

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
им. ГУМАРБЕКА ДАУКЕЕВА»  
Кафедра IT – инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»  
Зав. кафедрой PhD, доцент Досжанова А.А.  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2020 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка информационной обучающей системы для детей с расстройством аутистического спектра  
Специальность 5В060200 – Информатика  
Выполнила: Алипбаева А.А. Группа Инф-16-2  
Научный руководитель: PhD, доцент Маликова Ф.У

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2020 г.

по безопасности жизнедеятельности: ассистент Тыщенко Е.М.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2020 г.

по программному обеспечению: ст. преп. Майкотов М.Н.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2020 г.

Нормоконтролер: ст. преп. Абсатарова Б.Р.  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2020 г.

Рецензент: \_\_\_\_\_  
(учёная степень, звание, Ф.И.О.)  
\_\_\_\_\_ « \_\_\_\_ » \_\_\_\_\_ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Алипбаевой Алине Асхатовне

Тема проекта: Разработка информационной обучающей системы для детей с расстройством аутистического спектра

Утверждена приказом по университету № \_\_\_ от «\_\_\_» \_\_\_\_\_ 2020 г.

Срок сдачи законченного проекта «\_\_\_» \_\_\_\_\_ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): руководство по программированию на С#, интегрированная среда Visual Studio, данные преддипломной практики.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): имеется 10 таблиц и 43 иллюстрации.

Основная рекомендуемая литература:

1 Петцольд Ч. Программирование с использованием Microsoft Windows Forms. Мастер-класс / Пер. с англ. – М.: Русская редакция; СПб.: Питер, 2006. – 432 стр.

2 Шилдт Г. С#: Учебный курс – СПб.: Питер; Киев: ВНУ, 2003. — 512 с.

3 Лабор В. В. Си Шарп: Создание приложений для Windows/ В. В. Лабор. – Мн.: Харвест, 2003. - 384 с.

4 Мацяшек Л. А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML: пер. с англ. / Л. А. Мацяшек. - Москва [и др.]: Вильямс, 2002. – 432 с.

5 Инклюзивное образование. Настольная книга педагога, работающего с детьми с ОВЗ: Методическое пособие — М.: Гуманитарный изд. центр ВЛАДОС, 2011. - 167 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р	21.04.2020	
Безопасности жизнедеятельности	Тыщенко Е.М	30.04.2020	
Программная часть	Майкотов М.Н	14.05.2020	
Нормконтролер	Абсатарова Б.Р	13.05.2020 – 18.05.2020	

**ГРАФИК**  
подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	17.02.2020 – 26.02.2020	
Проектирование приложения	27.02.2020 – 23.03.2020	
Программная реализация	24.03.2020 – 30.04.2020	

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 2020 г.

Заведующий кафедрой \_\_\_\_\_ А. А. Досжанова  
(подпись)

Научный руководитель проекта \_\_\_\_\_ Ф. У. Маликова  
(подпись)

Задание принял к исполнению студент \_\_\_\_\_ А. А. Алипбаева  
(подпись)

## Аңдатпа

Дипломдық жоба аутистік спектрі бұзылған балаларға арналған ақпараттық оқыту жүйесін әзірлеуге арналған. Әзірленген бағдарлама сөйлеу және коммуникативтік қиындықтарды бастан кешірген балаға қарым-қатынас құралы ретінде балалардың базалық қажеттіліктерін көрсететін PECS визуалды карточкаларын пайдалану арқылы танымдық қабілеттерді және коммуникативтік дағдыларды дамытуға көмектеседі. Карточкалар дауыстық пікірмен сүйемелденеді, сондай-ақ кітапхананы жаңа бейнелермен өз бетінше толықтыру мүмкіндігі бар. Бағдарлама әр түрлі кітапханаларды қолдану арқылы Microsoft Visual Studio әзірлеудің біріктірілген ортасында C# бағдарламалау тілінде іске асырылды. Бағдарлама интерфейсі Windows Forms технологиясы арқылы терезелік бағдарлама түрінде жасалған, онда санаттар бойынша әр түрлі қойындылар бар.

Бұдан басқа, бітіру жұмысында PAC балаларға арналған қолда бар оқыту бағдарламаларына талдау жүргізіледі, сондай-ақ өмір сүру қауіпсіздігі және жүйені өндіріске енгізуден экономикалық тиімділік мәселесі қарастырылады.

## **Аннотация**

Дипломный проект посвящен разработке информационной обучающей системы для детей с расстройством аутистического спектра. Разработанная программа поможет ребёнку, испытывающему речевые и коммуникативные затруднения, развивать познавательные способности и коммуникативные навыки посредством использования визуальных карточек PECS, отображающих базовые потребности детей, в качестве средства взаимодействия. Карточки сопровождаются голосовым комментарием, а также имеется возможность самостоятельно дополнять библиотеку новыми изображениями. Программа была реализована на языке программирования C# в интегрированной среде разработки Microsoft Visual Studio с использованием различных библиотек. Интерфейс программы был создан с помощью технологии Windows Forms в виде оконного приложения, в котором располагаются различные вкладки по категориям.

Кроме того, в выпускной работе проводится анализ существующих обучающих программ для детей с РАС, а также рассматривается вопрос безопасности жизнедеятельности и экономической эффективности от внедрения системы в производство.

## **Abstract**

The graduation paper is devoted to development of information training system for children with autism spectrum disorders. The developed program will help a child experiencing speech and communication difficulties to develop cognitive abilities and communication skills by using visual PECS cards that reflect basic needs of children as a means of interaction. The cards are accompanied by voice commentary, and you can add new images to the library yourself. The program was developed in the C# programming language in the Microsoft Visual Studio integrated environment using various libraries. The program interface was created using Windows Forms technology in the form of a window application that contains various tabs by category.

In addition, the graduation paper analyzes existing training programs for children with ASD, as well as considers the issues of life safety and economic efficiency from introducing the system into production.

## Содержание

Введение.....	8
1 Аналитическая часть.....	10
1.1 Компьютерные обучающие системы.....	10
1.2 Понятие расстройства аутистического спектра и его общая характеристика.....	12
1.3 Анализ существующих программ для детей с РАС .....	22
2 Проектная часть .....	30
2.1 Описание предметной области.....	30
2.2 Обоснование выбора языка программирования.....	30
2.3 Среда .NET Framework.....	32
2.4 Интерфейс программирования Windows Forms.....	35
3 Программная реализация .....	37
3.1 Структурное представление программы.....	37
3.2 Построение UML-диаграмм вариантов использования.....	38
3.3 Установка Microsoft Visual Studio 2019.....	41
3.4 Разработка приложения .....	44
3.5 Описание интерфейса.....	50
4 Техничко-экономическое обоснование проекта .....	62
4.1 Определение объема и трудоемкости разработки ПП.....	62
4.2 Расчет затрат на разработку ПП.....	63
4.3 Смета затрат на разработку ПП.....	68
4.4 Определение договорной цены ПП .....	69
4.5 Оценка социально-экономических результатов функционирования программного продукта.....	69
4.6 Вывод по экономической части.....	70
5 Безопасность жизнедеятельности .....	72
5.1 Роль анализаторов в операторской деятельности .....	72
5.2 Сравнительный анализ средств отображения информации .....	73
5.3 Вывод по разделу безопасность жизнедеятельности .....	76
Заключение.....	77
Список литературы .....	78
Приложение А Техническое задание .....	80
Приложение Б Листинг программы.....	81
Приложение В Акт внедрения.....	96

## Введение

Развитие и внедрение современных систем поддержки и обучения для улучшения качества жизни детей с расстройством аутистического спектра является одной из приоритетных задач специального и инклюзивного образования в нашей стране.

Расстройство аутистического спектра (РАС) - это расстройство развития нервной системы, связанное с несколькими этиологиями и различной степенью тяжести. Основные характеристики проявления РАС у детей наблюдаются в неспособности распознавать эмоции и мимику людей, что непосредственно влияет на их когнитивные навыки, ответственные за социальное взаимодействие и экспрессивное поведение. Следовательно, необходимо раннее вмешательство и персонализированное лечение наряду с психолого-педагогической поддержкой.

С интенсивным развитием технологий и информированности общественности число таких детей растет, соответственно возникает острая необходимость разработки отечественных информационных систем обучения для детей с аутизмом как для повсеместного внедрения в специальные образовательные и инклюзивные учреждения, так и для использования в индивидуальном порядке с помощью родителей.

Целью дипломного проекта является разработка информационной обучающей системы для детей с расстройством аутистического спектра, которая поможет ребенку развивать коммуникативные навыки, речевую активность и познавательные способности.

Для достижения цели были поставлены следующие задачи:

- 1) исследовать теоретические основы расстройств аутистического спектра;
- 2) провести подробный анализ существующих обучающих программ для детей с РАС;
- 3) определить структуру программы;
- 4) разработать удобный и понятный интерфейс;
- 5) дать оценку экономической эффективности внедрения системы.

Предметом научной работы является информационная обучающая система для детей с расстройством аутистического спектра.

Объект исследования - информационно-обучающее пространство в системе специального и инклюзивного образования.

На сегодняшний день проблема аутизма является одной из распространенных и актуальных проблем мирового масштаба, так как наблюдается высокая частота развития аутистических нарушений, а отсутствие детально разработанной системы специализированной помощи привлекает к себе внимание психологов, педагогов и психиатров.

Развитие ИКТ положили начало использованию технологий в программах обучения и развития детей с РАС. Так как у детей с аутистическими проявлениями чаще всего наблюдаются нарушения



социального взаимодействия, трудности в установлении контакта с окружающими и стереотипное поведение, им легче всего будет взаимодействовать с компьютером, который сможет заинтересовать ребенка визуальным и звуковым сопровождением.

Апробация разработанной обучающей системы проводилась в КГУ «Кабинет психолого-педагогической коррекции № 1» города Талдыкорган и подтверждается актом внедрения.

## **1 Аналитическая часть**

### **1.1 Компьютерные обучающие системы**

Компьютерные обучающие системы (КОС) представляют собой значительный класс средств, которые относятся к образовательным информационным технологиям и используются для решения педагогических задач. В настоящее время они содействуют образовательному процессу наравне с традиционными учебными средствами. Однако по сравнению с традиционными средствами обучения компьютерные обучающие системы предоставляют уникальные возможности, а многие действующие функции осуществляются с более высоким качеством.

Компьютерные обучающие системы – это программные модули, которые разработаны для активного использования в учебном процессе. Данные системы предназначены для интенсификации познавательной деятельности обучаемого, развития и совершенствования его профессиональных знаний и умений. Историю появления компьютерных обучающих систем можно проследить с начала промышленного изготовления компьютеров первых поколений и их появлением в образовательных институтах и организациях. Таким образом, появляются новые подходы в обучении, а именно применение компьютерных технологий обучения. Первая система обучения была разработана в США в конце 1950-х гг. и носила название Plato. Система Plato была разработана и совершенствовалась в течение 20 лет в стенах компании Control Data Corporation, которая занимается производством ЭВМ и различных автоматизированных систем [1]. Начало 1980-х гг. ознаменовалось массовым созданием и использованием похожих обучающих систем, так как персональные компьютеры стали широко распространяться среди населения. С тех пор ЭВМ стали больше использоваться для образовательных целей, нежели для математических расчетов.

Мультимедийные возможности современных ПК делают их уникальными по своей специфике в качестве средств обучения. Персональный компьютер используется в обучении самым разнообразным дисциплинам и служит основой для создания большого количества новых информационных технологий обучения.

Главной отличительной особенностью персонального компьютера от других технических средств обучения, являются способность компьютера одновременно совмещать следующие функции:

- диалоговый режим работы – каждое действие человека сопровождается ответом компьютера;
- отличные графические и иллюстративные возможности, что позволяет сделать обучение наглядным и визуально привлекательным;

-«персональность» компьютера – небольшие габариты и относительно невысокая стоимость позволяют обеспечить компьютерами весь класс или учебную аудиторию;

- простота в эксплуатации и интуитивно понятный интерфейс;

Если компьютер используется в качестве обучающего средства, то его технические возможности могут позволить:

- повысить наглядность изложения материала;

- индивидуализировать обучение;

- работать каждому обучающемуся в индивидуальном темпе;

- активизировать процесс обучения;

- сместить акценты с теоретических знаний на их практическое использование (и наоборот).

Кроме того, графические возможности дисплеев компьютеров позволяют сделать компьютерное обучение визуально привлекательным и более эффективным. На экране дисплея можно показывать различные фигуры и конструкции, изображения реальных и вымышленных объектов, фотографии, рисунки, gif- и 3D-изображения и так далее. Все это может быть реализовано как статически, так и в движении. Компьютерная графика может сделать видимыми такие процессы и явления, которые невозможно увидеть в реальности, можно создать визуальный образ того, что на самом деле не имеет никакой видимости (например, изображение абстрактных понятий, принципов теории относительности и т.п.). Эта способность компьютеров основана на так называемой когнитивной компьютерной графике - особом направлении использования компьютеров в научных исследованиях, когда возможности компьютера используются для изучения закономерностей.

### **Типы обучающих программ**

В зависимости от того, какие задачи необходимо решить во время обучения, компьютерные программы можно разделить на четыре типа:

1) имитационные и моделирующие;

2) тренировочные и контролирующие;

3) информационные;

4) развивающие игры.

Главным назначением тренировочных программ является закрепление и дальнейшее совершенствование полученных знаний и сформированных навыков, а контролирующих – контроль над этими новыми умениями.

Информационные программы направлены на обеспечение обучающегося новым дидактическим материалом. Разные задачи и вопросы используются в этих программах для управления ходом обучения и организации человеко-машинного диалога. Таким образом, если ученик дает неверные ответы, программа может вернуться «назад» и продемонстрировать учебный материал для повторного изучения.

Основой программ моделирования являются видеографические и вычислительные способности компьютера, которые позволяют проводить компьютерный эксперимент. В дополнение, данные программы позволяют наблюдать за определенным процессом на экране дисплея и влиять на его ход, используя клавиатуру.

Развивающие игры представляются в виде некоторой воображаемой среды, реализованной с помощью компьютерных технологий и существующей непосредственно в самом компьютере. Использование средств, предусмотренных программой для реализации возможностей, которые связаны непосредственно с изучением игрового и реального мира в сравнении, приводит к самостоятельному открытию закономерностей, отношений объектов действительности, развитию познавательных способностей и формированию личности обучаемого [2].

Повсеместное использование компьютерных обучающих систем в учебном процессе сопровождается и некоторыми трудностями, которые связаны с затратами времени на разработку, со сложностью этих разработок, а также с ограниченным использованием всех возможностей современных компьютеров.

На это есть несколько причин:

1) слабая заинтересованность во внедрении таких программ в учебный процесс;

2) недостаток специалистов, которые способны спроектировать, а затем и разработать высокоуровневую компьютерную обучающую программу;

3) отсутствие государственного финансирования таких разработок.

Несмотря на эти недостатки, компьютер является эффективным средством обучения, который позволяет находить новые возможности в образовании:

- позволяет активизировать познавательные процессы учащихся и вовлечь их в активную работу;

- повышает мотивацию к обучению;

- увеличивает возможности представления информации за счет наглядности и иллюстративности;

- позволяет изменять и улучшать качество контроля деятельности обучающегося;

- развивает у учащихся навыки рефлексии самостоятельной деятельности.

## **1.2 Понятие расстройства аутистического спектра и его общая характеристика**

### **1.2.1 Общая характеристика РАС у детей**

Проблема обучения детей с расстройствами аутистического спектра (РАС) является актуальной как в Казахстане, так и во всем мире. По

статистическим данным благотворительного Фонда Булата Утемуратова, реализующего в Казахстане проект «Аутизм. Мир один для всех», число детей с РАС достигает 60 тысяч в Республике Казахстан [3]. С интенсивным развитием технологий и информированности общественности число таких детей растет, соответственно возникает острая необходимость разработки отечественных информационных систем обучения для детей с аутизмом как для повсеместного внедрения в специальные образовательные и инклюзивные учреждения, так и для использования в индивидуальном порядке.

Рассмотрим основные определения расстройства аутистического спектра, его характеристики и особенности обучения детей данной категории. Расстройство аутистического спектра (РАС) - это группа расстройств развития и поведения, которые характеризуются нарушенным социальным взаимодействием, трудностями в установлении контакта с окружающими, повторяющимися действиями и ограниченными увлечениями. Аутистический спектр нарушений возникает на генетическом уровне до рождения ребенка и влияет на развивающийся мозг. Однако их воздействие на формирующуюся психику может быть разным, соответственно, ребенок может стать полностью неуправляемым или же проявить уникальные способности в какой-то области. Симптомы такого нарушения диагностируются в любом возрасте, но проявляются на раннем этапе жизни.

Согласно Международной классификации болезней существует следующее разделение расстройств аутистического спектра, которые включены в подраздел «Первазивные (общие) расстройства психологического развития» [4]:

Таблица 1 – Первазивные расстройства психологического развития

Название нарушения	Краткое определение	Подвиды
Детский аутизм	Проявляется у детей до трех лет, характеризуется общей задержкой в развитии в связи с нарушением социального взаимодействия, коммуникации и поведения, которое отличается монотонностью и стереотипностью. Могут появиться фобии, вспышки раздражения и агрессивность.	Синдром Кеннера Детский психоз

Продолжение таблицы 1

Атипичный аутизм	Проявляется у детей после трех лет, нарушения могут быть недостаточно выражены в одной из трех областей (нарушенные навыки социального взаимодействия, общения и поведения). Чаще всего развивается у детей с задержкой психического или речевого развития.	Умственная отсталость с чертами аутизма Атипичный детский психоз
Синдром Ретта	Обнаруживается в основном у девочек, характеризуется нормальным ранним развитием, но с частичной или полной потерей речи, двигательных навыков, что приводит к тяжелым умственным нарушениям.	-
Синдром Аспергера	Характеризуется нарушенными функциями социального взаимодействия и общения, ограниченными интересами, но при этом сохраняется нормальное развитие познавательных и речевых способностей.	Шизоидное расстройство в детском возрасте Аутическая психопатия
Дезинтегративное расстройство детского возраста	Характеризуется нормальным развитием до появления симптомов расстройства, в том числе утраты интереса к окружающему миру, монотонного поведения, нарушений в сфере социального взаимодействия и общения.	Симбиозный психоз Детская деменция Синдром Геллера Дезинтегративный психоз
Гиперактивное расстройство, сочетающееся с умственной отсталостью и стереотипными движениями	Характерно для детей с выраженной умственной отсталостью, проявляется через нарушение внимания, гиперактивность и ограниченность поведения.	-
Другие общие расстройства развития	Проявляются в раннем возрасте, могут сопровождаться тихой слабой памятью, речью и гиперактивным поведением.	-

В дополнение, следует отметить, что основной особенностью этих детей является нарушение или задержка в развитии форм социализированного

поведения, которое предполагает взаимодействие с людьми, учет их чувств, эмоций и реакций. Общими характеристиками РАС можно считать триаду аномалий в развитии, изображенной на рисунке 1.1 [5].



Рисунок 1.1 – Триада основных нарушений в развитии у детей с РАС

Основную массу детей с РАС составляют дети с синдромом раннего детского аутизма (РДА). Такие дети имеют тяжелые нарушения в социально-личностном развитии и нуждаются в психолого-педагогической помощи. Детский аутизм выражается стереотипным поведением, эмоциональной холодностью и потерей контакта с окружающими, что отрицательно воздействует на восприятие мира и развитие таких психических функций как речь и мышление [6].

Таким образом, данные отклонения приводят к задержке умственного, речевого и общего развития в целом, поэтому любая обучающая система должна, прежде всего, основываться на развитии коммуникативных способностей и навыков социального взаимодействия, стереотипном поведении ребенка.

### **1.2.2 История и этимология расстройств аутистического спектра**

Впервые термин «аутизм» (от греч. autos - сам) был введен швейцарским психиатром Эйгеном Блейлером. Изначально аутизм анализировался как особенность шизофренического мышления, характерной чертой которого было уход в себя и бегство от реальности.

Впоследствии данный феномен заинтересовал других исследователей. Так, еще один швейцарский психиатр Лео Каннер впервые описал аутизм как отдельное от шизофрении заболевание на примере наблюдения и изучения

одиннадцати детей в своей работе «Аутистические нарушения аффективного контакта» (1943 г.). Ученый предложил назвать расстройство «ранним детским аутизмом», так как эти особые состояния проявляются у детей на раннем этапе развития. Основными симптомами автор указывал раннюю самоизоляцию, ограниченность интересов, расстройство речи, поведения и моторики [5].

Наряду с исследованиями Каннера появляется научная работа австрийского педиатра и психиатра Ганса Аспергера, посвященная аутистической психопатии детей. Первичным отличительным признаком Аспергер считал нарушение социального взаимодействия, которое остается постоянным на протяжении всей жизни. Однако можно выявить различия в описании речевых способностей и развитии моторики авторами. Швейцарский ученый отмечал потерю коммуникативной функции у детей и двигательную ловкость мелкой моторики. В то же время Ганс Аспергер констатировал свободное владение языком своих подопечных, неуклюжесть и слаборазвитые навыки мелкой моторики. Описанные австрийским исследователем черты аутистической психопатии впоследствии были названы «синдромом Аспергера». Похожие исследования проводила советский детский психиатр Г.Е. Сухарева, дав описание поведенческому синдрому, который отличается высоким уровнем развития интеллектуальных способностей, но слабыми навыками социального взаимодействия и коммуникации.

В итоге научные работы вышеуказанных ученых стали основой для выделения новых диагнозов, которые составили более обширную классификацию нарушений аутистического спектра.

### **1.2.3 Особенности развития детей с расстройством аутистического спектра**

Исходя из выше сказанного, лица с расстройством аутистического спектра, прежде всего, испытывают трудности в эмоционально-волевой сфере. РАС проявляется на начальном этапе жизни человека, поэтому дети уже в раннем возрасте не могут поддержать или понять сущность социальных взаимоотношений, не могут участвовать в творческих играх, испытывают трудности в распознавании эмоций и поведенческих реакций других людей, не могут завести друзей. Во-вторых, дети с РАС имеют проблемы в социальной коммуникации. Например, они не могут поддержать разговор, сблизиться с другими людьми, не разделяют их интересы и эмоции. В дополнение, при аутизме присутствуют проблемы в невербальной коммуникации и поведении. Так, детям трудно поддерживать зрительный контакт, распознавать значение жестов и языка тела, речевой интонации и мимики. При этом деятельность, которая не касается взаимодействия в обществе, в том числе интеллектуальная деятельность, фантазирование, конструирование, может протекать на достаточно высоком уровне [7].



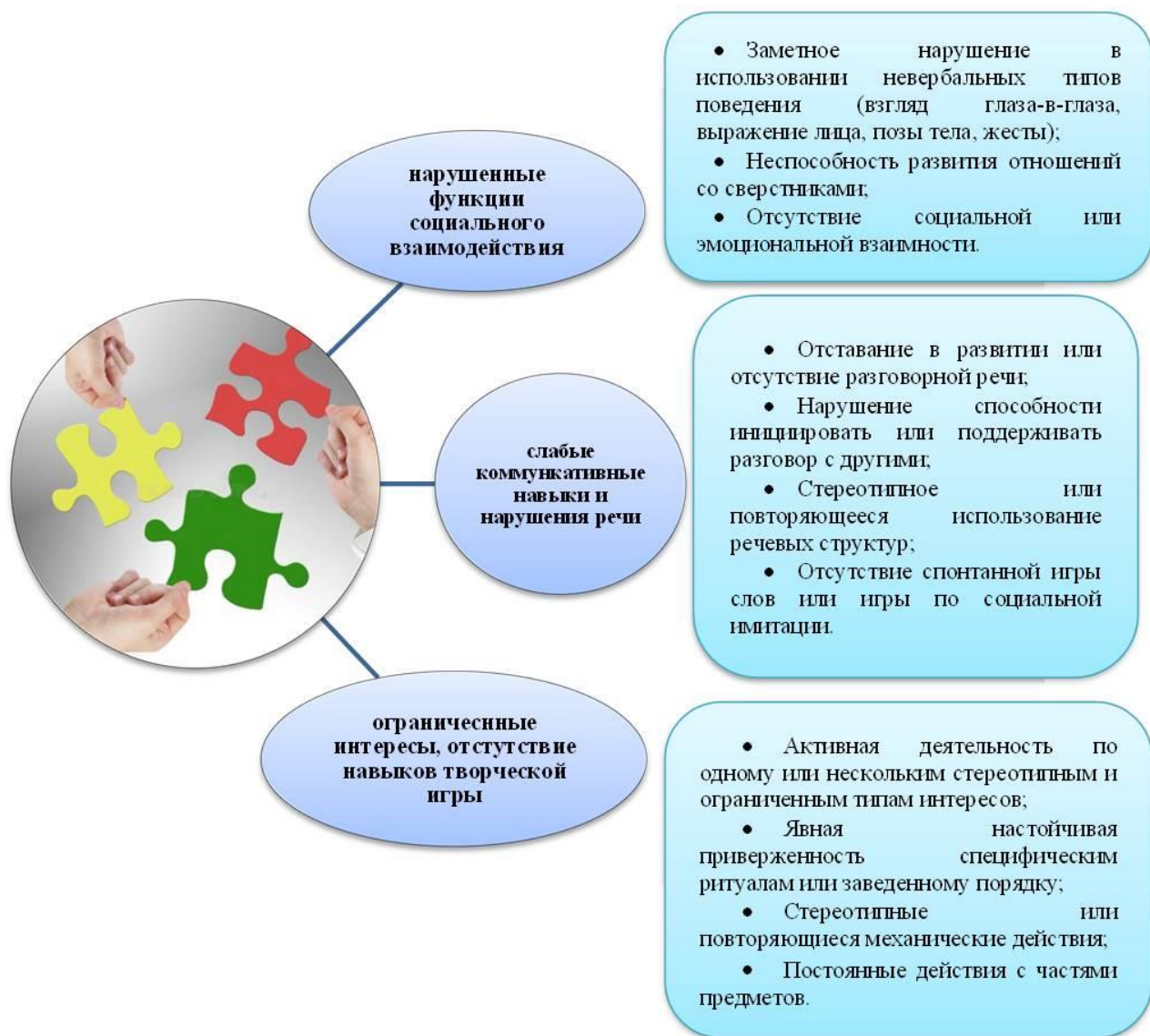


Рисунок 1.2 – Диагностические критерии согласно триаде нарушений при РАС

Первым характерным признаком наличия РАС у ребенка является нарушение социального взаимодействия. Несмотря на отсутствие совместного внимания, то есть они не привлекают внимание другого человека к объекту, дети могут использовать направленный взгляд для других целей. Например, они просят достать недостижимый предмет, при этом, не проявляя желания разделить внутренний опыт с другим человеком.

По результатам научных экспериментальных исследований Дж. Доусон аутичные дети затрудняются в имитировании мимики и телесных движений, ориентации на другого человека и проявлении эмпатии [8]. Кроме того, дети с аутизмом плохо ориентируются на слуховые сигналы, что свидетельствует о дефиците реагирования на людей и их поведение, о слаборазвитых эмоциональных связях с членами семьи.

Трудность опознавания эмоционального состояния является одним из главных нарушений социального воздействия. Проведение тестов на понимание эмоций (эмпатию) и реакции на эмоции других людей является одним из диагностических критериев детского аутизма и синдрома Аспергера [4]. Такое «безразличие» к эмоциям других интерпретируется, как неспособность воспринимать эмоциональную экспрессию других лиц и отвечать на нее.

На сегодняшний момент исследования отклонений в невербальном общении у детей с РАС описывают трудности установления глазного контакта, в том числе отсутствие неподвижного взгляда «сквозь» или «мимо», необычное использование жестов или отсутствие экспрессивной мимики. В целом, нарушения общения, как и нарушения социального взаимодействия, могут сильно варьироваться у индивидов с РАС, в зависимости от интеллектуальных способностей и возраста. Далее представим основные особенности нарушенного развития речи и коммуникативных навыков (рисунок 1.3).

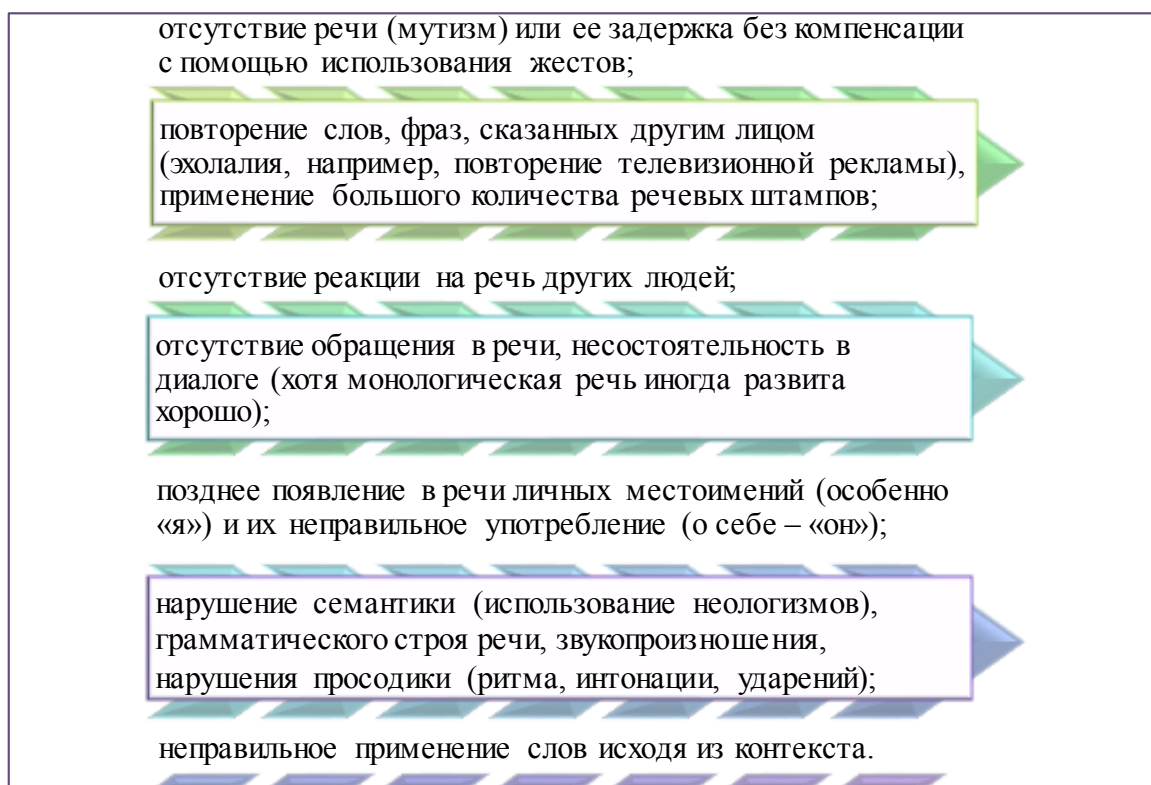


Рисунок 1.3 – Отличительные характеристики речи у детей с РАС

Последний элемент триады заключается в нарушенной способности к воображению и символической игре. Как мы знаем, ребенок в раннем возрасте умеет представлять несуществующие свойства объектов, например, готовить из глины или песка различные блюда, играть в доктора или представлять себя супергероем. Таким образом, ребенок заменяет один предмет другим. В тоже

время игры детей с РАС стереотипны и подчинены зрительной, слуховой или тактильной аутостимуляции - извлечение различных звуков из предметов за счет потряхивания или построение объектов в определённом порядке.

Стереотипные и механические повторяющиеся движения и действия связаны со стремлением сохранить привычные условия жизни. Самые распространённые примеры таких действий могут быть раскачивание телом, хождение кругом, открывание и закрывание тетради, повторение движений или фраз за людьми. Дети данной категории имеют определенные ритуалы, нарушение которых может привести к травмирующим изменениям в поведении. Результатом таких изменений может стать агрессивное и деструктивное поведение, жестокое обращение с животными или сверстниками, «уход в свой внутренний мир» [6].

С возрастом стереотипная игра обычно приводит к появлению у ребенка с аутизмом стереотипных увлечений, например, собирание пазлов или конструктора, увлечение флагами стран, схемами и т.п. Одной из удивительных форм проявления аутистического спектра является синдром Саванта. Несмотря на явные нарушения восприятия окружающего мира и их негативный эффект на работу мозга, дети с синдромом Саванта обладают уникальными талантами. При правильном подходе к обучению можно помочь развить одаренность или порой гениальность ребенка. Ученые выявляют поразительные интеллектуальные способности или знания в какой-либо сфере деятельности.

#### **1.2.4 Обучение детей с расстройством аутистического спектра**

Программы обучения и поддержки людей с РАС опираются на принцип преемственности на всех этапах жизни. Во всем мире сейчас внедряется и реализуется инклюзивное обучение - дети без и с ограниченными возможностями обучаются вместе. Основными условиями для эффективного инклюзивного обучения учащихся с расстройством аутистического спектра является, в первую очередь, квалифицированный педагогический состав, который владеет специальными методами и приемами, умеет использовать новые технологии, в том числе компьютерные обучающие системы.

Сегодня стратегия педагогического воздействия требует знания особенностей развития детей с аутизмом, постоянного контакта с родителями, адаптации программного материала, методики и окружающей среды для занятий, использование наглядных средств. При правильном подходе к обучению у детей должны успешно развиваться коммуникативные и социальные умения, навыки самообслуживания, если имеются, уникальные таланты и способности. На начальном этапе учебного процесса необходимо определить индивидуальные возможности и способности аутичного ребенка, применяя параметры, определенные на рисунке 1.4

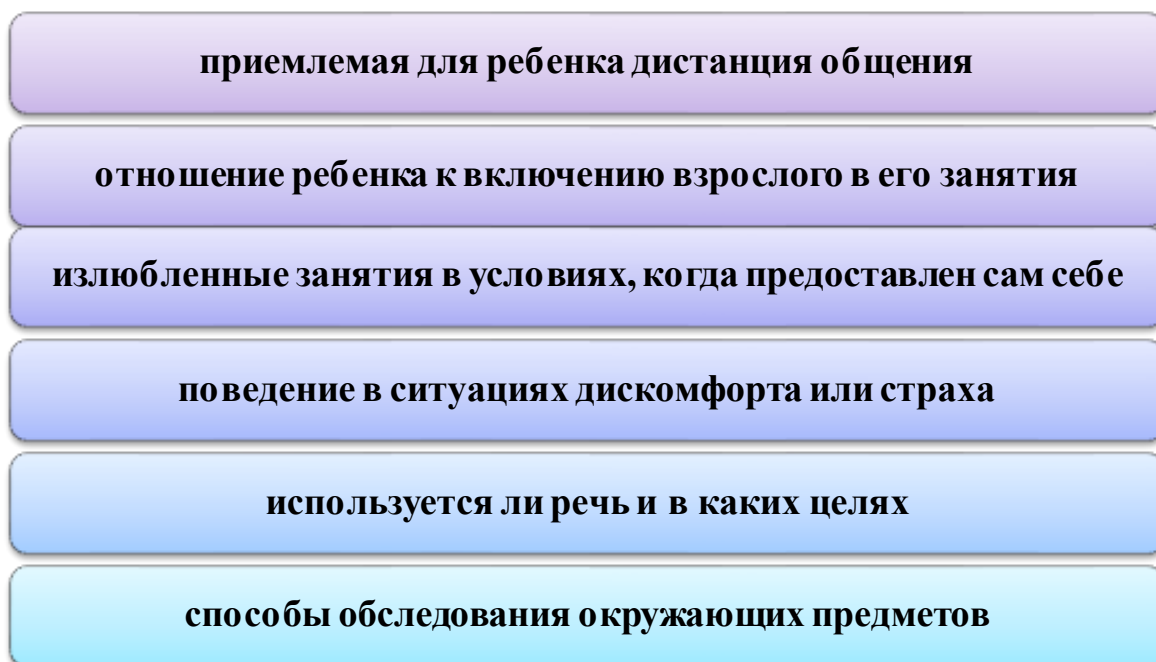


Рисунок 1.4 – Параметры определения индивидуальных способностей и возможностей детей с РАС

Первостепенным в общении с аутичным ребенком является личное отношение к нему: необходимо обращаться непосредственно к нему, использовать условные знаки (кивок головы, улыбка), обговорённые с ребенком заранее, постоянно повторять задания для такого ученика. Таким образом, ребенок постепенно осваивает социальную роль учащегося. Учитывая все особенности проявления аутистического спектра, можно представить следующую схему с педагогическими советами и приемами во время обучения детей с РАС (рисунок 1.5) [9].

<ul style="list-style-type: none"> <li>- может играть, выходить в сопровождении из комнаты;</li> <li>- разрешается манипулировать другими предметами во время занятия.</li> </ul>	<ul style="list-style-type: none"> <li>- задания выполняются постепенно без давления со стороны тьютора;</li> <li>- необязательно следовать правилам задания.</li> </ul>	<ul style="list-style-type: none"> <li>- используется спокойное освещение, музыкальное сопровождение в начале и в конце занятия, физ.развминку иногда.</li> </ul>
<ul style="list-style-type: none"> <li>- ребенку оказывается помощь в выполнении заданий, убеждая, что он уже может это делать, используются различные поощрения.</li> </ul>	<ul style="list-style-type: none"> <li>- используются жесты и наглядность, в том числе картинки с предметами или событиями, компьютерные программы, цветотерапия (буквы разных цветов для улучшения эмоционального состояния)</li> </ul>	<ul style="list-style-type: none"> <li>- стимулирование интеллектуальных интересов;</li> <li>- ребенок учится осознанно использовать память; - активизируются способности в математике и лингвистике, в т.ч. к иностранным языкам.</li> </ul>
<ul style="list-style-type: none"> <li>- поощряются короткие и осмысленные ответы;</li> <li>- ребенок учится устанавливать связь знаний с навыками, используя специальные карточки; произносить монологи через пересказ книги и фильмов, учит правила диалога.</li> </ul>	<ul style="list-style-type: none"> <li>- тьютор стимулирует произвольное внимание;</li> <li>- часто обращается по имени, контактируя глазами;</li> <li>- поглаживает по плечу, спине.</li> </ul>	<ul style="list-style-type: none"> <li>- развитие навыков чтения при помощи худ. литературы;</li> <li>- тьютор помогает понять чувства и внутренний мир героев, ассоциировать качества и эмоции героев со своими собственными и других людей.</li> </ul>

Рисунок 1.5 – Педагогические советы и приемы во время обучения детей с РАС

Кроме того, педагог и психолог всегда должны работать вместе, своевременно оказывать дополнительную помощь ребенку. Она необходима для назначения подходящего ребенку режима приобретения учебных умений и навыков. Трудности в обучении обычно связаны с прошлыми неудачными учебными попытками, нетерпеливостью родителей и давлением со стороны тьютора [10]. Первой предпосылкой преодоления этих сложностей является успешный опыт учебных достижений ребенка с расстройством аутистического спектра.

В следующей главе рассмотрим более подробно существующие компьютерные обучающие программы для детей с расстройством аутистического спектра.

### 1.3 Анализ существующих обучающих систем для детей с РАС

Расстройство аутистического спектра (РАС) описывается как расстройство нервно-психического развития, возникающее в социальной среде, характеризующейся неспособностью к общению и взаимодействию, а также ограниченным повторяющимся поведением и интересами. Аутизм также характеризуется когнитивными нарушениями, речевыми затруднениями и неспособностью к творческой деятельности. Настойчивое стремление ребенка к одинаковости является основной чертой аутизма. По сравнению с людьми, компьютеры - это инструменты, которые, как правило, обеспечивают такое сходство. Это основная причина, по которой дети с РАС предпочитают в общении механические предметы, а не людей. Следовательно, интерес аутичных детей к механическим предметам позволяет использовать компьютеры.

С внедрением технологий в образовательную жизнь использование компьютеров и интернета стало считаться полезным для поддержки когерентного подхода к детям с РАС, так как компьютерные технологии могут помочь детям с аутизмом в их функциональных недостатках, повысить их способности к обучению, позволят им быть независимыми и облегчат их общение.

Исследователи зарубежных и российских научных институтов выдвинули гипотезу о том, что использование информационно-коммуникационных технологий может значительно повысить эффективность обучения детей с РАС. В связи с этим был проведен ряд исследований, чтобы оценить, насколько достоверна данная гипотеза. Например, в Швеции был проведен научный эксперимент по обучению детей навыкам устной речи и чтения с помощью мультимедийно-интерактивной программы для компьютера. Исследование проводилось с участием 30 детей, из которых: 11 детей с РАС (средний возраст 9 лет), 9 детей, страдающих заболеваниями разнообразной этиологии (средний возраст 13 лет) и 10 нормальных дошкольников (средний возраст 6 лет). Результаты исследования показали, что контрольные показатели детей с РАС по фонологическим тестам и тестам чтения заметно улучшились в процессе компьютерного обучения, однако данный эффект не имел продолжительного действия после завершения процесса обучения и был крайне неустойчивым [11].

По сравнению с другими средствами обучения компьютерная программа позволяет реализовать такие значимые для образовательного процесса детей с РАС аспекты как:

- 1) умение работать «со смыслами»;
- 2) обеспечение ясности и наглядности;
- 3) дифференциация, поскольку каждый ребенок уникален.

В дополнение, планшет со специальным программным обеспечением может стать отличным инструментом развития навыков общения у детей с РАС, которым трудно установить контакт с окружающими, в особенности с

близкими людьми. С помощью таких образовательных технологий человек с аутизмом чувствует себя увереннее и спокойнее, ему не стоит бояться вопросов, на которые он не сможет ответить или беспокоиться о своем поведении и. Специально разработанные программы направлены на то, чтобы вывести ребенка с РАС из своей зоны комфорта.

Исследование существующего программного обеспечения для обучения детей с расстройством аутистического спектра позволило выделить следующие особенности внедрения в учебный процесс таких программ:

1) ПО, как правило, проприетарное, т.е. является частной собственностью авторов или правообладателей и требует финансовых инвестиций;

2) ПО применяется без методического сопровождения, снижая эффективность его применения;

3) ПО реализуется только на иностранных языках, т.к. большинство программ разработано за рубежом.

Американский психолог Кенетт Колби и компьютерный специалист Дэвид Смит подчеркивают важность использования компьютеров для детей с РАС. Компьютеры никогда не устают, они предсказуемы, всегда говорят одно и то же в одной и той же манере, никогда не сердятся и не скучают. К.Колби использовал компьютер для обучения детей, испытывающих речевые трудности. Он разработал игру, в которой используются символы в сочетании с человеческим голосом, которые появляются на экране при нажатии кнопок на клавиатуре. Исследование проводилось с участием 17 детей, испытывающих трудности с речью. Он проводил получасовые учебные занятия, и после 50 таких занятий наблюдались значимые улучшения в области развития речи. В этом обучении первостепенное значение придавалось использованию четких выражений и используемых предложений в ежедневном общении [12].

Кроме того, в поддержку развития социальных навыков Лизабет Эскобедо, Дэвид Нгуен и другие в 2012 году разработали вспомогательное приложение под названием MOSOCO (Mobil Social Compass) (рисунок 1.6). Данное приложение использует возможности дополненной реальности и визуальной поддержки, что позволяет детям с РАС практиковать социальные навыки в реальных ситуациях [13].



Рисунок 1.6 – Приложение MOSOCO

Исследование проводилось с участием 12 детей с диагнозом РАС. В ходе исследования особое внимание уделялось таким социальным навыкам, как установление зрительного контакта с учеником, общение со сверстниками и беседа о себе. Результаты исследования показывают, что MOSOCO способствует обучению и реализации социальных навыков, а также усиливает качественное и количественное социальное взаимодействие.

На территории СНГ ресурсов, ориентированных на детей с расстройством аутистического спектра, практически нет в свободном бесплатном доступе. АутизмСофт – первая компания в России, занимающаяся разработкой специализированного программного обеспечения для людей с РАС, была создана в 2013 году. ПО компании разрабатывается преимущественно для планшетов. Разработчики компании подчеркивают, что они стараются создавать свои программы, учитывая потребности будущих пользователей. В данный момент компания выпустила два программных продукта: «Визуальное расписание» и «Аутизм коммуникатор» [14].

«Аутизм коммуникатор» - приложение для детей с аутизмом, которое обеспечивает коммуникацию с другими людьми с помощью специальных карточек (рисунок 1.7). Приложение позволяет создавать собственный язык альтернативного общения, используя карточки, фотографии и даже записи голоса. Дети могут общаться с другими людьми, составляя фразы из картинок и предъявляя экран планшета собеседнику. Коммуникатор позволяет выстроить общение на основе тех понятий, которые известны и доступны ему непосредственно в данный момент, давая возможность добавления новых слов, сохраняя их в визуальном языке.





Рисунок 1.7 – Приложение «Аутизм коммуникатор»

«Визуальное расписание» - планировщик, впервые разработанный на русском языке, который отвечает нуждам детей с расстройством аутистического спектра (рисунок 1.8). Приложение оснащено яркими иллюстрациями, в том числе: фотографиями и картинками. Такое расписание позволяет определить четкую структуру последовательности событий и помогает снять тревогу ребенка перед будущим и способствует выработке определенного режима.

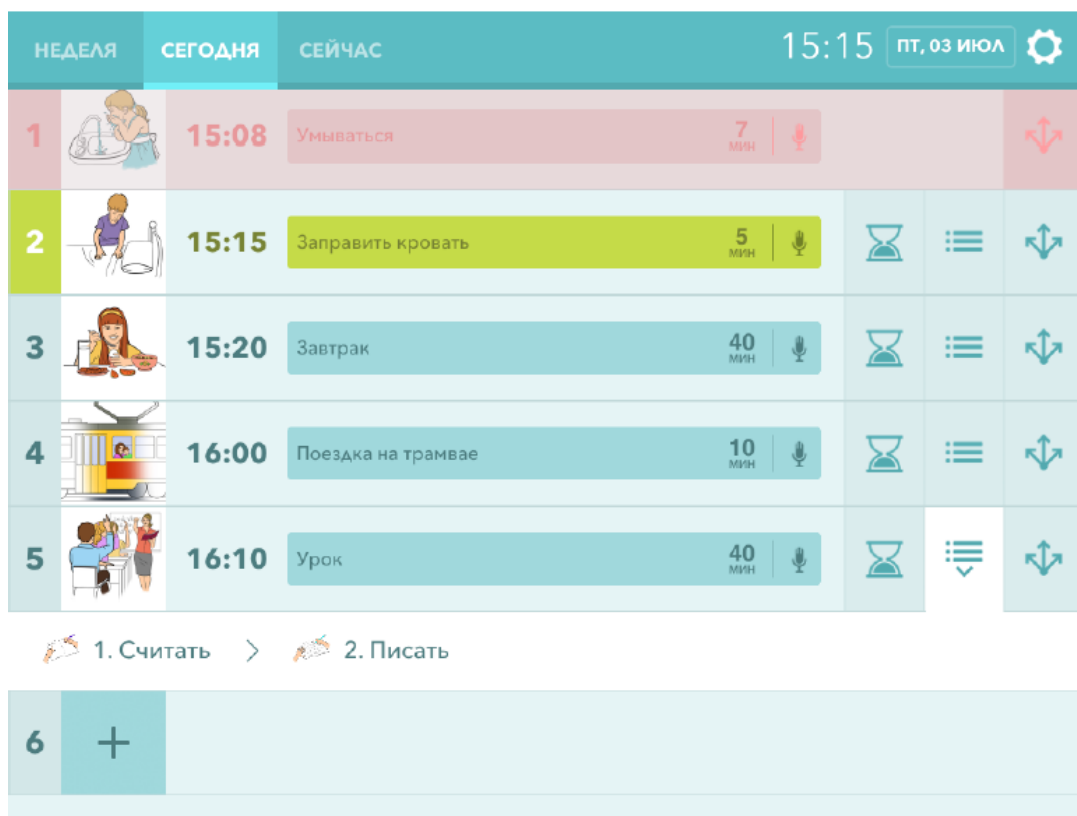


Рисунок 1.8 – Приложение «Визуальное расписание»

Недостаток приложений в том, что они ориентированы на платформу iOS, а также оба продукта являются платными. В конце июля 2015 года «Аутизм коммуникатор» был перенесен на платформу Android, оставаясь при этом платным.

Бесплатной альтернативой этих продуктов может служить уникальное приложение, разработанное в 2015 году международной благотворительной организацией «Дети. Аутизм. Родители» и компанией Velcom. Они представили мобильное приложение «Коммуникатор ДАР», которое позволяет обмениваться фразами с помощью пиктографии (рисунок 1.9). Приложение позволяет использовать для общения графические символы (картинки), понятные любому человеку. Каждое изображение обозначает какое-то понятие (эмоции, движения и т. д.) и имеет голосовое сопровождение [15].



Рисунок 1.9 – Приложение «Коммуникатор ДАР»

Еще одним бесплатным приложением является коммуникатор «Говори Молча: Аутизм Диалог» для детей, испытывающих проблемы с речью (аутизм, синдром Дауна, синдром Аспергера и др.) (рисунок 1.10). Приложение разработано IT-компанией Indigo Kids, которая специализируется на разработке образовательных и развивающих программ для детей с особыми потребностями [14]. Программа доступна для устройств на базе Android. Используя приложение, ребенок может выбирать желаемые предметы, обучаться общению и составлять сообщения и предложения-просьбы. Программа основана на визуальных карточках PECS, которые отображают базовые потребности детей. Особенность приложения в том, что имеется возможность отправки набранного предложения по смс и возможность добавления своих собственных слов. Исходя из отзывов, недостаток коммуникатора в том, что используемые карточки отличаются от стандартных карточек PECS внешне и по функционалу.



Рисунок 1.10 – Интерфейс приложения «Говори Молча: Аутизм Диалог»

В 2015 году группа российских разработчиков выпустила первый мессенджер на русском языке под названием «Сезам» (рисунок 1.11). «Сезам» – удобная и простая программа для детей и взрослых с нарушениями речи или письма, а также с такими расстройствами, как ДЦП, аутизм, синдром Дауна и другие. Программа позволяет обмениваться специальными пиктограммами, с помощью которых также можно составлять отдельные предложения. Для реализации такой альтернативной коммуникации используется социальная сеть «В контакте». Это открывает новые возможности для знакомств, общения и социальной адаптации. Стартовая версия приложения доступна для устройств, работающих на базе ОС Android, но предпочтительно использование на планшетах.



## 2 Проектная часть

### 2.1 Описание предметной области

Разработанная программа будет представлять собой приложение с оконным интерфейсом с различными вкладками и компонентами, внутри которых будет отображаться нужная информация. Вкладки будут разделяться по тематикам, а внутри вкладок будут находиться карточки PECS, отображающие базовые потребности детей и оснащенные звуковым сопровождением, которые будут использоваться в качестве средства коммуникации и взаимодействия. Основная направленность приложения будет на предоставление возможности ребёнку взаимодействовать с другими людьми посредством данных карточек, с возможностью изучения новых слов и способов коммуникации. Также будет возможность изучения базовых эмоций людей, которое затем закрепляется небольшой викториной. Доступ ко всем вкладкам и окнам будет осуществляться из главного меню.

### 2.2 Обоснование выбора языка программирования

Для разработки программы был выбран язык программирования C#, который является одним из популярных современных языков программирования. C# (произносится как "си шарп") - это объектно-ориентированный и типобезопасный язык программирования, разработанный компанией Microsoft. Он соответствует современным стандартам программирования и разрабатывался специально для поддержки развития технологии .NET Framework. Язык программирования был разработан в конце 1990-х годов группой инженеров во главе со Скоттом Вильтаумотом и Андерсом Хейлсбергом в компании Microsoft как основной язык разработки приложений для платформы Microsoft .NET (программной платформы от компании Microsoft, предназначенной для создания обычных программ и веб-приложений) [16, 17, 18].

Символ # (октоторп) в названии языка имеет несколько вариаций расшифровки: данный символ печатается на клавиатуре как Shift+3, что также может символизировать третью реализацию C. Кроме того, # можно интерпретировать как две пары плюсов ++++ - что может означать новый шаг в развитии языка по сравнению с C++.

Инструментарий C# позволяет решать широкий круг задач, поэтому язык используется при разработке:

- веб-приложений;
- различных игровых программ;
- мобильных приложений для Android и IOS;
- программ для ОС Windows.

Язык C# имеет много полезных функций, таких как:

- полиморфизм;
- инкапсуляция;
- наследование;

- события;
- атрибуты;
- свойства;
- итераторы;
- перегрузка операторов;
- обобщённые типы и методы;
- статическая типизация;
- исключения;
- комментарии в формате XML;
- сборка мусора.

C# будучи объектно-ориентированным языком программирования поддерживает все три его основных составляющих: инкапсуляцию, полиморфизм и наследование.

Инкапсуляция – механизм, который связывает вместе код и данные, которыми он манипулирует. Данный механизм позволяет ограждать этот код и данные от неправильного использования и внешнего доступа. То есть, это контроль доступа к полям и методам объекта с помощью модификаторов доступа: `private`, `public` и `protected`.

Полиморфизм – свойство системы, позволяющее нескольким объектам с общими характеристиками иметь доступ к одному интерфейсу. Обобщенно концепцию полиморфизма следует выразить так: «один интерфейс – множество методов» [18].

Наследование – такое свойство, с помощью которого один объект может приобретать свойства другого. То есть, появляется возможность наследовать одними классами свойства и поведение других классов.

Язык C# связан с такими языками программирования, как C, C++ и Java. В C++ была определена объектная модель, на которой теперь строится язык C#. Связь с языком C основывается на унаследованном от него синтаксисе, операторах и ключевых словах. Между C# и Java прямой связи нет, но оба языка были разработаны, чтобы создать специальный переносимый код. Данный код базируется на C и C++, используя их синтаксис и объектную модель. Владение навыками того или иного языка облегчит переход на любой из них.

Наиболее важная отличительная черта языка программирования C# – способность работать в многоязыковом окружении. Под этим понимается способность кода, написанного на разных языках, работать совместно, что является важной способностью при создании больших программ.

*У языка довольно много преимуществ:*

- поддержка большинства продуктов Microsoft;
- использование объектно-ориентированного подхода;
- наличие большого количества библиотек и шаблонов;
- типы данных имеют фиксированный размер: 32-битный `int` и 64-битный `long`, что повышает «мобильность» языка и упрощает программирование;

- схожесть синтаксиса языка с другими языками программирования, что облегчает переход для программиста (например, с Java или C++);
  - огромное количество специальных готовых конструкций («синтаксический сахар»), разработанных для понимания и написания кода, которые не имеют значения при компиляции;
  - автоматическая «сборка мусора». Это облегчает работу с освобождением памяти, так как общезыковая среда CLR сама вызовет сборщик мусора и очистит память;
  - возможность создавать кроссплатформенные приложения. С помощью Xamarin на C# можно писать программы и приложения для таких операционных систем, как iOS, Android, MacOS и Linux;
  - добавлено функциональное программирование (F#).
- Однако имеются и недостатки:*
- ориентированность, в основном, только на .NET (на Windows платформу);
  - язык бесплатен только для небольших компаний или групп учащихся. Крупной компании покупка лицензионной версии этого языка обойдётся недешево.

### **2.3 Среда .NET Framework**

Среда, в которой работает язык программирования C#, называется NET.Framework, и поэтому лучше всего рассматривать их во взаимосвязи. На это есть несколько причин:

- 1) C# изначально разрабатывался компанией Microsoft для создания кода .NET Framework;
- 2) сама среда .NET Framework определяет библиотеки, которые используются C#.

Так что же такое .NET Framework? Отвечая на вопрос, можно сказать, что .NET Framework – это среда, которая поддерживает развитие и выполнение платформонезависимых приложений. Платформа позволяет совместное использование в приложении различных языков программирования, а также может обеспечивать общую модель программирования для Windows и переносимость программ. Однако среда .NET Framework не ограничена только платформой Windows, а написанные программы могут быть перенесены на другие платформы [18].

.NET включает в себя как среду выполнения, так и основную библиотеку классов, используемую программами C#.

Часть среды выполнения называется общезыковой средой выполнения (обычно сокращенно CLR от Common Language Runtime), потому что данная среда поддерживает не только C#, но и любой язык .NET. Например, Microsoft также предлагает расширения Visual Basic, F# и .NET для C++. CLR имеет общую систему типов (CTS), которая позволяет коду из нескольких языков свободно взаимодействовать, что означает, что библиотеки .NET обычно можно использовать с любого языка .NET. Например, F# может использовать



библиотеки, написанные на C#, а C# может использовать библиотеки Visual Basic и так далее.

Помимо среды выполнения, существует обширная библиотека классов. Эта библиотека предоставляет программе доступ к среде исполнения, например, используя ее для ввода и вывода данных. Программа может быть запущена везде, где поддерживается среда исполнения .NET до тех пор, пока она ограничена характеристиками, определенными библиотекой классов .NET.

Библиотека классов предоставляет обертки для многих функций базовой операционной системы (ОС), но она также предоставляет значительное количество собственных функциональных возможностей, таких как классы коллекций или обработка JSON.

Многие другие системы предоставляют свои собственные библиотеки .NET. Например, существуют обширные библиотеки, которые позволяют программам C# использовать популярные облачные сервисы. Как и следовало ожидать, Microsoft предоставляет комплексные библиотеки .NET для работы со службами в своей облачной платформе Azure. C# предлагает различные механизмы для работы с другими видами API, такими как API в стиле C, доступные в Win32, macOS и Linux, или API, основанные на компонентной объектной модели (COM) в Windows. Таким образом, с помощью C# мы получаем сильный набор абстракций, встроенных в язык, а именно: мощная среда выполнения, легкий доступ к огромному количеству библиотек и платформа функциональности.

В течение многих лет наиболее распространенным способом работы компилятора была обработка исходного кода и получение выходных данных в форме, которая может быть выполнена непосредственно процессором компьютера. Компиляторы производили машинный код-серии инструкций в любом двоичном формате, который требовался для того типа процессора, которым располагал компьютер. Многие компиляторы все еще работают таким способом, но компилятор C# этого не делает. Вместо этого он использует модель, называемую управляемым кодом.

При использовании управляемого кода компилятор не генерирует машинный код, выполняемый процессором. Вместо этого компилятор создает форму двоичного кода, называемую промежуточным языком Microsoft (Microsoft Intermediate Language, MSIL). Исполняемый двоичный файл создается позже, обычно во время выполнения. Использование MSIL позволяет использовать функции, которые трудно или даже невозможно обеспечить в рамках более традиционной модели.

Возможно, наиболее заметным преимуществом управляемой модели является то, что выходные данные компилятора не привязаны к одной архитектуре процессора. Вы можете написать компонент .NET, который может работать на 32-разрядной архитектуре x86, используемой ПК в течение десятилетий, но он также будет хорошо работать в новом 64-разрядном обновлении этого дизайна (x64) и даже на совершенно других архитектурах,

таких как ARM. (Например, .NET Core представила возможность запуска на устройствах на базе ARM, таких как Raspberry Pi). С помощью .NET вы можете скомпилировать один компонент, который может работать на любом из них, и он даже сможет работать на платформах, которые не поддерживались во время компиляции кода, если в будущем появится подходящая среда выполнения. В более общем плане любое улучшение генерации кода CLR - будь то поддержка новых архитектур ЦП или просто повышение производительности существующих - мгновенно приносит пользу всем языкам .NET. Например, более старые версии CLR не использовали преимущества расширений векторной обработки, доступных на современных процессорах x86 и x64, но текущие версии теперь часто используют их при создании кода для циклов. От этого выигрывает весь код, запущенный в текущих версиях .NET Core, включая код, написанный за несколько лет до добавления этого улучшения.

Точный момент, в который CLR генерирует исполняемый машинный код, может варьироваться. Как правило, он использует подход, называемый just-in-time компиляцией (JIT), в котором каждая отдельная функция компилируется при первом запуске.

Управляемый код имеет повсеместную информацию о типе. Форматы файлов, диктуемые CLI, требуют, чтобы это присутствовало, потому что это включает определенные функции среды выполнения. Например, .NET предлагает различные службы автоматической сериализации, в которых объекты могут быть преобразованы в двоичные или текстовые представления их состояния, а затем эти представления могут быть возвращены в объекты, возможно, на другой машине.

Платформа .NET Framework предоставляет приложениям следующие службы (услуги):

- управление памятью. В приложениях платформы .NET Framework среда CLR предоставляет сервисы для управления ресурсами памяти автоматически;

- система общего типа. Система CTS (Common Type System) определяет базовые типы в платформе .NET Framework. К тому же, используются одни и те же базовые типы для всех языков .NET Framework;

- совместимость версий. Приложения определенной версии, могут выполняться без изменений на более поздней версии;

- взаимодействие языков. Языковые компиляторы .NET Framework компилируют приложение в промежуточный код, CIL (Common Intermediate Language), который затем компилируется во время исполнения приложения средой CLR. Следовательно, программы, написанные на одном языке, доступны в других языках;

- расширенная библиотека классов. Облегчает написание большого кода для стандартных операций, благодаря использованию легкодоступной библиотеки типов и членов из библиотеки классов .NET Framework;

- параллельное выполнение. .NET Framework помогает в разрешении конфликтов версий, разрешая установку нескольких версий среды CLR на одном компьютере.

При написании C#-программы разрабатывается так называемый управляемый код. Он всегда реализуется управляемой средой выполнения (CLR), а не непосредственно выполняется операционной системой. Управляемая среда выполнения предоставляет различные типы служб, такие как сбор мусора, проверка типов, обработка исключений, проверка границ и т. д. без вмешательства программиста. Он также обеспечивает выделение памяти, безопасность типов и другое для кода. Приложение пишется на языках Java, C#, VB.Net и т.д., так как они всегда нацелены на службы среды выполнения для управления, и код, написанный на этих типах языков, известен как управляемый код.

В случае .NET Framework компилятор всегда компилирует код управления на промежуточном языке (MSIL), а затем создает исполняемый файл. Когда программист запускает исполняемый файл, то Just In Time компилятор CLR компилирует промежуточный язык в машинном коде, который является специфичным для базовой архитектуры. Здесь этот процесс происходит в управляемой среде выполнения во время выполнения, поэтому эта среда отвечает за работу кода.

Код, который непосредственно выполняется операционной системой, называется неуправляемым кодом. Он всегда нацелен на процессорную архитектуру и зависит от архитектуры компьютера. Когда этот код компилируется, он всегда стремится получить определенную архитектуру и всегда работает на этой платформе, другими словами, всякий раз, когда вы хотите выполнить один и тот же код для другой архитектуры, вы должны перекомпилировать этот код снова в соответствии с этой архитектурой. Он всегда компилируется в машинный код, специфичный для данной архитектуры.

В неуправляемом коде распределение памяти и безопасность, в том числе безопасность типов, управляются разработчиком. Из-за этого возникает несколько проблем, связанных с памятью, таких как переполнение буфера, утечка памяти, переопределение указателя и т.д. Исполняемые файлы неуправляемого кода обычно находятся в двоичных образах, код x86, который непосредственно загружается в память. Приложение, написанное на VB 6.0, C, C++ и т. д., всегда находится в неуправляемом коде.

## **2.4 Интерфейс программирования Windows Forms**

Приложение разрабатывалось в виде оконного пользовательского интерфейса, который реализовывался при помощи методов визуального программирования и классов библиотеки Windows Forms.

Windows Forms представляет собой интерфейс программирования приложений (API), который используется для создания приложений, оснащенных графическим интерфейсом. Windows Forms является частью

Microsoft .NET Framework. Интерфейс упрощает доступ к визуальным компонентам Microsoft Windows за счет того, что создает обертки для существующего Win32 API в управляемом коде [19]. Управляемый код не зависит от языка разработки, что позволяет использовать Windows Forms как при написании программного обеспечения на C#, C++, так и на VB.Net, J# и других языках.

Основной класс Windows Forms – форма, экземплярами которого являются главные и диалоговые окна. Форма – экранный объект, представляемый в виде прямоугольной формы, использующийся в основном для предоставления информации пользователю или для обработки вводимой информации пользователем. Они могут иметь вид многодокументного интерфейса (MDI), стандартного диалогового окна и также поверхность для отображения графической информации [20].

Формы являются потомками класса Form, определенного в пространстве имен System.Windows.Forms. Интерфейс Windows Forms позволяет работать в режиме конструктора, добавляя элементы управления (кнопки, поля для ввода текста, поля для отображения текста, меню и прочие компоненты) простым «перетаскиванием» на поверхность формы. В коде они представлены как поля класса формы. Все элементы управления окна являются объектами классов, содержащихся в System.Windows.Forms и являющихся потомками базового класса Control. Ответная реакция программы на определенное действие (событие), связанное с элементом управления – обработчик события – оформляется в виде метода формы.

Некоторые элементы управления, которые можно располагать на формах:

- Button (Кнопка);
- Label (Надпись);
- Panel (Панель);
- CheckBox (Флажок);
- ListBox (Список);
- RadioButton (Переключатель);
- Menu (Меню);
- Toolbar (Панель инструментов);
- TabControl (Управление вкладками);
- DataGrid (Сетка данных);
- TreeView (Дерево);
- MessageBox (Окно сообщений);
- PictureBox (Изображение).

## 3 Реализация приложения

### 3.1 Структурное представление программы

Реализация приложения начинается с определения структуры будущего программного продукта. В первую очередь, это нужно для того, чтобы иметь представление о составе программы: наличие структурных компонентов окон программы, наличие подсистем, что поможет облегчить разработку, а также функции каждого компонента программы. В качестве структурных компонентов принимаются дополнительные вкладки, которые содержат необходимую для использования информацию.

Структурная схема отображает состав и взаимодействие по управлению частями разрабатываемого программного продукта. Структурная схема программы, отражающая специфику построения интерфейса приложения, представлена на рисунке 3.1. Переключение между вкладками не требует открытия нового окна, все действия выполняются в главном окне, что значительно повышает скорость работы приложения и облегчает пользование. Интерфейс выглядит просто и понятно.

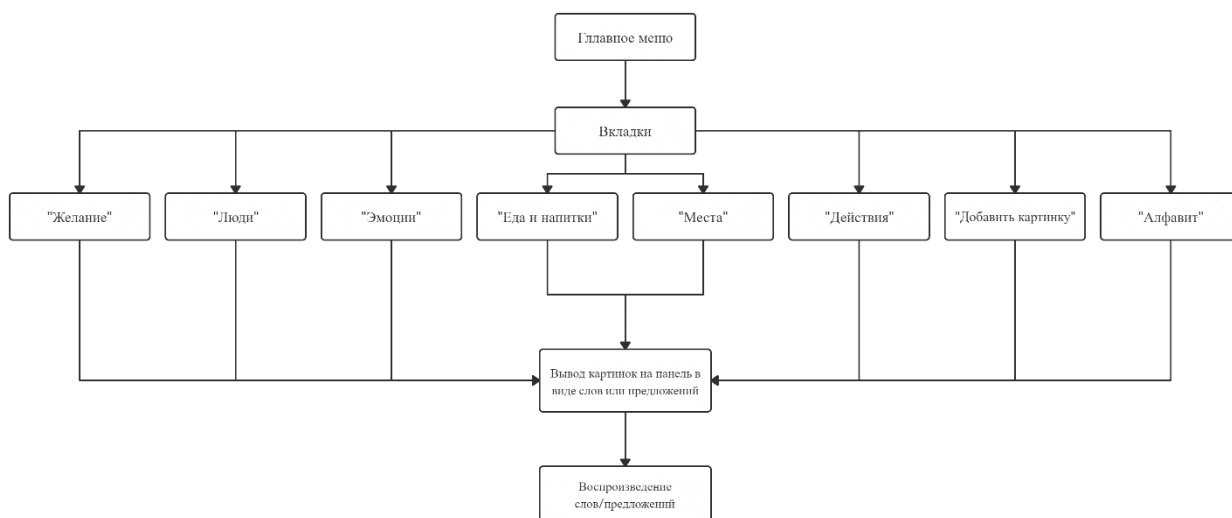


Рисунок 3.1 – Структурная схема программы

Каждая вкладка содержит определенные карточки по своей тематике, которые используются как средство коммуникации: можно составлять предложения/словосочетания, которые затем озвучиваются. При открытии любой из вкладок будет появляться дополнительная панель, скрытая на главном меню. Данная панель будет использоваться для отображения выбранных карточек и воспроизведения их.

### **3.2 Построение UML-диаграмм вариантов использования**

При проектировании программного продукта необходимо учитывать его специфику и предназначение. Для графического представления поведения системы используется так называемый UML-язык.

UML (Unified Modeling Language – унифицированный язык моделирования) – это визуальный язык, который используется для спецификации, визуализации, конструирования и документирования программных систем [22].

Диаграммы прецедентов (или вариантов использования) – один из пяти типов диаграмм, используемых в языке UML для моделирования вида системы с точки зрения внешнего наблюдателя. Они позволяют визуализировать поведение системы, подсистемы или класса для того, чтобы пользователи понимали, как их использовать. Диаграммы представляют совокупность прецедентов (вариантов использования) и актеров и отношения между ними.

Актер – это любая сущность, которая взаимодействует с системой извне. В нашем случае роль актера выполняет пользователь программы.

Прецедент (вариант использования) – описание последовательности действий, которые выполняются системой для того, чтобы привести актера к желаемому результату.

На рисунке 3.2 приводится диаграмма вариантов использования программного продукта.

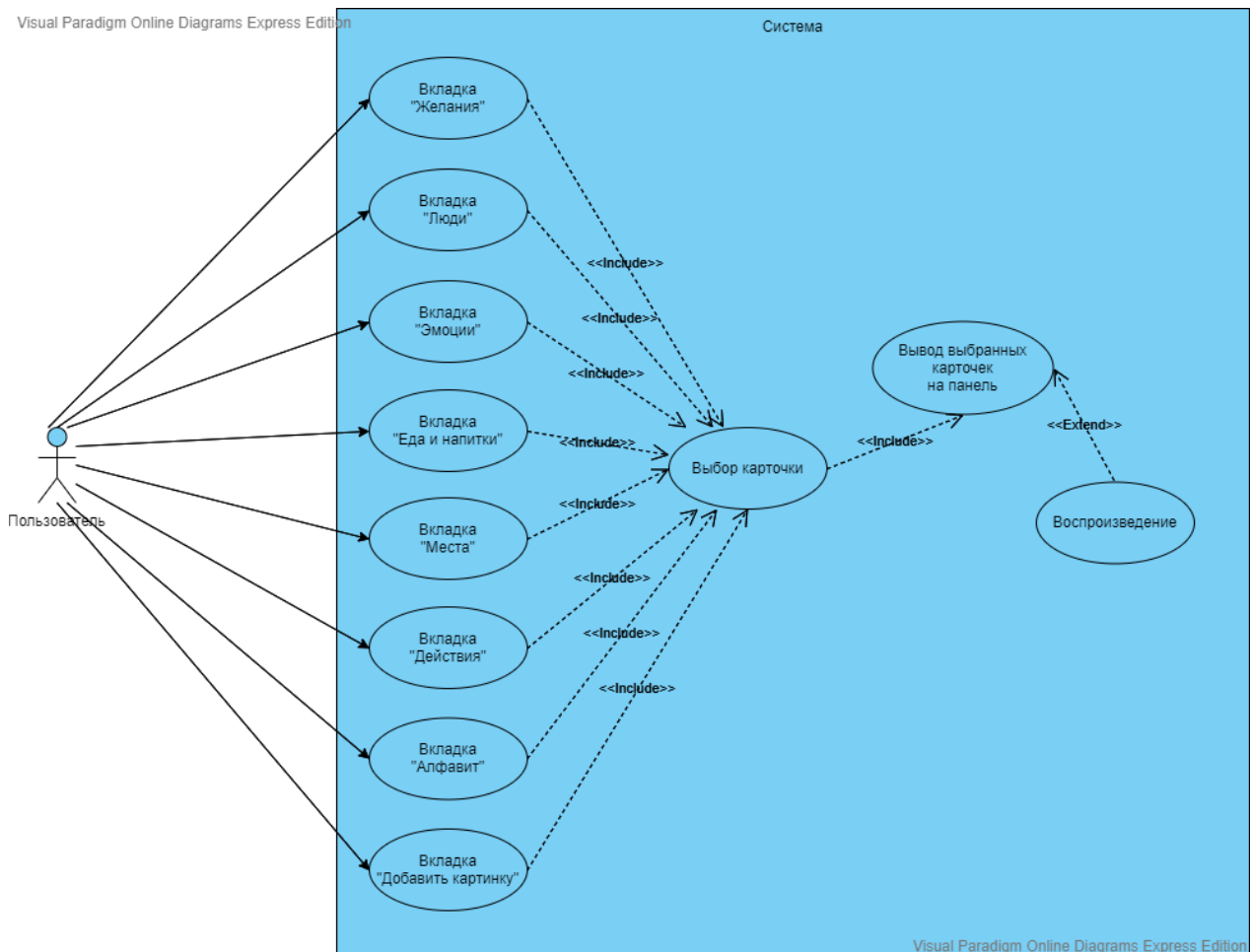


Рисунок 3.2 – Диаграмма вариантов использования

Сплошные линии обозначают отношения ассоциации, которые отражают возможность использования актером прецедента (варианта использования). То есть пользователь, запустив приложение, может переключать вкладки, которые содержат специальные карточки.

Пунктирными линиями обозначают отношения расширения (extend) или включения (include), которые возникают между прецедентами.

Отношение включения (include) появляется, когда поведение одного прецедента включается в некоторой точке в другой прецедент и становится составным компонентом. Включаемый прецедент становится обязательным для дополняемого. В нашем случае «Выбор карточки» становится составным компонентом.

Отношение расширения (extend) изображает возможное присоединение одного варианта использования к другому в некоторой точке. Расширяющий вариант использования выполняется только при определенных условиях и может быть не обязательным для выполнения основного прецедента. На рисунке 3.2 расширяющим элементом будет «Воспроизведение», которое не является обязательным по отношению к основному прецеденту – «Вывод выбранных карточек на панель».

Вариант использования вкладки «Добавить картинку» можно представить в виде UML-диаграммы, изображенной на рисунке 3.3

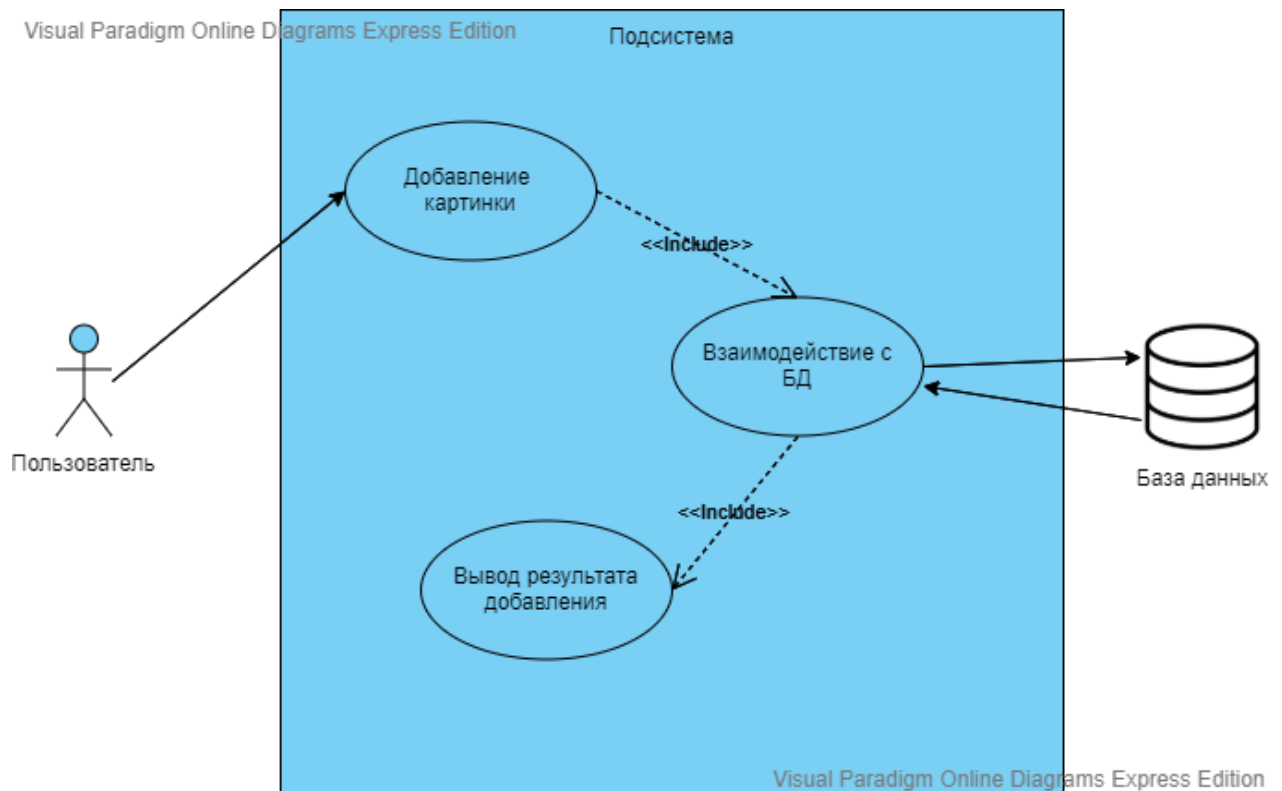


Рисунок 3.3 – Диаграмма взаимодействия с БД

В программе будет также возможность сыграть в викторину «Угадай эмоции», где пользователю будут поочередно показываться изображения различных эмоций и варианты ответа. UML-диаграмма варианта использования вкладки «Угадать эмоции» показана на рисунке 3.4



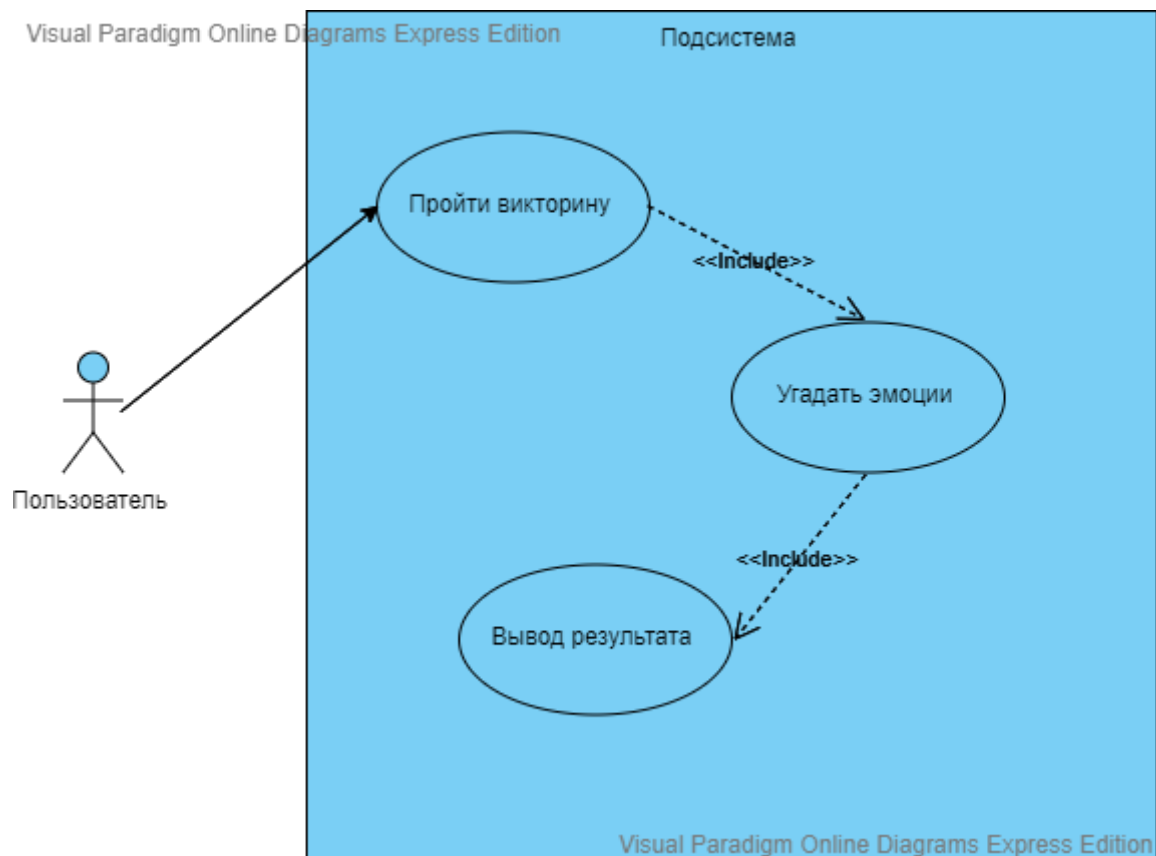


Рисунок 3.4 – UML-диаграмма прохождения викторины

### 3.3 Установка Microsoft Visual Studio 2019

Microsoft Visual Studio выпускается в трех редакциях: Community, Professional и Enterprise. Мы будем устанавливать Professional, так как данная редакция содержит профессиональные инструменты для разработки приложений. Перед установкой необходимо убедиться, что компьютер соответствует требованиям к установке Visual Studio 2019.

Системные требования Visual Studio 2019:

- операционная система: Windows 7/8/10;
- процессор с тактовой частотой не ниже 1,8 ГГц. Рекомендуется использовать как минимум двухъядерный процессор;
- 2 ГБ оперативной памяти, рекомендуется 8 ГБ (если устанавливать на виртуальную машину, то минимум 2.5 ГБ);
- свободного места на жестком диске от 800 мегабайт до 210 гигабайт, в зависимости от установленных компонентов;
- видеоадаптер с минимальным разрешением 1280 на 720 пикселей (для оптимальной работы Visual Studio рекомендуется разрешение 1366 на 768 пикселей или более высокое);

Скачать продукт можно на официальном сайте Microsoft (рисунок 3.5).

## Скачиваемые файлы

**Visual Studio 2019**  
Версия 16.5  
[Заметки о выпуске >](#)

Полнофункциональная интегрированная среда разработки (IDE) для приложений Android, iOS, Windows, веб- и облачных приложений

[Сравнение выпусков >](#)  
[Как выполнить установку в автономном режиме >](#)

**Сообщество**  
Эффективная среда IDE предоставляется бесплатно для студентов, участников создания открытого исходного кода и отдельных пользователей

[Скачать бесплатно ↓](#)  
[Скачать предварительную версию ↓](#)

**Professional**  
Профессиональная интегрированная среда разработки, оптимальная для небольших команд

[Бесплатная пробная версия ↓](#)  
[Скачать предварительную версию ↓](#)

**Корпоративное использование**  
Масштабируемое, комплексное решение для команд любого размера

[Бесплатная пробная версия ↓](#)  
[Скачать предварительную версию ↓](#)

Рисунок 3.5 – Официальный сайт загрузки Microsoft Visual Studio

После загрузки установочного файла необходимо запустить его от имени администратора. Установщик начнет выполнять подготовительные действия (рисунок 3.6). Нажимаем кнопку «Продолжить», соглашаясь с условиями лицензионного соглашения.

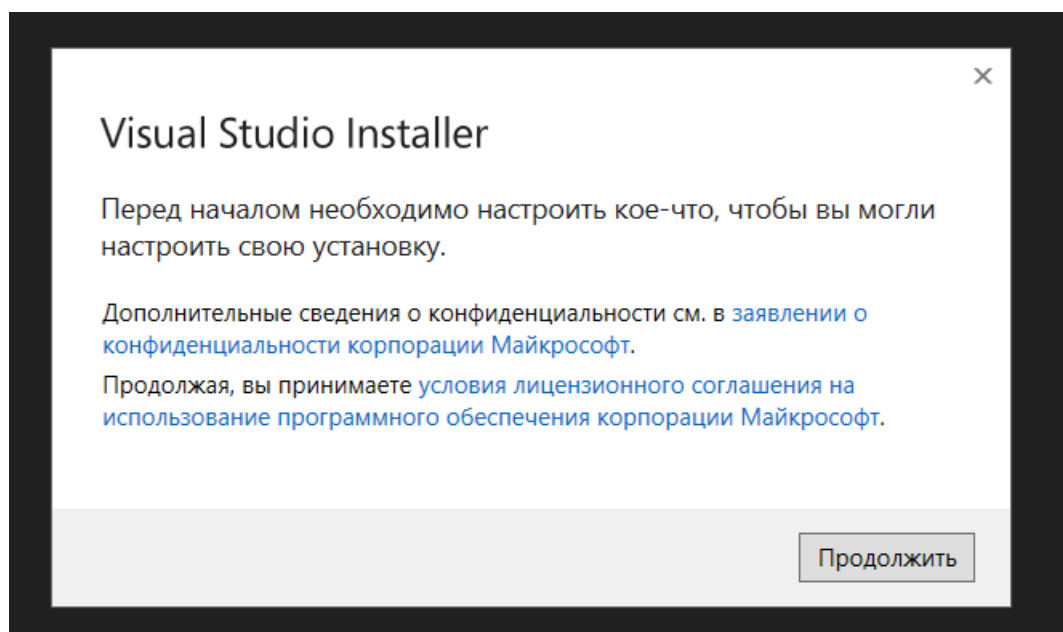


Рисунок 3.6 – Подготовка к установке

Далее происходит загрузка и установка необходимых файлов установщика (рисунок 3.7).

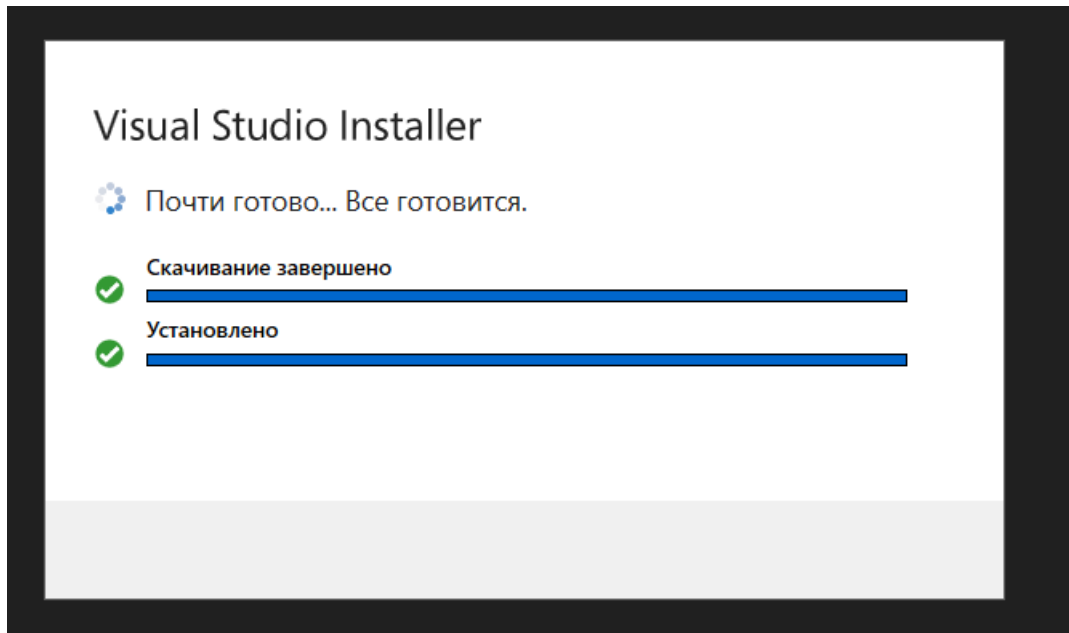


Рисунок 3.7 – Извлечение файлов установщика

После выполнения всех необходимых предварительных мероприятий запускается установщик. Открывается окно с набором компонентов и инструментов на выбор, которые необходимо загрузить (рисунок 3.8). Так как мы будем разрабатывать оконное приложение, во вкладке «Рабочие нагрузки» выбираем «Разработка классических приложений .NET». Данный пакет содержит полный набор инструментов и плагинов для создания приложений Windows с помощью Windows Forms.

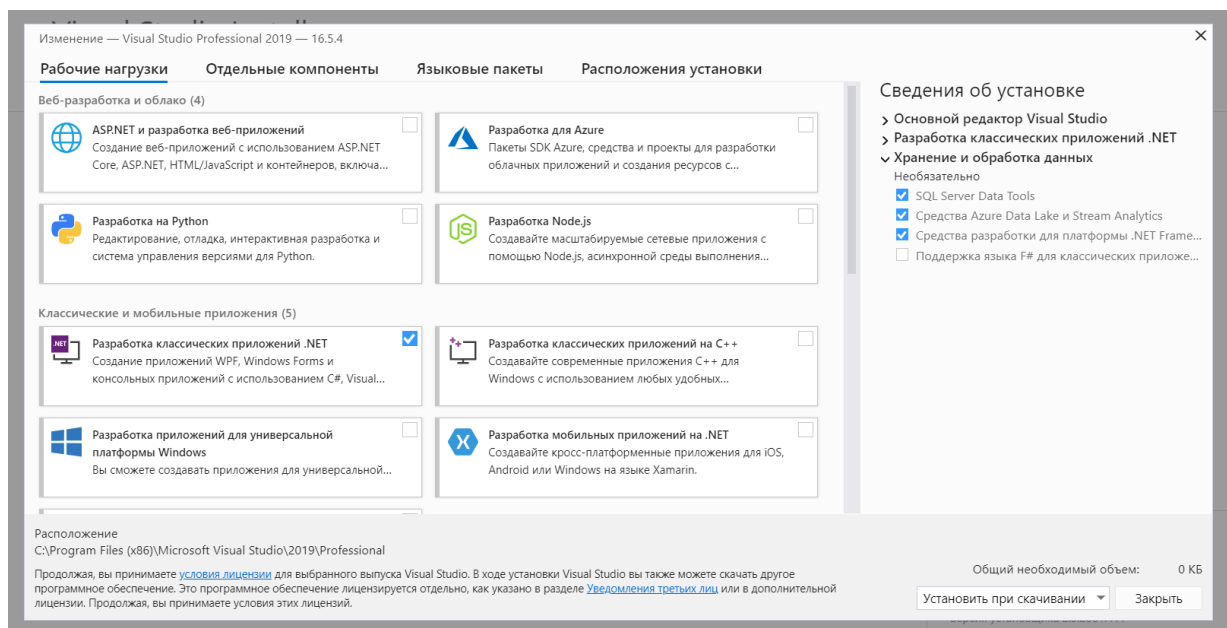


Рисунок 3.8 – Установка компонентов для создания приложения

После выбора необходимых компонентов, начинается процесс загрузки и установки. В зависимости от объема компонентов, скорости интернета и характеристик ПК продолжительность установки будет различаться. После завершения установки запускаем программу и приступаем к реализации приложения.

### 3.4 Разработка приложения

Реализация программы начинается с создания проекта приложения в интегрированной среде разработки Microsoft Visual Studio. После запуска в строке меню выбираем Файл-Создать-Проект. Диалоговые окна создания и настройки нового проекта показаны на рисунках 3.9 и 3.10

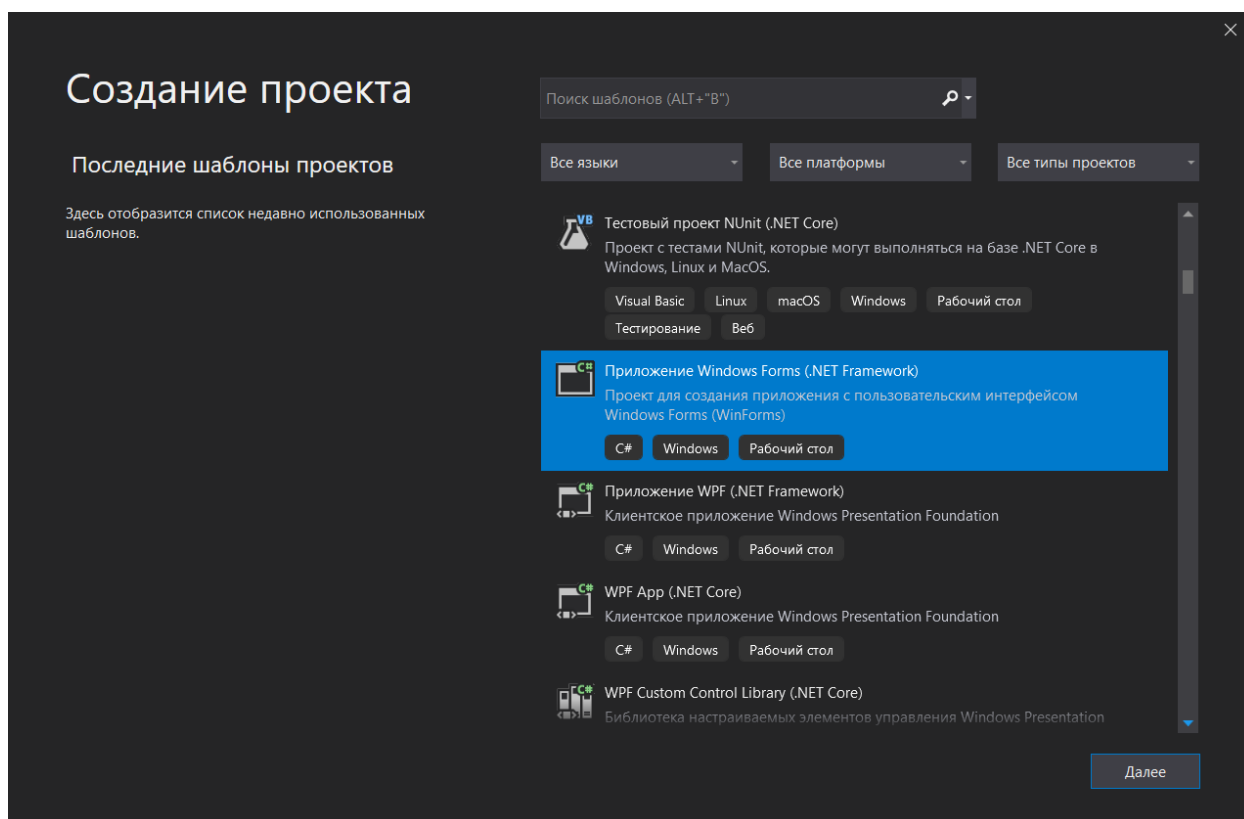


Рисунок 3.9 – Диалоговое окно «Создание проекта»

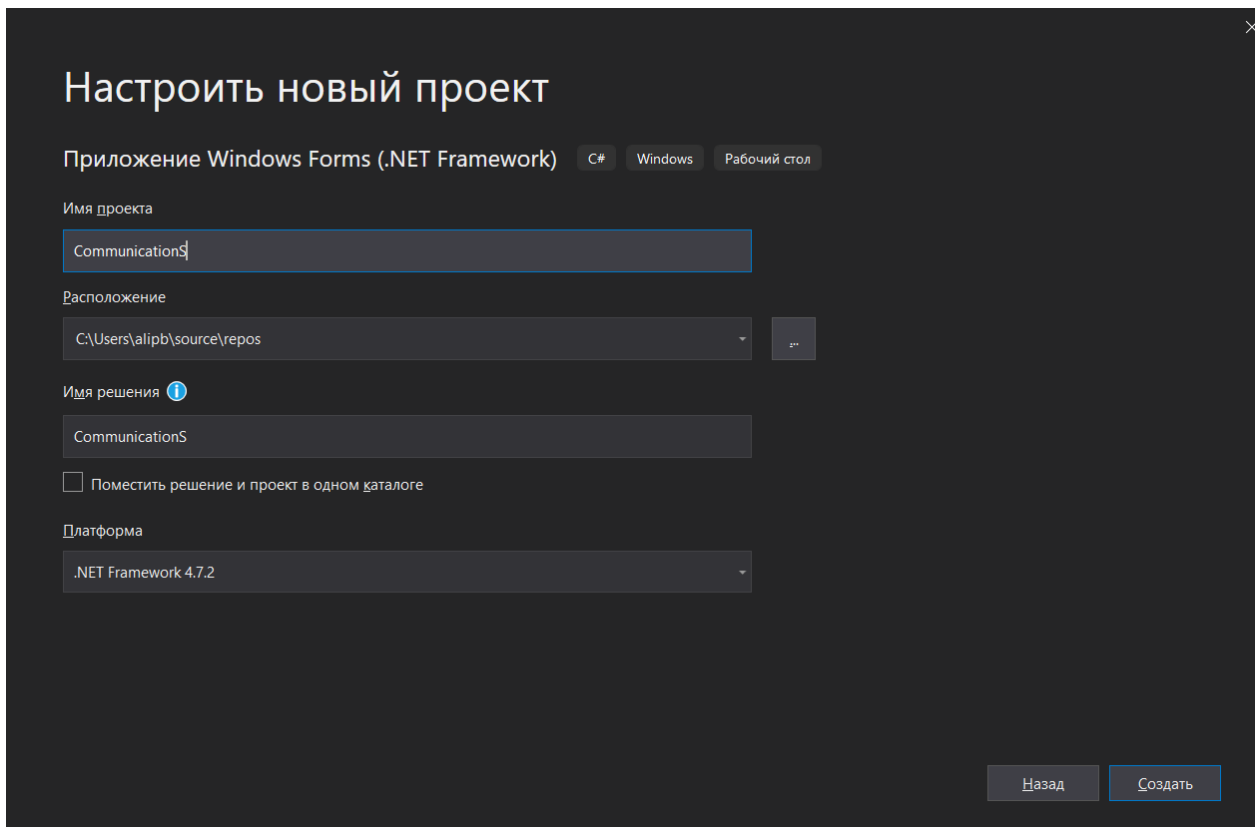


Рисунок 3.10 – Диалоговое окно «Настройка проекта»

В появившемся окне из списка шаблонов выбираем Приложение Windows Forms на языке программирования C#. Далее необходимо настроить проект: указать имя проекта, в нашем случае – CommunicationS, а также расположение (можно оставить по умолчанию). После этого нажимаем кнопку «Создать», и Visual Studio создает решение для программы. Решение представляет собой контейнер, в котором находятся все проекты и файлы, необходимые программе. В результате появляется окно, с графическим дизайнером, который содержит форму будущего приложения, представленное на рисунке 3.11. Графический дизайнер представляет собой визуальную поверхность, на которую можно добавлять элементы управления, например, кнопки, изображения и тому подобное, для создания удобного интерфейса. В Visual Studio конструктор форм называется Windows Forms Designer.

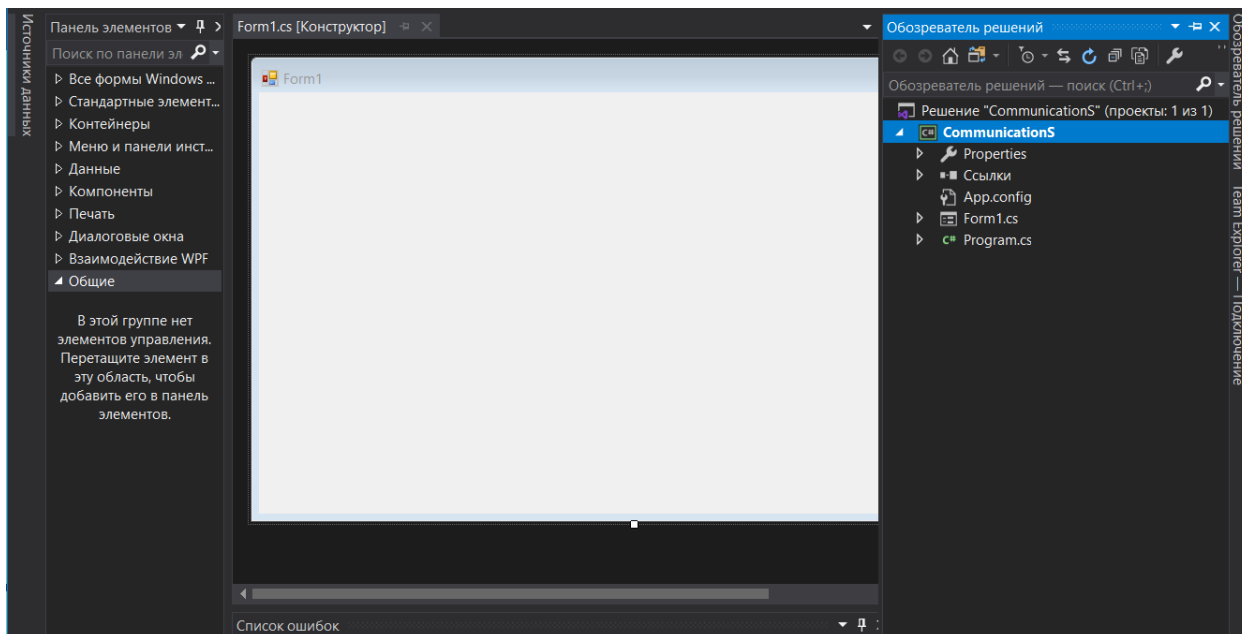


Рисунок 3.11 – Окно графического дизайнера

При разработке проекта визуальная среда подразделяется на три основных части: главное окно, Обозреватель решений и окно Свойств. В центре располагается главное окно, в котором создаются визуальные формы и пишется код. Справа размещается окно файлов проекта/решения - Solution Explorer (Обозреватель решений), в котором находятся все файлы, связанные с приложением. Оно позволяет перемещаться между проектами и их элементами [22]. На рисунке 3.12 показано готовое решение нашего проекта со всеми необходимыми формами и файлами.

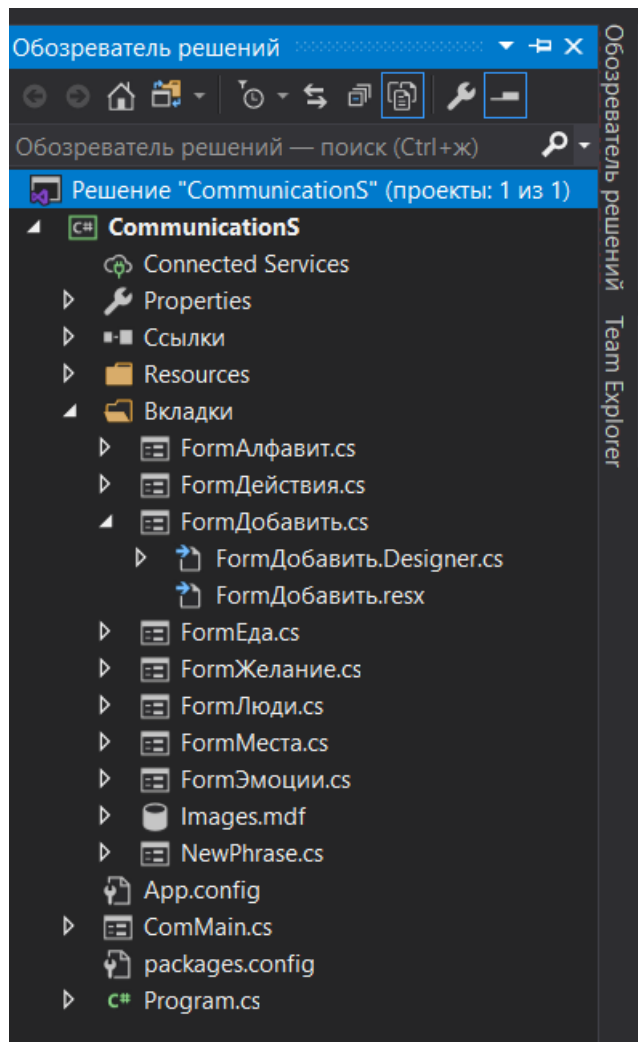


Рисунок 3.12 – Обозреватель решений

Properties содержит свойства объектов и решений. В нем будут записаны настройки интерфейса и значений форм.

Ссылки отображают ссылочные постоянные, на которые будет ссылаться программный код для упрощения реализации программного кода и корректной работы приложения.

В папке Resources находятся любые неисполняемые данные, которые запускаются вместе с приложением. Ресурсы могут содержать данные различных видов, такие как, символьные строки, изображения и объекты.

В папке Вкладки содержатся дополнительные формы с данными, представленные в приложении в виде вкладок.

ComMain – основная форма приложения.

Файл Program.cs содержит основной код программы.

Во время работы с графическим дизайнером справа открывается окно Свойств (Properties). Оно содержит список атрибутов объекта, выделенного в данный момент, свойства объектов и решений. С помощью этого окна можно произвести настройку интерфейса и значений форм (рисунок 3.13). Например,

можно установить определенный шрифт и размер шрифта, задать фоновое изображение, ширину и высоту главного окна, и другие свойства.

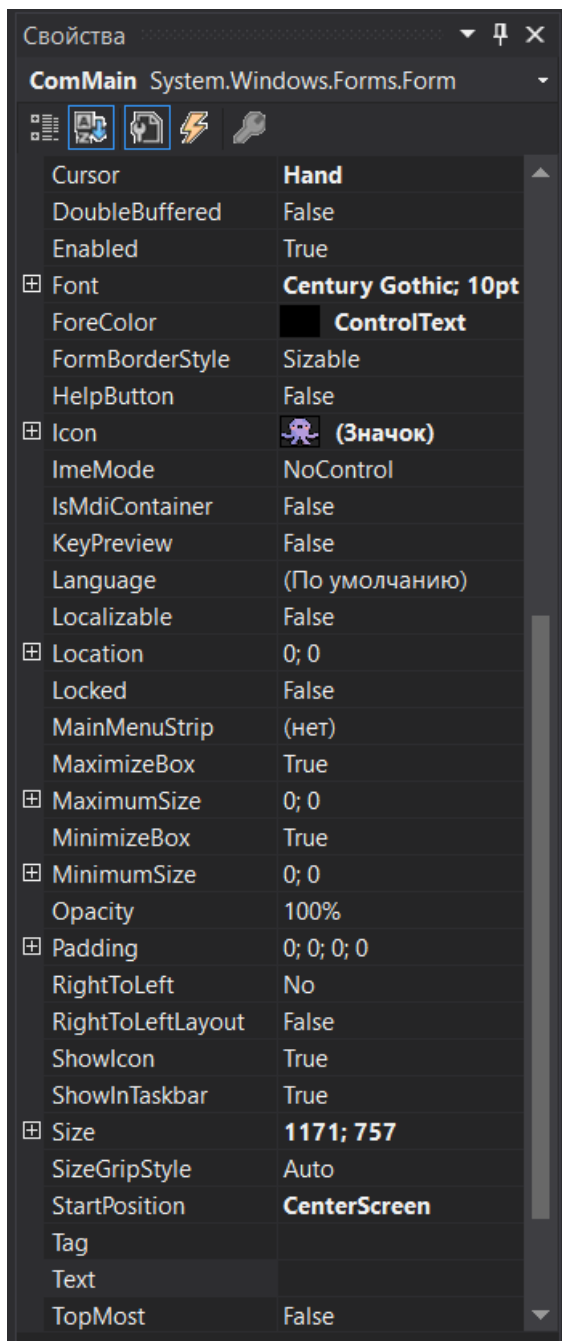


Рисунок 3.13 – Окно Свойства

Основное преимущество над другими интегрированными средами разработки состоит в том, что Visual Studio позволяет перемещать элементы сразу на форму при проектировании приложения. Все эти элементы находятся слева в окне Панель элементов (рисунок 3.14).



Для того, чтобы использовать какой-либо из компонентов необходимо щелкнуть и переместить желаемый компонент на форму или дважды щелкнуть по нему, тогда он автоматически добавит его экземпляр.

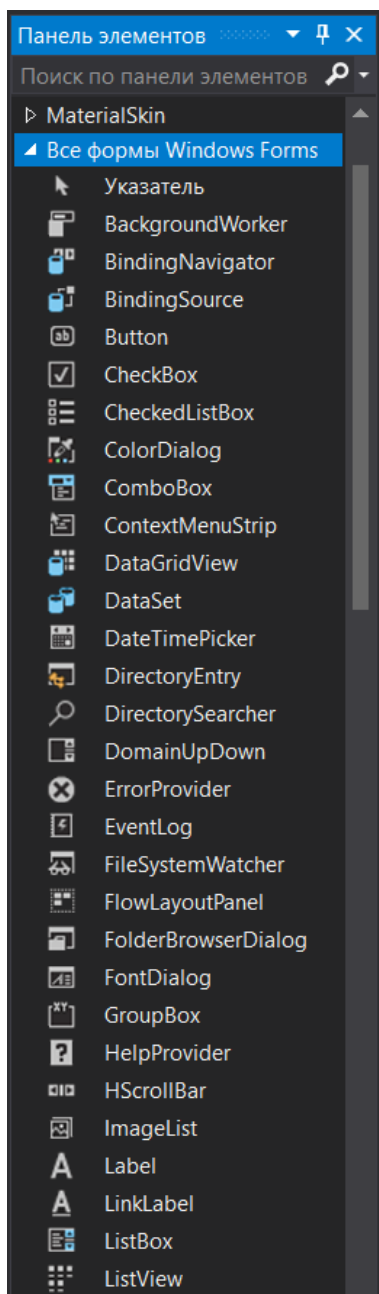


Рисунок 3.14 – Панель элементов

Для создания оконного приложения будем использовать вкладку «Все формы Windows Forms», в которой находятся все нужные компоненты. Основные компоненты, используемые в приложении:

1) Button – элемент управления, выступаемый в роли обработчика событий. При нажатии на кнопку осуществляется переход на другую вкладку;

2) Panel представляет собой панель, которая объединяет элементы в группы. Удобство элемента в том, что панель может сливаться со всей формой;

3) FlowLayoutPanel также используется для объединения элементов в группы. Элемент имеет дополнительную функциональность: можно изменять позицию и компоновку дочерних элементов при изменении размеров формы во время выполнения программы.

Важным этапом разработки приложения является создание дизайна. Дизайн подразумевает создание интерфейса и индивидуального стиля программы. Так как наше приложение будет ориентировано на детей дошкольного и школьного возраста, следовательно, интерфейс должен быть простым, привлекательным и интуитивно понятным. Цветовая палитра интерфейса играет ключевую роль, так как именно с помощью цвета можно привлечь внимание, повлиять на эмоции, восприятие и создать определенное настроение у ребенка. По данным статистики 75% детей предпочитают фиолетовый цвет всем остальным оттенкам. Поэтому мы остановили свой выбор именно на нем.

### 3.5 Описание интерфейса

Главное окно программы представляет собой форму, на которой располагаются вкладки (рисунок 3.15).

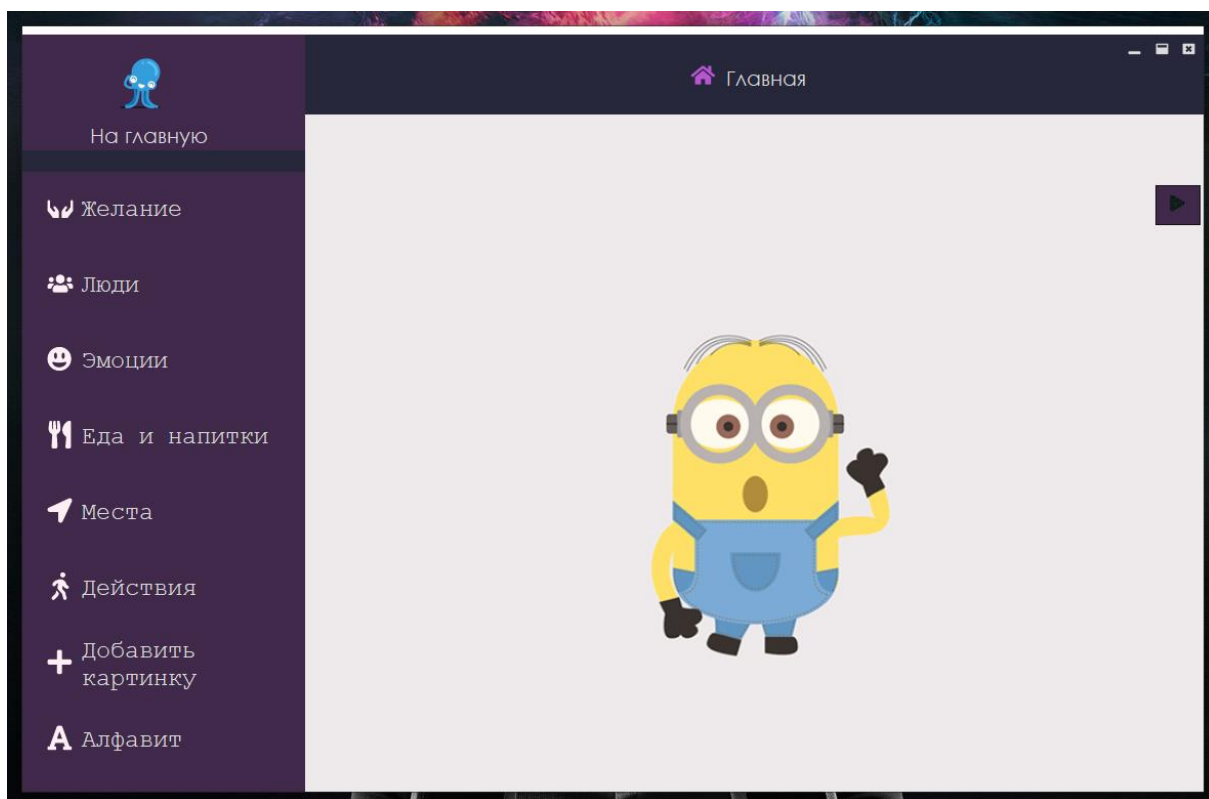


Рисунок 3.15 – Основная форма

В каждой вкладке имеются специальные карточки PECS (Picture Exchange Communication System), которые являются альтернативной системой для общения, когда используются не слова, а картинки. Данные карточки применяются среди аутичных детей, испытывающих коммуникативные затруднения.

При открытии любой вкладки появляется дополнительная панель, на которой будут располагаться выбранные карточки для составления предложений или отдельных слов. На панели расположена кнопка воспроизведения фразы. На рисунках 3.16 – 3.22 показаны скриншоты отдельных вкладок.

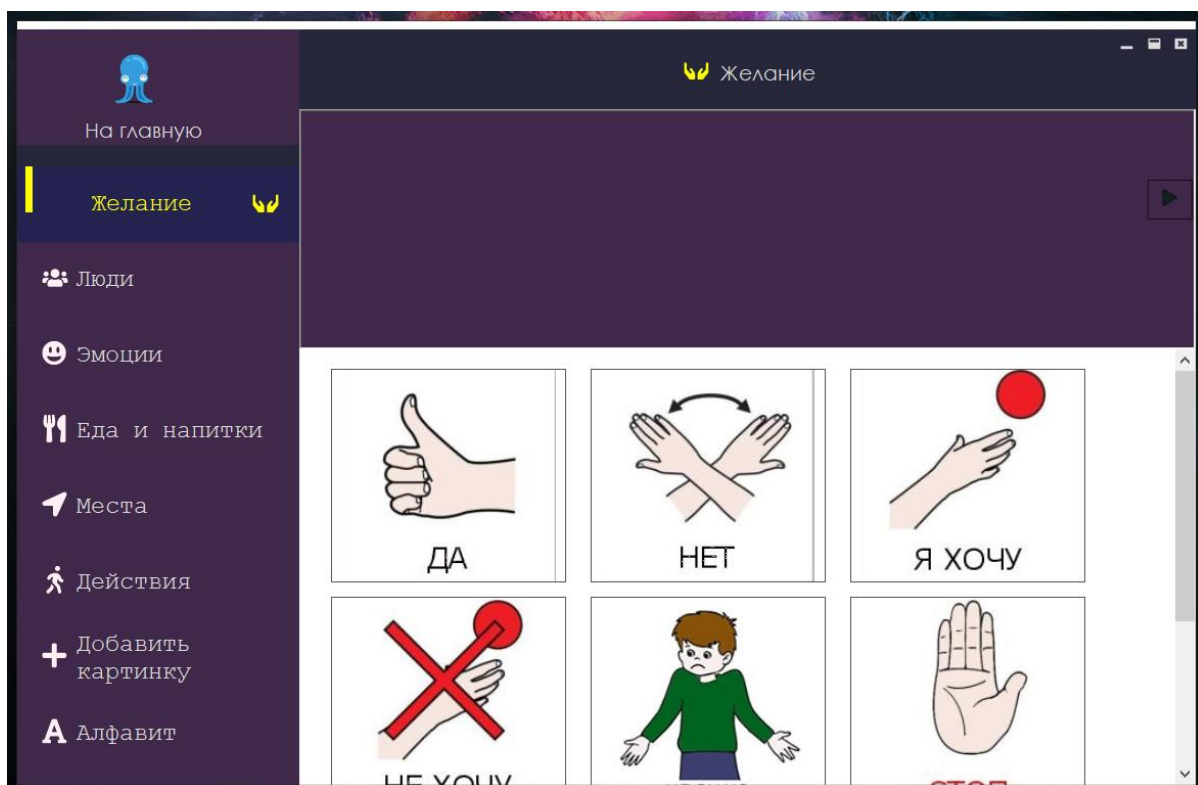


Рисунок 3.16 – Вкладка «Желание»

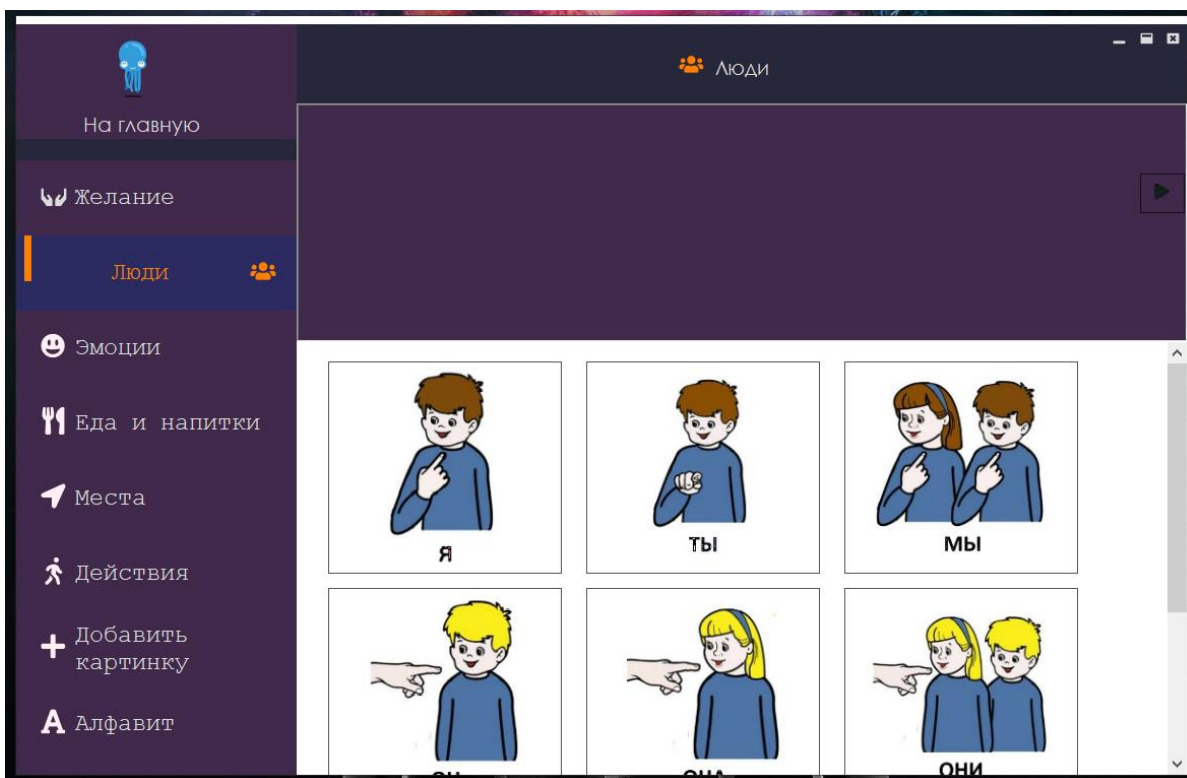


Рисунок 3.17 – Вкладка «Люди»

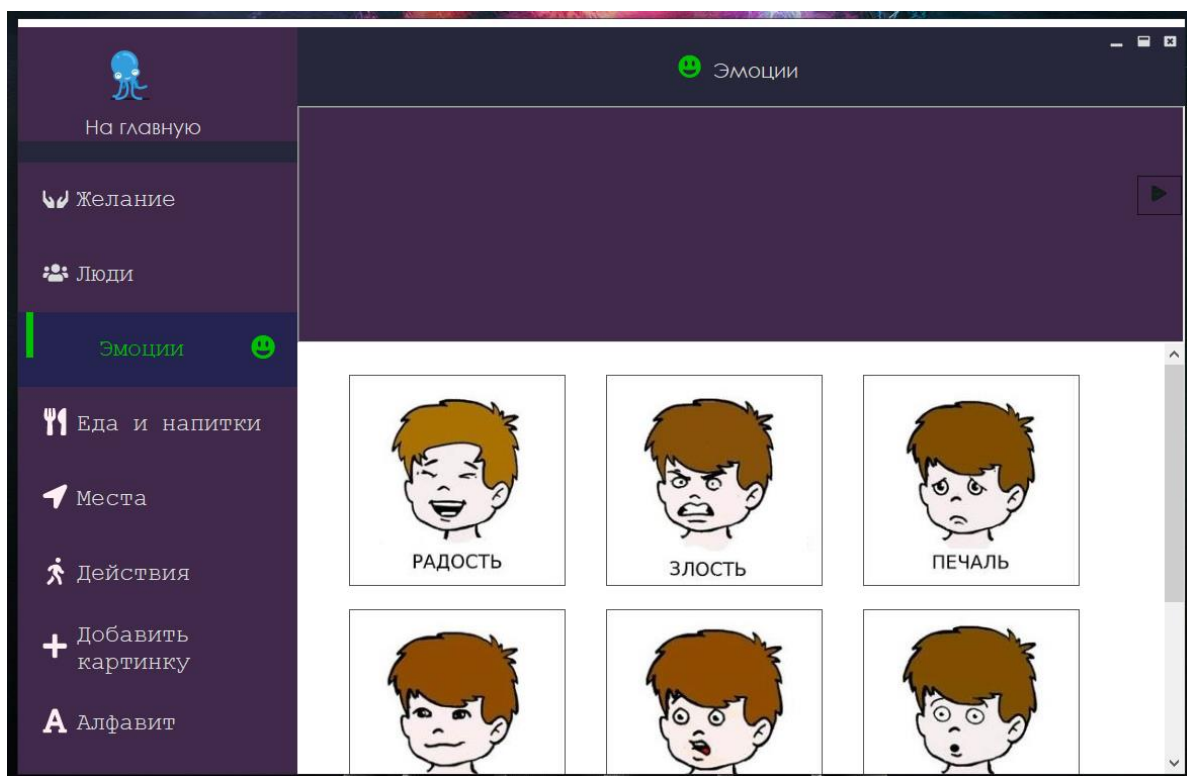


Рисунок 3.18 – Вкладка «Эмоции»

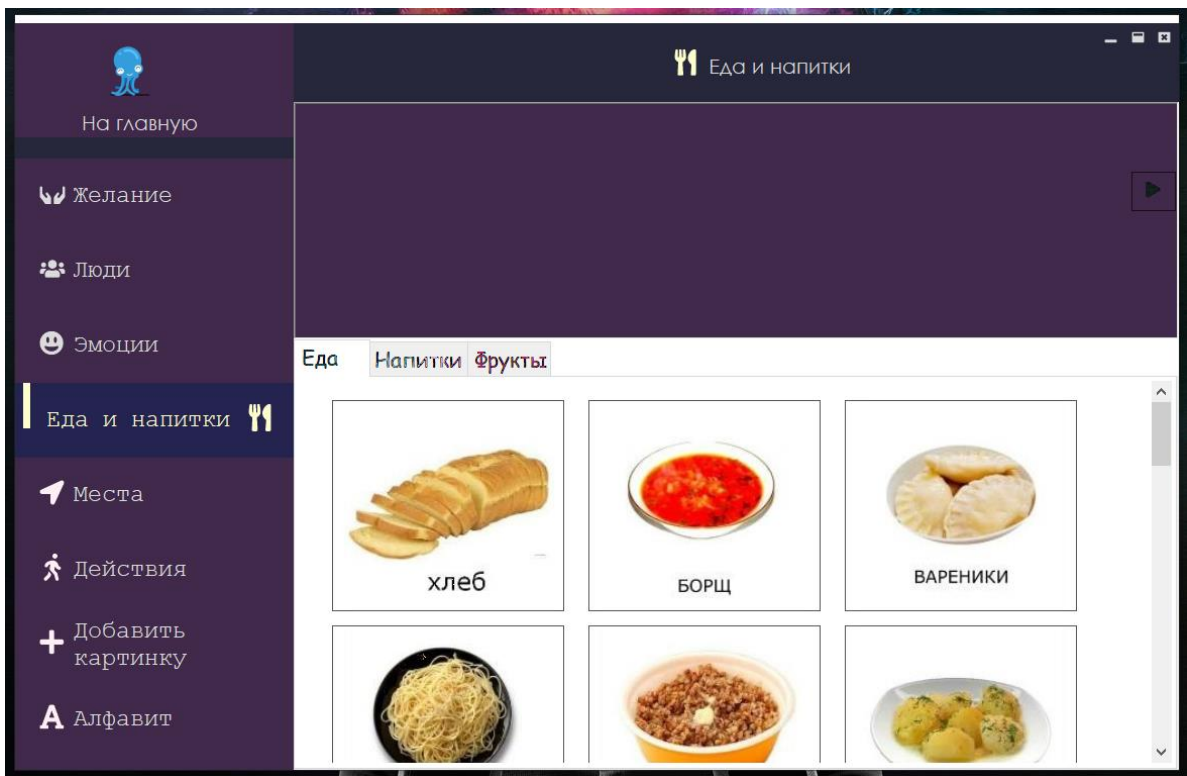


Рисунок 3.19 – Вкладка «Еда и напитки»

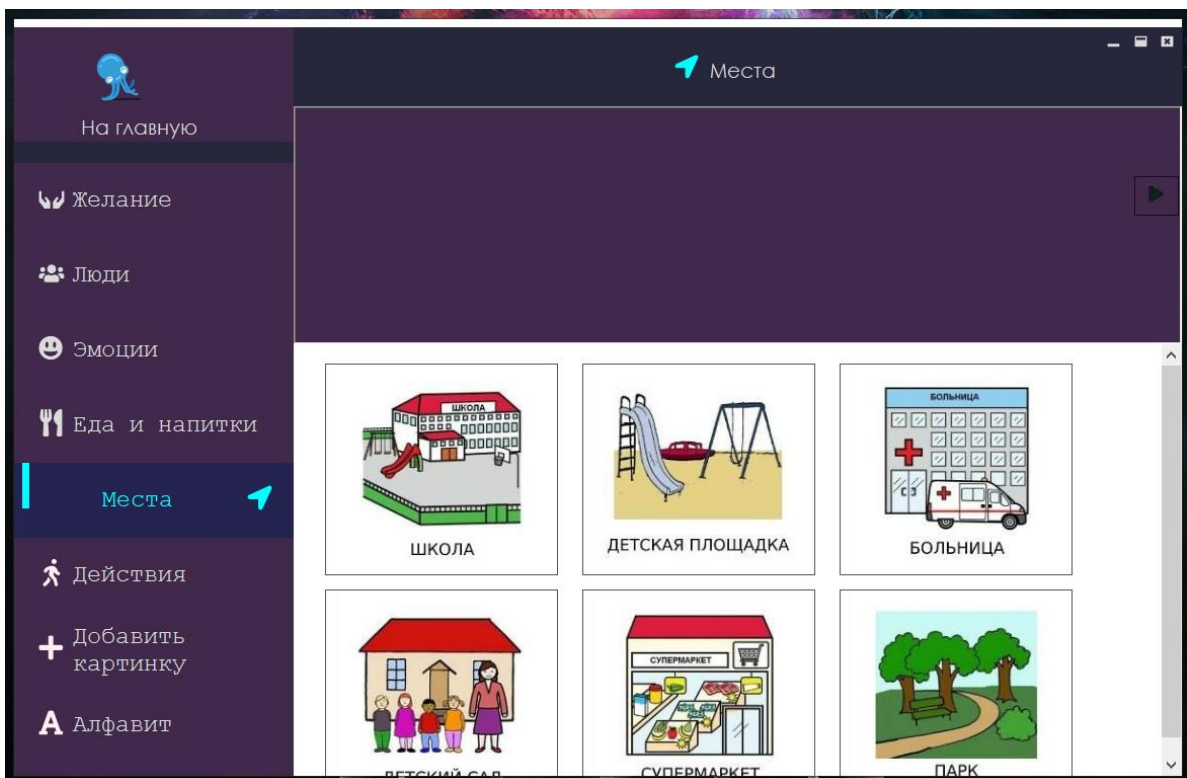


Рисунок 3.20 – Вкладка «Места»

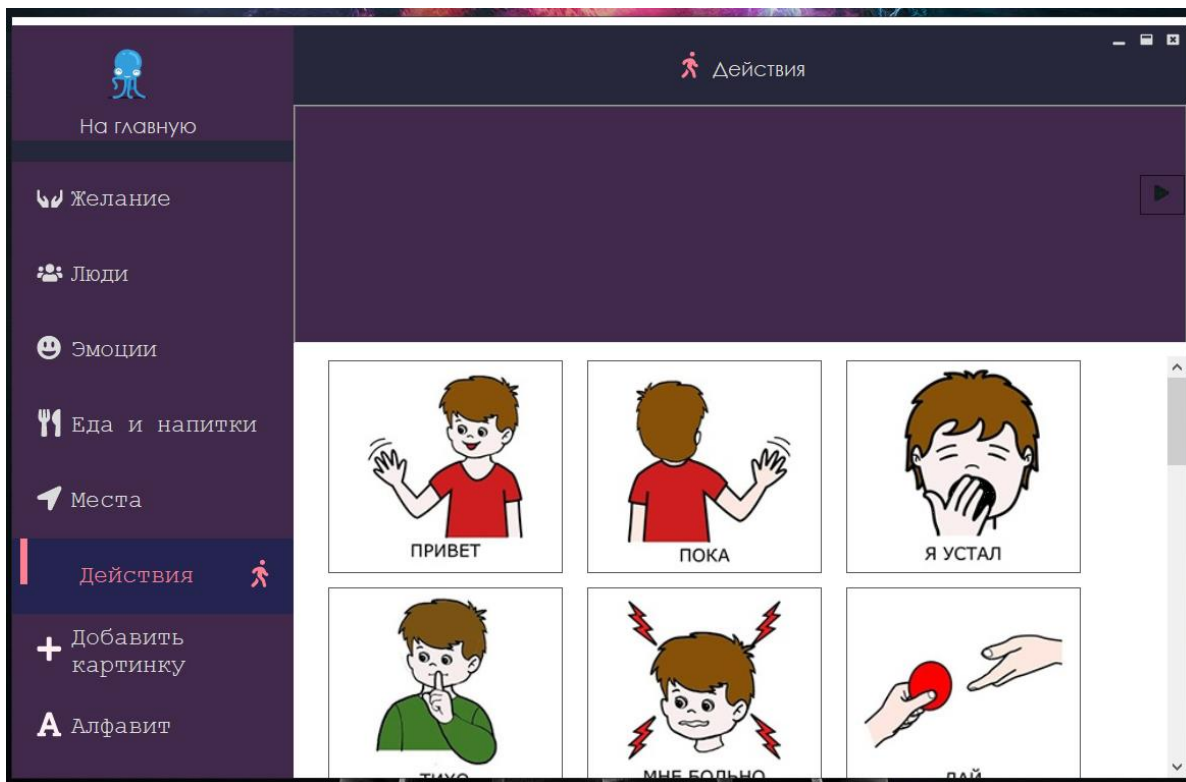


Рисунок 3.21 – Вкладка «Действия»

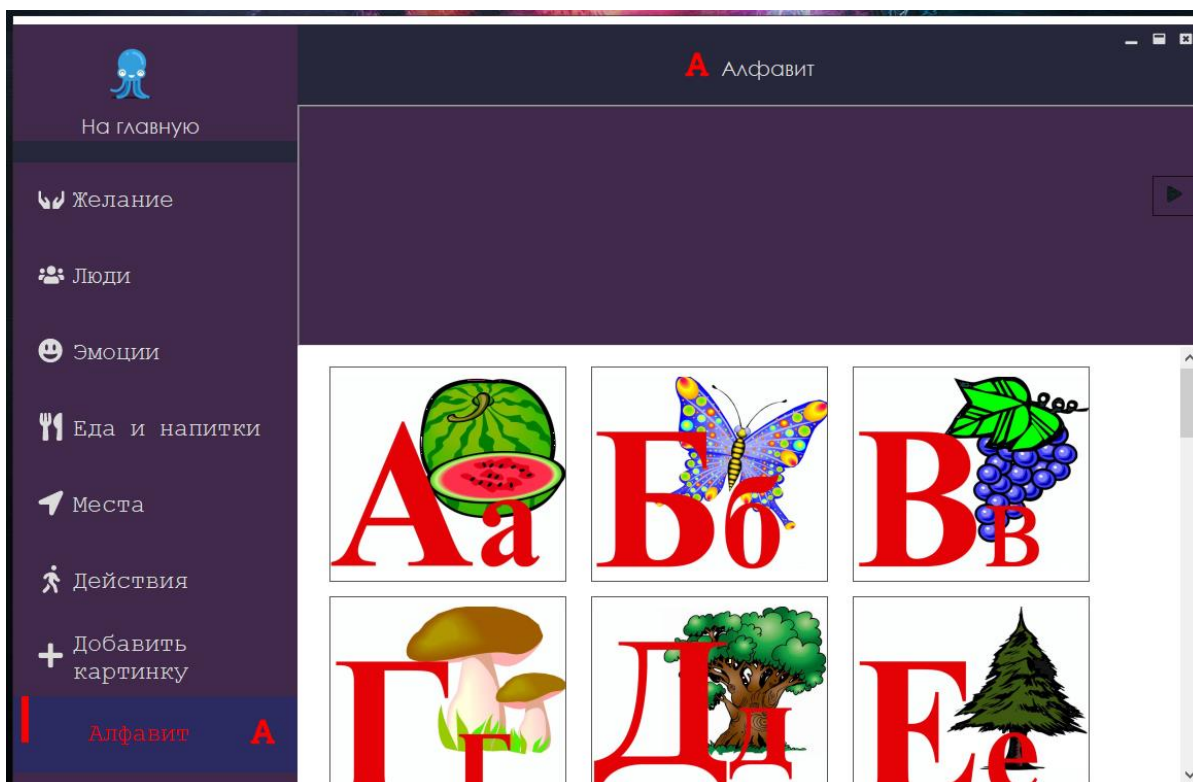


Рисунок 3.22 – Вкладка «Алфавит»

Во вкладке «Добавить картинку» пользователь может самостоятельно добавить свою картинку, а также привязать к ней звуковое сопровождение. Пример добавления новой картинки показан на рисунке 3.23.

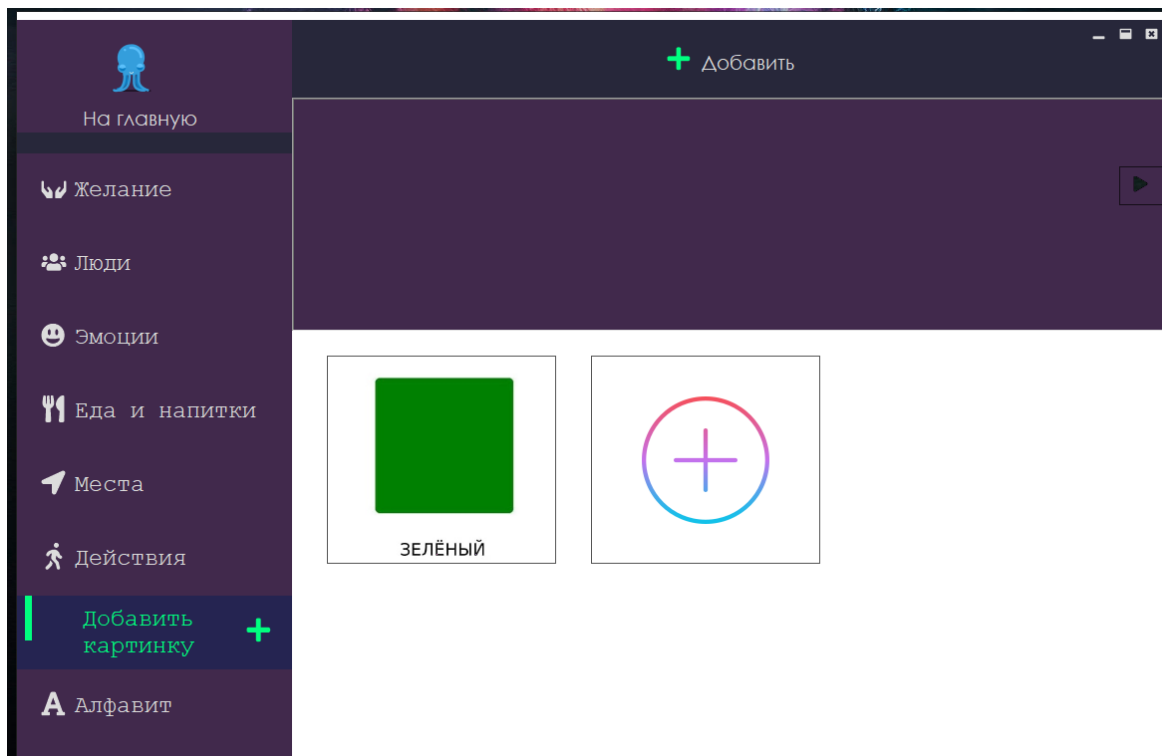


Рисунок 3.23 – Вкладка «Добавить картинку»

При нажатии на плюс появляется окно с доступом к файлам (рисунок 3.24).

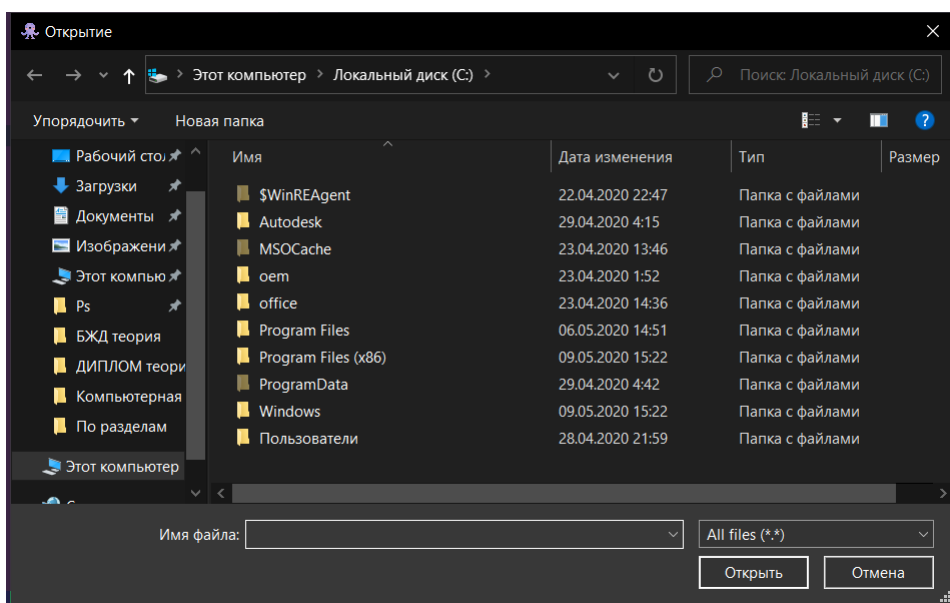


Рисунок 3.24 – Окно добавления новой картинки

После выбора нужной картинке предлагается ввести описание к ней, данное описание будет воспроизводиться при выборе картинке (рисунок 3.25).

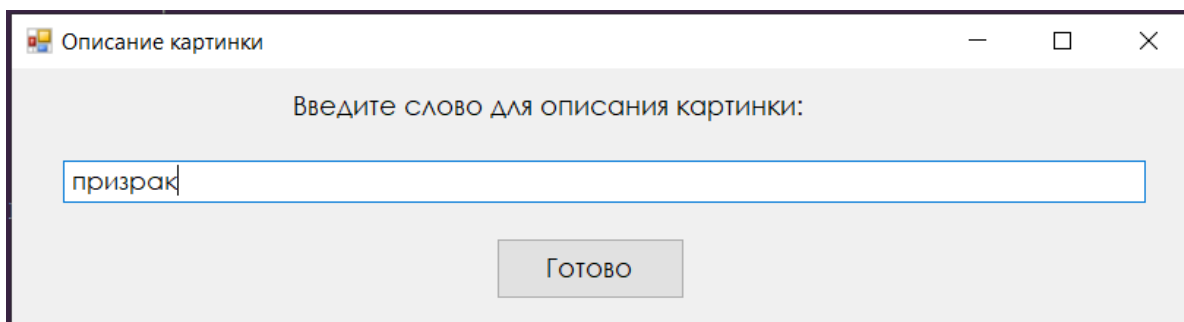


Рисунок 3.25 – Диалоговое окно описания картинке

Результат появления новой картинке представлен на рисунке 3.26.

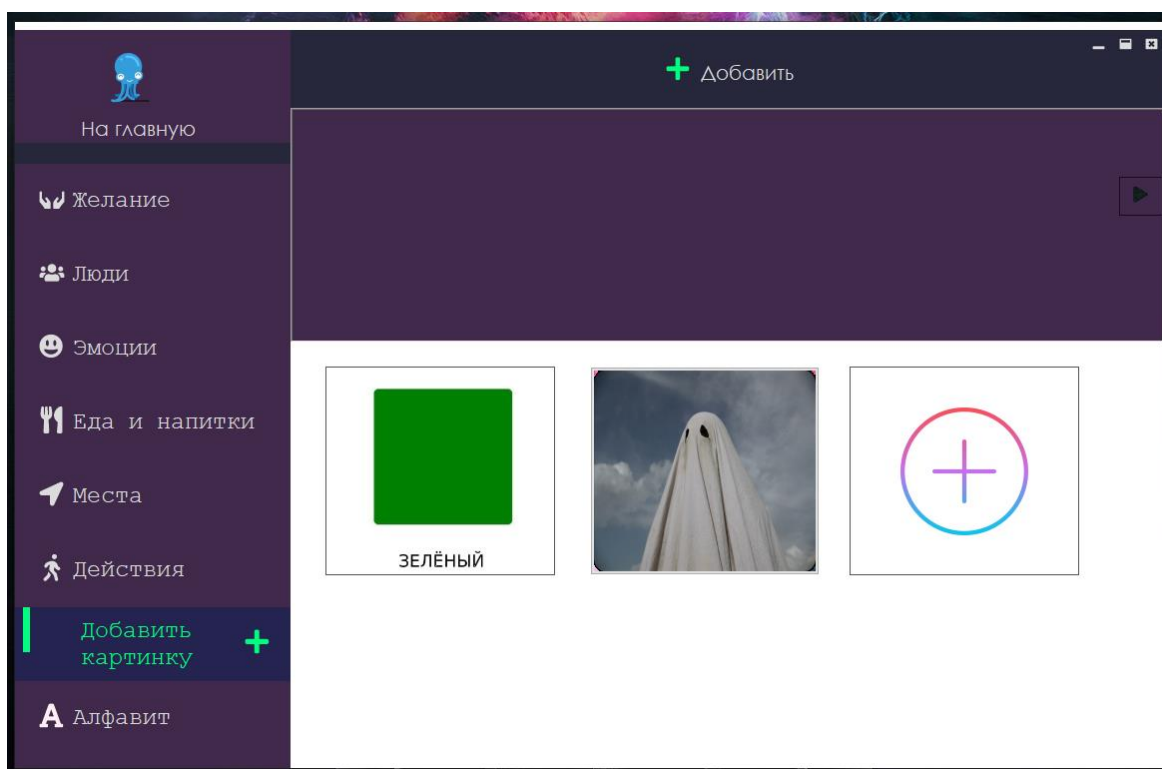


Рисунок 3.26 – Добавление новой картинке

Для составления предложения или просто слова достаточно лишь кликнуть по картинке, затем она отображается на специальной панели. После каждого клика карточка воспроизводится, а для озвучивания полной фразы из карточек необходимо нажать кнопку «Воспроизвести» на панели. Для воспроизведения используется синтез речи и экземпляр класса



SpeechSynthesizer. В листинге ниже приведен отрывок кода, выполняющий функцию воспроизведения.

```
public static void SpeakForMe(string phrase)
{
    // создание нового экземпляра синтезатора речи
    SpeechSynthesizer synth = new SpeechSynthesizer();
    using (MemoryStream streamAudio = new MemoryStream())
    {
        //создание экземпляра для воспроизведения выходного аудиофайла
        System.Media.SoundPlayer m_SoundPlayer = new
System.Media.SoundPlayer();
        // настройка синтезатора для вывода в аудиопоток
        synth.SetOutputToWaveStream(streamAudio);
        //воспроизведение фразы
        synth.Speak(phrase);
        streamAudio.Position = 0;
        m_SoundPlayer.Stream = streamAudio;
        m_SoundPlayer.Play();
        synth.SetOutputToNull();
    }
}
//кнопка воспроизведения
private void iconPlay_Click(object sender, EventArgs e)
{
    string ph = "";
    for (int i = 0; i < Phrase.Count; i++)
    {
        ph += Phrase[i];
    }
    if (ph != "") {SpeakForMe(ph); } else { MessageBox.Show("Фраза не
заполнена!", "Пустое поле!", MessageBoxButtons.OK, MessageBoxIcon.Information); }
    GC.Collect();
}
```

Пример составления предложения и использования программы приведен на рисунках 3.27 и 3.28



Рисунок 3.27 – Пример составления предложения

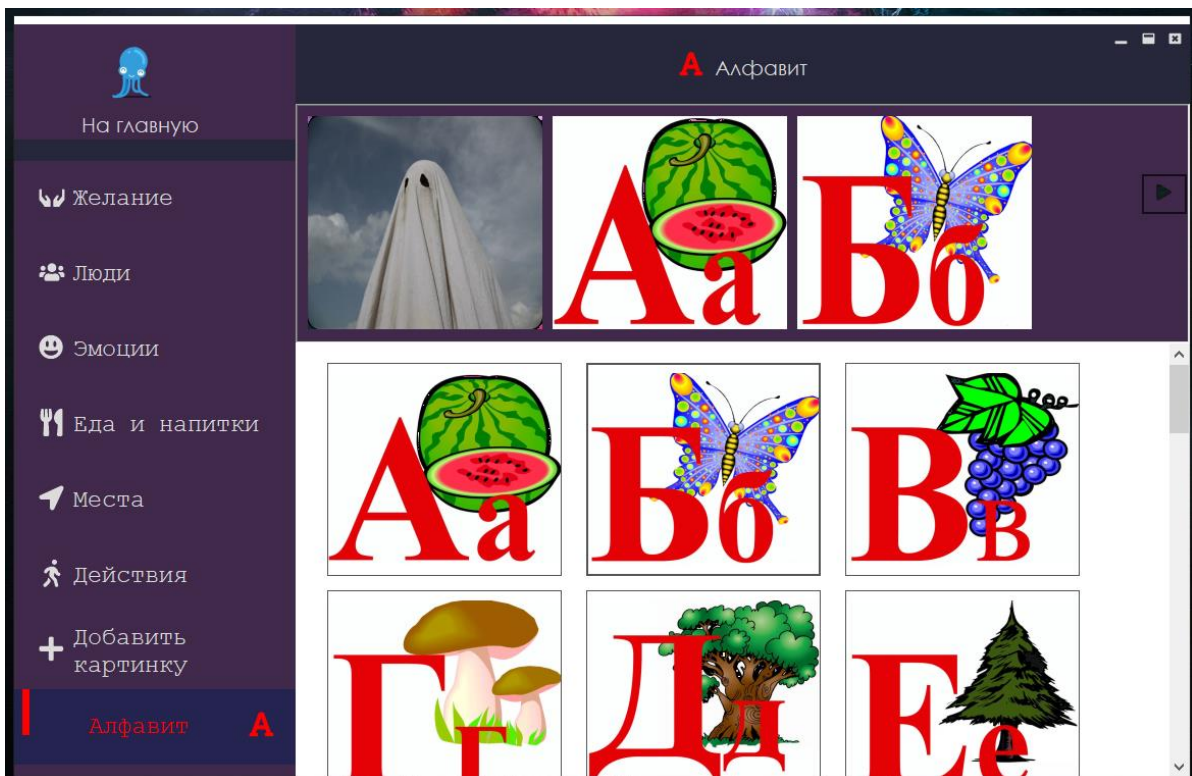


Рисунок 3.28 – Пример использования программы

При открытии вкладки «Угадай эмоции» появляется новое окно, в котором пользователю предлагается отгадать, какую из базовых эмоций испытывает ребёнок, изображенный на картинке (рисунок 3.29).

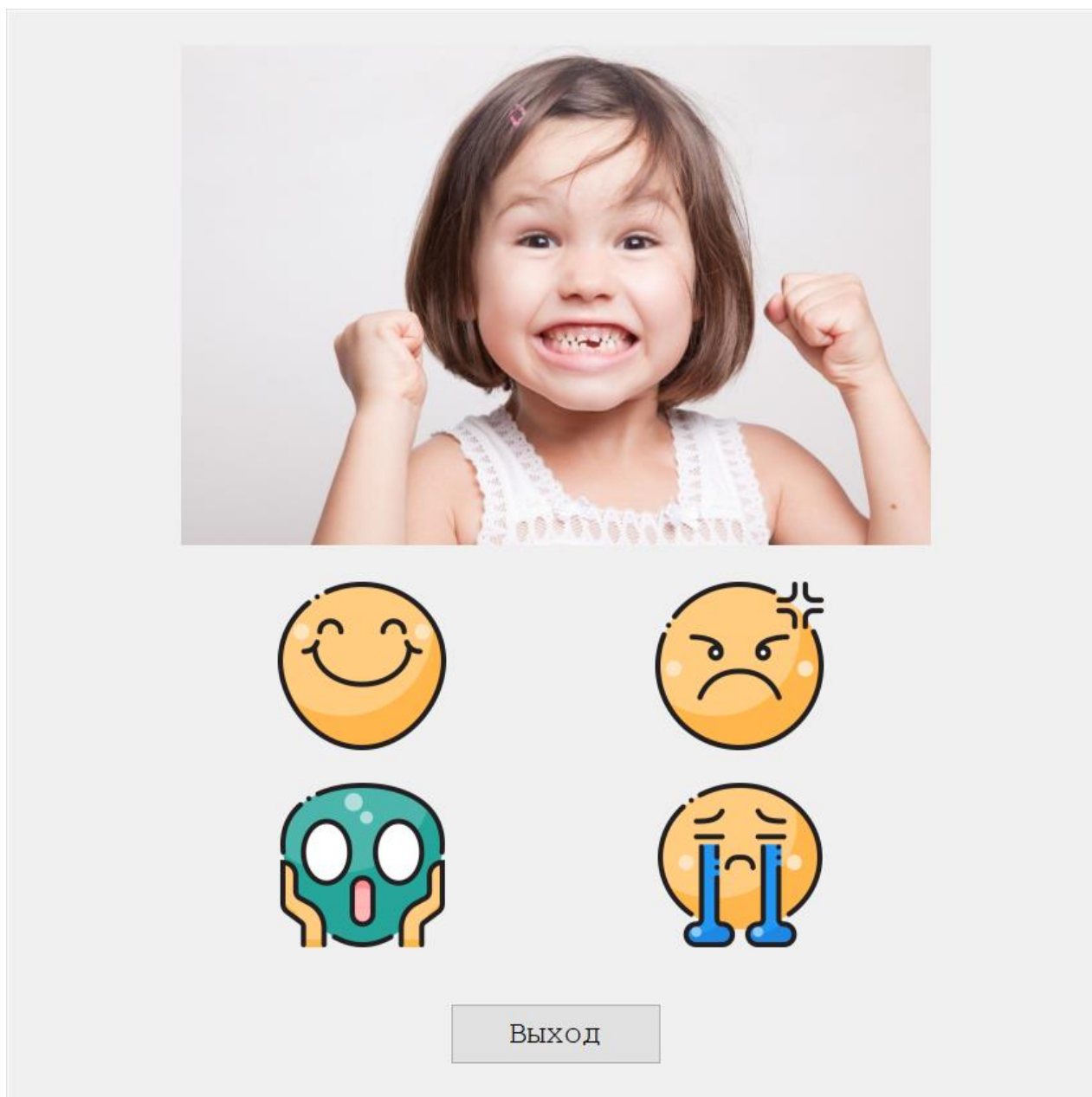


Рисунок 3.29 – Окно викторины «Угадай эмоции»

Поочередно будет показываться 9 изображений с различными эмоциями. Верный и неверный ответы будут сопровождаться специальным всплывающим окном с сообщением (рисунок 3.30 и 3.31).

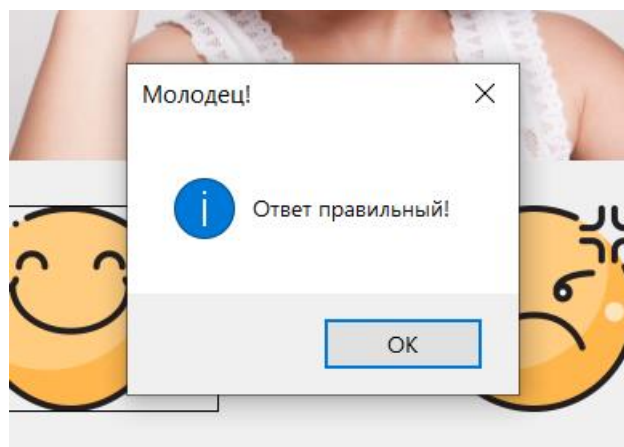


Рисунок 3.30 – Всплывающее окно, если ответ верный

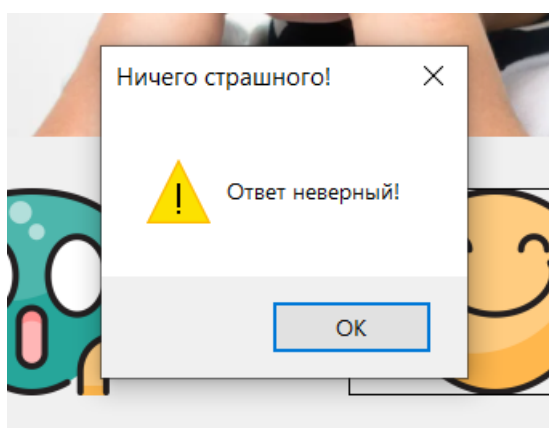


Рисунок 3.31 – Всплывающее окно, если ответ неверный

После завершения викторины результат также появляется в виде окошка, показывая количество правильных и неправильных ответов (рисунок 3.32).

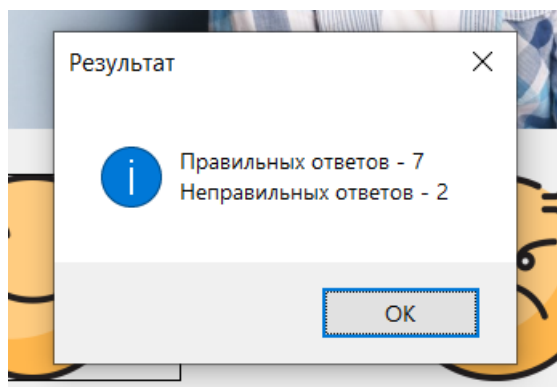


Рисунок 3.32 – Всплывающее окно результата

Если имеются неправильные ответы, то после подтверждения кнопки имеется возможность вернуться к ним ответам и постараться отгадать еще раз, тем самым закрепить материал и узнать свои ошибки.

Таким образом, данный программный продукт выполняет все функции коммуникатора для детей с расстройством аутистического спектра. С помощью него ребёнок, имеющий затруднения в общении, сможет озвучивать свои желания, составлять предложения и использовать данное приложение в качестве средства взаимодействия с людьми. Также имеется возможность изучения базовых эмоций посредством прохождения викторины.

Программа была полностью реализована в интегрированной среде разработки Microsoft Visual Studio без использования сторонних программных средств и приложений. Графический интерфейс создавался с использованием технологии Windows Forms платформы .NET. Данная технология является наиболее удобной и простой в использовании.

#### 4 Технико-экономическое обоснование

В данной дипломной работе рассматривается проект по разработке информационной обучающей системы для детей с расстройством аутистического спектра.

Целью дипломной работы является разработка информационной обучающей системы для детей с расстройством аутистического спектра, которая поможет ребенку развивать коммуникативные навыки и улучшить познавательные способности.

Программа представляет собой коммуникатор общения, в которой имеются специальные карточки PECS со звуковым сопровождением на различные категории, с помощью которых непосредственно осуществляется взаимодействие ребенка и взрослого. Ребенок будет иметь возможность составлять предложения или отдельные фразы при помощи данных карточек. Также имеется возможность изучать эмоции людей по картинкам, затем закреплять полученные знания викториной. Разработка программы производится на языке C# в среде разработки Visual Studio. Графический интерфейс программы создан с помощью технологии Windows Forms.

Технико-экономическое обоснование разработки и внедрения в данной работе будет содержать следующее:

- расчет затрат на разработку программного продукта;
- оценка экономической эффективности программного продукта;
- определение трудоемкости разработки программного продукта;
- определение возможной (договорной) цены программного продукта.

##### 4.1 Определение объема и трудоемкости разработки ПП

Трудоемкость программного продукта определяется на основе данных о предположительном распределении времени, которое было затрачено на разработку ПП.

Таблица 4.1 – Распределение работ вместо скобках ограничения по видам, этапам и количестве получают обучающихся оценка их трудоемкости

Этапы разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП	
		Чел. х час	Час х день
Этап 1	Постановка цели и задач	1 х 8	8 х 1
Этап 2	Анализ предметной области	1 х 24	8 х 3
Этап 3	Составление требований ПП	1 х 32	8 х 4

Продолжение таблицы 4.1

Этап 4	Проектирование структуры ПП	1 x 48	8 x 6
Этап 5	Разработка ПП	1 x 72	8 x 9
Этап 6	Отладка и тестирование	1 x 8	8 x 1
Этап 7	Обработка результатов	1 x 24	8 x 3
Итого трудоемкость выполнения проекта		1 x 216	8 x 27

## 4.2 Расчет затрат на разработку ПП

Для составления сметы определения затрат на разработку ПП необходимо рассчитать следующие затраты:

- 1) затраты на оплату труда;
- 2) материальные затраты;
- 3) затраты на электроэнергию;
- 4) социальный налог;
- 5) амортизация основных фондов.

### 4.2.1 Материальные затраты

Затраты по материалам, необходимым для программного продукта, рассчитываются с помощью таблицы 4.2 и 4.3.

Определим общую сумму затрат на материальные ресурсы ( $Z_M$ ) по следующей формуле:

$$Z_M = \sum_{i=1}^n P_i * C_i, \quad (4.1)$$

- где  $P_i$  – расход  $i$ -го вида материального ресурса, натуральные единицы;  
 $i$  – вид материального ресурса;  
 $C_i$  – цена за единицу  $i$ -го вида материального ресурса, тг;  
 $n$  – количество видов материальных ресурсов.

Таблица 4.2 – Стоимость оборудования и ПО

№	Наименование	Описание	Цена за единицу, тг	Сумма, тг
1	Ноутбук	Ноутбук HP 15-rb006ur	90 000	90 000
2	Операционная система	Microsoft Windows 10 Домашняя версия	Бесплатно	Бесплатно

Продолжение таблицы 4.2

3	Интегрированная среда разработки	Подписка Visual Studio Professional	19 125/мес.	38 250
4	Беспроводная мышь	Rapoo 3510, White	7 290	7 290
5	Принтер	CANON I-SENSYS MF4410	68 990	68 990
Итого затраты на оборудование и ПО				204 530

Таблица 4.3 – Затраты на материальные ресурсы

№	Наименование	Описание	Цена за единицу, тг	Сумма, тг
1	Бумага	A4	1 000	2 000
2	Картридж		5 000	5 000
Итого затраты на материальные ресурсы				7 000

Таким образом, общая сумма затрат на материальные ресурсы и оборудование составила 211 530 тенге.

#### 4.2.2 Затраты на электроэнергию

Эта статья \_\_\_\_\_ включает затраты на электроэнергию. \_\_\_\_\_ Общая сумма затрат \_\_\_\_\_ рассчитывается по формуле (4.2):

$$Z_{\text{э}} = \sum_{i=1}^n M_i * K_i * T_i * Ц, \quad (4.2)$$

где  $M_i$  – паспортная мощность  $i$ -го электрооборудования, кВт;  
 $K_i$  – коэффициент использования мощности  $i$ -го электрооборудования (принимается  $K_i=0,7$ );  
 $T_i$  – время работы  $i$ -го оборудования за весь период разработки ПП ч;  
 $Ц$  – цена электроэнергии, тг/кВт×ч;  
 $i$  – вид электрооборудования;  
 $n$  – количество электрооборудования.

С 1 января 2020 года цена на электроэнергию по тарифу ТОО «АлматыЭнергоСбыт» передачи в вашем данном составляет 19,17 тенге за 1 кВтч с НДС.

$$Z_{\text{э ноутбук}} = 0,4 * 0,7 * 216 * 19,17 = 1159,40 \text{ тг,}$$

$$Z_{\text{э принтер}} = 0,5 * 0,7 * 20 * 19,17 = 134,19 \text{ тг.}$$



Таблица 4.4 – Затраты на технологические нужды

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена электроэнергии г/кВт*ч	Сумма, тг
Ноутбук	0,4	0,7	216	19,17	1159,40
Принтер	0,5	0,7	20	19,17	134,19
Итого затраты на электроэнергию					1293,59

#### 4.2.3 Затраты на оплату труда

В этой статье рассчитывается основная и дополнительная заработная плата всех работников, которые были задействованы в разработке программного продукта. В разработке ПП был задействован один работник - разработчик-программист.

Средняя заработная плата разработчика-программиста на 2020 год составляет 250 000 тенге.

Часовая тарифная ставка будет рассчитываться по формуле:

$$З_ч = \frac{\Phi_{мес}}{Ч}, \quad (4.3)$$

где  $\Phi_{мес}$ . - среднемесячная заработная плата разработчика программы, тг.;

$Ч$  - среднее количество рабочих часов в месяц (принимается 160 часов).

Часовая ставка разработчика-программиста будет равняться:

$$З_ч = \frac{250\,000}{160} = 1562,5 \text{ тенге в час.}$$

Общая сумма затрат на оплату труда рассчитывается по формуле (4.4):

$$З_{тр} = \sum_{i=1}^n ЧС_i \times T_i, \quad (4.4)$$

где  $ЧС_i$  – часовая ставка  $i$ -го работника, тг;  
 $T_i$  – трудоемкость разработки ПП, чел.×ч;  
 $i$  – категория работника;

n – количество работников, занятых разработкой ПП.

Трудоемкость разработки программного продукта составляет 216 часов (на основе таблицы 4.1).

Тогда, общая сумма затрат на оплату труда составляет:

$$З_{тр} = 1562,5 \times 216 = 337\,500 \text{ тг.}$$

Таблица 4.5 – Затраты на оплату труда

Работник	Трудоемкость разработки ПП, чел. х ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик - программист	1 х 216	1562,5	337 500
Итого затрат на оплату труда			337 500

Дополнительная заработная плата принимается равной 10% от основной заработной платы работников. Расчет производится по формуле:

$$З_{доп} = З_{тр} \times 10\% \quad (4.5)$$

Отсюда,

$$З_{доп} = 337500 \times 0,1 = 33\,750 \text{ тг.}$$

Фонд заработной платы вычисляется по формуле:

$$\Phi_{зп} = З_{тр.} + З_{доп} \quad (4.6)$$

Следовательно, фонд заработной платы разработчика-программиста составляет:

$$\Phi_{зп} = 337\,500 + 33\,750 = 371\,250 \text{ тг.}$$

#### 4.2.4 Социальный налог

Социальный налог включает сумму, которая рассчитывается как 9,5% от затрат на оплату труда всех работников, занятых разработкой программного продукта. Пенсионные отчисления будут составлять 10% от затрат на оплату труда и не будут облагаться социальным налогом.

Затраты на налоговые отчисления рассчитаны в таблице 4.6

Таблица 4.6 – Затраты на налоговые отчисления

СО (Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	11 694,38
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	7 425,00
СН (Социальный налог)	9,5	(ЗП - ОПВ - ВОСМС)*9,5%-СО	19 694,81
Всего уплаченные налоги			38 814,19

#### 4.2.5 Амортизация основных фондов

Как известно, амортизация основных фондов состоит из суммы амортизационных отчислений от стоимости программного обеспечения (ПО) и оборудования, используемых при разработке ПП.

Сумму \_\_\_амортизационных отчислений необходимо определить \_\_\_по формуле (4.7):

$$Z_{AM} = \sum_{i=1}^n \frac{\Phi_i \times H_{Ai} \times T_{НИРi}}{100 \times T_{Эфи}}, \quad (4.7)$$

где  $\Phi_i$  – стоимость  $i$ -го ОФ, тг;

$H_{Ai}$  – годовая норма амортизации  $i$ -го ОФ, %;

$T_{НИРi}$  – время работы  $i$ -го ОФ за весь период разработки ПП, ч;

$T_{Эфи}$  – эффективный фонд времени работы  $i$ -го ОФ за год, ч/год (за 2020 год  $T_{Эфи} = 1968$ );

$n$  – количество ОФ;

$i$  – вид ОФ.

Годовая норма \_\_\_амортизации ОФ определяется с учетом возможного срока полезного использования ОФ:

$$H_{Ai} = \frac{100}{T_{Hi}}, \quad (4.8)$$

где  $T_{Hi}$  – возможный срок использования  $i$ -го ОФ, год.

Возможный срок использования ОФ может быть принят от 3 до 10 лет.

Возможный срок полезного использования ноутбука принят 5 лет, беспроводной мыши - 3 года, принтера - 6 лет.

$$H_{Ai} = \frac{100}{5} = 20\%,$$

$$H_{Ai} = \frac{100}{3} \approx 33,33\%,$$

$$H_{Ai} = \frac{100}{6} \approx 16,67\%.$$

Используя формулу (4.7), заполним таблицу 4.6 для отображения амортизации основных фондов.

Расчеты амортизации:

$$Z_{AM1} = \frac{90\,000 \times 20 \times 216}{100 \times 1968} = 1975,61 \text{ тг},$$

$$Z_{AM2} = \frac{7\,290 \times 33,33 \times 216}{100 \times 1968} = 266,68 \text{ тг},$$

$$Z_{AM3} = \frac{68\,990 \times 16,67 \times 20}{100 \times 1968} = 116,88 \text{ тг}.$$

Таблица 4.7 – Амортизация основных фондов

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Ноутбук	90 000	20	216	1975,61
Беспроводная мышь	7 290	33,33	216	266,68
Принтер	68 990	16,67	20	116,88
Итого амортизация основных фондов				2359,17

### 4.3 Смета затрат на разработку ПП

Посчитав все затраты на разработку программного продукта, составляется смета затрат по форме, приведенной в таблице 4.8.

Таблица 4.8 – Смета затрат на разработку ПП

Статья затрат	Сумма, тг
Материальные затраты	211 530

*Продолжение таблицы 4.8*

Затраты на оплату труда	337 500
Налоги от заработной платы	38 814,19
Затраты на электроэнергию	1293,59
Амортизация основных фондов	2359,17
Итого по смете	591 496,95

#### **4.4 Определение возможной (договорной) цены ПП**

Величина возможной (договорной) цены программного продукта будет устанавливаться в зависимости от сроков выполнения работы, качества и эффективности разработанного ПП.

Договорная цена (Ц<sub>д</sub>) для прикладных ПП рассчитывается по формуле:

$$Ц_{д} = З_{\text{НИР}} \times \left(1 + \frac{P}{100}\right), \quad (4.9)$$

где  $Z_{\text{НИР}}$  – затраты на разработку ПП (из таблицы 4.9), тг;

$P$  – средний уровень рентабельности ПП – 25%.

$$Ц_{д} = 591\,496,95 \times (1 + 0,25) = 739\,371,19 \text{ тг.}$$

Далее необходимо определить цену реализации с учетом налога на добавленную стоимость (НДС) при ставке, установленной в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Ц_{р} = Ц_{д} + Ц_{д} \times \text{НДС}, \quad (4.10)$$

$$Ц_{р} = 739\,371,19 + 739\,371,19 \times 0,12 = 828\,096 \text{ тг.}$$

#### **4.5 Оценка социально-экономических результатов функционирования программного продукта**

Основываясь на произведенных расчетах, капиталовложения проекта получились небольшими, следовательно, срок окупаемости будет коротким, что выгодно для моего предприятия.

Срок окупаемости разработанного программного продукта рассчитывается исходя из затрат на заработную плату работников, участвующих в процессе. Например, средний показатель заработной платы консультанта-дефектолога составляет 120 000 тг, налоговые отчисления 12

552 тенге, совокупные издержки в месяц - 132 552 тенге, в годовом исчислении - 1 590 624 тенге.

Определим условно-годовую экономию, экономическую эффективность и срок окупаемости продукта.

Так как после внедрения информационной обучающей системы для детей с расстройством аутистического спектра не понадобится консультант-дефектолог, то условно-годовая экономия составит 1 590 624 тенге.

Экономическая эффективность проекта будет:

$$E_p = \frac{\mathcal{E}_{yg}}{K}, \quad (4.11)$$

где  $\mathcal{E}_{yg}$  - условно-годовая экономия - 1 590 624 тг;

$K$  - капиталовложения - 828 096 тг.

Тогда,

$$E_p = \frac{\mathcal{E}_{yg}}{K} = \frac{1\,590\,624}{828\,096} = 1,92 \text{ или } 192\%.$$

С каждого вложенного тенге будет получена прибыль 1,92 тенге.  
Расчетный срок окупаемости капитальных вложений составляет:

$$T_p = \frac{1}{E_p}, \quad (4.12)$$

где  $E_p$  - коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{E_p} = \frac{1}{1,92} = 0,52 \text{ года или } 6,25 \text{ месяц.}$$

#### 4.6 Выводы по экономической части

В данной работе проведены расчеты затрат и общая экономическая оценка проекта по разработке информационной обучающей системы для детей с расстройством аутистического спектра. Приведен расчет затрат, связанных с закупкой оборудования и программного обеспечения, расчет эксплуатационных расходов: фонд оплаты труда, социальный налог, расходы на электроэнергию, амортизационные отчисления.

Стоимость разработки программного продукта составила 828 096 тенге, в нее заложены все возможные затраты при разработке. Исходя из полученных расчетов, можно утверждать, что реализация проекта будет экономически целесообразна при условии высокого качества продукта, что позволит получить прибыль по истечению определенного времени.

Таким образом, создание и реализация данного программного продукта будет экономически выгодным, эффективным и рентабельным.

## 5 Безопасность жизнедеятельности

### 5.1 Роль анализаторов в операторской деятельности

Исследование факторов, с помощью которых можно исследовать человеческие раздражители, является физиологической основой формирования информационной модели. Информация, полученная с помощью исследуемых показателей, называется сенсорной (чувственной), а процесс ее получения называется сенсорным восприятием или сенсорной активностью.

Любой исследуемый фактор включает в себя:

- рецептор;
- проводящие нервные пути;
- центр в коре больших полушарий головного мозга [24].

Основной функцией рецептора является превращение активного раздражителя в нервный процесс. Вход рецептора имеет приспособление к приему сигналов определенного вида – вибрация, звук, свет и т.п. Его выход передает сигнал, который является равным для любого входа нервной системы. Это позволяет рассматривать рецепторы как устройство для кодирования информации.

Самое большое значение для профессиональной деятельности оператора имеет орган зрения – 90% всей информации поступает именно через него. За ним по значимости стоит орган слуха, а на третьем месте – тактильный (или осязательный). Участие других органов чувств в профессиональной деятельности оператора намного меньше.

Основные характеристики любого органа чувств – это пороги. Они подразделяются на:

- оперативные;
- абсолютные (верхний и нижний);
- дифференциальные.

Нижний абсолютный порог чувствительности – это минимальная величина раздражителя, вызывающая едва заметное ощущение. Сигналы, величина которых меньше нижнего абсолютного порога чувствительности, человеком не считываются.

Увеличение сигнала до более верхнего абсолютного порога чувствительности вызывает у человека болевые ощущения. Интервал между нижним и верхним абсолютными порогами чувствительности называется охватом чувствительности анализатора.

Дифференциальный порог – это наименьшая разница между двумя раздражителями или между двумя состояниями одного раздражителя, которое вызывает еле заметное различие ощущений.

Оперативный порог определяется как минимальная величина разницы между сигналами, при которой точность и скорость различения достигает своего максимального значения.



Основываясь на вышесказанном, можно сформулировать общие условия к сигналам раздражителям:

1) различия между сигналами не должны превышать оперативный порог чувствительности, в противном случае возникает усталость;

2) важные и значимые сигналы должны располагаться зонах, которые соответствуют участкам рецепторной поверхности с максимальной чувствительностью;

3) при проектировании индикаторных устройств необходимо правильно выбрать тип (модальность) сигнала и модальность работы анализаторов;

4) интенсивность сигнала должна соответствовать средним значениям покрытия чувствительности анализатора, что создает оптимальные условия для приема и обработки информации;

5) должна быть обеспечена разница между сигналами, которая превышает дифференциальный порог чувствительности;

6) при проектировании средств отображения информации (СОИ), помимо изучения возможностей конкретных органов чувств, необходимо учитывать межанализаторные связи. Это необходимо делать при предъявлении оператору полимодальных сигналов, то есть сигналов различной модальности.

Полимодальные сигналы можно использовать в таких ситуациях как:

1) дублирование – сигнал передается одновременно на несколько органов чувств для того, чтобы повысить надежность передачи информации (сигналы тревоги);

2) распределение поступающей информации между органами чувств для того, чтобы избежать их перегрузки, требует учета пропускных способностей органов чувств;

3) переключение активности органов чувств – борьба с монотонностью операторской работы.

## **5.2 Сравнительный анализ средств отображения информации**

Человек-оператор получает информацию с помощью средств отображения информации (СОИ), где ход процесса или состояние объекта наблюдения представлены в закодированном виде в форме, удобной для восприятия человеком.

Средства отображения информации могут быть использованы для следующих целей:

1) считывания количественных и качественных показателей;

2) контрольного считывания показателей;

3) регуляция настройки параметров.

Любые СОИ должны соответствовать следующим инженерно-психологическим условиям:

1) предоставить работнику необходимую и достаточную информацию для оценки ситуации и возможности принятия правильного решения, а также осуществление контроля за его выполнением;

2) информация должна быть представлена в тот момент, когда это становится необходимым;

3) форма предоставления информации должна учитывать психофизиологические возможности оператора в плане восприятия, специфику его профессиональной деятельности и условий труда;

4) полученная информация должна правильно отражать положение и состояние объекта управления, быть обеспечена определенным количеством времени, достаточным для ее обработки;

5) предоставлять оператору дополнительные сведения по запросу, а также обеспечивать надежное восприятие сигналов тревоги;

6) информационный поток должен соответствовать пропускной способности оператора [25].

Наиболее важные эргономические требования, предъявляемые к средствам отображения информации можно оценить по техническим характеристикам мониторов. Проведем сравнительный анализ 24-дюймовых мониторов различных фирм-производителей. На основе сравнительных данных выбирается монитор с наилучшими эргономическими показателями.

Таблица 5.1 – Сравнительный анализ мониторов по техническим характеристикам на соответствие эргономическим требованиям

Параметр (эргономический показатель)	Технические характеристики мониторов	Марки мониторов		
		DELL P2418D	Philips 245E1S	Samsung S24D300H
Отношение ширины к высоте	Габаритные параметры (Размер видимой области экрана)	525,7 мм х 295,7 мм	526,9 мм х 296,4 мм	531,36 мм х 298,89 мм
Одномоментная переработка информации	Время отклика пикселя	8 мс (обычный режим); 5 мс (ускоренный режим) (от серого к серому)	4 мс (серый к серому)*	2 мс
Контрастность изображения (пределы регулировки)	Динамическая контрастность	8M:1 AMD FreeSync	8M:1 AMD FreeSync	5M:1 AMD FreeSync
Интервал яркости	Тип подсветки монитора	Система W-LED	Система W-LED	Система W-LED

Продолжение таблицы 5.1

Число градаций яркости	Величина контраста и яркости	1000:1 (номинал) 300 кд/м <sup>2</sup> (номинал)	1000:1 (номинал) 250 кд/м <sup>2</sup>	1000:1 (номинал) 250 кд/м <sup>2</sup>
Обеспечение четкости изображения	Угол обзора по вертикали и горизонтали, разрешение монитора	178° по вертикали/178° по горизонтали, разрешение 2560x1440 (16:9) при 60 Гц	178° по вертикали/178° по горизонтали, разрешение 2560x1440 (16:9) при 75 Гц	160° по вертикали/170° по горизонтали, разрешение 1920x1080 (16:9)
Частота обновления кадров	Максимальная частота обновления экрана	60 Гц	48 Гц – 75 Гц	56 Гц – 75 Гц
Соотношение яркости полезного изображения и яркости помех	Тип покрытия матрицы	антибликовое покрытие с твердостью 3H	антиблик, жесткость 3H, матовость 25 %	матовое
Критическая частота мелькания	Тип матрицы	IPS (планарная коммутация)	IPS (планарная коммутация)	TN (Twisted Nematic)

При эргономическом обосновании выбора монитора следует особое внимание уделять следующим показателям:

1) тип матрицы – влияет на качество (контрастность, яркость, цветопередача, углы обзора и т.п.):

а) TN – одна из дешевых матриц, небольшое время отклика, низкая цветопередача и минимальный угол обзора, вероятность появления битых пикселей (белых точек на экране). Подойдет для работы с офисными программами и приложениями, для работы с графикой – нет. Из-за низкой цветопередачи данный тип матрицы уступает IPS;

б) IPS – более лучшая цветопередача, чем у TN, однако цена в разы выше. Хорошие углы обзора: картинка не искажается, и цвета не теряют своей яркости и оттенков. Время отклика высокое (около 60 мс). Хорошо подойдет для профессиональной деятельности (работа с графикой, фотографиями и т.д.);

2) использование технологии улучшения изображения. Технология AMD FreeSync – технология, которая динамически синхронизирует количество кадров с частотой обновления монитора, улучшая изображение. Благодаря этому изображение кажется более плавным, так же за счет технологии контроля резкости картинка получается четкая и реалистичная;

3) время отклика отвечает за плавность воспроизведения картинки в динамических сценах, чем меньше значение этого параметра, тем лучше качество изображения;

4) угол обзора по вертикали и горизонтали проявляется в ухудшении изображения при взгляде на экран под углом: падает контрастность и снижается точность передачи цветов. Оптимальное значение углов обзора по вертикали и горизонтали, позволяющее пользоваться монитором без особых ограничений, составляет 165-175 градусов;

5) разрешение монитора – размер изображения, отображаемого на экране монитора, измеряемое в пикселях. Чем больше количество пикселей, тем четче и качественней отображение картинки, меньшее число пикселей вызывает ее интерполяцию и снижает уровень четкости. Для мониторов размером 24-дюйма обычно устанавливается разрешение 1920x1080 или 2560x1440 пикселей, что обозначает количество точек по вертикали к количеству точек по горизонтали. Чем выше разрешение, тем больше информации может отображаться на мониторе;

6) яркость характеризует интенсивность свечения экрана и измеряется в канделах на квадратный метр (кд/кв.м). Она является одной из важных деталей при работе за компьютером, так как сказывается на усталости глаз. Низкий показатель яркости монитора будет напрягать глаза, что приводит к дискомфорту при работе за компьютером;

7) контрастность определяется как отношение яркости белого цвета на экране к яркости черного. Высокая контрастность делает изображение более «осязаемым» и «живым». Минимальный рекомендуемый уровень контрастности для монитора – 500:1.

### **5.3 Вывод по разделу безопасность жизнедеятельности**

В результате сравнения мониторов Dell, Philips и Samsung по техническим характеристикам на соответствие эргономическим требованиям наиболее эргономичным является Philips 245E1S, у которого время отклика наименьшее по сравнению с другими двумя, так же у данной марки тип матрицы IPS, который исключает эффект мерцания экрана, что является очень важным показателем при длительной работе за монитором. Монитор поддерживает технологию улучшения качества изображения AMD FreeSync, которая позволяет получить плавную смену кадров на мониторе с максимально возможной производительностью и комфортом, улучшает качество изображения после масштабирования (увеличения), картинка получается четкой и не размытой.

## Заключение

В результате выполнения дипломного проекта была разработана информационная обучающая система для детей с расстройством аутистического спектра. Следовательно, была достигнута поставленная цель. Данная система будет служить коммуникатором между ребенком и другими людьми, которая будет помогать развивать коммуникативные навыки, речевую активность и познавательные способности. Были реализованы такие задачи как:

- изучены теоретические аспекты расстройств аутистического спектра;
- проведен подробный анализ существующих обучающих программ для детей с РАС;
- определена структура данной программы;
- разработан удобный и понятный интерфейс;
- дана оценка экономической эффективности внедрения системы.

Проблема обучения детей расстройствами аутистического спектра является актуальной как в Казахстане, так и во всем мире. Высокая частота развития аутистических нарушений требует создания и внедрения современных систем поддержки и обучения для улучшения качества жизни детей с РАС, что также является одной из приоритетных задач специального и инклюзивного образования в нашей стране.

Таким образом, изучив теоретические аспекты обучения детей с РАС и проведя сравнительный анализ существующих обучающих программ, предназначенных для детей, была разработана полнофункциональная и готовая к использованию обучающая система, которая позволит развивать:

- функции социального взаимодействия;
- коммуникативные навыки;
- спектр интересов;
- понимание эмоций;
- познавательные способности;
- речевые навыки.

Разработанная обучающая система может быть внедрена в специальные и инклюзивные образовательные учреждения, в том числе вспомогательные школы, коррекционные кабинеты, специальные дошкольные учреждения, образовательные центры. Кроме того, программа пригодна для индивидуального использования в домашних условиях, например, родителями.

## Список литературы

- 1 Могилев А.В., Пак Н.И., Хеннер Е.К. Информатика. 2003 – 848 с.
- 2 Учебное пособие к курсу лекций по дисциплине «Компьютерные технологии» Часть 2 / Сост. П.С. Сабуров., Владим. гос. ун-т., Владимир, 2012 – с. 103
- 3 Проект «Аутизм. Мир один для всех». Фонд Булата Утемуратова <https://utemuratovfund.org/projects/autizm-mir-odin-dlya-vsekh>
- 4 Международная классификация болезней 10-го пересмотра (МКБ-10) <https://mkb-10.com/>
- 5 Frith U. Autism: Very short introduction. Published in the United States by Oxford University Press Inc., New York, 2008, 145 p.
- 6 Инклюзивное образование. Настольная книга педагога, работающего с детьми с ОВЗ: Методическое пособие — М.: Гуманитарный изд. центр ВЛАДОС, 2011. - 167 с.
- 7 Питерс Т. Аутизм: от теоретического понимания к педагогическому воздействию. СПб.: Институт специальной педагогики и психологии, 2009, - 192 с.
- 8 Dawson G. et al. Children with autism fail to orient to naturally occurring social stimuli //Journal of autism and developmental disorders. 1998, Vol 28, No. 6, p. 479-485.
- 9 Никольская О., Фомина Т., Цыпотан С. Вместе с учителем. Ребенок с аутизмом в обычной школе. М.: "Чистые пруды", 2006. - 32 с.
- 10 Тебенова Қ.С. Арнайы психология: оқулық / Қ. С.Тебенова, А.Р. Рымханова ; ҚР Білім және ғылым м-гі. - Алматы : ЖШС РПБК "Дәуір", 2011. - 262 с.
- 11 Бабаева Ю.Д., Войскунский А.Е. Проблемы детей, одаренных в области ИТ// Аутизм. URL: <http://www.autism.ru/read.asp?id=121&vol=0>
- 12 Ülkü Pişkin Abidoğlu et al. Importance of Computer-Aided Education for Children with Autism Spectrum Disorder (ASD), EURASIA Journal of Mathematics Science and Technology Education, 2017, 13(8):4957-4964 DOI: 10.12973/eurasia.2017.00975a
- 13 Escobedo L., David H. Nguyen et al. MOSOCO: A Mobile Assistive Tool to Support Children with Autism Practicing Social Skills in Real-Life Situations, 2012 DOI: 10.1145/2207676.2208649
- 14 Обзор бесплатных программ-коммуникаторов. URL: <https://autizm-aura.ru/?p=3912>
- 15 Коммуникатор ДАР. URL: <https://www.kv.by/content/339783-samoe-strashnoe-eto-kogda-tebya-ne-ponimayut-mobilnoe-prilozhenie-kommunikator-dar-so>
- 16 McGrath M. C# Programming in easy steps. Published by: Easy Steps Limited, 2016. – 357 p.
- 17 Griffiths I. Programming C# 8.0. Published by O'Reilly Media, Inc., USA, 2019. – 800 p.

- 18 Шилдт Г. С#: Учебный курс – СПб.: Питер; Киев: ВНУ, 2003. – 512 с.
- 19 Петцольд Ч. Программирование с использованием Microsoft Windows Forms. Мастер-класс / Пер. с англ. – М.: Русская редакция; СПб.: Питер, 2006. – 432 стр.
- 20 Лабор В.В. Си Шарп: Создание приложений для Windows/ В. В. Лабор. – Мн.: Харвест, 2003. - 384 с.
- 21 Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML: пер. с англ. / Л. А. Мацяшек. - Москва [и др.]: Вильямс, 2002. – 432 с.
- 22 Савихин О.Г. Основы разработки приложений в Microsoft Visual Studio .NET
- 23 Боканова Г.Ш.. Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 – Вычислительная техника и программное обеспечение, 5В070300 – «Информационные системы», 5В060200 – «Информатика». – Алматы: АУЭС, 2020 – 35 с.
- 24 Эргономика: учебное пособие / сост. А.И. Фех; Томский политехнический университет. – Томск: Изд-во Томского политехнического университета, 2014. – 119 с.
- 25 Новиков С.Г., Маслова Т.Н., Копылова Л.Н. Электронный учебно-методический комплекс «Безопасность жизнедеятельности», Московский энергетический институт (ТУ). URL: [http://ftemk.mpei.ru/bgd/bgd\\_start.htm](http://ftemk.mpei.ru/bgd/bgd_start.htm)
- 26 СанПиН «Санитарно-эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека»
- 27 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы: Санитарно-эпидемиологические правила и нормативы. – М: Федеральный центр госсанэпиднадзора Минздрава России, 2003. – 54 с.

# Приложение А

## Техническое задание

### 1 Общие сведения

#### 1.1 Наименование системы

Полное наименование системы:

Информационная обучающая системы для детей с расстройством аутистического спектра

#### 1.2 Сроки начала и окончания работ

Дата начала: 17.02.2020

Дата окончания: 09.05.2020

### 2 Назначение и цели создания системы

#### 2.1 Назначение системы

Обучающая система предназначена для обучения детей с расстройством аутистического спектра, развития коммуникативных навыков, речевой активности и познавательных способностей.

### 3 Рекомендации к разработке программы

Обучающая система может быть разработана на языке С# с использованием СУБД Microsoft SQL Server.

### 4 Требование к внешнему виду системы

Обязательным условием является простота и понятливость интерфейса, не яркая, приятная цветовая гамма.

### 4 Технические требования

Специальных технических требований для ПК для работы приложения нет, кроме как Windows 7 или. Подойдёт любой ПК, на котором может работать Windows.

### 5 Экономические требования

- возможная (договорная) цена продукта составила 828 096 тг;
- стоимость разработки продукта составила 591 497 тг.



## Приложение Б (ЛИСТИНГ ПРОГРАММЫ)

```
Форма ComMain.cs
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
using FontAwesome.Sharp;
using System.Runtime.InteropServices;
using CommunicationS.Вкладки;
using System.Speech.Synthesis;
using System.IO;
namespace CommunicationS
{
    public partial class ComMain : Form
    {
        //поля
        private IconButton curBtn;
        private Panel leftBordBtn;
        private Form CurMainForm;
        public static List<string> Phrase = new List<string>();
        public static void SpeakForMe(string phrase)
        {
            SpeechSynthesizer synth = new SpeechSynthesizer();//создание объекта
            using (MemoryStream streamAudio = new MemoryStream())
            {
                System.Media.SoundPlayer m_SoundPlayer = new System.Media.SoundPlayer()
                synth.SetOutputToWaveStream(streamAudio);
                synth.Speak(phrase);
                streamAudio.Position = 0;
                m_SoundPlayer.Stream = streamAudio;
                m_SoundPlayer.Play();
                synth.SetOutputToNull();
            } }
        public ComMain()
        {
            Program.mainf = this;
            InitializeComponent();
            leftBordBtn = new Panel();
            leftBordBtn.Size = new Size(7, 45);
            panelMenu.Controls.Add(leftBordBtn);
            //форма
            this.Text = string.Empty;
            this.ControlBox = false;
            this.DoubleBuffered = true;
            this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;
        }
        //структура
        private struct RGBc {
```

*Продолжение приложения Б*

```
public static Color color1 = Color.FromArgb(255, 255, 0);
public static Color color2 = Color.FromArgb(255, 128, 0);
public static Color color3 = Color.FromArgb(0, 192, 0);
public static Color color4 = Color.FromArgb(255, 255, 204);
public static Color color5 = Color.FromArgb(0, 255, 255);
public static Color color6 = Color.FromArgb(255, 128, 149);
public static Color color7 = Color.FromArgb(0, 255, 127);
public static Color color8 = Color.FromArgb(255, 0, 0);
}
//методы
private void ActivateBtn(object senderBtn, Color color)
{
    if (senderBtn != null)
    {
        DisableBtn();
        //button
        curBtn = (IconButton)senderBtn;
        curBtn.BackColor = Color.FromArgb(37, 36, 81);
        curBtn.ForeColor = color;
        curBtn.TextAlign = ContentAlignment.MiddleCenter;
        curBtn.IconColor = color;
        curBtn.TextImageRelation = TextImageRelation.TextBeforeImage;
        curBtn.ImageAlign = ContentAlignment.MiddleRight;
        //left border button
        leftBordBtn.BackColor = color;
        leftBordBtn.Location = new Point(8, curBtn.Location.Y);
        leftBordBtn.Visible = true;
        leftBordBtn.BringToFront();
        //Current Child Form Icon
        iconCurMainForms.IconChar = curBtn.IconChar;
        iconCurMainForms.IconColor = color;
    } }
private void DisableBtn()
{
    if (curBtn != null)
    {
        curBtn.BackColor = Color.FromArgb(65, 41, 76);
        curBtn.ForeColor = Color.Gainsboro;
        curBtn.TextAlign = ContentAlignment.MiddleLeft;
        curBtn.IconColor = Color.Gainsboro;
        curBtn.TextImageRelation = TextImageRelation.ImageBeforeText;
        curBtn.ImageAlign = ContentAlignment.MiddleLeft;
    } }
private void OpenMainForm(Form mainForm)
{
    if(CurMainForm != null)
    {
        //открываем только форму
```

```
    CurMainForm.Close();
}
CurMainForm = mainForm;
//конец
mainForm.TopLevel = false;
mainForm.FormBorderStyle = FormBorderStyle.None;
mainForm.Dock = DockStyle.Fill;
panelDesk.Controls.Add(mainForm);
panelDesk.Tag = mainForm;
mainForm.BringToFront();
mainForm.Show();
labelHome.Text = mainForm.Text;
}
//кнопки вкладок
private void iconButton1_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color1);
    OpenMainForm(new FormЖелание());
    panel2.Visible = true;
    iconPlay.Visible = true;
}
private void iconButton2_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color2);
    OpenMainForm(new FormЛюди());
    panel2.Visible = true;
}
private void iconButton3_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color3);
    OpenMainForm(new FormЭмоции());
    panel2.Visible = true;
}
private void iconButton4_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color4);
    OpenMainForm(new FormЕда());
    panel2.Visible = true;
}
private void iconButton5_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color5);
    OpenMainForm(new FormМеста());
    panel2.Visible = true;
}
private void iconButton6_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color6);
    OpenMainForm(new FormДействия());
```

*Продолжение приложения Б*

```
        panel2.Visible = true;
    }
//название панели
private void Reset()
{
    DisableBtn();
    leftBordBtn.Visible = false;
    iconCurMainForms.IconChar = IconChar.Home;
    iconCurMainForms.IconColor = Color.MediumPurple;
    labelHome.Text = "Главная";}
//возможность перетаскивания формы
[DllImport("user32.DLL", EntryPoint = "ReleaseCapture")]
private extern static void ReleaseCapture();
[DllImport("user32.DLL", EntryPoint = "SendMessage")]
private extern static void SendMessage(System.IntPtr hWnd, int wMsg, int wParam, int
iParam);
private void panelTitle_MouseDown(object sender, MouseEventArgs e)
{
    ReleaseCapture();
    SendMessage(this.Handle, 0x112, 0xf012, 0);
}
private void btnExit_Click(object sender, EventArgs e)
{
    Application.Exit();
}
private void btnMax_Click(object sender, EventArgs e)
{
    if (WindowState == FormWindowState.Normal)
        WindowState = FormWindowState.Maximized;
    else WindowState = FormWindowState.Normal;
}
private void btnMin_Click(object sender, EventArgs e)
{
    WindowState = FormWindowState.Minimized;
}
//удаляем прозрачную границу в развернутом состоянии
private void FormMainMenu_Resize(object sender, EventArgs e)
{
    if (WindowState == FormWindowState.Maximized)
        FormBorderStyle = FormBorderStyle.None;
    else
        FormBorderStyle = FormBorderStyle.Sizable;
}
private void BtnHome_Click_1(object sender, EventArgs e)
{
    if (CurMainForm != null)
    {
        CurMainForm.Close();
    }
    Reset();
}
```

*Продолжение приложения Б*

```
panel2.Visible = false;
}
private void panelLogo_Click(object sender, EventArgs e)
{
    if (CurMainForm != null)
    {
        CurMainForm.Close();
    }
    Reset();
    panel2.Visible = false;
}
private void iconPlay_Click(object sender, EventArgs e)
{
    string ph = "";
    for (int i = 0; i < Phrase.Count; i++)
    {
        ph += Phrase[i];
    }
    if (ph != "") {SpeakForMe(ph); } else { MessageBox.Show("Фраза не заполнена!",
"Пустое поле!", MessageBoxButtons.OK, MessageBoxIcon.Information); }
    GC.Collect(); }
private void iconButton7_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color7);
    OpenMainForm(new FormДобавить());
    panel2.Visible = true;
}
private void iconButton8_Click(object sender, EventArgs e)
{
    ActivateButn(sender, RGBc.color8);
    OpenMainForm(new FormАлфавит());
    panel2.Visible = true;
} } }
```

Форма FormЖелание.cs

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using System.Speech.Synthesis;
using System.IO;
namespace CommunicationS.Вкладки
{
    public partial class FormЖелание : Form
    {
        public FormЖелание()
        {
            Program.fwish = this;
            InitializeComponent(); }
    }
```

*Продолжение приложения Б*

```
public static List<PictureBox> pictureboxes = new List<PictureBox>();
public static void SpeakForMe(string phrase)
{
    SpeechSynthesizer synth = new SpeechSynthesizer();
    using (MemoryStream streamAudio = new MemoryStream())
    {
        System.Media.SoundPlayer m_SoundPlayer = new System.Media.SoundPlayer();
        synth.SetOutputToWaveStream(streamAudio);
        synth.Speak(phrase);
        streamAudio.Position = 0;
        m_SoundPlayer.Stream = streamAudio;
        m_SoundPlayer.Play();
        synth.SetOutputToNull();
    }
}
private int picturesCount()
{
    pictureboxes.Clear();
    for (int i = 0; i < 30; i++)
    {
        PictureBox pic = Program.mainf.panel2.Controls.Find("pictureBox" + (i + 1),
true).FirstOrDefault() as PictureBox;
        if (pic != null)
        {
            pictureboxes.Add(pic);
        }
    }
    return pictureboxes.Count;
}
void AddPhrase(Button button)
{
    int PosX = 10;
    for (int i = 0; i < picturesCount(); i++)
    {
        PosX += pictureboxes[i].Size.Width + 10;
    }
    PictureBox pictureBox1 = new PictureBox();
    pictureBox1.Name = "pictureBox" + (picturesCount() + 1);
    pictureBox1.Location = new Point(PosX, Program.mainf.panel2.Location.Y + 10);
    pictureBox1.Size = new Size(button.Size.Width, button.Size.Height);
    pictureBox1.BackgroundImage = button.BackgroundImage;
    pictureBox1.BackgroundImageLayout = ImageLayout.Stretch;
    pictureBox1.Anchor = (AnchorStyles.Top | AnchorStyles.Left);
    pictureBox1.Click += new EventHandler(pictureBox1_Click);
    Program.mainf.panel2.Controls.Add(pictureBox1);
    pictureBox1.BringToFront();
    GC.Collect();
}
private void pictureBox1_Click(object sender, EventArgs e)
```

```
{
    PictureBox p = sender as PictureBox;
    if (Convert.ToInt32(p.Name.Split(new[] { "pictureBox" }, StringSplitOptions.None)[1])
== picturesCount())
    {
        Program.mainf.panel2.Controls.Remove(p);
        ComMain.Phrase.RemoveAt(Convert.ToInt32(p.Name.Split(new[] { "pictureBox" },
StringSplitOptions.None)[1]) - 1);
    }
    GC.Collect();
}
private void button1_Click(object sender, EventArgs e)
{
    SpeakForMe("Да ");
    AddPhrase(button1);
    ComMain.Phrase.Add("Да ");
}
private void button2_Click(object sender, EventArgs e)
{
    SpeakForMe("Нет ");
    AddPhrase(button2);
    ComMain.Phrase.Add("Нет ");
}
private void button3_Click(object sender, EventArgs e)
{
    SpeakForMe("Я хочу ");
    AddPhrase(button3);
    ComMain.Phrase.Add("Я хочу ");
private void button4_Click(object sender, EventArgs e)
{
    SpeakForMe("Не хочу ");
    AddPhrase(button4);
    ComMain.Phrase.Add("Не хочу ");
}
private void button5_Click(object sender, EventArgs e)
{
    SpeakForMe("Не знаю ");
    AddPhrase(button5);
    ComMain.Phrase.Add("Не знаю ");
}
private void button6_Click(object sender, EventArgs e)
{
    SpeakForMe("Стоп ");
    AddPhrase(button6);
    ComMain.Phrase.Add("Стоп ");
}
private void button7_Click(object sender, EventArgs e)
{
    SpeakForMe("Помоги ");
```

*Продолжение приложения Б*

```
AddPhrase(button7);  
ComMain.Phrase.Add("Помоги ");  
}}}
```

Форма FormДобавить.cs

```
using System;  
using System.Collections.Generic;  
using System.Data;  
using System.Data.SqlClient;  
using System.Drawing;  
using System.Drawing.Imaging;  
using System.IO;  
using System.Linq;  
using System.Windows.Forms;  
namespace CommunicationS.Вкладки  
{  
    public partial class FormДобавить : Form  
    {  
        public List<Button> customButtons = new List<Button>();  
        public Button RightClickedButton = new Button();  
        public static List<PictureBox> pictureboxes = new List<PictureBox>();  
        public FormДобавить()  
        {  
            Program.fadd = this;  
            InitializeComponent();  
  
            //связь с БД  
            SqlConnection connection = new SqlConnection("Data  
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\  
CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True");  
            string sql = "SELECT * FROM UserImages";  
            SqlCommand sCom = new SqlCommand(sql, connection);  
            List<int> IDs = new List<int>();  
            List<byte[]> Images = new List<byte[]>();  
            connection.Open();  
            using (SqlDataReader read = sCom.ExecuteReader())  
            {  
                while (read.Read())  
                {  
                    IDs.Add(int.Parse(read["Id"].ToString()));  
                    Images.Add((byte[])read["Image"]);  
                }  
            }  
            connection.Close();  
            for (int i = 0; i < IDs.Count; i++)  
            {  
                Button button1 = new Button();  
                button1.Name = "customButton" + IDs[i];  
                button1.Location = new Point(buttonplus.Location.X, buttonplus.Location.Y);  
                button1.Size = new Size(buttonplus.Size.Width, buttonplus.Size.Height);  
                button1.BackgroundImage = ByteToImage(Images[i]);  
            }  
        }  
    }  
}
```



*Продолжение приложения Б*

```
        button1.BackgroundImageLayout = ImageLayout.Stretch;
        button1.Anchor = (AnchorStyles.Top | AnchorStyles.Left);
        button1.Click += new EventHandler(customButton1_Click);
        button1.MouseDown += new MouseEventHandler(mouseCustomButton1_Click);
        this.Controls.Add(button1);
        button1.BringToFront();
        buttonplus.Location = new Point(buttonplus.Location.X + 260, buttonplus.Location.Y);
    } }
private int customButtonsCount()
{
    customButtons.Clear();
    for (int i = 0; i < 30; i++)
    {
        Button btn = this.Controls.Find("customButton" + (i + 1), true).FirstOrDefault() as
Button;
        if (btn != null)
        {
            customButtons.Add(btn);
        }
    }
    return customButtons.Count;
}
private int picturesCount()
{
    pictureboxes.Clear();
    for (int i = 0; i < 30; i++)
    {
        PictureBox pic = Program.mainf.panel2.Controls.Find("pictureBox" + (i + 1),
true).FirstOrDefault() as PictureBox;
        if (pic != null)
        {
            pictureboxes.Add(pic);
        }
    }
    return pictureboxes.Count;
}
void AddPhrase(Button button)
{
    int PosX = 10;
    for (int i = 0; i < picturesCount(); i++)
    {
        PosX += pictureboxes[i].Size.Width + 10;
    }
    PictureBox pictureBox1 = new PictureBox();
    pictureBox1.Name = "pictureBox" + (picturesCount() + 1);
    pictureBox1.Location = new Point(PosX, Program.mainf.panel2.Location.Y + 10);
    pictureBox1.Size = new Size(button.Size.Width, button.Size.Height);
    pictureBox1.BackgroundImage = button.BackgroundImage;
    pictureBox1.BackgroundImageLayout = ImageLayout.Stretch;
    pictureBox1.Anchor = (AnchorStyles.Top | AnchorStyles.Left);
```

*Продолжение приложения Б*

```
pictureBox1.Click += new EventHandler(pictureBox1_Click);
Program.mainf.panel2.Controls.Add(pictureBox1);
pictureBox1.BringToFront();
GC.Collect();
}
private void pictureBox1_Click(object sender, EventArgs e)
{
    PictureBox p = sender as PictureBox;
    if (Convert.ToInt32(p.Name.Split(new[] { "pictureBox" }, StringSplitOptions.None)[1])
== picturesCount())
    {
        Program.mainf.panel2.Controls.Remove(p);
        ComMain.Phrase.RemoveAt(Convert.ToInt32(p.Name.Split(new[] { "pictureBox" },
StringSplitOptions.None)[1]) - 1);
    }
    GC.Collect();
}
private void button1_Click(object sender, EventArgs e)
{
    ComMain.SpeakForMe("Зелёный ");
    AddPhrase(button1);
    ComMain.Phrase.Add("Зелёный ");
}
private void customButton1_Click(object sender, EventArgs e)
{
    Button b = sender as Button;
    AddPhrase(b);
    SqlConnection connection = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True");
    string sql = "SELECT Phrase FROM UserImages where Id = " +
Convert.ToInt32(b.Name.Split(new[] { "customButton" }, StringSplitOptions.None)[1]);
    SqlCommand sCom = new SqlCommand(sql, connection);
    string ButtonPhrase = "";
    connection.Open();
    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            ButtonPhrase = read["Phrase"].ToString();
        }
    }
    connection.Close();
    ComMain.Phrase.Add(ButtonPhrase.Replace(" ", "") + " ");
}
private void mouseCustomButton1_Click(object sender, MouseEventArgs e)
{
    Button b = sender as Button;
    if (e.Button == MouseButton.Right)
    {
```

```
RightClickedButton = b;
ContextMenuStrip removeMenu = new ContextMenuStrip();
removeMenu.Name = "removeMenu1";
removeMenu.Location = new Point(e.Location.X, e.Location.Y);
removeMenu.Size = new Size(100, 20);
removeMenu.Text = "RemoveMenu";
b.ContextMenuStrip = removeMenu;
removeMenu.Show();
ToolStripMenuItem RemovePicture = new ToolStripMenuItem("File");
RemovePicture.Text = "Удалить картинку";
RemovePicture.TextAlign = ContentAlignment.BottomRight;
RemovePicture.ToolTipText = "Удалить фразу";
RemovePicture.Click += new EventHandler(RemovePictureClick);
RemovePicture.Image = b.BackgroundImage;
removeMenu.Items.Add(RemovePicture);
} }
private void RemovePictureClick(object sender, EventArgs e)
{
    if (MessageBox.Show("Вы хотите удалить выбранную картинку?", "Удаление",
        MessageBoxButtons.YesNo) == DialogResult.Yes)
    {
        int CustomButtonID = Convert.ToInt32(RightClickedButton.Name.Split(new[] {
            "customButton" }, StringSplitOptions.None)[1]);
        SqlConnection connection = new SqlConnection("Data
        Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
        CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True");
        string sql = "DELETE FROM UserImages WHERE Id = " + CustomButtonID + """;
        SqlCommand sCom = new SqlCommand(sql, connection);
        connection.Open();
        sCom.ExecuteNonQuery();
        connection.Close();
        if ((customButtonsCount() - CustomButtonID) > 0)
        {
            for (int i = (customButtonsCount() - (customButtonsCount() - CustomButtonID)); i <
                customButtonsCount(); i++)
            {
                customButtons[i].Location = new Point(customButtons[i].Location.X - 260,
                customButtons[i].Location.Y);
            } }
            buttonplus.Location = new Point(buttonplus.Location.X - 260, buttonplus.Location.Y);
            this.Controls.Remove(RightClickedButton);
            int count = customButtonsCount();
            for (int i = 0; i < count; i++)
            {
                string connectionString = "Data
                Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
                CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True";
                connection = new SqlConnection(connectionString);
                SqlCommand sCommand = new SqlCommand();
```

*Продолжение приложения Б*

```
sCommand.Connection = connection;
connection.Open();
sCommand.CommandType = CommandType.Text;
sCommand.CommandText = String.Format("UPDATE UserImages SET Id =
@NewID where Id = " + Convert.ToInt32(customButtons[i].Name.Split(new[] { "customButton"
}, StringSplitOptions.None)[1]));
sCommand.Parameters.AddWithValue("@NewID", (i + 1));
sCommand.ExecuteNonQuery();
connection.Close();
customButtons[i].Name = "customButton" + (i + 1);
}
MessageBox.Show("Картинка №" + CustomButtonID + " успешно удалена.",
"Успешное удаление!", MessageBoxButtons.OK, MessageBoxIcon.Information);
}}
private void buttonplus_Click(object sender, EventArgs e)
{
    bool isCancelled;
    Button button1 = new Button();
    button1.Name = "customButton" + (customButtonsCount() + 1);
    button1.Location = new Point(buttonplus.Location.X, buttonplus.Location.Y);
    button1.Size = new Size(buttonplus.Size.Width, buttonplus.Size.Height);
    button1.BackgroundImage = null;
    button1.BackgroundImageLayout = ImageLayout.Stretch;
    button1.Anchor = (AnchorStyles.Top | AnchorStyles.Left);
    button1.Click += new EventHandler(customButton1_Click);
    button1.MouseDown += new MouseEventHandler(mouseCustomButton1_Click);
    this.Controls.Add(button1);
    button1.BringToFront();
    using (OpenFileDialog openFileDialog = new OpenFileDialog())
    {
        openFileDialog.InitialDirectory = "c:\\";
        openFileDialog.Filter =
"**.bmp|*.bmp|*.ico|*.ico|*.gif|*.gif|*.jpg|*.jpg|*.jpeg|*.jpeg|*.png|*.png| All files (*.*)|*.*";
        openFileDialog.FilterIndex = 7;
        openFileDialog.RestoreDirectory = true;
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            button1.BackgroundImage = Image.FromFile(openFileDialog.FileName);
            ImageFormat thisFormat = GetImageFormat(openFileDialog.FileName);
            SaveImage(Image.ToByte(Image.FromFile(openFileDialog.FileName), thisFormat),
"Image");
            isCancelled = false;
        }
        else
        {
            isCancelled = true;
            this.Controls.Remove(button1);
        }
    }
    if (isCancelled == false)
```

*Продолжение приложения Б*

```
{
    NewPhrase PhraseAskForm = new NewPhrase();
    PhraseAskForm.ShowDialog();
    SqlConnection connection = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True");
    string sql = "SELECT * FROM UserImages";
    SqlCommand sCom = new SqlCommand(sql, connection);
    int max = 0;
    connection.Open();
    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            if (int.Parse(read["Id"].ToString()) > max)
            {
                max = int.Parse(read["Id"].ToString());
            }
        }
    }
    connection.Close();
    string connectionString = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True";
    connection = new SqlConnection(connectionString);
    SqlCommand sCommand = new SqlCommand();
    sCommand.Connection = connection;
    connection.Open();
    sCommand.CommandType = CommandType.Text;
    sCommand.CommandText = String.Format("UPDATE UserImages SET Phrase =
@phrase where Id = " + max);
    sCommand.Parameters.AddWithValue("@phrase", NewPhrase.NewPicturePhrase);
    sCommand.ExecuteNonQuery();
    connection.Close();
    buttonplus.Location = new Point(buttonplus.Location.X + 260, buttonplus.Location.Y);
}
}

void SaveImage(byte[] imagen, string valueName)
{
    SqlConnection connection = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True");
    string sql = "SELECT * FROM UserImages";
    SqlCommand sCom = new SqlCommand(sql, connection);
    int max = 0;
    connection.Open();
    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            if (int.Parse(read["Id"].ToString()) > max)
            {
```

## Продолжение приложения Б

```
        max = int.Parse(read["Id"].ToString());
    } } }
    max++;
    connection.Close();
    string connectionString = "Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\alipb\\Desktop\\ДИПЛОМ\\
CommunicationS\\CommunicationS\\Вкладки\\Images.mdf;Integrated Security=True";
    connection = new SqlConnection(connectionString);
    SqlCommand sCommand = new SqlCommand();
    sCommand.Connection = connection;
    SqlParameter param = new SqlParameter("@0", DbType.Binary);
    param.Value = imagen;
    sCommand.Parameters.Add(param);
    connection.Open();
    sCommand.CommandType = CommandType.Text;
    sCommand.CommandText = String.Format("INSERT INTO UserImages(Id," +
valueName + ") VALUES (" + max + ",@0)");
    sCommand.ExecuteNonQuery();
    connection.Close();
}

private static ImageFormat GetImageFormat(string fileName)
{
    string extension = Path.GetExtension(fileName);
    if (string.IsNullOrEmpty(extension))
        throw new ArgumentException(
            string.Format("Unable to determine file extension for fileName: {0}", fileName));
    switch (extension.ToLower())
    {
        case @".bmp":
            return ImageFormat.Bmp;
        case @".gif":
            return ImageFormat.Gif;
        case @".icon":
            return ImageFormat.Icon;
        case @".jpg":
        case @".jpeg":
            return ImageFormat.Jpeg;
        case @".png":
            return ImageFormat.Png;
        default:
            throw new NotImplementedException();
    }
}

public Image ByteToImage(byte[] imageBytes)
{
    MemoryStream ms = new MemoryStream(imageBytes, 0, imageBytes.Length);
    ms.Write(imageBytes, 0, imageBytes.Length);
    Image image = new Bitmap(ms);
    return image;
}
```

*Продолжение приложения Б*

```
}  
public byte[] ImageToByte(Image image, System.Drawing.Imaging.ImageFormat format)  
{  
    using (MemoryStream ms = new MemoryStream())  
    {  
        image.Save(ms, format);  
        byte[] imageBytes = ms.ToArray();  
        return imageBytes;  
    }  
}  
private void flowLayoutPanel1_Paint(object sender, PaintEventArgs e)  
{  
    buttonplus.Visible = true;  
}}}
```

## Приложение В Акт внедрения

Утверждаю  
Руководитель КППК №1  
г. Талдыкорган  
«08» мая 2020 г.

### АКТ ВНЕДРЕНИЯ

Настоящий акт составлен о том, что результаты выпускной работы студента НАО АУЭС гр. Инф-16-2 очной формы обучения Алипбаевой А.А. на тему *«Разработка информационной обучающей системы для детей с расстройством аутистического спектра»* внедрены в КГУ «Кабинет психолого-педагогической коррекции № 1» города Талдыкорган и используются для работы с детьми с расстройством аутистического спектра. Использование разработанной обучающей системы Алипбаевой А.А. поможет в развитии социальных навыков и когнитивных способностей детей с расстройством аутистического спектра.

Руководитель КГУ «Кабинет  
психолого-педагогической  
коррекции № 1» г. Талдыкорган



*Г.Б. Омиржанова*  
Г.Б. Омиржанова