "Введите запомнившееся стимулы"));

textBox\_stimuls.Invoke(new Action(() => textBox\_stimuls.Visible = true));

Button\_Yes.Invoke(new Action(() => Button\_Yes.Visible = false));

Button\_NO.Invoke(new Action(() => Button\_NO.Visible = false));

return;

}

Result result = new Result();

result.StudName = textBox1.Text;

result.StudGroup = textBox2.Text;

result.ExpNumber = chosedTest;

result.ExpDate = DateTime.Now.ToString();

result.First\_part\_words = words;

result.First\_part\_answers = first\_part\_answer;

result.Second\_part\_answers = second\_part\_answers;

XmlWorker.saveResults(result);

showResult(result);

tabConrol1.Invoke(new Action(() => tabConrol1.SelectTab("results")));

XmlWorker.saveDoc(result);

}

private void Button\_Yes\_Click(object sender, EventArgs e)

{

second\_part\_answers.Add(stim\_words\_label.Text, "1");

pressed = true;

}

private void Button\_NO\_Click(object sender, EventArgs e)

{

second\_part\_answers.Add(stim\_words\_label.Text, "0");

pressed = true;

}

private void showResult(Result res)

{

List<string> fpw = res.First\_part\_words.Split(',').ToList<string>();

fpw[fpw.Count-1] = fpw[fpw.Count-1].Remove(fpw[fpw.Count-1].Length - 1);

List<string> fpa = res.First\_part\_answers.Split(' ').ToList<string>();

int fpa\_count = -1;

List<string> fpa\_buffer = new List<string>();

fpa\_buffer.AddRange(fpa);

foreach (string s in fpa)

{

fpa\_count++;

if (s.Equals(""))

{

fpa\_buffer.RemoveAt(fpa\_count);

fpa\_count--;

}

}

fpa.Clear();

fpa.AddRange(fpa\_buffer);

fpa\_count = fpa.Count;

if (fpw.Count != fpa.Count)

for (int j = 0; j < fpw.Count - fpa\_count; j++)

fpa.Add("--Не введено--");

int i = 0;

foreach (KeyValuePair<string, string> k in res.Second\_part\_answers)

{

if (i < fpw.Count)

{

if (Int32.Parse(k.Value) == 1)

dataGridView1.Invoke(new Action(() => dataGridView1.Rows.Add(fpw[i], fpa[i], k.Key, "Да")));

else

dataGridView1.Invoke(new Action(() => dataGridView1.Rows.Add(fpw[i], fpa[i], k.Key, "Нет")));

i++;

}

else

{

if (Int32.Parse(k.Value) == 1)

dataGridView1.Invoke(new Action(() => dataGridView1.Rows.Add("", "", k.Key, "Да")));

else

dataGridView1.Invoke(new Action(() => dataGridView1.Rows.Add("", "", k.Key, "Нет")));

}

}

}

private void showResultPic(Result\_pic res) {

tabConrol1.SelectTab("result\_pic");

pictureBox13.Image = new Bitmap(@"Data\Cashe\" + res.First\_part\_pic);

pictureBox12.Image = new Bitmap(@"Data\Cashe\" + res.First\_part\_user\_pic);

label17.Text = res.Second\_part;

}

List<Result> resultsForAdmin = new List<Result>();

private void button\_showResults\_Click(object sender, EventArgs e)

{

button\_formuls.Visible = false;

dataGridView2.Rows.Clear();

resultsForAdmin.Clear();

button\_delete.Visible = true;

if (chosedTest != 4)

{

List<Result> results = XmlWorker.getResults();

foreach (Result res in results)

{

if (chosedTest == res.ExpNumber)

{

dataGridView2.Rows.Add(res.StudName, res.StudGroup, res.ExpDate);

resultsForAdmin.Add(res);

}

}

}

else

{

List<Result\_pic> results = XmlWorker.getResultsPic();

foreach (Result\_pic res in results)

{

dataGridView2.Rows.Add(res.StudName, res.StudGroup, res.ExpDate);

}

}

tabConrol1.SelectTab("results");

dataGridView2.Visible = true;

}

private void dataGridView2\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

button\_formulsPic.Visible = false;

dataGridView1.Rows.Clear();

dataGridView2.Visible = false;

if (chosedTest != 4)

{

List<Result> results = resultsForAdmin;

showResult(results[e.RowIndex]);

}

else

{

List<Result\_pic> results = XmlWorker.getResultsPic();

showResultPic (results[e.RowIndex]);

}

button\_delete.Visible = false;

}

private void button\_delete\_Click(object sender, EventArgs e)

{

if (chosedTest != 4)

{

List<Result> results = resultsForAdmin;

if (dataGridView2.CurrentRow != null)

{

XmlWorker.deleteResult(results[dataGridView2.CurrentRow.Index]);

button\_showResults\_Click(new object(), new EventArgs());

}

else

MessageBox.Show("Нечего удалять.", "Неудача", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

List<Result\_pic> results = XmlWorker.getResultsPic();

if (dataGridView2.CurrentRow != null)

{

XmlWorker.deleteResult(results[dataGridView2.CurrentRow.Index]);

button\_showResults\_Click(new object(), new EventArgs());

}

else

MessageBox.Show("Нечего удалять.", "Неудача", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

int pic\_count = 0;

private void buttons\_click\_image(object sender, EventArgs e)

{

if (((Button)sender).FlatAppearance.BorderColor == Color.White)

{

if (pic\_count < 9)

{

pic\_count++;

((Button)sender).FlatAppearance.BorderColor = Color.Lime;

}

}

else {

pic\_count--;

((Button)sender).FlatAppearance.BorderColor = Color.White;

}

label16.Text = label16.Text.Remove(label16.Text.Length - 1) + pic\_count.ToString();

return;

}

//----create or instruction change---

bool create = false;

private void button25\_Click(object sender, EventArgs e)

{

textBox\_dbName.Visible = false;

label\_test\_num.Text = "Введите набор стимулов через пробел ";

create = false;

label\_test\_num.Text = "Описание опыта " + (chose\_test\_admin.SelectedIndex + 1);

textBox\_test\_instr.Text = XmlWorker.getSetting(chose\_test\_admin.SelectedIndex.ToString()).Instruction;

tabConrol1.SelectTab("create\_change");

}

private void button\_create\_base\_Click(object sender, EventArgs e)

{

create = true;

tabConrol1.SelectTab("create\_base");

}

private void button29\_Click(object sender, EventArgs e)

{

tabConrol1.SelectTab("createBaseIMG");

}

private void button28\_Click(object sender, EventArgs e)

{

create = true;

textBox\_dbName.Visible = true;

label\_test\_num.Text = "Введите имя базы данных ";

textBox\_test\_instr.Text = "";

tabConrol1.SelectTab("create\_change");

}

private void button\_save\_Click(object sender, EventArgs e)

{

if (create)

{

if (textBox\_test\_instr.Text != "")

{

List<string> DBwords = new List<string>();

DBwords = textBox\_test\_instr.Text.Split(' ').ToList();

XmlWorker.createDataBase(DBwords, textBox\_dbName.Text);

textBox\_test\_instr.Text = "";

}

tabConrol1.SelectTab("admin\_mode");

}

else {

Settings setting = new Settings();

setting = XmlWorker.getSetting(chose\_test\_admin.SelectedIndex.ToString());

setting.Instruction = textBox\_test\_instr.Text;

XmlWorker.editSetting(setting);

TextLabel\_instruct.Text = XmlWorker.getSetting("0").Instruction;

textBox\_test\_instr.Text = "";

tabConrol1.SelectTab("admin");

}

back\_button.Enabled = true;

}

List<string> imgFiles = new List<string>();

private void button\_choseFile\_Click(object sender, EventArgs e)

{

imgFiles.Clear();

openFileDialog1.Filter = "JPEG Files (\*.jpeg)|\*.jpeg|PNG Files (\*.png)|\*.png|JPG Files (\*.jpg)|\*.jpg|GIF Files (\*.gif)|\*.gif";

if (openFileDialog1.ShowDialog() != DialogResult.OK) return;

foreach (string name in openFileDialog1.FileNames) {

imgFiles.Add(name);

}

if (imgFiles.Count < 20)

{

label21.Text = "Выбрано меньше 20 изображений.";

label21.ForeColor = Color.Red;

imgFiles.Clear();

return;

}

else {

label21.Text = "Выбрано.";

label21.ForeColor = Color.Gold;

}

}

private void button30\_Click(object sender, EventArgs e)

{

Directory.CreateDirectory(@"Library\Pictures\" + textBox4.Text);

if (imgFiles.Count != 0)

if (textBox4.Text != "")

{

foreach (string name in imgFiles)

{

File.Copy(name, @"Library\Pictures\" + textBox4.Text + name.Substring(name.LastIndexOf('\\')), true);

}

tabConrol1.SelectTab("admin\_mode");

label21.Text = "FILES ARE NOT SELECTED";

label21.ForeColor = Color.Black;

}

else

toolTip1.Show("Заполни это, Нигга!", textBox4, 3000);

else

label21.Text = "NOT CHOSEN";

}

private void button\_deleteBase\_Click(object sender, EventArgs e)

{

DirectoryInfo directoryInfo = new DirectoryInfo(@"Library\Words");

if (MessageBox.Show("Вы уверены, что хотите удалить " + comboBox2.SelectedItem + "?", "Удаление...", MessageBoxButtons.OKCancel, MessageBoxIcon.Question) == DialogResult.OK)

if (chosedTest == 0)

{

if (directoryInfo.GetFiles().Length == 1)

MessageBox.Show("Нельзя удалить последнюю базу стимулов", "Внимание!", MessageBoxButtons.OK , MessageBoxIcon.Warning);

else

File.Delete(@"Library\Words\" + comboBox2.SelectedItem);

}

else

{

directoryInfo = new DirectoryInfo(@"Library\Pictures");

if (directoryInfo.GetDirectories().Length == 1)

MessageBox.Show("Нельзя удалить последнюю базу стимулов", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

else

Directory.Delete(@"Library\Pictures\" + comboBox2.SelectedItem, true);

}

chose\_test\_admin\_SelectedIndexChanged(new object(), new EventArgs());

}

private void registration\_Paint(object sender, PaintEventArgs e)

{

textBox1.Select();

}

}

}