

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ЮЖНО-УРАЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУВО «ЮУрГГПУ»)

ФАКУЛЬТЕТ МАТЕМАТИКИ, ФИЗИКИ, ИНФОРМАТИКИ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МЕТОДИКЕ
ОБУЧЕНИЯ ИНФОРМАТИКЕ

**Развитие алгоритмического мышления посредством
изучения создания мобильных приложений в
App Inventor**

**Выпускная квалификационная работа по направлению
44.04.01 Педагогическое образование**

Направленность программы магистратуры

«Информатика и робототехника в образовании»

Форма обучения заочная

Проверка на объем заимствований:
88,41 % авторского текста
Работа рекомендована к защите
рекомендована/не рекомендована
«12» сентября 20__ г.
зав. кафедрой
ИИТиМОИ ЮУрГГПУ
Рузаков Андрей
Александрович

Выполнил:
Студент группы ЗФ-313-276-2-1
Мякушева Дарья Петровна Мяку
Научный руководитель:
к.п.н., доцент кафедры ИИТиМОИ
Дмитриева Ольга
Александровна

Челябинск
2023 год



МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ЮЖНО-УРАЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУВО «ЮУрГГПУ»)

ФАКУЛЬТЕТ МАТЕМАТИКИ, ФИЗИКИ, ИНФОРМАТИКИ
КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И МЕТОДИКЕ
ОБУЧЕНИЯ ИНФОРМАТИКЕ

**Развитие алгоритмического мышления посредством
изучения создания мобильных приложений в
App Inventor**

Выпускная квалификационная работа по направлению

44.04.01 Педагогическое образование

Направленность программы магистратуры

«Информатика и робототехника в образовании»

Форма обучения заочная

Проверка на объем заимствований:
_____ % авторского текста
Работа _____ к защите
рекомендована/не рекомендована
«__» _____ 20__ г.
зав.кафедрой
ИИТиМОИ ЮУрГГПУ
_____ Рузаков Андрей
Александрович

Выполнил:
Студент группы ЗФ-313-276-2-1
Мякушева Дарья Петровна
Научный руководитель:
к.п.н., доцент кафедры ИИТиМОИ
_____ Дмитриева Ольга
Александровна

Челябинск

2023 год

СОДЕРЖАНИЕ

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ ШКОЛЬНИКОВ ПОСРЕДСТВОМ СОЗДАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В СРЕДЕ ВИЗУАЛЬНОЙ РАЗРАБОТКИ ANDROID ПРИЛОЖЕНИЙ MIT APP INVENTOR.....	13
1.1 Сущность и содержание понятия «алгоритмическое мышление».....	13
1.2 Структура и содержание учебной программы содержательной линии «Компьютерное мышление» обновленного содержания образования Республики Казахстан по предмету «Информатика»	19
Вывод по главе 1	35
ГЛАВА 2. РАЗВИТИЕ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ ШКОЛЬНИКОВ ПОСРЕДСТВОМ ИЗУЧЕНИЯ СОЗДАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В MIT APP INVENTOR.....	37
2.1 Возможности среды визуальной разработки Android-приложений MIT App Inventor как средство развития алгоритмического мышления школьников.	37
2.2 Разработка программы внеурочной деятельности «Основы программирования мобильных приложений под ОС Android»	42
2.3 Разработка программы «Основы программирования мобильных приложений под ОС Android» на образовательной платформе stepik.org ..	52
Выводы по главе 2.....	57
ГЛАВА 3. ПЕДАГОГИЧЕСКИЙ ЭКСПЕРИМЕНТ И ОЦЕНКА ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ МЕТОДИКИ.....	59
3.1 Организация и проведение педагогического эксперимента.....	59
3.2 Анализ результатов исследования уровня развития алгоритмического мышления школьников посредством создания мобильных приложений во внеурочной деятельности	61

Выводы по 3 главе.....	75
ЗАКЛЮЧЕНИЕ	77
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	81
ПРИЛОЖЕНИЕ 1	87
ПРИЛОЖЕНИЕ 2	92

ВВЕДЕНИЕ

В современном Казахстане формируется новая образовательная система, ориентированная на глобальное образовательное пространство. Сегодня в современном обществе возрастает потребность в людях самостоятельных, активных, неординарно мыслящих, способных быстро адаптироваться к изменяющимся ситуациям, строить алгоритмы и следовать им, творчески подходить к решению проблем, добиваться поставленных целей. Система образования отказывается от традиционного представления результатов обучения в виде формирования знаний, умений и навыков. Обучение в школе должно быть построено так, чтобы выпускники могли самостоятельно ставить и достигать целей, умело реагировать на разные жизненные ситуации. Таков социальный заказ государства школе сегодня. Но готова ли система образования к этому вызову?

Академик А. П. Ершов, отмечая значимость законов обработки информации, способов перехода от знания к действию, способности строить программы и рассуждать о них, предвидеть результаты их выполнения в поступательном развитии человеческого интеллекта, предложил включить эти вопросы в ряд фундаментальных компонентов общего образования вместе с математическими и лингвистическими концепциями [19]. Высказывая идею, академик А. П. Ершов писал, что «программирование – это вторая грамотность», ученый предполагал, что со временем цифровые технологии неизбежно окажут огромное влияние на интеллектуальное развитие человечества, содержание образования, основные положения теории и практики обучения [19]. Возможно, поэтому вопросы формирования и развития вычислительных навыков и алгоритмического мышления все больше привлекают внимание исследователей.

Предмет информатика обладает максимальным потенциалом для формирования и развития алгоритмического мышления учащихся. В учебной программе по предмету «Информатика» которая разработана в соответствии с Государственными общеобязательными стандартами дошкольного воспитания и обучения, начального, основного среднего и общего среднего, технического и профессионального, послесреднего образования, которая утверждена приказом Министра просвещения Республики Казахстан от 3 августа 2022 года № 348 «Об утверждении государственных общеобязательных стандартов дошкольного воспитания и обучения, начального, основного среднего и общего среднего, технического и профессионального, послесреднего образования» четко обозначаются задачи развивать алгоритмическое мышление [31]. Одной из задач обучения является «развивать алгоритмическое и операционное мышление, логические, интеллектуальные и творческие способности учащихся средствами информационно-коммуникационных технологий». Таким образом, можно сделать вывод: формирование алгоритмического мышления у школьников является важной целью обучения на разных ступенях обучения информатики в школе.

Алгоритмизация и программирование является важным разделом учебной дисциплины курса информатики в школе. Изучение программирования в школьном курсе информатики является основой алгоритмической подготовки учащихся и создает предпосылки для развития алгоритмического мышления школьников. Уже с появлением первых языков программирования, разработчики и педагоги стремились к тому, чтобы максимально упростить обучение. Блочное программирование – самая последняя разработка. Процесс программирования стал больше похож на собирание конструктора, где каждая деталь имеет свои имя и назначение. Если конструктор собрать правильно, то получится настоящий рабочий код. До недавнего времени проблема обучения школьников

средних классов программированию казалась неразрешимой – прежде всего из-за отсутствия инструмента, который, с одной стороны, был бы достаточно прост в освоении, а с другой – позволял бы создавать действительно стоящие продукты. Попытки поголовно обучать школьников Бейсику или Паскалю приводили только к тому, что предмет «информатика» оказывался по зубам лишь очень узкому кругу учащихся – тем, кому в силу интеллектуальных особенностей, семейного воспитания или чрезвычайного везения с учителем удавалось продвинуться в освоении программирования дальше, чем другим. Для большинства остальных школьников информатика так и оставалась чем-то недоступным.

Обстановка начала меняться в начале 2000-х годов, с возникновением и развитием визуальных языков программирования, первооткрывателем которых по праву считается язык Scratch. Этот язык совершил настоящий переворот в школьном преподавании программирования. Программировать на Scratch легко, как складывать детскую мозаику-пазл. Операторы языка представляют собой цветные блоки. Перетаскивая и соединяя их, мы создаём программы. Ошибку в написании программы на язык Scratch сделать просто невозможно — если блоки не встанут рядом, пазл просто не сложится.

Естественным развитием этого подхода стало создания облачной среды визуальной разработки приложений для платформы OS Android MIT App Inventor, разработанной профессором технологического института (MIT) Халом Абелсоном в 2010 году. В основе его – тот же принцип перетаскивания визуальных кирпичиков и собирания программы из блоков. Отличие MIT App Inventor от Scratch состоит в том, что MIT App Inventor не на настольное использование, а предназначен для создания приложений под мобильное устройство – смартфон или планшет с ОС Android. Приложение умеет многое, например, «понимать» данные

акселерометра мобильного гаджета, управлять встроенной камерой, видеть, как ориентирован телефон в пространстве и многое другое.

В MIT App Inventor разноцветные блоки дают подсказку, что с помощью них можно выполнить. Программирование – одновременно творческий и логический процесс, с чем новичкам справиться нелегко. Применение блоков минимизирует когнитивную нагрузку до восприятия считанного числа цветных деталей. Обучаемый больше сосредотачивается на том, как именно их расставить правильно. Ошибку в коде всегда найти нелегко. Представьте, как тяжело ребенку, когда его программа не работает просто потому, что он забыл поставить точку с запятой в конце строки. Множество мелких ошибок часто отбивает желание у новичков продолжать программировать. Блоки уменьшают количество таких ошибок. Достаточно логически думать, как правильно соединить их друг с другом. Таким образом, можно смело сказать, что блочное программирование идеально подходит для детей. Программирование в облачной среде визуальной разработки приложений осуществляться через сознательную и активную деятельность учащихся, поскольку решаемые задачи имеют для них практическую направленность и персональную значимость. Это полностью соответствует требованиями ГОСО, в которых предусматривается вовлечение обучающихся в конкретную практическую деятельность для достижения личностных, метапредметных и предметных образовательных результатов. Таким образом, создавая мобильные приложения учащиеся смогут развивать алгоритмическое мышление.

Обзор научной литературы по разработке мобильных приложений позволили выявить необходимость создания программы, позволяющей организовать процесс обучения учащихся разработке мобильных приложений. Исходя из вышесказанного, актуальность исследования определяется наличием противоречия между: требованиями ГОСО к уровню развития алгоритмического мышления учащихся и недостаточной

практической реализацией возможностей по использованию облачной среды визуальной разработки приложений во внеурочной деятельности для удовлетворения данных требований.

Из противоречия вытекает **проблема исследования**: как результативно применять облачную среду визуальной разработки приложений во внеурочной деятельности для развития алгоритмического мышления школьников?

Можно сделать вывод, что **актуальность исследования** задается современными тенденциями развития информатики.

Объект исследования: алгоритмическое мышление учащихся.

Предмет исследования: развитие алгоритмического мышления обучающихся средствами визуальной среды разработки Android-приложений MIT App Inventor.

Цель исследования: разработать, обосновать и экспериментально проверить программу внеурочной деятельности «Основы программирования мобильных приложений под ОС Android» для учащихся, которая позволит повысить уровень алгоритмического мышления обучающихся общеобразовательной школы.

Гипотеза исследования: развитие алгоритмического мышления обучающихся будет более эффективно, если разработать и внедрить для учащихся седьмых классов программу внеурочной деятельности «Основы программирования мобильных приложений под ОС Android», направленную на развитие алгоритмического мышления.

Поставленная цель и выдвинутая гипотеза предполагают **решение следующих задач**:

1. Рассмотреть сущность и содержание понятия «алгоритмическое мышление».
2. Определить значение содержательной линии «Компьютерное

мышление» для развития алгоритмического мышления на уроках информатики.

3. Рассмотреть основные подходы к обучению программирования в школьном курсе информатики.

4. Охарактеризовать среду визуальной разработки приложений MIT App Inventor для ОС Android, как средство обучения программированию учащихся и развития алгоритмического мышления.

5. Подобрать диагностический инструментарий и проанализировать уровень развития алгоритмического мышления школьников.

6. Разработать программу внеурочной деятельности для школьников по созданию мобильных приложений в MIT App Inventor и разместить ее на образовательной платформе stepik.org.

7. Организовать и провести экспериментальную апробацию результатов исследования в учебном процессе.

8. Проанализировать результаты эксперимента и оценить эффективность введения программы во внеурочную деятельность.

База исследования: КГУ «Общеобразовательная школа № 23 им. М. Козыбаева отдела образования города Костаная» Управления образования акимата Костанайской области.

Для решения поставленных задач использовались следующие **методы исследования:** теоретический анализ литературы по проблематике исследования; сравнение и обобщение педагогических исследований, системный анализ и классификация, психодиагностические, анкетирование, статистические методы обработки экспериментальных данных, методы количественной обработки данных (метод математической обработки Т-критерий Вилкоксона), педагогический эксперимент.

Этапы исследования:

Первый этап – поисково-подготовительный (2020 г.). Был проведен теоретический анализ проблемной области, изучена литература по

проблеме исследования, сформулирована гипотеза, объект и предмет исследования.

Второй этап – опытно-экспериментальный (2020-2021 гг.). Разработана и реализована программа внеурочной деятельности по обучению школьников созданию мобильных приложений в MIT App Inventor, обеспечивающих повышение уровня развития алгоритмического мышления обучающихся; проведен констатирующий эксперимент, определена экспериментальная группа.

Третий этап – контрольно-обобщающий (2022 г.). Завершен качественный и количественный анализ полученных результатов; статистическая обработка данных; результаты оформлены в виде диссертационной работы.

Теоретико-методологической базой исследования являются:

– работы ученых, рассматривавших значимость алгоритмического стиля мышления в обществе (И.Н. Антипов, В.М. Монахов, А.П. Ершов, М.П. Лапчик)

– теоретико-методические основы обучения алгоритмизации и программированию (А.П. Ершов, М.П. Лапчик, А.А. Кузнецов, А.Г. Гейн, Е.К. Хеннер, И. Г. Семакин, Л.Л. Босова, Н.Д. Угринович, Т.А. Степанова, А.А. Дуванов)

– диссертационным исследованиям по формированию и развитию алгоритмического мышления (В.В. Калитина, А. И. Газейкина, И.Н. Слинкина, Т.Н. Лебедева), позволившим выделить приоритетные методы и приемы обучения алгоритмизации.

Научно-практическая ценность и значимость исследования заключатся в том, что разработана программа внеурочной деятельности для школьников по созданию мобильных приложений в MIT App Inventor и размещена на образовательной платформе stepik.org. Материалы, размещенные на платформе, могут быть использованы учителями

информатики общеобразовательных школ, преподавателями колледжей и ВУЗов, студентами.

Базой для проведения педагогического эксперимента послужила КГУ «Общеобразовательная школа 23 им. М. Козыбаева отдела образования города Костаная» Управления образования акимата Костанайской области.

Диссертационная работа состоит из введения, трех глав, раскрывающих последовательность и суть исследования, описание результатов педагогического эксперимента и синтез данных, полученных в результате теоретического анализа, заключения. К работе прилагаются список использованной литературы и материалы, дополняющие основной текст работы.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ РАЗВИТИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ ШКОЛЬНИКОВ ПОСРЕДСТВОМ СОЗДАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В СРЕДЕ ВИЗУАЛЬНОЙ РАЗРАБОТКИ ANDROID ПРИЛОЖЕНИЙ MITAPP INVENTOR

1.1 Сущность и содержание понятия «алгоритмическое мышление»

Проблема развития алгоритмического мышления многогранна. Чтобы полностью понять значение, нам необходимо рассмотреть такие важные понятия, как «мышление» и «алгоритм». Любая деятельность человека представляет собой решение определенной последовательности задач. При выполнении задач разум играет главную роль. В школьном возрасте мышление становится доминирующей функцией и играет особую роль в жизни ребенка. Л.С. Выготский определял мышление как «психические процессы, отражающие объективную действительность и представляющие собой высшую ступень человеческого познания» [7]. Он пишет, что один из типов мышления, который развивается у детей младшего школьного возраста – это алгоритмическое мышление [7].

В научной психологической литературе алгоритмическое мышление не рассматривается как отдельный и специфический тип. Некоторые исследователи рассматривают этот тип мышления (наряду с логическим мышлением) как тип теоретического мышления. Имеется суждение, что алгоритмическое мышление – это особый способ мышления, характерный для определенной деятельности. В этом подходе мышление понимается, в частности, как «открытая система интеллектуальных стратегий, приемов, навыков и операций, к которой человек склонен в силу своих индивидуальных особенностей (от систем ценностей и мотиваций до характерных признаков)» [13]. В этом отношении мышление обладает шестью качествами, зависящими от личностных характеристик человека:

быстротой мышления, гибкостью мышления, глубиной мышления, критичностью мышления, самостоятельностью, широтой мышления [13].

Проанализировав все точки зрения по основным научным работам и подходам, которые описаны Л. С. Выготским, С.Л. Рубинштейном, В.С. Мерлиным, О.Г. Леонтьевым, В.В. Давыдовым, можно сделать вывод, что данные авторы определяют стиль мышления как «систему действий, приемов, методов и соответствующих стратегий мышления, направленных на решение определенного типа задач и определяемых этими задачами». Исходя из этого, алгоритмически стиль мышления – это «система действий, приемов, методов и соответствующих стратегий мышления, направленных на решение теоретических и практических задач, в результате чего алгоритмы становятся конкретным продуктом человеческой деятельности» [16].

Если рассматривать работы ученых, исследователей, которые в своих трудах уделяют внимание мышлению, в частности, алгоритмическому мышлению, то можно выделить разнообразные точки зрения по определению данного понятия. В настоящее время разработано достаточно большое количество определений алгоритмического мышления. Это зависит, с какой стороны рассматривать это мышление.

Рассмотрим определения алгоритмического мышления, которые раскрывают различные авторы в своих трудах. Отечественные психологи Д.Н. Богоявленский и П.Я. Гальперин выделяют такие виды как «логическое мышление» и «логико-алгоритмическое мышление». По их мнению, логико-алгоритмическое мышление определяется в индуктивном и дедуктивном мышлении, умениях создавать логические цепочки, обращаться за помощью к поисковым системам составлять свои намерения так, чтобы они были записаны на некотором алгоритмическом языке [18].

В методических обоснованиях педагога Т. Н. Лебедевой, вслед за Д. Н. Богоявленским и П. Я. Гальпериним, алгоритмическое мышление

определено как «когнитивный процесс, который характеризуется четкой, целесообразной (или рациональной) последовательностью мыслительных процессов с присущими им детализацией и оптимизацией укрупненных блоков, сознательной фиксацией процесса получения конечного результата на языке исполнителя в формализованной форме с признанными семантическими и синтаксическими правилами» [13].

Психолог П.Я. Гальперин [12] различает «логическое мышление» и «логико-алгоритмическое мышление». Автор утверждает, что логико-алгоритмическое мышление определяется в индуктивном и дедуктивном мышлении, способностью создавать логические цепочки, обращаться за помощью к поисковым системам и составлять свои намерения так, чтобы они были записаны на некотором алгоритмическом языке [18].

Основоположник теории системного программирования А.П. Ершов использовал понятие операционного мышления для определения алгоритмического мышления. Оно включает в себя умение строить информационные модели для объектного и системного анализа, умение искать информацию, а в нужных случаях обращаться к ЭВМ для решения задач в предметной области [19]. Ю.А. Первин, Н.В. Беленов использовали этот подход при определении алгоритмического мышления.

Алгоритмическое мышление имеет свои общие и специфические характеристики по сравнению с другими стилями мышления. К общим свойствам алгоритмического мышления относятся полнота и результативность. Это помогает увидеть проблему в целом и включает в себя создание предварительного образа результатов решения проблемы. Свойства специфические алгоритмического мышления включают дискретность, абстракцию и сознательную фиксацию в языковой форме. Эти свойства представляют собой пошаговое выполнение алгоритма, что позволяет нам абстрагироваться от некоторых исходных данных, решить задачу в общем виде и представить алгоритм с использованием

формализованного языка. Составной частью алгоритмического мышления является способность формализовать задачу и разделить ее на дискретные составляющие логические блоки.

Кандидат педагогических наук А.И. Газейкина в своей статье «Формирование когнитивных универсальных учебных действий при обучении робототехнике учащихся основной школы» конкретизировала концепцию операционно-алгоритмического образа мышления. В этом случае операционный образ мышления характеризуется способностью учащегося действовать по заданному алгоритму, с умением его выполнять, «алгоритмическое мышление» – это особый тип мышления, подразумевающий умение создавать алгоритмы, требующие обдумывания решений, приводящих к концепции задачи в целом, обдумывания решения задачи большими блоками, а затем разработки процесса конечного результата и сознательной фиксации его в языковой форме [10].

В своей работе мы будем опираться на определение кандидата педагогических наук Анны Ивановны Газейкиной, и понимать алгоритмическое мышление как это особый тип мышления, при котором реализуются умения планировать свои действия, умения учитывать различные ситуации и действовать соответственно, умение легко рассуждать об алгоритмических процессах [10].

Проанализируем компоненты алгоритмического мышления, которые выделяют авторы.

Профессор Александр Георгиевич Гейн в книге для учителя «Методика изучения алгоритмизации с помощью учебных исполнителей» выделяет следующие компоненты алгоритмического мышления:

- исполнительский: умение точно и правильно создавать определённый продукт по известной схеме, образцу;
- технологический: умение самому придумать схему создания продукта (алгоритм);

- экспертный: умение дать качественную оценку готовому продукту, полученному по определённой схеме (алгоритму);
- аналитико-синтезирующий: умение на основе уже готового продукта, предлагать изменения в структуре самого продукта, тем самым меняя его алгоритм [16].

Ирина Николаевна Слинкина, кандидат педагогических наук, в своей диссертации «Использование компьютерной техники в процессе развития алгоритмического мышления у младших школьников» выделяет следующие компоненты алгоритмического мышления:

- анализ требуемого результата и выбор исходных данных;
- выделение операций, необходимых для решения той или иной задачи;
- выбор исполнителя, способного осуществить данные операции;
- упорядочение операций и выбор нужного решения;
- реализация процесса решения и соотнесение результатов с тем, что должны были получить [44].

Анна Ивановна Газейкина определяет алгоритмическое мышление следующими компонентами:

- наличие знаний об алгоритме и способе решения задач определённого уровня;
- выбор определённых операций для решения задач;
- построение модели процесса решения;
- реализация процесса, проверка правильности и результативности;
- анализ и коррекция исходных данных в случае несовпадения готовым результатом [11].

Рассмотрев различные мнения по поводу компонентов алгоритмического мышления, мы пришли к выводу, что можно обобщить предложенные компоненты: наличие знаний, исходных данных для выполнения операций над конкретной задачей, возможность построения

модели, возможность реализации последовательности действий и анализ, коррекция, проверка, соотнесение готового результата с тем, что должно получиться.

Подобно И.Н. Слинкиной, А.Г. Гейну, рассмотрим компоненты алгоритмического мышления:

- когнитивный (познавательный),
- деятельностный,
- рефлексивно-оценочный (рефлексивный).

Теоретическая модель исследуемого понятия представлена в таблице 1.

Таблица 1 – Модель понятия «алгоритмическое мышление»

Компоненты алгоритмического мышления		
познавательный	деятельностный	рефлексивный
Знания понятия «алгоритм», «исполнитель»	Умение применять знания об алгоритмах, использовать готовые алгоритмы, составлять свои алгоритмы	Умение формулировать цель, планировать, реализовывать, оценивать, рефлексировать свою деятельность

Таким образом, нами рассмотрено большое количество различных трактовок понятия «алгоритмическое мышление». Ученые, исследователи подходят с разнообразных точек зрения к определению данного понятия и выделяют разные компоненты алгоритмического мышления.

В структуре алгоритмического мышления можно выделить следующие компоненты: когнитивный, деятельностный и рефлексивно-оценочный.

1.2 Структура и содержание учебной программы содержательной линии «Компьютерное мышление» обновленного содержания образования Республики Казахстан по предмету «Информатика»

Алгоритмическое мышления формируется при изучении содержательной линии «Компьютерное мышление» на уроках информатики.

В Государственной программе «Цифровой Казахстан» по направлению «Развитие человеческого капитала» намечена реализация преобразований, охватывающих создание креативного общества для обеспечения перехода к новой реальности – доступности цифровых навыков у людей в экономике знаний, их эффективное использование в необходимых условиях, развитие и рост [33]. В связи с этим Министерство просвещения Республики Казахстана с учетом новых требований к подрастающему поколению и адаптации системы образования к потребностям новой индустриализации, в целях развития алгоритмического мышления и технических навыков пересмотрены и утверждены типовые учебные программы обновленного содержания по предмету «Информатика» общеобразовательных школ.

Изучение предмета информатики способствует пониманию учащимися того, что:

- навыки вычислительного мышления и моделирования, приобретенные в курсе информатики, могут быть использованы при анализе различных ситуаций;
- системы могут быть смоделированы посредством абстракции, алгоритмов и программирования;
- применение знаний из курса информатики может иметь ключевое влияние на развитие науки, техники, медицины, образования и культуры;

- использование навыков программирования позволяет создавать приложения, которые могут улучшить текущую деятельность и дать возможность появлению новых идей.

Обновление содержания образования характеризуется изменением содержания и структуры образования, определяющих качество образования в общеобразовательных школах страны, эффективной реализацией активных методов и способов обучения, системой критериального оценивания учебных достижений. В учебную программу также внесены значительные изменения и в содержание учебного предмета, прописаны новые цели обучения, отличные от прежних, предлагается изучение нового языка программирования. В методических рекомендациях анализируются особенности учебной программы обновленного содержания образования по предмету «Информатика». Формы и методы организации преподавания предмета, пути эффективной организации преподавания информатики приводятся на конкретных примерах.

На рисунке 1 представлено сравнение структур учебных программ 2013 и 2019 годов.

Целью изучения учебного предмета «Информатика» является обеспечение обучающихся базовыми знаниями, умениями и навыками в области компьютерных систем, информационных процессов, компьютерного мышления для эффективного использования современных информационных технологий на практике.

Структура типовой учебной программы для 5-9 классов уровня основного среднего образования 2013 год



Структура учебной программы для 5-9 классов уровня основного среднего образования по обновленному содержанию 2019 жыл



Рисунок 1 – Структура учебных программ 2013 и 2019 годов

Соответственно целям определены следующие задачи программы:

1. Формирование у обучающихся понимания роли информационных процессов в обществе, технических возможностей и перспектив.

2. Использование информационных технологий в различных сферах человеческой деятельности.

3. Развитие умений эффективно использовать информационные технологии в повседневной жизни, в учебе и дальнейшей трудовой деятельности.

4. Усвоение обучающимися базовых принципов работы компьютеров для анализа системы, разработки решения, формирования программного приложения и оценки своей продукции.

5. Развитие умения решать разнообразные задачи посредством анализа, абстракций, моделирования и программирования.

6. Развитие у обучающихся логического, алгоритмического, а также вычислительного мышления, включающего способность к обобщению и аналогии, разложению задачи на составные части и выделению общих закономерностей, нахождению эффективных и рациональных способов решения поставленных задач.

7. Формирование у обучающихся информационной культуры – соблюдение общепринятых правил, учет интересов личности и всего казахстанского общества.

8. Обогащение понятийного аппарата по предмету и овладение обучающимися академическим языком.

Максимальный объем учебной нагрузки учебного предмета «Информатика» в 5-9 классах составляет 1 час в неделю, что соответствует 36 часам в учебном году. Содержание учебного предмета организовано по разделам обучения. Разделы далее разбиты на подразделы.

В рисунке 2 приведено изменение в содержании учебной программы 2019 года в сравнении с программой 2013 года.

В новом содержании разделы «Моделирование», «Алгоритмизация», «Программирование» объединены в раздел «Компьютерное мышление». В этот раздел включен и подраздел «Робототехника».

Термин «компьютерное (вычислительное) мышление» появился в 1980 году. Его ввел специалист по разработке искусственного интеллекта доктор Сеймур Пайперт из Массачусетского технологического института [33].



Рисунок 2 – Содержание учебной программы 2019 года

Но сам стиль этого мышления существовал давно, а стал широко распространяться при создании первых электронно-вычислительных машин. С развитием технических и программных средств расширялось и количество решаемых задач, что приводило к новому содержанию и смысловому наполнению понятия вычислительное или компьютерное мышление. Одно из определений этого термина следующее: компьютерное или вычислительное мышление – это мыслительные процессы, участвующие в постановке проблем и представлении их решения в форме, которая может быть эффективно реализована с помощью человека или компьютера. По применению и смысловому наполнению компьютерное мышление включает навыки создания алгоритмов, разработки тестов,

поиска и исправления ошибок, умение сводить сложное к простому, переходить от конкретного к абстрактному, а также организовать последовательность и распознавание паттернов [33]. Навыки компьютерного мышления находят своё применение в самых разных сферах. Чтобы принимать верные решения, нацеленные на определённый результат, всегда полезно выделить главные элементы проблемы и понять, как они связаны с более мелкими её частями. Мыслительным инструментом для этого является алгоритмическое мышление. Отсюда можно сделать вывод, что термин «компьютерное (вычислительное) мышление» можно приравнять к алгоритмическому мышлению.

Раздел «Компьютерное мышление» исследует вышеупомянутые инструменты в сочетании со свободным и творческим мышлением и включает следующие подразделы:

1. Моделирование.
2. Алгоритмы.
3. Программирование.
4. Робототехника.

Основными принципами организации программы обучения с обновленным содержанием являются:

- спиральность к оформлению тематического контента;
- иерархия целей обучения согласно таксономии Блума, классифицированная по наиболее важным типам манипуляций с объектами на основе законов познания;
- педагогическое целеполагание по уровням образования и в процессе обучения. Это дает возможность максимально рассмотреть внутри предметную коммуникацию;
- наличие «сквозных тем» между предметами в рамках одного образовательного поля и в реализации межпредметных связей;

- соответствие тематических разделов и тематического содержания духу времени с упором на развитие социальных навыков.

Спиралевидная форма обучения предполагает, что повторение материала, который усложняется в процессе обучения, дает большие преимущества для развития современных учащихся.

Рассмотрим содержание учебной программы по разделу «Компьютерное мышление» в разрезе классов с 5 по 11 класс.

Таблица 2 – Базовое содержание учебного предмета «Информатика» для 5 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Программирование	<p>Определение углового наклона робота. Поворот робота на заданные градусы. Датчик цвета для организации движения робота. Датчик ультразвука для нахождения объекта.</p>	<p>(Начальное знакомство с роботами, алгоритмами, исполнителями линейных алгоритмов в игровой среде программирования (Scratch) введены в начальные классы).</p>
Робототехника	<p>Определение робота. Примеры разновидностей роботов и области их применения. Примеры технических достижений человечества в области робототехники. Принцип работы гироскопического датчика.</p>	<p>Робототехника впервые включена в программу 5 класса (Сборка базовой модели образовательного робота, загрузка и запуск программы для робота, управление движением и поведением робота введены в начальные классы).</p>

Таблица 3 – Базовое содержание учебного предмета «Информатика» для 6 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Алгоритмы.	Линейные алгоритмы на языке Python	Изучались на языке Паскаль/ Язык Python введен впервые.
Программирование.	Алфавит и синтаксис языка программирования. Типы данных.	Изучались на языке Паскаль/ Изучение языка Python введено впервые.

Таблица 4 – Базовое содержание учебного предмета «Информатика» для 7 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Алгоритмы	Запись разветвляющихся алгоритмов на языке программирования Python	Изучались в 8-м классе на языке Паскаль / Язык Python введен впервые.
Программирование	Чтение и запись файла, использование вложенных условий, использование составных условий на языке программирования Python.	Изучались в 8-м классе на языке Паскаль / Язык Python введен впервые.

Таблица 5 – Базовое содержание учебного предмета «Информатика» для 8 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Алгоритмы	Трассировка алгоритма.	Как отдельная тема рассматривалась
Программирование	Операторы цикла (цикл с параметром, цикл с предусловием, цикл с постусловием). Инструкции управления циклом.	Изучались в 9-м классе на языке Паскаль
Моделирование	Создание моделей задач на языке программирования Python.	Модели создавались языке Паскаль.

Таблица 6 – Базовое содержание учебного предмета «Информатика» для 9 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Программирование	Создание программы на языке программирования Python с использованием одномерных, двумерных массивов.	Массивы изучались на языке
Программирование	Подключение и использование готовых модулей библиотеки PyGame для создания окна, фона игры. Загрузка готовых персонажей для игры. Движение персонажей. Программирование игры по готовому сценарию.	Паскаль/ Создание игр на языке Python введено впервые

Таблица 7 – Базовое содержание учебного предмета «Информатика» для 9 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Алгоритмизация и программирование	Пользовательские функции и процедуры. Работа со строками. Работа с файлами. Методы сортировки. Алгоритмы на графах.	Строки и сортировки массивов изучались на Языке Паскаль. Пользовательские функции и процедуры, алгоритмы на графах изучаются впервые

Таблица 8 – Базовое содержание учебного предмета «Информатика» для 9 класса по разделу «Компьютерное мышление»

Подраздел	Темы	Сравнение с программой 2013 года
Искусственный интеллект	Искусственный интеллект. Проектирование искусственного интеллекта.	Темы введены в программу впервые
Мобильные приложения	Создание мобильного приложения. Умный дом. Разработка проекта умного дома.	Темы введены в программу впервые

Подразделы содержат в себе цели обучения по классам в виде ожидаемых результатов: навыки, умения, знания, понимания.

Таблица 9 – Цели обучения и уровни учебных целей по таксономии Блума подраздела «Алгоритмы»

Класс	Цели обучения	Уровни учебных целей по таксономии Блума
6	6.3.2.1 записывать линейные алгоритмы на языке программирования Python.	Знание, Понимание
7	7.3.2.1 записывать разветвляющиеся алгоритмы на языке программирования Python.	Знание, Понимание
8	8.3.2.1 осуществлять трассировку алгоритма	Понимание, Применение
9	9.3.2.1 применять алгоритмы сортировки.	Понимание, Применение, Анализ
10	10.5.1.4 реализовывать алгоритмы сортировки для решения практических задач 10.5.1.5 реализовывать алгоритмы поиска на графах для решения практических задач.	Понимание, Применение
11	11.3.4.3 проектировать нейронную сеть в электронных таблицах/ программах математического моделирования по готовому алгоритму.	Понимание, Применение, Анализ

Часовая нагрузка на изучение подраздела «Программирование» по классам приведены в таблице 11.

Таблица 11 – Часовая нагрузка на изучение подраздела «Программирование».

Класс	Количество часов
5	10 часов
6	10 часов
7	10 часов
8	20 часов
9	20 часов
10	15 часов
11	15 часов
Итого	100 часов

Хотелось бы отметить, что в обновленной программе произошел перенос изучения пропедевтического курса информатики из средних классов в начальные классы основной школы. Этот переход предполагает не только адаптацию предметного содержания к возрастным особенностям школьников, но и изменение изучения языков программирования, STEM элементов (робототехника, виртуальная реальность, 3D-печать).

Программирование – это одно из важнейших разделов «Информатики», А. П. Ершов говорил: «Программирование – вторая грамотность» [19]. Любая среда программирования способствует развитию алгоритмического типа мышления, значимость которого отметил в своих работах учёный Н. М. Амосов, считавший необходимым включение в школьную программу разработку алгоритмов, а также показавший, что использование алгоритмов помогает не только организовать мыслительную деятельность, но и описывать с их помощью процессы [3]. Всё человечество понимает важность, значимость и перспективность среды ИТ, но в мире всё равно существует недостаток программистов. Поэтому одной из главных задач сейчас является подготовка таких специалистов. Этой проблемой занимаются множество популярных компаний, которые предлагают обучение и стажировку с дальнейшим трудоустройством, например компании «Google» [5], «Яндекс» [5] или «Mail.ru» [5].

Министерство просвещения Республики Казахстан в целях решения данной проблемы за последние несколько лет увеличило количество грантов в высших учебных заведениях по направлениям ИТ. В Единый национальный экзамен для выпускников в 2022 году предмет «Информатика» включили в перечень профильных дисциплин ЕНТ (Единое национальное тестирование). Абитуриентам предстоит выполнить 35 тестовых заданий, 20 из них будут с выбором одного правильного ответа, 5 – с выбором одного правильного ответа в практическом задании,

10 – с выбором одного или нескольких правильных ответов. Максимальное количество баллов – 45. Профильный предмет «Информатика» на ЕНТ (Единое национальное тестирование) будут сдавать абитуриенты, которые будут поступать в вузы по следующим направлениям: «Подготовка учителей информатики»; «Информационные технологии»; «Информационная безопасность».

Благодаря этому количество абитуриентов по направлению «информационные технологии» увеличилось. Учебные заведения общего среднего образования, также способствуют решению этой проблемы. Во многих школах с 2015 года проводится акция «Час кода», в рамках которой школьникам рассказывают о профессии программиста и новых информационных технологиях, а также предлагают на практике попробовать себя в роли программистов. В 2019 году в Казахстане в пятый раз проводилась ежегодная глобальная акция «Час Кода», в рамках которой в общеобразовательных школах по всему Казахстану более 360 000 школьников приняли участие в часовых уроках информатики – это каждый десятый школьник страны и каждый четвертый школьник целевого в рамках акции возраста. Всего за все пять лет существования акции было пройдено более 900000 сессий «Часа Кода».

Развитое алгоритмическое мышление помогает не только в образовательной деятельности, но и в быту, например, при приготовлении обеда, уборке квартиры. Таким образом, изучение раздела «Компьютерное мышление» в школе становится жизненно необходимым для школьников.

1.3 Основные подходы к обучению программированию в школьном курсе информатики

Программирование с самых истоков введения информатики в школьный курс составляет одну из главных дидактических линий данной учебной дисциплины. Изучение основ программирования в школьном

курсе информатики составляет фундаментальную основу алгоритмической подготовки учащихся.

При обучении информатике учитель отталкивается от планируемых результатов, представленных в учебной программе. При изучении раздела «Компьютерное мышление» обучающийся должен научиться:

- определять результат исполнения алгоритма над входными данными;
- распознавать алгоритмы обработки числовых данных;
- создавать на основе алгоритмов обработки числовых данных программы анализа;
- анализировать простые программы на выбранном для изучения языке программирования или алгоритмическом языке;
- воспроизводить алгоритмы пошагового управления исполнителями;
- писать на выбранном языке программирования программы для решения задач базового уровня из различных научных областей, используя основные конструкции;
- определять затрачиваемое время на запуск программы, объём занимаемой оперативной памяти;
- применять компьютерно-математические модели;
- визуально представлять результаты математического моделирования;
- разрабатывать программы в определённой среде разработки программного обеспечения;
- использовать основные конструкции управления в последовательном программировании, использовать библиотеки, выполнять созданные программы;
- использовать алгоритмы сортировки и поиска данных, верно, применять их для решения поставленной задачи;

- понимать, что для решения одной задачи может существовать множество вариаций алгоритмов;
- создавать и применять программные математические модели;
- оценивать входные и выходные числовые значения в моделируемых объектах;
- объяснять полученное путём моделирования решение.

В современном программировании сформировались четыре основных способа разработки алгоритмов. Такие способы в специализированной литературе получили название парадигма программирования. Выделяют четыре парадигмы: процедурная, объектно-ориентированная, логическая и функциональная. Под эту классификацию подходят все известные на сегодняшний день языки программирования. У реализации каждой парадигмы есть свои положительные и отрицательные стороны.

В определенной степени парадигмы использовались в качестве основы для обучения программированию. Процедурная парадигма являлась основой обучения в большинстве курсов программирования. Опыт этой работы отражен в работах таких исследователей, как А.П. Ершов, А.Г. Гейн, В.М. Монахов и другие [10].

Парадигма объектно-ориентированного программирования частично реализована в работах Н.Д. Угринович, Н.Н. Истоминой и ряда других исследователей. Построение курса обучения на основе логической парадигмы программирования реализовано в ряде работ С.Г. Григорьева, А.Г. Щеголева, Д.П. Федюшина. В литературе практически не проработаны вопросы использования функциональной парадигмы в качестве основы для обучения программированию.

Традиционно в школе преподается процедурное программирование. Считается, что первый язык программирования (это, как правило, Питон, С, С++) должен иллюстрировать именно алгоритмические структуры, а

затем уже можно осваивать более сложные, объектно-ориентированные языки.

Рассмотрим традиционный дидактический подход к обучению программированию в целом, и в общеобразовательной школе в частности. Начиная с самых первых учебников по программированию, появились неизменные до настоящего времени каноны преподавания. Вначале рассматриваются переменные, типы данных, основные встроенные функции и синтаксические конструкции (операторы), затем структурные типы и средства работы с ними. Как можно заметить, основой является изучение конструкций линейных, ветвления (условный оператор, оператор выбора) и повторения (операторы циклов) и алгоритмов, строящихся на их основе. Такой подход оправдывает себя, если речь идёт об обучении специалиста, да и то лишь в случае хорошо подготовленного, мотивированного на изучение языка программирования. Оно не несет смысловой нагрузки для ученика, поскольку ему заранее даются ответы на ещё незадаанные вопросы, а предлагаемые задания не являются личносно значимыми.

В общеобразовательных школах существует «задачный» подход к обучению программированию. Это тот случай, когда постановке задачи предшествует изучение каждой языковой структуры и проблема решается с использованием этой структуры. Эти задачи обычно математические и редко связаны друг с другом. Большинство из них короткие, ограниченные и требуют от вас «найти» решение за один урок. Решение этих проблем привело к появлению небольших прикладных программ, которые могут использовать различные данные для поиска решений математических формул, вводить диалоговые формы для интерпретации полученных данных и выполнять ряд логических действий. Рассматриваемая задача часто не имеет отношения к жизни и не представляет интереса для ученика. Потому что школьники не понимают, как эта задача может

помочь им решить их жизненные проблемы. Классикой жанра можно считать решение квадратного уравнение, которая является типовой задачей, предлагаемой школьникам при изучении любого языка программирования. Для многих учеников остается не понятным вопрос: «Зачем нужно писать программу если квадратное уравнение легко решается вручную» Таким образом, структура урока сводится к этапам постановки задачи, лекция, нахождения решения, то есть применяется репродуктивный метод.

Однако, в последнее время намечаются серьёзные изменения в обучении программированию, связанные с тем, что на смену Basic, Pascal, C++ приходят Python и Scratch. А если в обучении программированию еще задействовать еще и телефоны учащихся, то можно вполне вернуть интерес учеников к программированию. Сейчас почти у каждого ребенка есть персональное мобильное устройство, которым они активно пользуются, нужно лишь начать активнее использовать его в учебном процессе. Такую возможность дает сервис-конструктор мобильных приложений MIT App Inventor, позволяющий разрабатывать приложения под Android.

MIT App Inventor – визуальной разработки Android-приложений, требующая от пользователя минимальных знаний программирования. MIT App Inventor использует графический пользовательский интерфейс (GUI), очень похожий на языки программирования Скретч. Использование данной среды в школе было бы логичным следствием учебной программы «Цифровая грамотность» для учащихся 1-4 классов, так как в начальной школе учащиеся работают с языком программирования Скретч. В настоящей школьной программе школьники знакомятся со средой MIT App Inventor в 11 классе, где учащиеся имеют достаточный уровень знаний по программированию и высокий уровень мотивации, а вот в основной школе интерес к программированию пропадает, потому что у школьников

недостаточная математическая подготовка и уровень мотивации нестабилен, слабо развито алгоритмическое мышление.

Поэтому логично изучать со школьниками то, что не только будет им интересно, но и может быть использовано ими в профессиональной деятельности, тем более, если это действительно востребовано. Отсюда следует, что необходимо ввести программу внеурочной деятельности, которая будет основана на современном направлении в программировании.

В ходе обучения, не прибегая к сложным математическим формулам, сложным алгоритмическим конструкциям обучающиеся на практике создают востребованные программные продукты. При решении практико-ориентированных задач, обучающийся на примере видит, как применяется алгоритмическая конструкция языка, что способствует развитию алгоритмического мышления.

Вывод по главе 1

Алгоритмическое мышление – это комплекс мыслительных операций приемов, направленных на решение как практических, так и теоретических задач, итогом которых являются алгоритмы как специфические продукты деятельности человека. Нами были проанализированы разные взгляды и формулировки понятия «алгоритмическое мышление». При изучении сущности и содержания понятия «алгоритмическое мышление», ввели компоненты алгоритмического мышления. В результате исследования можно сказать, что содержательная линия «Компьютерное мышление» является системообразующей темой курса информатики.

Огромным потенциалом для развития алгоритмического мышления школьников имеет MIT App Inventor – среда визуальной разработки только Android-приложений, требующая от пользователя минимальных знаний программирования. Изучив особенности работы в ней, мы выявили

возможности развития алгоритмического мышления. Чтобы достичь всех поставленных результатов будет разработана программа внеурочной деятельности по обучению программированию посредством изучения создания мобильных приложений в MIT App Inventor.

ГЛАВА 2. РАЗВИТИЕ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ ШКОЛЬНИКОВ ПОСРЕДСТВОМ ИЗУЧЕНИЯ СОЗДАНИЯ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В MIT APP INVENTOR

2.1 Возможности среды визуальной разработки Android-приложений MIT App Inventor как средство развития алгоритмического мышления школьников.

Мы живем в эпоху бурного развития технологий и лавины информации в разных областях знаний. Конечно, этот факт не мог не сказаться на системе образования в целом, и особенно на предмете «Информатика». Очевидно, что необходимо срочно искать образовательные инновации, модернизирующие содержание и приводящие качество подготовки выпускников в соответствие с запросами современного общества. Что касается курса информатики в школе, то это влечет за собой разработку новых понятий, связанных с новыми возможностями информации и коммуникации, особенно мобильных технологий. Алан Кей, американский ученый в области теории систем и один из пионеров в области объектно-ориентированного программирования и графических интерфейсов, внимательно следил за образованием детей «Мы должны дать нашим детям мощные способности, инструмент для размышлений «как можно скорее» [41]. Основная цель этого инструмента – создание связей между новыми знаниями и известными, развитие как синтетического, так и алгоритмического мышления. В связи с этим значительно возросла роль школьных курсов информатики. У учителей информатики сегодня есть уникальная возможность формировать познавательные интересы, развивать алгоритмическое мышление школьников в процессе обучения программирования с использованием различных программных сред.

Например, при изучении программирования на уроке информатики преподаватели активно используют визуальные среды программирования (Scratch, Kodu, Alice и т.д.) и делают упор на разработку «настольных» приложений. Но помимо компьютеров, учащиеся все чаще используют свои мобильные устройства. Учителя информатики сталкиваются с проблемой, как превратить любимую игрушку в средство обучения. Одним из способов популяризации программирования мобильных приложений является MIT App Inventor, облачная среда разработки приложений для платформы ОС Android. Операционная система Android является одной из ведущих операционных систем. Вот цифры рыночной доли Android, собранные аналитиками Downloadastro [18]. Android является самой крупной ОС в мире с более чем 70% долей рынка. Старые версии ОС Android более популярны среди пользователей Android. GooglePlayStore заработали почти \$34 млрд за последний квартал 2021 года.

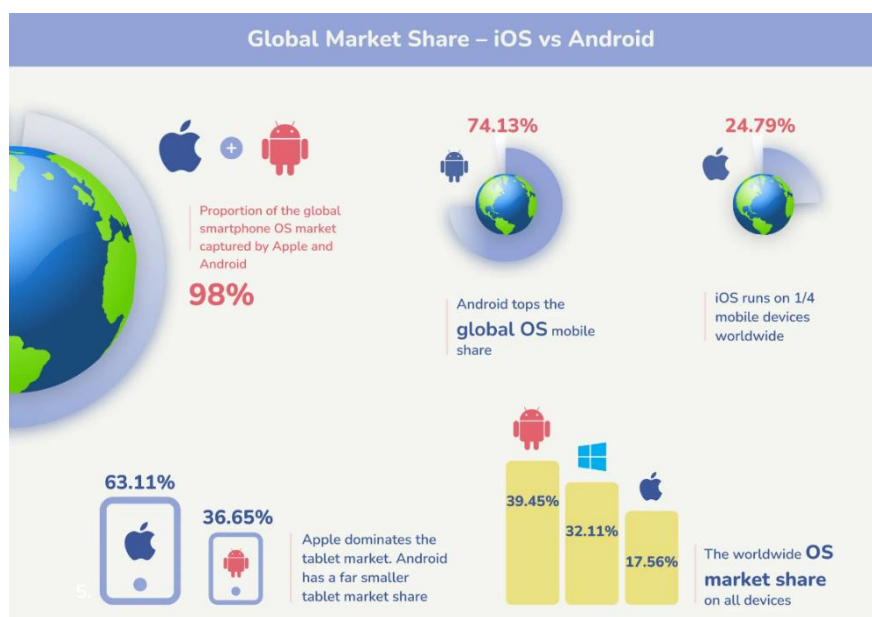


Рисунок 3 – Инфографика: рыночная доля Android и iOS – статистика на 2022 год

Разработка мобильных приложений для ОС Android очень важна несмотря на то, что для Android разрабатывается множество приложений самых разных форматов и жанров. На практике найти подходящее

приложение бывает непросто, и у пользователей часто возникает идея создать собственное. Создание приложения для Android требует определенных навыков программирования, что делает создание приложения практически невозможным для тех, кто незнаком с основами программирования. Поэтому этим вопросом занялась команда Массачусетского технологического института (MIT). В стенах института была создана среда, в которой каждый школьник мог создать приложение для Android. Эта разработка базируется на популярной алгоритмической среде Scratch, также разработанной в Массачусетском технологическом институте.

MIT App Inventor – визуальная среда разработки приложений для Android, разработанная в Google Labs и переданная в дар Массачусетскому технологическому институту. Разработка приложений с помощью MIT App Inventor не требует знания языка программирования Java и Android SDK. Достаточно базовых знаний алгоритмических основ. Достоинством этой среды является то, что с ее помощью всего лишь за несколько минут можно создать свое элементарное приложение, а за час – построить довольно сложную программу с несколькими экранами. Язык программирования среды MIT App Inventor очень прост, поскольку он с самого начала создавался для того, чтобы его использовали школьники. При программировании на нем не нужно писать строчки кода, как это происходит во «взрослых» языках программирования. Чтобы создавать приложения в MIT App Inventor, достаточно просто перетаскивать блоки, собирая их в программу, как пазлы. В среде можно работать из любого современного браузера. Это создает дополнительные преимущества, поскольку не требует установки никаких дополнительных программ. Заметим, что специфика работы в среде состоит в необходимости авторизации через Google-аккаунт. Еще одним достоинством является то, что MIT App Inventor – это облачная среда, в ней можно работать из

любого браузера. Таким образом, после авторизации приложение MIT App Inventor и все сохраненные проекты будут доступны на любом компьютере.

В среде два режима работы:

- в режиме «Дизайнер» осуществляется разработка интерфейса и внешнего вида приложения, его отображение на устройстве;
- в режиме «Блоки» происходит непосредственно разработка самого приложения, а именно написание программы с помощью блоков.

Уникальной особенностью MIT App Inventor является возможность тестировать приложения, разработанные для мобильных устройств, в режиме реального времени без необходимости их предварительной компиляции и установки на мобильное устройство. Для этого просто установите на свое мобильное устройство специальное приложение MIT AI2 Companion. Кроме того, вы можете протестировать свое приложение на эмуляторе Android для ПК. Для создания приложения необходимо представить на экране необходимые элементы интерфейса. После завершения разработки вы можете получить готовый APK-файл (в виде заархивированного исполняемого файла Android-приложения) для установки на своем устройстве или QR-код со ссылкой для скачивания.

Что важно при организации обучения с использованием среды MIT App Inventor? Понятно, что мотивация и познавательный интерес имеют не только доступность среды, но и практическую ценность. Она также определяется свободой выбора деталей своего приложения с точки зрения выполняемых им функций.

Каждое занятие – практическая работа по планированию, созданию, программированию и апробированию результата.

Возможности среды для развития уровня алгоритмического мышления:

- планирование и постановка задач;

- описание логической последовательности действий;
- целеполагание и достижение целей;
- реализация математических и логических операций;
- программирование в виде блоков;
- конструирование приложения;
- конструирование приложения по алгоритму;
- программирование с опорой на алгоритмические структуры, блок-схемы.

Каждый урок предполагает формирование и развитие алгоритмического мышления.

Рассмотрим план занятия по созданию мобильных приложений:

1. Знакомство с приложением.
2. Обсуждение (тема урока, постановка цели, построение плана действий (разработка алгоритма или анализ готового алгоритма)).
3. Конструирование приложения.
4. Программирование.
5. Тестирование приложения.
6. Оценка (работы, рефлексия).

То есть мы видим, что на всех этапах урока возможно развивать алгоритмическое мышление.

Формат работы со средой MIT App Inventor доступный и достаточно мобильный для разных возрастов. Он идеален для использования в обучении программированию школьников в школе.

Мы считаем, возможным организовать в рамках внеурочной деятельности программирование мобильных приложений с использованием визуальной среды для разработки MIT App Inventor, переход на которую после Scratch происходит достаточно легко. Использование указанной среды программирования позволяет

организовать применение современных информационных технологий в образовательном процессе, что соответствует стратегии информатизации образования.

Актуальность изучения данной среды программирования обусловлена следующими обстоятельствами:

- создание приложений для мобильных устройств является одним из популярных и востребованных направлений программирования в современном мире;

- среды визуального программирования позволяют научить создавать мобильные приложения учеников с разными навыками в области алгоритмизации и программирования;

- вместо текстового языка программирования в таких средах используются визуальные логические блоки с уже готовыми действиями, функциями для работы с социальными сетями, веб-сайтами или сенсорами устройства;

- визуальная среда разработки позволяет обрабатывать мультимедийный контент, распознавать речевые команды и синтезировать речь.

2.2 Разработка программы внеурочной деятельности «Основы программирования мобильных приложений под ОС Android»

Программа разработана для организации внеурочной деятельности учащихся. Программа расширяет и дополняет раздел алгоритмизации и программирования курса информатики в основной школе. Программа нацелена на развитие интереса учеников к практической работе с мобильными устройствами и программами, формирование представлений об основных правилах и методах программирования мобильных устройств, развитие у учащихся алгоритмического мышления, конструкторских способностей в процессе моделирования и экспериментов.

Программа «Основы программирования мобильных приложений под ОС Android» построена на принципах погружения в среду визуального программирования. Ученики изучают основы программирования на практических примерах, разбирая реальные существующие задачи, выполняя на каждом занятии учебные проекты.

В рамках занятий ученики познакомятся со средой визуального программирования MIT App Inventor, научатся создавать мобильные приложения для Android, протестируют эти приложения на своих мобильных устройствах и по окончании обучения смогут разработать собственный итоговый учебный продукт – мобильное приложение.

В основу программы положены следующие принципы обучения:

- от простого к сложному,
- через практику к теории,
- самостоятельного обучения,
- коллективного взаимообучения.

Актуальность: сейчас мир становится все более «мобильным». При помощи гаджетов люди узнают о погоде, пробках на дорогах, читают книги, смотрят фильмы, заказывают и покупают товары в интернет-магазинах. Ну и конечно же – общаются друг с другом. Это удобно! Можно выходить в интернет, находясь в любой точке мира, всегда быть на связи с родными и близкими, оплачивать покупки с доставкой на дом, делиться любимой музыкой и фотографиями в социальных сетях. Операционная система Android является одной из лидирующих. Разработка мобильных приложений под ОС Android очень актуальна даже несмотря на то, что под Android разработано множество приложений самых разных форматов и жанров. Дело в том, что подчас трудно найти нужное приложение, а часто бывает и так, что у пользователя рождается идея создания собственного уникального приложения. Для создания приложений под Android необходима определенная квалификация в

программировании, что делает практически невозможным создание приложений людьми, не познавшими основы программирования. Создание приложений для мобильных устройств является одним из популярных и востребованных направлений программирования в современном мире. Среды визуального программирования позволяют научить создавать мобильные приложения учеников с разными навыками в области алгоритмизации и программирования. Вместо текстового языка программирования в таких средах используются визуальные логические блоки с уже готовыми действиями, функциями для работы с социальными сетями, веб-сайтами или сенсорами устройства и др. Визуальная среда разработки позволяет обрабатывать мультимедийный контент, распознавать речевые команды и синтезировать речь.

Направленность программы: имеет инженерно-техническую направленность, в связи с этим рассматриваются следующие аспекты изучения:

1. Содержание программы рассматривается как средство формирования образовательного потенциала, позволяющего развивать наиболее передовые на сегодняшний день технологии — информационные, интегрирующие в себе науку, технологию, инженерное дело.

2. Обучение по данной программе создает благоприятные условия для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

3. Содержание программы рассматривается как средство развития алгоритмического мышления, умения анализировать, выявлять сущности и отношения, описывать планы действий и делать логические выводы.

Новизна, целесообразность программы.

Занятия по программе позволяют сформировать как технические навыки разработки программного обеспечения, так и развить алгоритмическое мышление, интеллектуально-творческие способности обучающихся в процессе работы над проектами. Программа предполагает работу над индивидуальными и групповыми проектами на занятиях. Проектная деятельность позволяет развивать у обучающихся кругозор, критическое и алгоритмическое мышление и практические навыки, умения в использовании алгоритмических конструкций.

Педагогическая целесообразность: Педагогическая целесообразность данной программы состоит в отражении содержательных линий «Компьютерное мышление»:

- формирование навыков информационно-поисковой деятельности;
- формирование алгоритмической культуры;
- формирование коммуникативных компетенций в области информационной деятельности;
- развитие системного, алгоритмического, операционного и критического мышления;
- подготовка к жизни в информационном обществе.

Практическая значимость программы отвечает возрастным особенностям и запросам подростков, характеризующимся повышенной потребностью в самоопределении и самовыражении, стремлении доказать себе и окружающим умение самостоятельно принимать и реализовывать собственные решения. Постигая азы алгоритмизации и программирования, дети знакомятся с ведущими профессиями (программиста, тестировщика) и имеют возможность проживать эти роли, реализуясь и самовыражаясь на каждом учебном занятии.

Отличительные особенности программы. В процессе реализации данной программы, обучающиеся создают проекты в облачной среде

разработки MIT App Inventor. Среда разработки MIT App Inventor позволяет создавать приложения различной сложности, в том числе с использованием библиотек.

Цель: Развитие навыков и умений по созданию простых мобильных приложений для ОС Android на основе визуального конструктора в среде App Inventor, а также развитие у учащихся алгоритмического мышления, творческих, аналитических способностей. Сформировать у учащихся основные представления о языках программирования, алгоритмах, исполнителях и способах написания алгоритмов; организовывать научно-педагогическую деятельность по проектам творческого характера, формировать у учащихся познавательный интерес к обучению и исследовательские умения.

Для достижения поставленной цели планируется выполнение следующих задач:

- сформировать общее представление о создании мобильных приложений на базе операционной системы Android;
- сформировать представления о структуре и функционировании среды App Inventor;
- сформировать умения и навыки построения различных видов алгоритмов в среде разработки MIT App Inventor;
- овладеть понятиями «объект», «событие», «управление», «обработка событий»;
- сформировать умение использовать инструменты и компоненты среды разработки MIT App Inventor для создания мобильных приложений;
- сформировать навыки создания приложений от идеи до реализации;
- сформировать ключевые компетенции проектной и исследовательской деятельности;
- развивать алгоритмическое и логическое мышление;

- развивать умение постановки задачи, выделения основных объектов, математическое моделирование задачи;
- развивать умение поиска необходимой учебной информации;
- развивать навыки планирования проекта, умение работать в группе;
- сформировать мотивацию к изучению программирования;
- воспитывать умение работать индивидуально и в группе для решения поставленной задачи;
- воспитывать трудолюбие, упорство, желание добиваться поставленной цели;
- воспитывать информационную культуру.

Программа рассчитана на учащихся в возрасте от 11 до 16 лет, возраст – становления интересов, активного обучения, саморазвития и самоопределения. В основу Программы положен принцип интеграции теоретического обучения с процессом практической исследовательской деятельности обучающихся с учётом возрастных и индивидуальных особенностей детей.

Уровень сложности освоения программы базовый. Занятия проводятся в группах, продолжительность занятия 45 минут, общая продолжительность программы – 36 часов. Программа предназначена для занятий с обучающимися, имеющими склонность к алгоритмическому мышлению, увлекаются IT-технологиями.

Форма проведения занятий. Организационные формы: индивидуальная, пары постоянного состава, групповая (большая или малая постоянного состава). Каждое занятие делится на теоретическую и практическую части. На теоретической части занятия создаются математические модели приложений и разбираются алгоритмы их создания. В ходе практической работы учащиеся создают приложения, тестируют их на мобильных устройствах, выполняют отладку приложения.

Изложение теории построено так, что сначала у обучающихся формируется общее понятие на основе имеющихся знаний, затем оно формализуется, и, наконец, демонстрируется его применение при решении конкретной задачи. Закрепление теоретического материала достигается путем создания каждым обучающимся собственного приложения на основе примера или с расширенными функциями. Каждый учащийся решает свою задачу, с учетом уровня способностей и полученных знаний. Итогом обучением является выполнение индивидуального проекта-приложения для мобильного устройства в любой области знаний.

Ниже представлено содержание программы.

Содержание программы внеурочной деятельности

Тема 1. Основы создания программ для мобильных устройств.

Введение в среду программирования приложений для мобильных устройств MIT App Inventor. Основные структурные блоки программирования. Принципы разработки мобильных приложений. Интерфейс программной среды MIT App Inventor. Режимы «Дизайн» и «Блоки». Основные компоненты среды программирования. Свойства компонентов. Сохранение и установка приложений на мобильные устройства.

Тема 2. Создание рисунков и анимационных игр.

Переменные. Алгоритмические конструкции. Способы создания приложений с использованием компонента «Холст».

Настройка параметров холста. Рисование круга и линий. Вывод текста на холст. Компоненты «Шар» и «Изображения Спрайта»: свойства, события и действия при их использовании. Анимация движения объектов по экрану: с помощью сенсоров, с помощью компонента «Часы», посредством управления объектом с помощью кнопок. Действия при наложении объектов. Локальные и глобальные переменные внутри приложения.

Тема 3. Использование сенсоров в приложении.

Типы сенсоров, используемые в приложении. Определение сенсоров на мобильных устройствах.

Тема 4. Распознавание речи и чтение текста

Распознавание речи в мобильных приложениях. Особенности построения приложений и использования программ.

Тема 5. Массивы и списки в приложении

Использованием массивов при создании приложений с большими наборами однотипных компонент. Создание пустых списков. Создание списков изображения. Использование операции конкатенации для формирования имени компонентов массивов.

Тема 6. Использование мультимедиа в приложении.

Использование компонентов палитры Медиа. Разработка приложений, содержащих мультимедиа-объекты (изображения и аудио ресурсы).

Тема 7. Передача данных

Экран приложения и его свойства. Принципы создания приложений с несколькими экранами. Передача данных между экранами. Использование компонента TinyDB и начального значения экрана.

Тема 8. Итоговый проект. Разработка и отладка мобильного приложения

Разработка прототипа приложения. Создание таблицы компонент приложения, описание действий и событий приложения. Оформление и описание приложения. Презентация и взаимооценка созданных приложений.

Учебно-тематическое планирование программы представлено в таблице, где указаны количество теоретических и практических занятий.

Таблица 12 – Учебно-тематический план

№	Тема	Всего часов	Теория	Практика
1	2	3	4	5
1	Тема 1. Основы создания программ для мобильных устройств.	3	1	2
2	Введение в среду программирования приложений для мобильных устройств MIT App Inventor. Основные структурные блоки программирования. Установка приложений на мобильные устройства	2	1	1
3	Выполнение проекта «Привет, Коди»	1		1
4	Тема 2. Создание рисунков и анимационных игр.	7	2	5
5	Способы создания приложений с использованием компонента «Холст» Холст и координатная сетка Настройка параметров холста. Рисование круга и линий. Вывод текста на холст.	1	1	
6	Выполнение проекта «PaintPot»	1		1
7	Компоненты «Шар» и «Изображения Спрайта», «Часы»: свойства, события и действия при их использовании. Анимация движения объектов по экрану. Выполнение проекта «Анимация шара»	2	1	1
8	Выполнение проекта «MoleMash»	1		1
9	Выполнение проекта «LadybugChase»	1		1
10	Выполнение проекта «Соберите монеты»	1		1
11	Тема 3. Использование сенсоров в приложении.	4	1	3
12	Типы сенсоров, используемые в приложении. Определение сенсоров на мобильных устройствах. Выполнение проекта «Где я?»	2	1	1
13	Выполнение проекта «Компас»	1		1
14	Выполнение проекта «Шагомер»	1		1
15	Тема 4. Распознавание речи и чтение текста	4	1	3
16	Распознавание речи в мобильных приложениях Особенности построения приложений и использования программ.	2	1	1
17	Выполнение проекта «Строчка»	1		1
18	Выполнение проекта «Голосовой калькулятор»	1		1
19	Тема 5. Массивы и списки в приложении	4	2	2
20	Использованием массивов при создании приложений с большими наборами однотипных компонент. Создание пустых списков. Создание списков изображения.	4	2	2

Продолжение таблицы 12

1	2	3	4	5
21	Тема 6. Использование мультимедиа в приложении.	4	1	3
22	Использование компонентов палитры Медиа. Разработка приложений, содержащих с мультимедиа-объекты (изображения и аудио ресурсы). Выполнение проекта «Видеоплеер»	2	1	1
23	Выполнение проекта «Мр3 плеер»	1		1
24	Выполнение проекта «Фотокамера»	1		1
25	Тема 7. Передача данных	4	2	2
26	Экран приложения и его свойства. Принципы создания приложений с несколькими экранами. Выполнения проекта «Викторина»	2	1	1
27	Передача данных между экранами. Использование компонента TinyDB и начального значения экрана.	2	1	1
28	Тема 8. Разработка и отладка мобильного приложения.	6	1	5
29	Разработка прототипа приложения. Создание таблицы компонент приложения, описание действий и событий приложения. Оформление и описание приложения. Презентация и взаимооценка созданных приложений.	4	1	5
30	Итого	36	11	25

Планируемые результаты освоения программы обучающимися.

Предметные:

- сформировать представления о создании мобильных приложений на базе платформы Android;
- сформировать понятия «моделирование», «программирование», «визуальная среда программирования», «компоненты приложения», «дизайн приложения», «блоки программирования»;
- сформировать представления о структуре и функциях среды MIT App Inventor: основные принципы создания и отладки мобильных приложения в среде визуального программирования MIT App Inventor;
- сформировать и способности для создания различных алгоритмов с использованием блоков в среде MIT App Inventor;

- сформировать умение создавать мобильные приложения с использованием компонентов, блоков и их комбинаций в среде MIT App Inventor.

Личностные:

- развивать коммуникативные навыки,
- развивать целеустремлённость и настойчивость в творческом, исследовательском и образовательном процессе.

Метапредметные (развитие алгоритмического мышления)

- обосновать выбранные подходы инструменты, используемые для достижения образовательных результатов;
- определять необходимые действия согласно учебно-познавательным задачами создавать алгоритмы их выполнения.
- выбирать из предложенных вариантов и самостоятельно искать средства/ресурсы для решения проблемы/достижения цели.
- разрабатывать план решения проблемы (проектная реализация, исследование).

2.3 Разработка программы «Основы программирования мобильных приложений под ОС Android» на образовательной платформе stepik.org

Одним из приоритетных направлений национального проекта «Цифровой Казахстан» [33] является развитие дистанционного образования, которое позволит самостоятельно строить индивидуальную учебную траекторию обучения в инновационной цифровой среде. Дистанционные курсы предназначены для того, чтобы ученики имели доступ к образованию и могли учиться каждый в своем ритме. Онлайн – курсы позволяют приобрести наиболее востребованные на рынке труда навыки и умения, особенно в перспективных высоко технологичных и быстрорастущих отраслях. Онлайн обучение – признанный тренд нашего

времени. Онлайн платформы отличаются от систем дистанционного обучения своей философией и инструментами.

При разработке программы нами был проведен сравнительный анализ образовательных платформ по ряду обобщенных критериев.

Таблица 13 – Сравнительный анализ образовательных платформ по ряду обобщенных критериев

Характеристика	Образовательные платформы			
	Открытое образование	Coursera	Stepik	Лекториум
Страна	Россия	США	Россия	Россия
Профиль	По теме образовательных программ высшего образования	Бизнес, компьютерные науки, анализ данных, саморазвитие	Информационные технологии и другое	Различные тематики
Аудитория	Студенты профессионалы	Молодые профессионалы	Профессионалы в сфере ИТ, школьники студенты	школьники, учителя, студенты
Создатели курсов	Университеты партнеры ассоциации «Национальная платформа открытого образования»	Вузы и организации партнеры, прошедшие одобрение советом ведущих вузов	Школы, вузы, организации, зарегистрированный пользователь	Школы, университет организации партнеры
Задания	Тесты, задания на взаимное оценивание, задания на соотнесение и программирование	Тесты, задания на взаимное оценивание, вопросы, встраиваемые в видео	Тесты, задания на соотнесение, Программирование задания на взаимную и самооценку, установите порядок	Тесты, задания на взаимное оценивание, задания на соотнесение и программирование
Обеспечение обучения	Отсутствует адаптивный контент	Отсутствует адаптивный контент	Возможность добавления адаптивного контента	Отсутствует адаптивный контент
Сертификация	Платная, с подтверждением личности	Платная, с подтверждением личности	Бесплатная, без подтверждения личности	Платная, с подтверждением личности

На основе сравнительного анализа, приведенного в таблице 13 в рамках нашего исследования, была выбрана платформа Stepik для разработки программы «Основы программирования мобильных

приложений под ОС Android». Образовательная платформа подходит по всем характеристикам: аудитория, сертификация, профиль, обеспечение обучения.

Программа «Основы программирования мобильных приложений под ОС Android» состоит из следующих разделов:

Тема 1. Основы создания программ для мобильных устройств.

Тема 2. Создание рисунков и анимационных игр.

Тема 3. Использование сенсоров в приложении.

Тема 4. Распознавание речи и чтение текста.

Тема 5. Массивы и списки в приложении.

Тема 6. Использование мультимедиа в приложении.

Тема 7. Передача данных.

Тема 8. Итоговый проект.

Курс состоит из нескольких модулей, каждый из которых включает в себя лекционный блок, мультимедийный блок (ссылки на видео, аудио), лекционный материал для чтения и записи, блок тестирования/опроса/решения задач. Последний блок предназначен для контроля знаний учащихся по пройденной лекции.

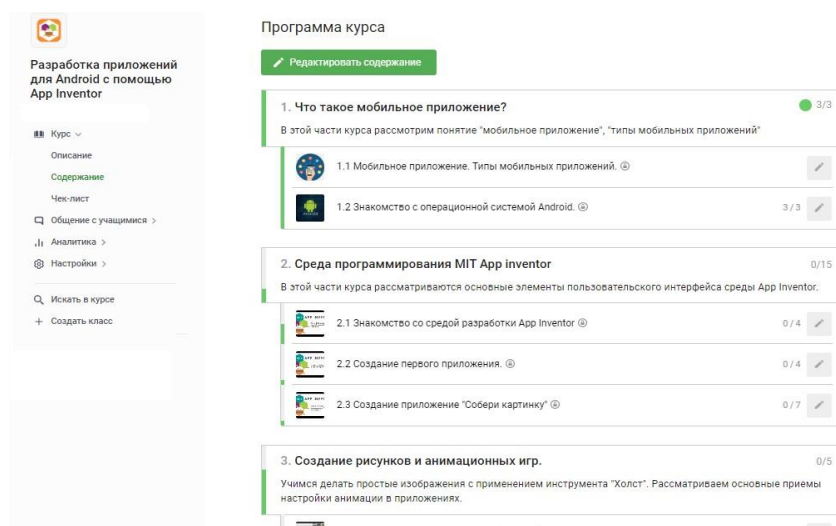


Рисунок 4 – Программа «Основы программирования мобильных приложений под ОС Android»

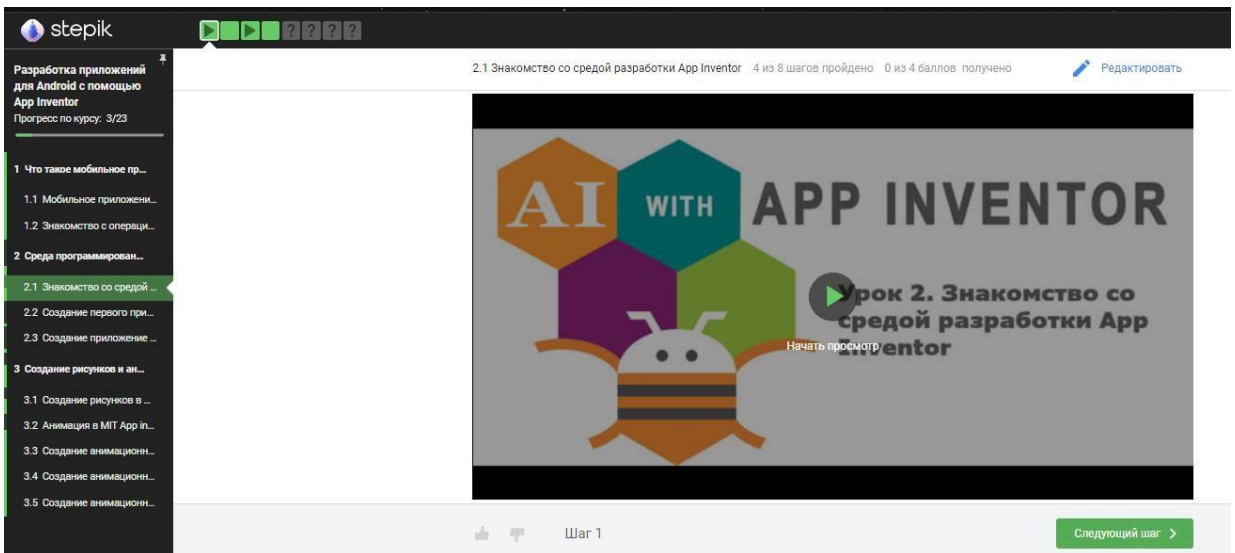


Рисунок 5 – Модуль курса

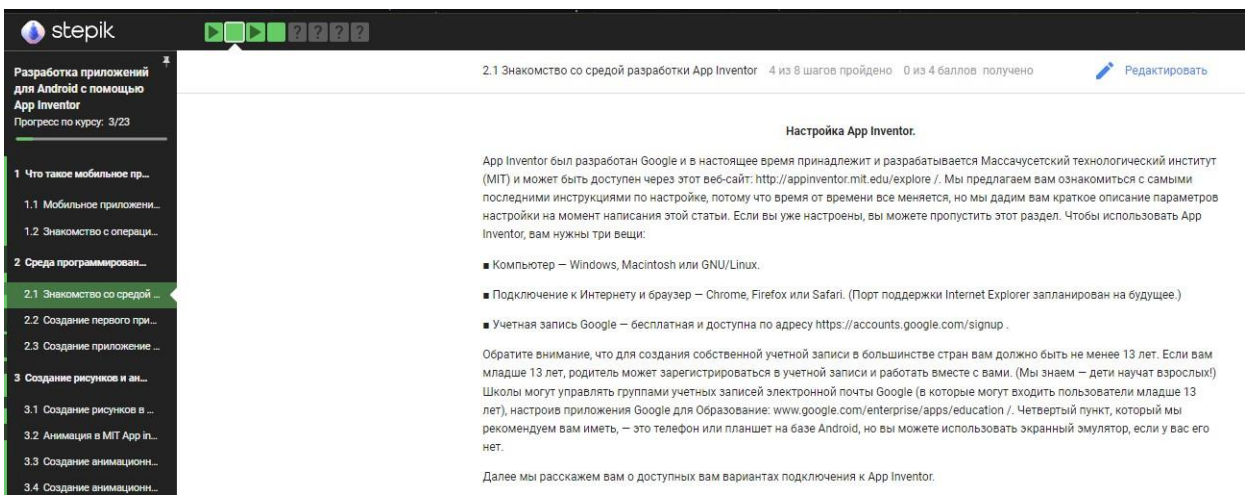


Рисунок 6 – Модуль курса

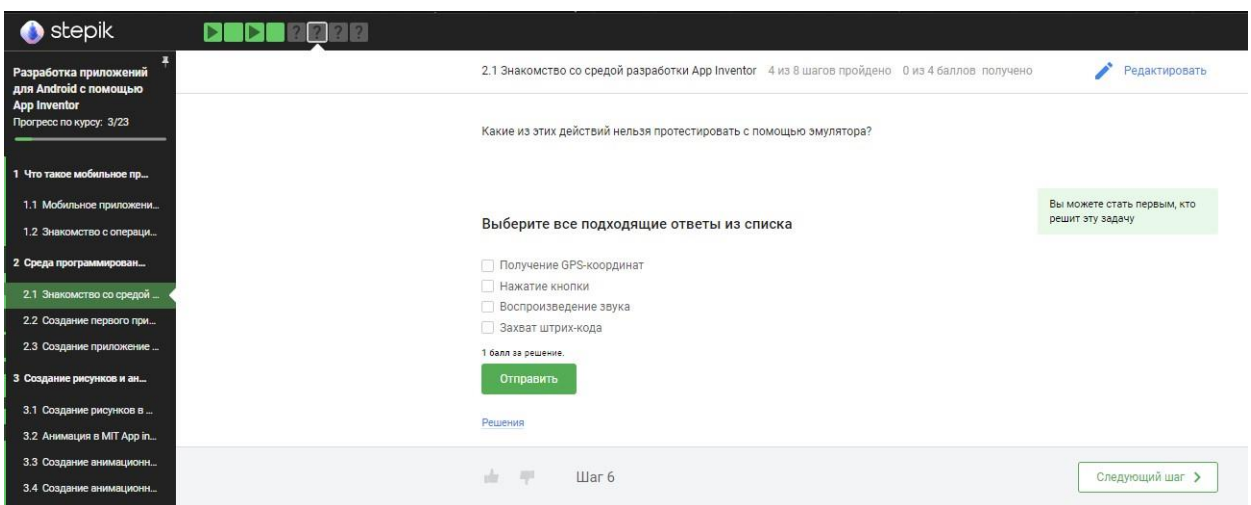


Рисунок 7 – Модуль курса

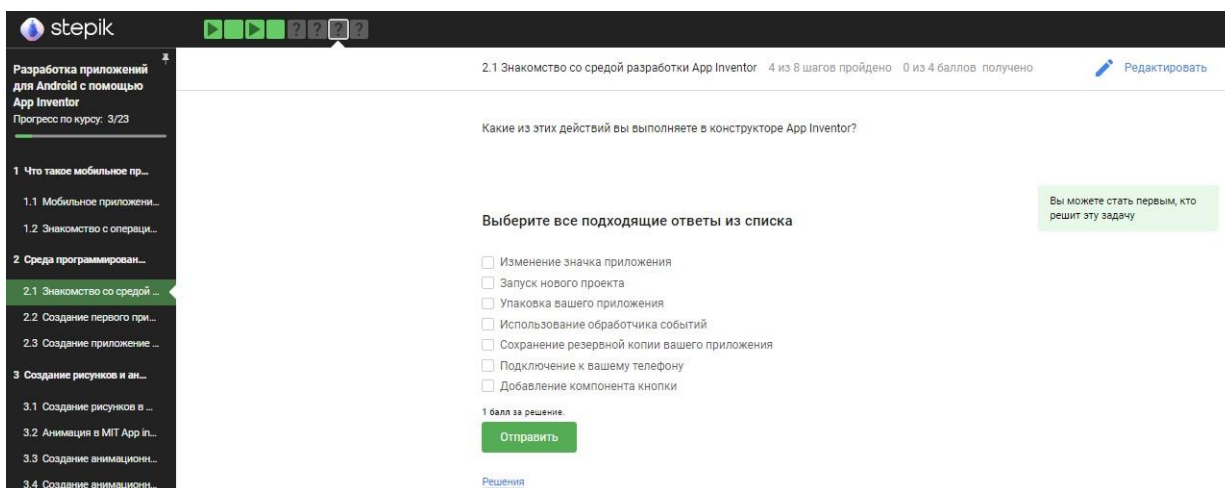


Рисунок 8 – Модуль курса

Модули имеют определенную логическую завершенность по отношению к установленным целям и результатам обучения. Каждый модуль состоит из 3 основных блоков: теоретического, практического, обратной связи.

Теоретический блок присутствует в каждом модуле и содержит систематизированный и обобщенный учебный материал в соответствии с изучаемой программой обучения. Теоретический блок представлен в виде отдельных уроков по рассматриваемым темам. В свою очередь уроки разбиты на шаги – небольшие по объему порции материала (видеолекции, текстовые страницы, страницы с заданиями).

После каждой темы в теоретическом блоке присутствуют тестовые задания разных типов, предназначенные для самопроверки полученных знаний по пройденному материалу.

Блок практико-ориентированных заданий организует деятельность слушателя курса и обеспечивает формирование необходимых умений и навыков.

Таким образом, предложенный онлайн-курс позволит обучающимся самостоятельно выстраивать индивидуальную образовательную траекторию.

Для того, чтобы подключиться к обучающему курсу нужно быть зарегистрированным в системе Stepik.org, иметь ссылку на курс или просто найти подходящий для себя курс и подключиться.

Таким образом, анализ возможностей платформы Stepik, а также опыт создания программы показали удобство работы с данной системой. При разработке программы «Основы программирования мобильных приложений под ОС Android» (<https://stepik.org/course/131338>) не возникало никаких проблем, сервис адаптирован под создание, теоретического и практического материала.

Выводы по главе 2

В рамках диссертационного исследования разработана программа «Основы программирования мобильных приложений под ОС Android». Программа расширяет и дополняет раздел алгоритмизации и программирования курса информатики в основной школе. Программа курса нацелена на развитие интереса учеников к практической работе с мобильными устройствами и программами, формирование представлений об основных правилах и методах программирования мобильных устройств, развитие у учащихся логического мышления, конструкторских способностей в процессе моделирования и экспериментов. Программа включающий пояснительную записку, тематическое планирование, планируемые результаты изучения курса, разработан электронный образовательный ресурс в поддержку программы. Разработанная нами программа направлена на раскрытие потенциала учащегося, вовлечение его в изучение программирования и развитие алгоритмического мышления. Программа направлена на развитие интереса обучающихся к программированию и робототехнике, способствует формированию представлений о практической значимости программирования. В ходе обучения, не прибегая к сложным математическим формулам, сложным

алгоритмических конструкциям обучающиеся на практике создают востребованные программные продукты. При решении практико-ориентированных задач, обучающийся на примере видит, как применяется алгоритмическая конструкция языка, что способствует развитию алгоритмического мышления и повышению мотивации школьников к изучению программирования. Программа ориентирован на изучение его во внеурочной деятельности. Программа рассчитана на 36 часов по 1 часу в неделю.

ГЛАВА 3. ПЕДАГОГИЧЕСКИЙ ЭКСПЕРИМЕНТ И ОЦЕНКА ЭФФЕКТИВНОСТИ ВНЕДРЕНИЯ МЕТОДИКИ

3.1 Организация и проведение педагогического эксперимента

В наше время педагогические эксперименты возможно проводить при помощи различных методов, инструментов, средств измерения и оценки результатов. Для подтверждения педагогического эксперимента, который прошел успешно необходимо использование методов, обеспечивающих проведение анализа исследуемой проблемы. Результаты анализа можно считать достоверным фактическим материалом. В основе планирования педагогического эксперимента лежит подготовительная работа, в ходе которой были определены методы исследования, методы анализа, объекты измерения, а также выбор измерителей. Основной целью проведения эксперимента являлось подтверждение выдвинутой в работе гипотезы: развитие алгоритмического мышления обучающихся будет более эффективно, если разработать и внедрить для учащихся седьмых классов программу внеурочной деятельности «Основы программирования мобильных приложений под ОС Android», направленную на развитие алгоритмического мышления.

Цель экспериментально-исследовательской работы: определить уровень развития алгоритмического мышления учащихся до изучения программы внеурочной деятельности по созданию мобильных приложений, разработать, апробировать программу внеурочной деятельности «Основы программирования мобильных приложений под ОС Android», определить уровень алгоритмического мышления после прохождения программы внеурочной деятельности.

Достижение поставленной цели осуществлялось посредством реализации следующих задач:

1. Опираясь на результаты анкетирования, собрать данные об интересе школьников к мобильным приложениям и их разработке.

2. Опираясь на результаты выполнения проверочной работы, собрать данные об уровне развития алгоритмического мышления, используя критерии сформированности алгоритмического мышления, определенные Н.Н. Тулькибаевой.

3. Апробировать программу и сопоставить статистические показатели констатирующего и контролирующего тестирования путем математического анализа и использовать Т-критерий Вилкоксона, тем самым подтвердить или опровергнуть выдвинутую ранее гипотезу.

Основной базой для проведения педагогического эксперимента была выбрана КГУ «Общеобразовательная школа № 23 им. М. Козыбаева г. Костаная».

Этапы исследования. Исследование проводилось в три этапа в течение 2021-2022 гг.

Первый этап – подготовительный (январь 2021 – май 2021 г.)
Формулирование цели и гипотезы диссертации. Была проанализирована литература по проблематике диссертации, изучены статьи по вопросам создания мобильных приложений и преподавания в школе курсов программирования. Изучен опыт преподавания внеурочных и дополнительных занятий коллегами из учреждений дополнительного образования и школ.

Второй этап – разработка программы и внедрение в учебный процесс внеурочных занятий (сентябрь 2021 г. – май 2022 г.) Разработана программа внеурочной деятельности, собран теоретический материал и практические задания, с использованием образовательной платформы Stepik для учащихся 7 классов.

Третий этап – экспериментально-аналитический (июнь-август 2022 г.) Проведена проверка гипотезы с помощью математической

статистики. Осуществлён анализ расчётов, сформировано и оформлено исследование. Составлено заключение, соответствующее результату.

3.2 Анализ результатов исследования уровня развития алгоритмического мышления школьников посредством создания мобильных приложений во внеурочной деятельности

Эксперимент проводился на двух группах обучающихся 7 классов. Для внедрения программы внеурочной деятельности в образовательный процесс нам было необходимо набрать группу учащихся седьмых классов, которые захотят посещать занятия по внеурочной программе, поэтому среди учащихся было проведено анкетирование «Использование мобильных приложений».

Анкета «Использование мобильных приложений»

1. Какая операционная система у Вашего смартфона?
а) Android б) IOS в) Windows г) Затрудняюсь ответить.
2. Сколько часов в день Вы пользуетесь своим мобильным телефоном?
а) Более 8 часов б) 5-8 часов в) 3-4 часа г) 1-2 часа д) Менее 1 часа.
3. С какой целью Вы чаще всего используете свои мобильные устройства?
а) Мобильные приложения (соц. сети, игры) б) Мобильная связь
в) Фотосъёмка г) Другое (запишите).
4. Довольны ли вы стандартными мобильными приложениями на Вашем смартфоне?
а) Да б) Нет в) Не всеми.
5. Сколько приложений на Вашем мобильном устройстве?
а) Менее 10 б) 10-15 в) 15-20 г) 20-25 д) Более 25.
6. Хотели бы Вы создавать собственные мобильные приложения?
а) Да б) Нет в) Затрудняюсь ответить.

7. Знакомы ли Вы уже с мобильной разработкой?

а) Да б) Нет.

8. Знакомы ли Вы со средой визуального программирования мобильных приложений для Android MIT App Inventor?

а) Да б) Нет.

9. Нравятся ли вам изучать язык программирования Питон на уроках информатики?

а) Да б) Нет.

10. Испытываете ли трудности в решении задач на языке программирования Питон?

а) Да б) Нет в) Часто г) Иногда.

Рассмотрим результаты анкетирования учащихся 7 «А» и 7 «Б» классах. Данные проведённого анализа показали, что: при ответе на первый вопрос: 36 учеников имеют телефоны на операционной системе Android, остальные 12 человек – обладатели телефонов с Apple IOS и 2 ученика – не знали ОС своего телефона. Отсюда делаем вывод, что большинство учащихся пользуются ОС Android.

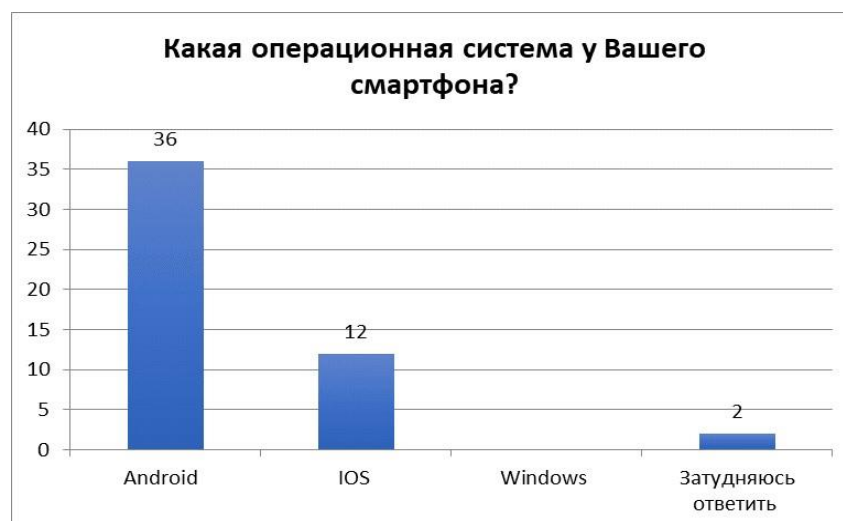


Рисунок 9 – Диаграмма ответов на 1 вопрос

В рамках второго вопроса мы хотели узнать сколько часов в день школьники пользуются своими мобильными устройствами.



Рисунок 10 – Диаграмма ответов на 2 вопрос

Более 8 часов в день телефоном пользуется – 28 человек, 12 школьников – 5 часов, 6 человек – 3 – 4 часа, 1 – 2 часа – 3 человека и менее 1 часа – 1 школьник.



Рисунок 11 – Диаграмма ответов на 3 вопрос

Ученики используют телефоны, чаще всего, с целью использования мобильных приложений (социальные сети, игры) – 44 человек – все опрошенные школьники используют мобильную связь и 3 человека для фотосъёмки.

На следующем вопросе нам стало интересно все ли школьники довольны стандартными мобильными приложениями, установленными на их смартфонах.

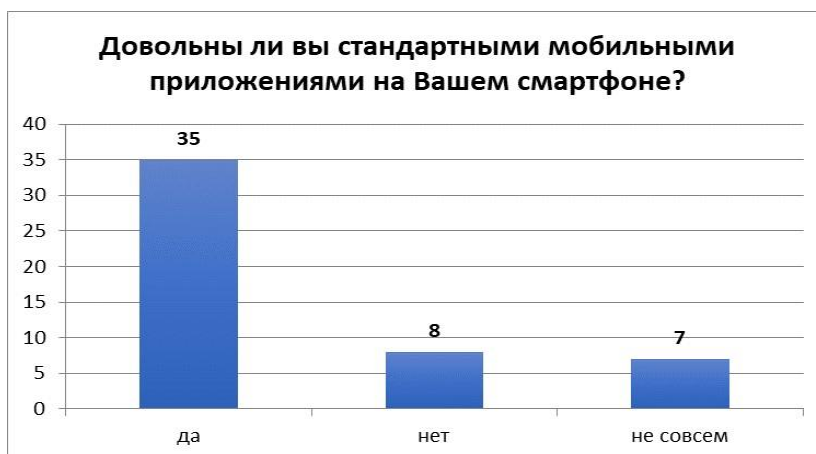


Рисунок 12 – Диаграмма ответов на 4 вопрос

Более половины довольны стандартным набором приложений на их телефоне, но не всеми 7 человек, а 8 человек – не довольны.

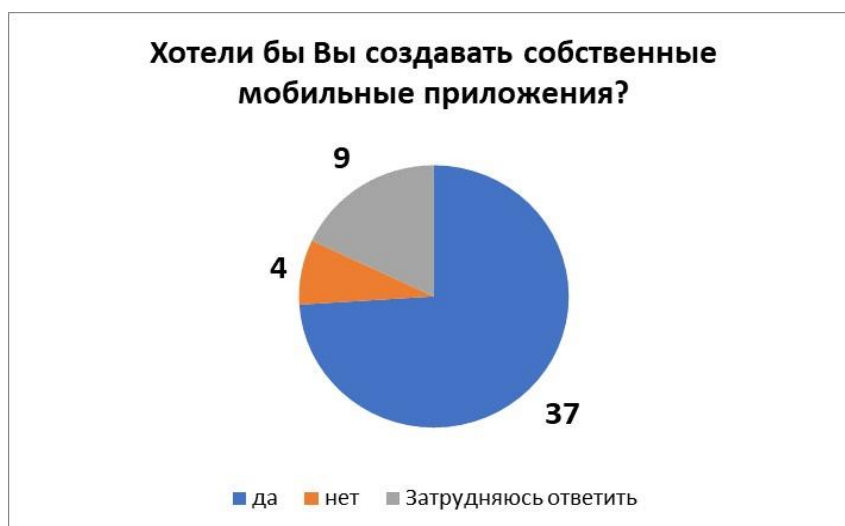


Рисунок 13 – Диаграмма ответов на 6 вопрос

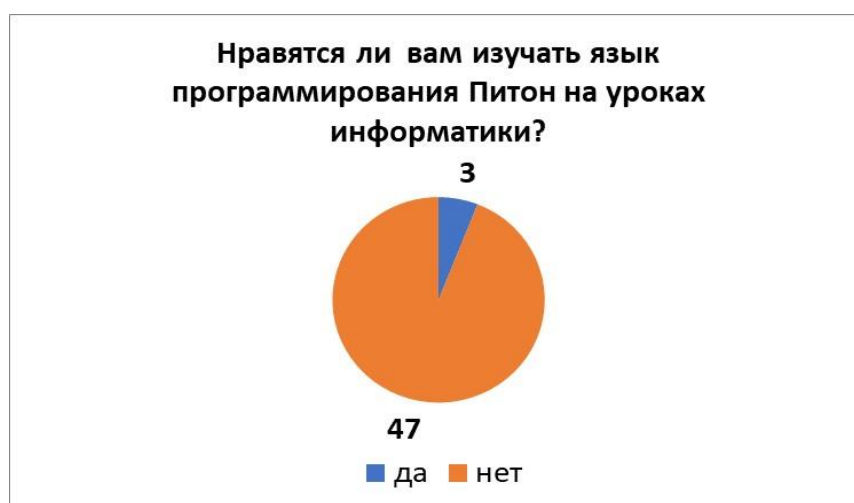


Рисунок 14 – Диаграмма ответов 9 вопрос

При ответе на вопрос: 47 учеников отметили низкий уровень интереса к изучению языка программирования Питон и только 3 ученика ответили – да.

На ответ на это вопрос мы видим, что большинство школьников не справляются с решением задач на программирования, используя классический способ программирования.

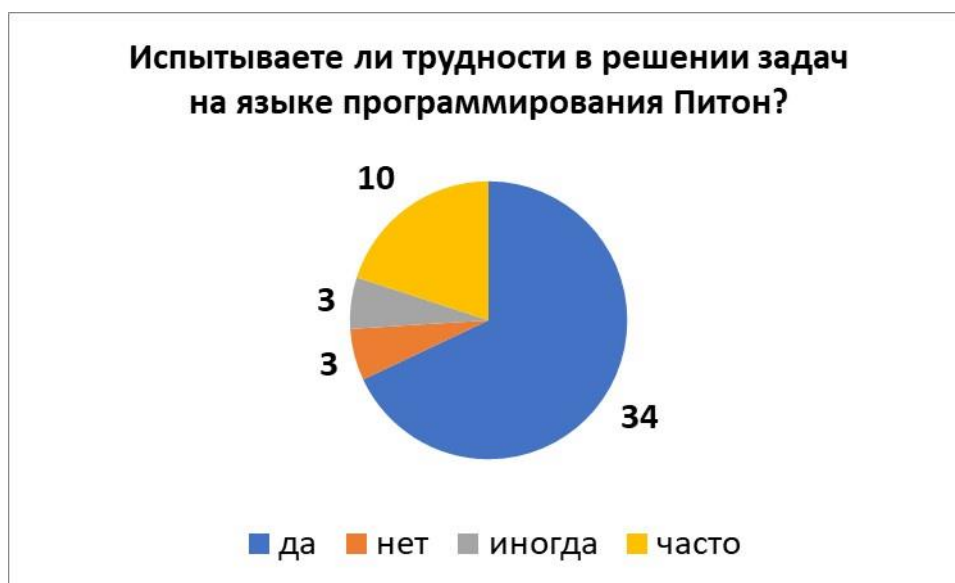


Рисунок 14 – Диаграмма ответов на 10 вопрос

Результаты анкетирования позволяют сделать следующие выводы: телефоны на платформе Android наиболее популярны среди учеников; приложения и телефон в целом имеют огромное значение и занимают достаточно много времени в жизни каждого подростка; ученикам интересна мобильная разработка, но пробовали разрабатывать лишь 5 человек; большинство испытывают трудности при изучении основ структурного программирования на языке программирования Питон.

Опрос 50 учащихся показал, что изучение мобильной разработки приложений обоснованно, так как больше 74% процентов учащихся проявили интерес к возможности созданию мобильных приложений и готовы к изучению в рамках внеурочной деятельности.

Для проведения эксперимента из 50 учащихся была сформирована экспериментальная группа из учеников 7 «А» и 7 «Б» класса. После того,

как мы сформировали экспериментальную группу из 25 учеников, была сформирована контрольная группа из оставшихся 25 учеников.

Для анализа об уровне развития алгоритмического мышления, используем критерии сформированности алгоритмического мышления, определенные Н.Н. Тулькибаевой.

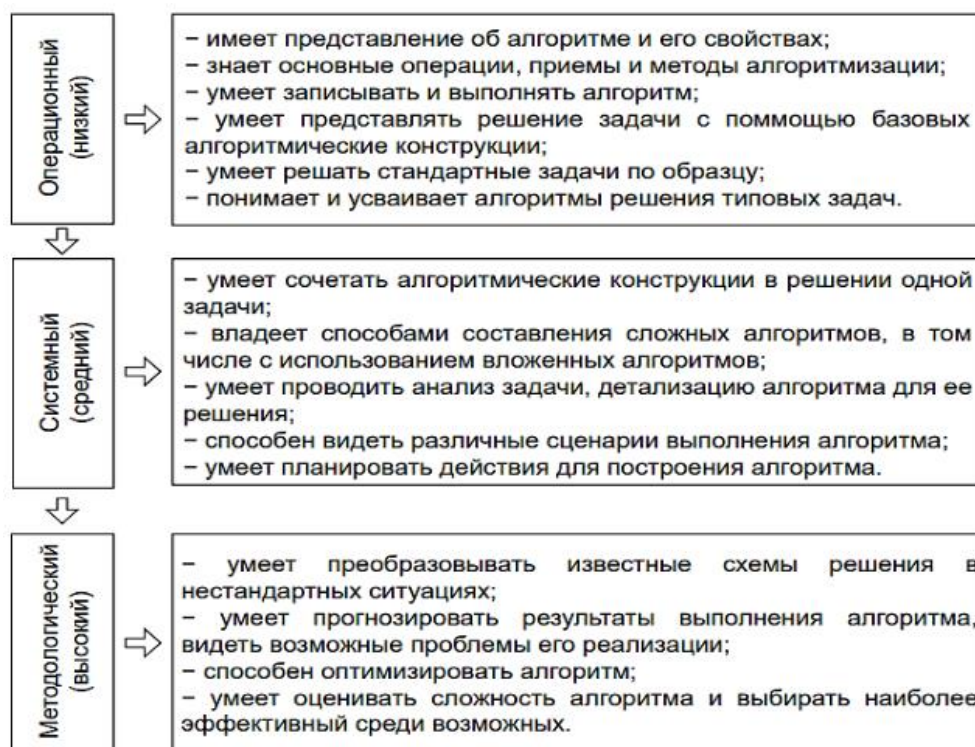


Рисунок 15 – Уровни и критерии сформированности алгоритмического мышления

Для проверки эффективности разработанной нами программы внеурочной деятельности разработке мобильных приложений нужно определить уровень алгоритмического мышления. Для сбора данных об уровне развития алгоритмического мышления, использовали критерии сформированности алгоритмического мышления, определенные Н.Н. Тулькибаевой. Сбор данных двух групп проводился на основе решения стартовой проверочной работы по разделу «Программирование на языке Python». Работа состояла из решения задач на программирование (приложение 1). Задачи взяты из учебника «Информатика. Учебник для учащихся 6 класса общеобразовательной школы», Р.А. Кадиркулов, Г.К.

Нурмуханбетова. Задания были взяты за курс шестого класса, так как сбор данных проводился в начале учебного года. Работа оценивалась с помощью дескрипторов. Максимальный балл составил 10 баллов, что составляет 100%. Операционный (низкий) уровень 1-4 балла, системный (средний) уровень 5-7 баллов, методический (высокий) 8-10 баллов.

Результат проведения констатирующего этапа эксперимента, представлены в таблице 14.

Таблица 14 – Констатирующий этап эксперимента. Результаты контрольной проверки уровня сформированности алгоритмического мышления учащихся 7 класса

Ученики контрольная группа	Сумма баллов	Уровень сформированности алгоритмического мышления	Ученики экспериментальная группа	Сумма баллов	Уровень сформированности алгоритмического мышления
ученик 1	4	низкий	ученик 1	3	низкий
ученик 2	5	средний	ученик 2	6	средний
ученик 3	3	низкий	ученик 3	3	низкий
ученик 4	6	средний	ученик 4	6	средний
ученик 5	4	низкий	ученик 5	3	низкий
ученик 6	4	низкий	ученик 6	4	низкий
ученик 7	3	низкий	ученик 7	5	средний
ученик 8	5	средний	ученик 8	5	средний
ученик 9	7	средний	ученик 9	4	низкий
ученик 10	4	низкий	ученик 10	8	высокий
ученик 11	8	высокий	ученик 11	7	средний
ученик 12	8	высокий	ученик 12	4	низкий
ученик 13	4	низкий	ученик 13	3	низкий
ученик 14	5	средний	ученик 14	3	низкий
ученик 15	7	средний	ученик 15	9	высокий
ученик 16	4	низкий	ученик 16	5	средний
ученик 17	3	низкий	ученик 17	4	низкий
ученик 18	3	низкий	ученик 18	3	низкий
ученик 19	3	низкий	ученик 19	6	средний
ученик 20	4	низкий	ученик 20	4	низкий
ученик 21	4	низкий	ученик 21	6	средний
ученик 22	5	средний	ученик 22	4	низкий
ученик 23	7	средний	ученик 23	8	высокий
ученик 24	4	низкий	ученик 24	4	низкий
ученик 25	5	средний	ученик 25	5	средний

Из таблицы мы видим, что большинство учащихся групп смогли набрать 3-7 балла. Процент выполнения проверочной работы контрольной группы составил 45%, а у экспериментальной группы 40%, из этого можно сделать вывод, что уровень развития алгоритмического мышления учащихся находится на операционном (низком) уровне.

После внедрения разработанной программы внеурочной деятельности проводим проверочную работу в двух группах по итогам изучения разделов «Программирование на языке Python», «Практическое программирование». (соответствует 3 и 4 четверти – 18 учебных часов). Задачи взяты из учебника «Информатика. Учебник для учащихся 7 класса общеобразовательной школы», Р.А. Кадиркулов, Г.К. Нурмуханбетова. Работа оценивалась с помощью дескрипторов. Максимальный балл составил 10 баллов, что составляет 100%. В таблице представлены результаты проверочной работы на контрольно-оценочном этапе эксперимента.

Таблица 15 – Контрольно-оценочный этап эксперимента. Результаты итоговой проверки уровня сформированности алгоритмического мышления учащихся 7 класса

Ученики контрольная группа	Сумма баллов	Уровень сформированности алгоритмического мышления	Ученики экспериментальная группа	Сумма баллов	Уровень сформированности алгоритмического мышления
1	2	3	4	5	6
ученик 1	5	средний	ученик 1	7	средний
ученик 2	6	средний	ученик 2	7	средний
ученик 3	4	низкий	ученик 3	6	средний
ученик 4	6	средний	ученик 4	8	высокий
ученик 5	4	низкий	ученик 5	5	средний
ученик 6	4	низкий	ученик 6	4	низкий
ученик 7	4	низкий	ученик 7	6	средний
ученик 8	5	средний	ученик 8	7	средний
ученик 9	5	средний	ученик 9	5	средний
ученик 10	4	низкий	ученик 10	10	высокий
ученик 11	9	высокий	ученик 11	9	высокий
ученик 12	9	высокий	ученик 12	7	средний
ученик 13	5	средний	ученик 13	4	низкий

Продолжение таблицы 15

1	2	3	4	5	6
ученик 14	5	средний	ученик 14	6	средний
ученик 15	5	средний	ученик 15	8	высокий
ученик 16	4	низкий	ученик 16	7	средний
ученик 17	3	низкий	ученик 17	5	средний
ученик 18	3	низкий	ученик 18	6	средний
ученик 19	3	низкий	ученик 19	7	средний
ученик 20	4	низкий	ученик 20	5	низкий
ученик 21	4	низкий	ученик 21	6	средний
ученик 22	6	средний	ученик 22	4	низкий
ученик 23	7	средний	ученик 23	10	высокий
ученик 24	3	низкий	ученик 24	7	средний
ученик 25	5	средний	ученик 25	7	средний

На данном этапе результат у контрольной группы составил 50%, а у экспериментальной группы 70%.

По данным таблиц построена гистограмма, на которой отображены уровни сформированности алгоритмического мышления в контрольной и экспериментальной группах (рисунок 16).

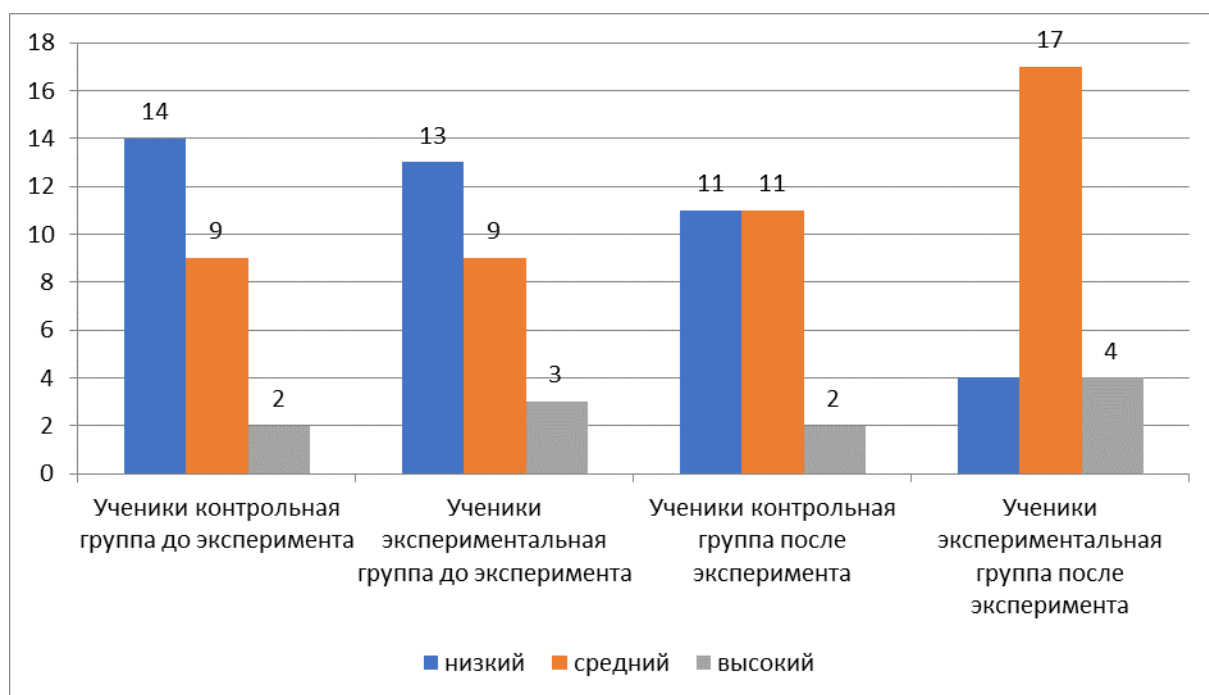


Рисунок 16 – Уровни сформированности алгоритмического мышления в контрольной и экспериментальной группах (чел.)

Получив данные результаты, можно приступать к доказательству достоверности этих измерений. Для этого мы будем использовать Т-

критерий Вилкоксона. Данный критерий использован для сравнения полученных данных тестирования экспериментальной группы до прохождения курса и после него, что не противоречит его предназначению – измерение на одной выборке обучающихся, но с разными условиями. Он даёт возможность распознать направленность измерений и их выраженность. Также, определяется интенсивность сдвига показателей по направлениям. Производится выделение двух гипотез.

Гипотеза H_0 : показатели уровня развития алгоритмического мышления после проведения программы внеурочной деятельности превышают значения показателей до эксперимента и являются статистически значимыми.

Гипотеза H_1 : показатели уровня развития алгоритмического мышления после проведения программы внеурочной деятельности не превышают значения показателей до эксперимента и не является статистически значимым.

Для подсчета этого критерия нет необходимости упорядочивать ряды значений по нарастанию признака. Первый шаг в подсчете Т-критерия – вычитание каждого индивидуального значения «до» из значения «после» представлены в таблице 16.

Таблица 16 – Вычитание значений «констатирующего этапа» из значений «контрольного этапа»

До измерения, $t_{до}$	После измерения, $t_{после}$	Разность ($t_{до}-t_{после}$)	Абсолютное значение разности
1	2	3	4
3	7	4	4
6	7	1	1
3	6	3	3
6	8	2	2
3	5	2	2
4	4	0	0
5	6	1	1
5	7	2	2

Продолжение таблицы 16

1	2	3	4
4	5	1	1
8	10	2	2
7	9	2	2
4	7	3	3
3	4	1	1
3	6	3	3
9	8	-1	1
5	7	2	2
4	5	1	1
3	6	3	3
6	7	1	1
4	5	1	1
6	6	0	0
4	4	0	0
8	10	2	2
4	7	3	3
5	7	2	2

Исключим нулевые сдвиги. Так как в таблице имеются связанные ранги (одинаковый ранговый номер) 1-го ряда, произведем их переформирование. Переформирование рангов производится без изменения важности ранга, то есть между ранговыми номерами должны сохраниться соответствующие соотношения (больше, меньше или равно). Также не рекомендуется ставить ранг выше 1 и ниже значения равного количеству параметров (в данном случае $n = 22$). Переформирование рангов производится в таблице 17.

Таблица 17 – Переформирование рангов

Номера мест в упорядоченном ряду	Расположение факторов по оценке эксперта	Новые ранги
1	2	3
1	1	4.5
2	1	4.5
3	1	4.5
4	1	4.5
5	1	4.5
6	1	4.5
7	1	4.5
8	1	4.5
9	2	12.5
10	2	12.5

Продолжение таблицы 17

<i>1</i>	<i>2</i>	<i>3</i>
11	2	12.5
12	2	12.5
13	2	12.5
14	2	12.5
15	2	12.5
16	2	12.5
17	3	19
18	3	19
19	3	19
20	3	19
21	3	19
22	4	22

Следующий шаг в подсчете Т-критерия предполагает вычисление рангового номера разности (таблица 18).

Таблица 18 – Ранговый номер разности

До измерения, $t_{до}$	После измерения, $t_{после}$	Разность ($t_{до} - t_{после}$)	Абсолютное значение разности	Ранговый номер разности
3	7	4	4	22
6	7	1	1	4.5
3	6	3	3	19
6	8	2	2	12.5
3	5	2	2	12.5
5	6	1	1	4.5
5	7	2	2	12.5
4	5	1	1	4.5
8	10	2	2	12.5
7	9	2	2	12.5
4	7	3	3	19
3	4	1	1	4.5
3	6	3	3	19
9	8	-1	1	4.5
5	7	2	2	12.5
4	5	1	1	4.5
3	6	3	3	19
6	7	1	1	4.5
4	5	1	1	4.5
8	10	2	2	12.5
4	7	3	3	19
5	7	2	2	12.5
Сумма				253

Сумма по столбцу рангов равна $\Sigma=253$. Проверка правильности составления матрицы на основе исчисления контрольной суммы:

$$\sum x_{ij} = \frac{(1+n)n}{2} = \frac{(1+22)22}{2} = 253$$

Сумма по столбцу и контрольная сумма равны между собой, значит, ранжирование проведено правильно.

Теперь отметим те направления, которые являются нетипичными, в данном случае – отрицательными. Сумма рангов «редких» направлений составляет эмпирическое значение критерия T:

$$T = \sum R_t = 4.5 = 4.5$$

Находим критические значения для T-критерия Вилкоксона для $n=22$:

$$T_{кр} = 55 \quad (p \leq 0.01)$$

$$T_{кр} = 75 \quad (p \leq 0.05)$$

Зона значимости в данном случае простирается влево, действительно, если бы "редких", в данном случае положительных, направлений не было совсем, то и сумма их рангов равнялась бы нулю.

В данном же случае эмпирическое значение T попадает в зону значимости: $T_{эмп} < T_{кр} (0,01)$.

Гипотеза H_0 принимается. Показатели после эксперимента превышают значения показателей до опыта.

Ось значимости:

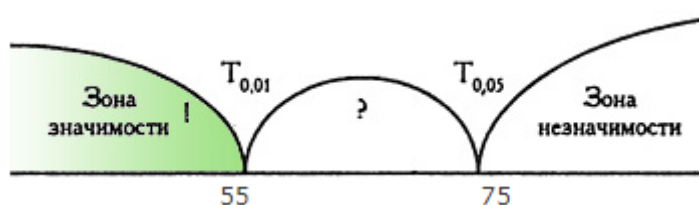


Рисунок 17 – Ось значимости T-критерий Вилкоксона.

Полученное эмпирическое значение $T_{эмп}$ находится в зоне значимости.

Гипотеза H_0 подтверждается, показатели уровня развития алгоритмического мышления после проведения программы внеурочной деятельности превышают значения показателей до эксперимента.

Вывод: сформулированная нами гипотеза о том, развитие алгоритмического мышления обучающихся будет более эффективно, если разработать и внедрить для учащихся седьмых классов программу внеурочной деятельности «Основы программирования мобильных приложений под ОС Android», направленную на развитие алгоритмического мышления подтверждена.

По завершению занятий курса, с целью выявления эффективности курса, проведено анкетирование «Отношение к курсу» (приложение 2)

Результаты анкетирования:

1. Всем участникам понравилась программа внеурочной деятельности.
2. За время занятий учащиеся научились разрабатывать приложения и работать в среде визуального программирования MIT APP Inventor.
3. Самым интересным, большинство отметило тестирование приложений на собственных мобильных приложениях.
4. Самым сложным – составить алгоритм для приложения.
5. Двадцать человек заинтересованы в мобильной разработке, 5 человек затрудняются ответить.
6. Двадцать один человек хотел бы продолжать изучать мобильную разработку в рамках курса, 4 человека затрудняются ответить.
7. 20 из 25 человек будут продолжать разрабатывать мобильные приложения самостоятельно.

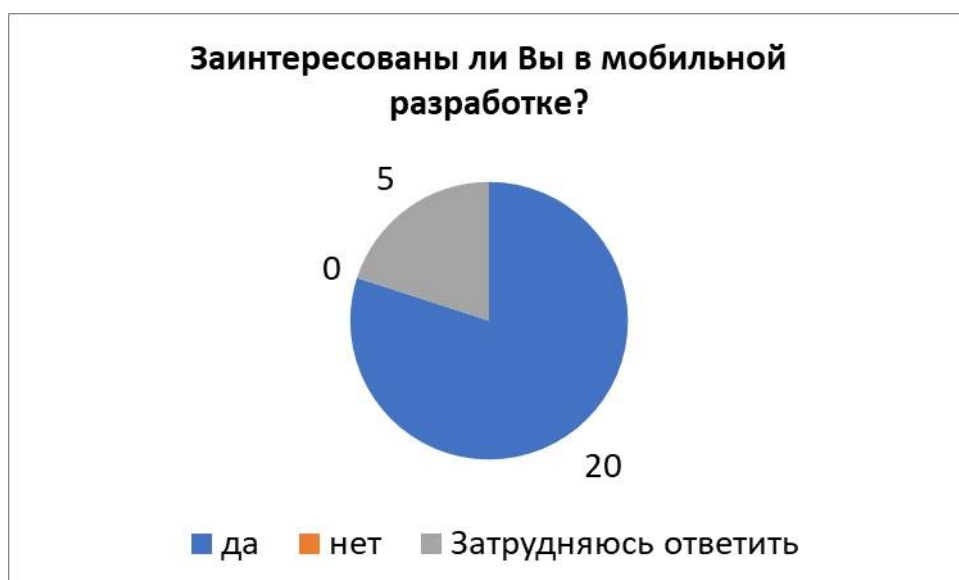


Рисунок 18 – Диаграмма ответов на 5 вопрос

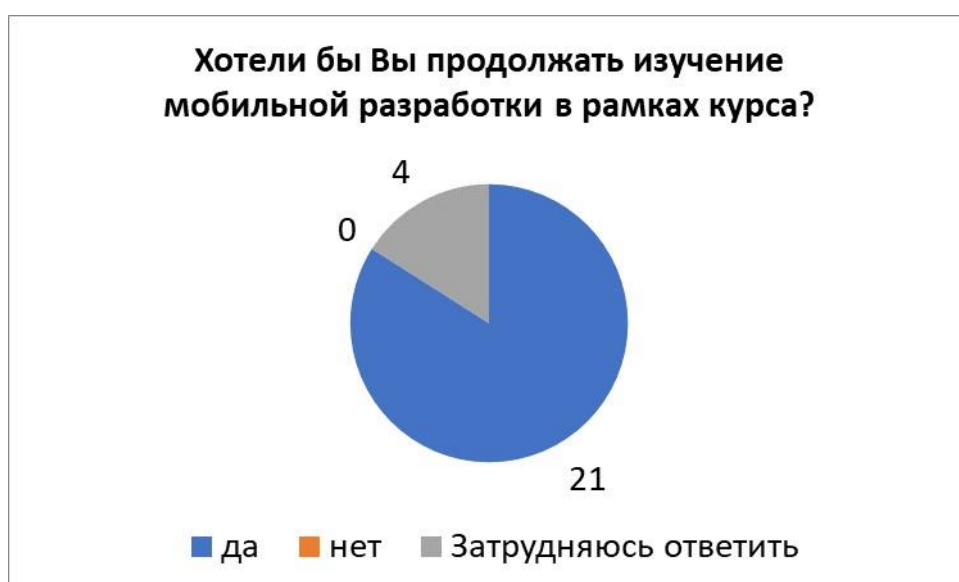


Рисунок 19 – Диаграмма ответов на 6 вопрос

Выводы по 3 главе

Экспериментальная работа проводилась в КГУ «Общеобразовательная школа № 23 им. М. Козыбаева г. Костаная» в три этапа. На первом этапе (2021 г.) осуществлялся анализ учебно-методической литературы и нормативно-правовых документов в изучаемой области; изучался существующий опыт в области разработки курсов по программированию для школьников; обосновывались и

формулировались принципы формирования содержания программы внеурочной деятельности «Основы программирования мобильных приложений под ОС Android»; подбирались методы обучения и педагогические технологии, адекватные целям исследования. На втором этапе (2021-2022 гг.) проводилось экспериментальное подтверждение результативности применения разработанной программы внеурочной деятельности для школьников. Программа позволяет развивать алгоритмическое мышление, повысить уровень знаний учащихся в рамках изучения раздела «Программирование на языке Питон», повысить уровень мотивации обучающихся к изучению программирования.

На третьем этапе (2022 г.) проведена статистическая обработка данных и их интерпретация; формулировались выводы; оформлялось диссертационное исследование. Результат педагогического эксперимента позволил сделать вывод, что содержание программы внеурочной деятельности «Основы программирования мобильных приложений под ОС Android» позволило обеспечить положительную динамику изменения уровня алгоритмического мышления, повысить мотивацию к изучению программирования учащихся 7-х классов в КГУ «Общеобразовательная школа № 23 им. М. Козыбаева г. Костаная».

ЗАКЛЮЧЕНИЕ

В ходе проведения педагогического эксперимента были рассмотрены и проанализированы теоретическое обоснование и практическую проверку результативности методики развития алгоритмического мышления школьников на занятиях по созданию мобильных приложений во внеурочной деятельности.

В результате рассмотрения теоретической стороны проблемы мы смогли рассмотреть сущность и содержание понятия «алгоритмическое мышление» с разных сторон. Таким образом, понятие «алгоритмическое мышление» мы рассматриваем как особый тип мышления, при котором реализуются умения планировать свои действия, умения учитывать различные ситуации и действовать соответственно, умение легко рассуждать об алгоритмических процессах». В структуре алгоритмического мышления можно выделить компоненты, которые выделили на основе анализа работ И.Н. Слинкиной, А.И. Газейкиной А.Г. Гейна: когнитивный, деятельностный и рефлексивно-оценочный. Показателями развития когнитивного компонента является полнота знаний о понятии «алгоритм». Показатели деятельностного компонента – сформированность умений работать с алгоритмами. Рефлексивно-оценочный компонент диагностируем по следующим умениям умения формулировать цель, планировать реализовывать, контролировать, оценивать и рефлексировать свою деятельность.

Анализ учебной программы в рамках обновленного содержания образования в рамках изучения информатики раздел «Компьютерное мышление» важно, так как это основа для изучения программирования. Мы выявили методические особенности преподавания алгоритмизации в предмете информатика, а также проанализировали различные методические подходы к изучению алгоритмизации. Современное общество требует от нового поколения умения планировать свои действия,

находить необходимую информацию для решения задачи, моделировать будущий процесс. Поэтому программа развивающая мышление, формирующий соответствующий стиль мышления, является важным и актуальным. По этой причине информатика и её основная часть – программирование, должны быть одними из базовых предметов для изучения. Изучив теоретические аспекты проблемы развития изучаемого процесса школьников посредством создания мобильных приложений, мы предположили, что развитие алгоритмического мышления школьников во внеурочной деятельности посредством создания мобильных приложений будет результативным, если методика организации внеурочной деятельности включает следующие условия: представление учителем алгоритма по достижению цели, формулирование цели учащимися, составление плана действий, составление учащимися простых и сложных алгоритмов для достижения цели. Для достижения этой цели через обучение программированию и создана программа «Основы программирования мобильных приложений под ОС Android».

Для проверки выдвинутой гипотезы нами была проведена опытно-экспериментальная работа. Эксперимент проводился в 2021 – 2022 учебном году среди учащихся 7-х классов в КГУ «Общеобразовательная школа № 23 им. М. Козыбаева г. Костаная».

В начале эксперимента на констатирующем этапе эксперимента средний уровень сформированности алгоритмического мышления составил 45% и 46% контрольной и экспериментальной группы. Для повышения уровня алгоритмического мышления была разработана программа дополнительного образования «Основы программирования мобильных приложений под ОС Android». После чего был проведен контрольно–оценочный этап эксперимента, который показал, что средний уровень знаний экспериментальной группы увеличился на 24%, а в сравнении с контрольной группой на 7%. Мы выяснили, что между

организацией занятий повнеурочной деятельности по созданию мобильных приложений и развитием алгоритмического мышления школьников есть высокая прямая связь. Таким образом, в нашем исследовании мы реализовали поставленные цели и задачи и обосновали возможность развития алгоритмического мышления школьников на занятиях по созданию мобильных приложений во внеурочной деятельности. Гипотеза, поставленная нами в начале исследования, получила свое подтверждение в педагогическом эксперименте.

Построенное таким образом изучение программирования способствует не только достижению предметных результатов.

Таким образом, использование среды MIT App Inventor при изучении темы «Алгоритмизация и программирование» способствует достижению предметных, метапредметных и личностных результатов образования. Полученные нами результаты исследования и разработанная программа может быть использована педагогами школы, студентами, педагогами дополнительного образования.

Педагогический эксперимент проходил в три этапа.

Первый этап – поисково-подготовительный (2020 г.). Был проведен теоретический анализ проблемной области, изучена литература по проблеме исследования, сформулирована гипотеза, объект и предмет исследования.

Второй этап – опытно-экспериментальный (2020 – 2021 гг.). Разработана и реализована программа внеурочной деятельности по обучению школьников созданию мобильных приложений в MIT App Inventor, обеспечивающих повышение уровня развития алгоритмического мышления обучающихся; проведен констатирующий эксперимент, определена экспериментальная группа.

Третий этап – контрольно-обобщающий (2022 г.). Завершен качественный и количественный анализ полученных результатов;

статистическая обработка данных; результаты оформлены в виде диссертационной работы.

Итоги проведения педагогического эксперимента подвели к заключению о том, что проведение программы внеурочной деятельности по созданию мобильных приложений позволило обеспечить положительную динамику изменения уровня алгоритмического мышления учащихся. Результаты эксперимента подтверждают гипотезу исследования: развитие алгоритмического мышления обучающихся будет более эффективно, если разработать и внедрить для учащихся седьмых классов программу внеурочной деятельности «Основы программирования мобильных приложений под ОС Android», направленную на развитие алгоритмического мышления.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Автоматический расчет T – критерия Вилкоксона [Электронный ресурс]. – <https://www.psychol-ok.ru/statistics/mannwhitney/>, 2022.
2. Алешкина О.В. Развитие алгоритмического мышления в процессе изучения темы «Циклы» [Электронный ресурс]/ О.В. Алешкина //Социальная сеть работников образования. – Режим доступа: <https://nsportal.ru/shkola/informatika-i-ikt/library/2013/03/26/razvitie-algoritmicheskogo-myshleniya-v-protssesse> (дата обращения 14.11.2022)
3. Амосов Н.М. Моделирование мышления и психики / Амосов Н.М.. – Киев: Наукова думка, 2014. - 303 с.
4. Арменков А.Г. Обучение программированию школьников. Создание приложения в среде MIT App Inventor // Вестник науки и образования, 18 (72). – 2019, С. 73-76.
5. Босова Л.Л. Непрерывный курс информатики: проблемы и возможности / Л.Л. Босова//Информатика в школе: прошлое, настоящее и будущее: материалы Всеросс. науч. -метод. конф. по вопросам применения ИКТ в образовании / отв. за вып. Ю. А. Аляев, И. Г. Семакин; Перм. гос. нац. исслед.ун-т. – Пермь. - 2014. – С.228-231
6. Виноградов А.В. Программируем игры для мобильных телефонов. - М. – Триумф, 2007. – 272с.
7. Воронцова Л.А. Из опыта обучения алгоритмизации и программированию в основной школе/ Л.А. Воронцова // Информатика в школе. – 2012. – №9. – С. 44-48
8. Выготский Л.С. Педагогическая психология / Под ред. В.В. Давыдова. – М.: Педагогика-Пресс, 1999. – 536 с.
9. Газейкина А.И. Обучение программированию будущего учителя информатики // Педагогическое образование в России. 2012. № 5. С. 45-48.
10. Газейкина А.И. Формирование когнитивных универсальных учебных действий при обучении робототехнике учащихся основной школы

//Педагогическое образование в России. 2015. № 7. С. 42—49

11. Газейкина А.И. Формирование научного мировоззрения будущего ИТ — специалиста в процессе обучения программированию // Педагогическое образование в России. 2015. №7. С. 36—41.

12. Газейкина А.М. Обучение будущего учителя информатики конструированию учебных заданий, направленных на формирование метапредметных результатов обучения // Педагогическое образование в России. 2014. № 8. С. 159 -164.

13. Гальперин П.Я. Психология мышления и учение о поэтапном формировании умственных действий. Москва: Изд-во Москва ун-та, 1976. 150 с.

14. Гальперин П.Я. Психолого-педагогические проблемы программированного обучения на современном этапе: Материалы всесоюзной конференции по программированному обучению / П.Я. Гальперин, З.А. Решетова, Н.Ф. Талызина. – Москва, 1966.

15. Гафурова Н.В. Методика обучения информационным технологиям. Теоретические основы: учебное пособие / Н.В. Гафурова, Е.Ю. Чурилова. – Красноярск: Сибирский федеральный университет, 2012– 111 с. – URL: <http://biblioclub.ru/index.php?page=book&id=229302>. – (дата обращения: 15.05.2021)

16. Гейн А.Г. Методика изучения алгоритмизации с помощью учебных исполнителей: книга для учителя. Москва: Просвещение, 2011.144с.

17. Гриффитс Д.Д. HeadFirst. Программирование для Android / Д.Д. Гриффитс. – Санкт-Петербург: Питер, 2016. – 704 с.: – (Серия «HeadFirstO'Reilly»).

18. Давыдов В.В. Определение мышления // Теория и методология. Под ред. Л.В. Берифаи. 2006. № 2. С. 3-16.

19. Елисеев Р.П. Разработка приложений для ОС Android / Р. Елисеев //НОУ «ИНТУИТ», 2003 – 2016. – URL: <http://www.intuit.ru/studies/courses/3703/945/info> – (дата обращения: 15.05.2020).

20. Ершов А.П. Введение в теоретическое программирование: учебное пособие / Ершов А.П. – Москва: Наука, 1977. – 280 с.: ил. – URL: <http://bookre.org/reader?file=1499378>. – (дата обращения: 16.05.2021). – Текст: электронный.

21. Ершов А.П. Программирование – вторая грамотность. URL: http://ershov.iis.nsk.su/ru/second_literacy/article (дата обращения 11.08.2021).

22. Краснокутская, Н.В. Элективные курсы как средство построения индивидуальных образовательных программ / Амурский научный вестник. - 2015. – №4. – С. 62-70.

23. Измestьев Н.С., Сокольская, М. А. Особенности формирования и развития алгоритмического мышления с помощью объектно-ориентированного программирования // Материалы конференции «Современные тенденции и проекты развития информационных систем и технологий. – 2015. – № 4 (13). – С. 167–172.

24. Калитина, В.В. Алгоритмические ментальные карты как эффективное средство обучения программированию / В. В. Калитина, Т.П. Пушкарева, Т. А. Степанова // Сборник статей Международной научно-практической конференции «Фундаментальные и прикладные научные исследования». - Москва: РИО ЕФИР. - 2015. - С.179-181

25. Копаев А. В. О практическом значении алгоритмического стиля мышления / А.В. Копаев // Школа и производство. – 2003. – № 6. – С. 6–11

26. Кузнецов, А. А. Элективные курсы образовательной области «Информатика»/Элективные курсы в профильном обучении: Образовательная область «Информатика». – Москва: Вита-Пресс, 2004.С. 5 – 20.

27. Кузнецов А. А. Школьная информатика: вчера, сегодня, завтра / А. А. Кузнецов, Т. Б. Захарова // Информатика и образование. – 2014. – № 10. – С. 3–6.

28. Кушниренко А.Г., Лебедев, Г.В. Информатика: 12 лекций о том, для чего нужен школьный курс информатики / А.Г Кушниренко // Хрестоматия. – Москва : МГУ, 1982. – 489 с. ISBN 5-222-02404-0.

29. Лапчик М. П. Методика преподавания информатики / М. П. Лапчик и др.; под общей ред. М.П. Лапчика. Москва: – Академия, 2001. – 624 с. ISBN 5-09-000598-2.

30. Ливенец М.А., Ярмахов Б.Б. Программирование мобильных приложений в MIT App Inventor: Практикум. – 2016. – 100 с. 7.

31. Малеев В. В. Общая методика преподавания информатики: учебно-методическое пособие / В. В. Малаев. – Воронеж : ВГПУ, 2005. – 271 с. ISBN 5-88519-365-7

32. Малев, В.В. Практикум по методике преподавания информатики: практикум / В.В. Малев, А.А. Малева. – Воронеж: ВГПУ, 2006 – 146 с. – URL: <http://biblioclub.ru/index.php?page=book&id=103304>. – (дата обращения: 15.05.2020). – Текст: электронный Моррисон М. Создание игр для мобильных телефонов. – М.: ДМК Пресс, 2006. - 494 с. 4.

33. О внесении изменений в приказ Министерства просвещения Республики Казахстан от 16 сентября 2022 года № 399 «Об утверждении типовых учебных программ по общеобразовательным предметам и курсам по выбору уровней начального, основного среднего и общего среднего образования» Информационно-правовая система нормативных правовых актов Республики Казахстан [сайт]. – 2018. – URL: <https://adilet.zan.kz/rus/docs/V1800017757> (дата обращения: 16.03.2022).

34. Паперт С.Д. Переворот в сознании: дети, компьютеры и плодотворные идеи. М.: Педагогика, 1989

35. Поляков К. Ю. Пояснительная записка к «Информатика» для 10-11 классов общеобразовательных учреждений (углубленный уровень) / К. Ю. Поляков, Е.А. Еремин. – Москва: БИНОМ, 2013. – 8 с.

36. Поляков К.Ю. Информатика. Углубленный уровень: учебник для 10 класса: в 2 ч. Ч. 2 / К.Ю. Поляков, Е.А. Еремин. – Москва: БИНОМ, 2013. – 304 с. 28.

37. Пьянзина И.Н. Программируем с MIT App Inventor 2. Информатика в школе. 2018. №3. – С. 19-22.

38. Пять платформ для создания и размещения онлайн-курсов [Электронный ресурс]. – Режим доступа <https://test.ru/2015/11/27/5resources-formooc/>.

39. Рагулина М.И. Элективные курсы информатики: классификация и спецификация содержания / М.И. Рагулина, Л.В. Смолина // Вестник Омского государственного педагогического университета: эл. науч. Журнал. – Омск. – Вып. 2006.

40. Семакин И.Г. Информатика. Углубленный уровень: учебник для 11 класса: в 2 ч. Ч.2 / И. Г. Семакин, Е. К. Хеннер, Л. В. Шестакова. – Москва: БИНОМ, 2014. – 216 с.: ил.

41. Семакин, И.Г. Пояснительная записка к «Информатика и ИКТ» (профильный уровень) для 10 класса общеобразовательной школы/ И.Г. Семакин, Т.Ю. Шеина, Л.В. Шестакова. – Москва: БИНОМ, 2014.12 с.

42. Скаткин М.Н., Логинов Н.Н. Исследование по алгоритмизации обучения//Советская педагогика. 1967, №4. С. 145-147.

43. Слинкина И.Н. Использование компьютерной техники в процессе развития алгоритмического мышления у младших школьников: автореф. дис. канд. пед. наук/ Уральский гос. педагогический ун-т. Екатеринбург, 2000. 22 с

44. Соколова В.В. Разработка мобильных приложений: учебное пособие для среднего профессионального образования / В. В. Соколова. —

Москва: Издательство Юрайт, 2020. — 175 с. – (дата обращения: 10.06.2020). – Текст: электронный

45. Хрестоматия по психологии: под ред. Ю.Б. Гиппенрейтер, В.Ф. Спиридонова, М. В. Фаликман, В.В. Петухова. Москва: Лабиринт, 2008. 493с.

46. Шалина О.Н. Некоторые аспекты обучения школьников созданию Android-приложений // Достижения и приложения современной информатики, математики и физики. Материалы VI Всерос. науч.практ. заоч. конф. - Уфа: Издательство «Башкирский государственный университет», 2017 – С. 134-139.

47. Ярмахов Б.Б. MIT App Inventor – каждый может создать мобильное приложение. URL: newtonew.com/app/mit-app-iventor-intro.

ПРИЛОЖЕНИЕ 1

Стартовая проверочная работа по разделу «Программирование на языке Python».

Задачи взяты из учебника «Информатика. Учебник для учащихся 6 класса общеобразовательной школы», Р.А. Кадиркулов, Г.К. Нурмуханбетова.

Задание № 1

Игра в баскетбол состоит из 4 периодов. Баллы, набранные обеими командами в каждом периоде, были показаны по окончании периода. Составь программу, которая выводит итоговые результаты игры на экран в представленном формате.

Пример	Результат	Примечание
15 7	71 : 73	15 + 12 + 25 + 19 = 71
12 18		7 + 18 + 33 + 15 = 73
25 33		
19 15		

Дескриптор:

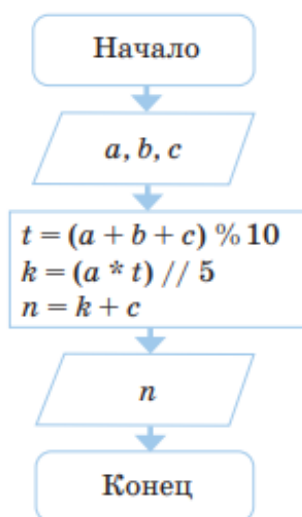
- Составляет программный код, использует оператор ввода данных.
- Составляет программный код, использует оператор присваивания
- Составляет программный код, использует оператор вывода.

(3 балла)

Задание № 2

Разработай программу на языке Python по блок-схеме. Вычисли результат, подставив заданные значения переменных a, b, c в блок-схему.

а) a = 5, b = 2, c = 6, d = 3 б) a = 1, b = 1, c = 2, d = 2



Дескриптор:

- Составляет программный код, использует оператор ввода данных.
- Составляет программный код, использует оператор присваивания
- Составляет программный код, использует оператор вывода.
- Вычисляет результат, используя значения переменных.

(4 балла)

Задание № 3

Составь текст задачи с переменными a , b , c по рисунку. Составь программу и блок-схему по составленному условию задачи.



Дескриптор:

- Составляет текст задачи, используя рисунок.
- Составляет блок-схему по составленному условию задачи.
- Составляет программный код на языке программирования.

(3 балла)

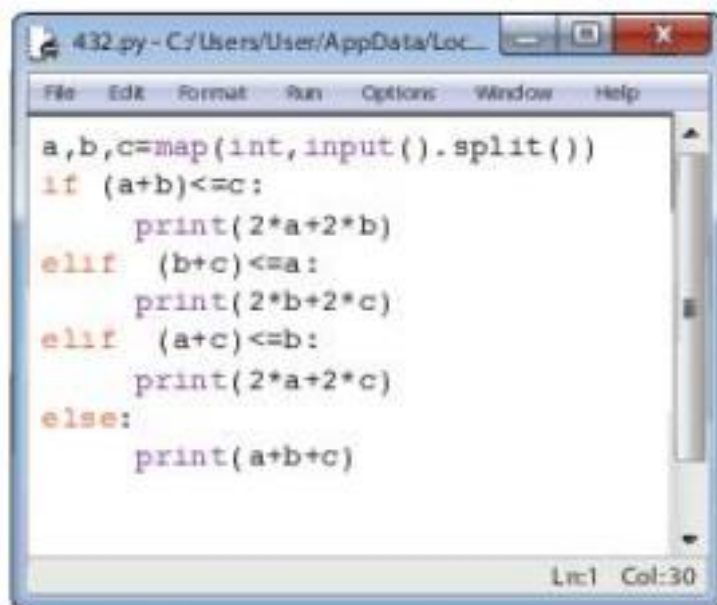
Проверочная работа по разделу «Программирование на языке Python» (после эксперимента)

Задачи взяты из учебника «Информатика. Учебник для учащихся 7 класса общеобразовательной школы», Р.А. Кадиркулов, Г.К. Нурмуханбетова.

Задание № 1.

Сегодня день рождения Динары. К ней в гости придут друзья. Чтобы угостить гостей Динаре нужно сходить в 2 магазина возле дома. Расстояние от дома Динары до 1 магазина, a метров, до второго магазина b метров, расстояние между двумя магазинами c метров. Помогите Динаре найти самый короткий путь чтобы сходить в оба магазина и вернуться домой.

1. Проанализируйте программный код. Определите результаты работы программы, если 1) $a = 4, b = 5, c = 10$ 2) $a = 5, b = 5, c = 9$



```
a,b,c=map(int,input().split())
if (a+b)<=c:
    print(2*a+2*b)
elif (b+c)<=a:
    print(2*b+2*c)
elif (a+c)<=b:
    print(2*a+2*c)
else:
    print(a+b+c)
```

2. Проанализируйте полученные результаты и дайте объяснение.

3. Составьте блок-схему к задаче. К какому виду относится алгоритмическая структура составленной блок-схемы? Дайте комментарии.

4. Оцените значение представления алгоритма в графическом виде при переходе от построения алгоритма к процессу разработки программы поставленной задачи.

Дескриптор:

- Выполняет трассировку алгоритма.
- Анализирует результаты, дает объяснение.
- Составляет блок-схему к задаче.
- Оценивает значение представления алгоритма.

(4 балла)

Задание № 2

Аружан очень капризная девочка. На прошлых летних каникулах она побывала в Таразе и в Алматы, и ей очень понравилось. Поэтому на следующие летние каникулы она снова хочет посетить два разных города. При этом Аружан хочет снова побывать в Таразе или в Алматы, но не в обоих городах одновременно - повторить прошлогодний маршрут полностью ей будет скучно. Определите, подходит ли предлагаемый маршрут под требования Аружан.

Входные данные:

Две строки — названия городов.

Выходные данные:

Выводится "ДА", если два города удовлетворяют пожеланиям Аружан или выводится "НЕТ", если два города не удовлетворяют пожеланиям Аружан.

Определите переменные, тип.

Составьте математическую модель решения задачи.

Составьте блок-схему решения задачи.

Напишите программный код решения задачи.

Составьте варианты значений входных данных для тестирования условий, соответствующих решению.

Дескриптор:

- Описывает переменные к задаче
- Составляет модель решения задачи.
- Строит блок – схему решения задачи.
- Пишет программный код решения задачи, используя условный оператор, оператор ввода – вывода.
- Составляет тесты к задаче.

(6 баллов)

ПРИЛОЖЕНИЕ 2

Анкета «Отношение к курсу»

Анкета проводится в рамках программы внеурочной деятельности «Разработка приложений для мобильных устройств под ОС Android».

Пожалуйста, уделите несколько минут и заполните анкету. Ваши ответы помогут определить, насколько интересным и полезным был предложенный Вам курс, и есть ли у Вас желание продолжить заниматься разработкой мобильных приложений.

Вопросы:

1. Понравился ли Вам предложенный курс?

а) Да б) Нет в) Затрудняюсь ответить.

2. Чему вы научились за время занятий?

3. Что было самым интересным?

Что было самым сложным в разработке приложений?

4. Заинтересованы ли Вы в мобильной разработке?

а) Да б) Нет в) Затрудняюсь ответить.

5. Хотели бы Вы продолжать изучение мобильной разработки в рамках курса?

а) Да б) Нет в) Затрудняюсь ответить.

6. Будете ли Вы самостоятельно продолжать разрабатывать мобильные приложения?

а) Да б) Нет в) Затрудняюсь ответить.

Спасибо за Ваши ответы.