

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
им. В.П. АСТАФЬЕВА
(КГПУ им. В.П. Астафьева)

Институт математики физики и информатики
Кафедра технологии и предпринимательства

БИРЮЛЕВА ДАРЬЯ ЮРЬЕВНА

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**РАЗВИТИЕ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ
ВОСЬМИКЛАССНИКОВ НА ВНЕУРОЧНЫХ ЗАНЯТИЯХ РУТНОН-
ПРОГРАММИРОВАНИЕ**

Направление подготовки 44.03.01 Педагогическое образование

Направленность (профиль) образовательной программы
Технология с основами предпринимательства

ДОПУСКАЮ К ЗАЩИТЕ

Заведующий кафедрой канд. технич. наук,
доцент Бортновский С.В.

Научный руководитель канд. технич. наук,
доцент Шадрин И.В.

Дата защиты

Обучающийся

Бирюлева Д.Ю.

Оценка

Красноярск 2023

Оглавление

ВВЕДЕНИЕ	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ФОРМИРОВАНИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ	6
1.1. Понятие алгоритма и алгоритмического мышления	6
1.2. Особенности когнитивной деятельности обучающихся 8 классов	12
1.3. Анализ языков программирования в контексте развития алгоритмического мышления	16
Выводы по главе 1	26
ГЛАВА 2. РАЗРАБОТКА КУРСА ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON	27
2.1. Анализ онлайн-курсов на языке программирования Python.....	27
2.2. Разработка курса по программированию на языке Python.....	31
Вывод по главе 2.....	38
ЗАКЛЮЧЕНИЕ.....	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	41

ВВЕДЕНИЕ

В постиндустриальном обществе, где информация и компьютерные технологии выходят на первый план, человеку необходимо обладать определенным количеством умений и навыков. Начать стоит с того, что люди каждый день живут по алгоритмам, идут на работу, принимают пищу, едут в машине и т.д. Алгоритмы являются неотъемлемой частью жизни любого человека. Частым явлением является то, когда человек перед выполнением какой-либо сложной задачи начинает теряться, переживать, опускать руки. Дело в том, что одним из главных навыков, которым должен овладеть человек является разбивать большую задачу на подзадачи, для более быстрого выполнения задания. Этот навык необходим любому для более эффективного функционирования в современном мире, поэтому важно развивать алгоритмическое мышление.

В основе алгоритмического мышления лежат алгоритмы. Алгоритм – это точное предписание, определяющее последовательность действий, обеспечивающую получение требуемого результата из исходных данных.

В свою очередь, алгоритмическое мышление – это совокупность мыслительных действий и приемов, нацеленных на решение задач, в результате которых создает алгоритм, являющийся продуктом человеческой деятельности.

Каким образом развивать алгоритмическое мышление? Самой эффективной, для развития алгоритмического мышления, является область программирования. Как известно, изучение области программирования в школах начинается на уроках информатики в 8 классе.

Необходимо заметить, что в российских школах ученики демонстрируют невысокие результаты при решении задач по программированию. Причин этого может быть несколько – недостаточное количество часов в базовом курсе информатики на изучение алгоритмизации и программирования, низкая мотивация учеников к изучению программирования, а также выбор языка программирования играет важную роль.

Однако стоит заметить, что язык Python в настоящее время изучается в общеобразовательных учреждениях наравне с языком Pascal, значит популярность такого языка программирования как Python действительно велика.

Python имеет несколько особенностей, которые делают его отличным выбором для обучения программированию в школе - это универсальность и простота. Также язык обладает рядом преимуществ – код программы может быть написан на любой платформе, свободная лицензия и открытый доступ к исходным кодам, имеется большой репозиторий, хранящий колоссальное количество подключаемых библиотек для решения широкого круга задач от создания простейших программ до разработок игр и приложений.

Одним из методов развития алгоритмического мышления является внеурочный курс по программированию на языке Python. На курсе ученики учатся разрабатывать алгоритмы решения задач и затем писать код. Различные задания формируют у учеников навыки логического мышления, учат решать проблемы шаг за шагом, развивают навыки аналитического и алгоритмического мышления и повышают их уверенность в программировании.

Актуальность работы обусловлена необходимостью развития алгоритмического мышления у школьников для повышения эффективности их функционирования в процессе обучения и реализации своих навыков в будущей учебной или профессиональной деятельности.

Объект исследования: процесс развития алгоритмического мышления.

Предмет исследования: курс по Python-программированию во внеурочной деятельности восьмиклассников

Цель: выявить педагогические условия способствующие развитию алгоритмического мышления восьмиклассников во внеурочной деятельности посредством программирования на языке Python.

Для достижения поставленной цели были определены следующие **задачи:**

1. Выявить сущность понятия алгоритмического мышления.

2. Проанализировать научно-педагогическую и учебно-методическую литературу по программированию для основной школы, выявить образовательные результаты обучающихся.
3. Провести сравнительный анализ языков программирования.
4. Разработать курс по Python-программированию для восьмиклассников.
5. Разработать методические рекомендации по использованию курса.

Для достижения цели и решения поставленных задач были выбраны следующие методы исследования:

- Анализ – последовательное описание объекта на основе его умозрительного разложения на составные части.
- Синтез – характеристика объекта через описание связей и зависимостей его частей (обобщение, соединение).
- Сравнение – сопоставление для выявления различий или сходств явлений, предметов.

Объем и структура работы: выпускная квалификационная работа состоит из введения, двух глав, заключения, списка литературы, включающего 34 источника.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ФОРМИРОВАНИЯ АЛГОРИТМИЧЕСКОГО МЫШЛЕНИЯ

1.1. Понятие алгоритма и алгоритмического мышления

В современном мире алгоритмы используются повсюду, от поисковых систем до систем безопасности. Они также используются в медицине для анализа результатов тестирования и прогнозирования рисков для пациента. Алгоритмы являются частью жизни человечества, каждый человек ежедневно использует алгоритмы для разных целей, а кто-то создает новые алгоритмы для упрощения жизни людей. Без составления алгоритмов человеку будет сложнее пользоваться современными технологиями. Следовательно, составление человеком алгоритмов выполняемых действий становится частью его культуры мышления и поведения.

Алгоритм – это точное предписание, определяющее последовательность действий, обеспечивающую получение требуемого результата из исходных данных [4].

Не любая последовательность действия может считаться алгоритмом. Каждый алгоритм обладает следующими свойствами: дискретность, определенность, результативность, массовость.

- дискретность (прерывность, раздельность). Алгоритм должен представлять процесс решения задачи как последовательное выполнение простых(или ранее определенных) шагов. Каждое действие, предусмотренное алгоритмом, исполняется только после того, как закончилось исполнение предыдущего;
- определенность. Каждое правило алгоритма должно быть четким, однозначным и не оставлять место для произвола;
- результативность (конечность). Алгоритм должен приводить к решению задачи за конечное число шагов;
- массовость. Алгоритм решения задачи разрабатывается в общем виде, то есть он должен быть применим для некоторого класса задач, различающихся только исходными данными [26].

При составлении алгоритмов необходимо соблюдать ряд правил:

- Прежде всего, необходимо задать множество объектов, с которыми будет работать алгоритм;
- Для работы алгоритма требуется память;
- Алгоритм строится из отдельных шагов (действий, операций, команд);
- После каждого шага необходимо указывать, какой шаг выполняется следующим, либо давать команду остановки;
- Алгоритм должен завершать работу после конечного цикла шагов.

Всё вышперечисленное доказывает, что создание алгоритма требует определенных навыков, именно поэтому необходимо развивать алгоритмическое мышление. Для того, чтобы углубиться в понятие алгоритмического мышления, необходимо обратиться к сути понятия мышление.

Понятие мышление изучают многие науки, каждая со своей точки зрения. В частности, в философии рассматриваются проблемы соотношения бытия и мышления, в том числе возможности и пути познания мира с помощью мышления [25]. Кибернетика рассматривает мышление как информационный процесс, акцентируя внимание на общем и различном в работе ЭВМ и мыслительной деятельности человека. Психология изучает мышление как познавательную деятельность человека, разделяя ее на виды в зависимости от уровня обобщения и характера используемых средств.

Мышление — опосредованное отражение внешнего мира, которое опирается на впечатления от реальности и даёт возможность человеку в зависимости от усвоенных им знаний, умений и навыков правильно оперировать информацией, успешно строить свои планы и программы поведения [5].

Существуют различные подходы к определению алгоритмического стиля мышления. Многие отечественные и зарубежные ученые рассматривали данное понятие. Например, А. П. Ершовым, Г. А. Звенигородским, Ю. А. Первиным оно определяется, как умение планировать структуру действий, необходимых для достижения цели, при помощи фиксированного набора средств; умение строить информационные модели для описания объектов и систем; умение

организовывать поиск информации, необходимой для компьютерного решения поставленной задачи [8]. А.Г. Кушниренко и Г.В. Лебедев связывают алгоритмический стиль мышления с курсом информатики и понимают как метод и способ, которые необходимы для перехода от непосредственного управления к программному, от умения сделать к умению записать алгоритм [13].

Т.Н. Лебедева под алгоритмическим мышлением понимает познавательный процесс, характеризующийся наличием четкой, целесообразной (или рациональной) последовательности совершаемых мыслительных процессов с присущей детализацией и оптимизацией укрупненных блоков, осознанным закреплением процесса получения конечного результата, представленного в формализованном виде на языке исполнителя с принятыми семантическими и синтаксическими правилами [14].

Определение в широком смысле представлено в работе О.В. Чебуриной, которая определила, что алгоритмическое мышление – это совокупность мыслительных действий и приемов, нацеленных на решение задач, в результате которых создается алгоритм, являющийся продуктом человеческой деятельности [30].

Опираясь на перечисленные определения, можно сделать вывод, что алгоритмическое мышление, по сути направлено на создание эффективного алгоритма для конкретной ситуации, то есть позволяет принимать оптимальные решения в любой сфере человеческой деятельности.

Все описанные интерпретации определения алгоритмического мышления дополняют друг друга, в результате чего, можно выделить следующие особенности данного типа мышления:

- использование уже имеющихся знаний и навыков для построения эффективного алгоритма (плана) решения задачи;
- умение видеть проблему в целом, а ее решение поэтапно;
- достижение конечного результата, определяемого условием задачи или практической ситуации.

Обобщение всех подходов и особенностей к определению понятия «алгоритмическое мышление», выделение его главных черт, позволяют сформулировать следующую трактовку:

Алгоритмическое мышление – это стиль мышления, включающий систему мыслительных операций, приемов, мыслительных способов действий, которые направлены на поиск эффективного решения теоретических или практических задач, результатом которого является алгоритм, четкий план или инструкция.

Алгоритмическое мышление имеет свои общие и специфические свойства по сравнению с другими стилями мышления. В число общих свойств алгоритмического мышления входят целостность и результативность, помогающие увидеть поставленную проблему в целом виде и предполагают создание предварительного образа результата решения поставленной проблемы.

К специфическим свойствам относятся дискретность, абстрактность и осознанная закреплённость в языковых формах. Эти свойства представляют собой пошаговое исполнение алгоритма, дают возможность абстрагироваться от конкретных исходных данных, перейти к решению задачи в общем виде и представить алгоритм при помощи некоторого формализованного языка. Компонентами алгоритмического мышления являются умение формализовать задачу и разбивать её на отдельные составные логические блоки [6].

Данный тип мышления необходимо развивать в процессе обучения в общеобразовательных учреждениях. Самой эффективной, для развития алгоритмического мышления, является область программирования. Основой программирования является написание кода программы на любом языке программирования. Код программы является реализацией алгоритма, он состоит из набора действий, выстроенных в определенной последовательности и с определенными условиями, что в итоге позволяет добиться поставленной цели. Соответственно, если при написании программы учащихся научить разбивать задачу на элементарные действия, а также логически выстраивать их последовательность, то они овладеют теми универсальными учебными действиями, которые прописаны в ФГОС.

Стоит заметить, что в федеральном государственном образовательном стандарте основного общего образования четко сформулированы итоговые результаты, целевые ориентиры, которыми должен овладеть каждый учащийся в процессе обучения. Важными являются следующие навыки: умение самостоятельно определять цели своего обучения; осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач; определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией.

Развитие алгоритмического мышления напрямую связано с овладением вышеперечисленных навыков.

В работе Т.Н. Губиной представлены три основных уровня развития алгоритмического мышления: операционный(низкий), системный(средний), методологический(высокий) [15].

Операционный (низкий) уровень характеризуется тем, что обучающиеся имеют представление об алгоритме и свойствах алгоритма, а также знают основные операции, приемы и методы алгоритмизации. Они умеют производить разрозненные операции и записывать алгоритм, имеют способность представить решение задачи через базовые алгоритмические конструкции. Учащиеся владеют способами составления простых алгоритмов, имеют представление об исполнителе и системе его команд. Умеют решать стандартные задачи по образцу, выполнять алгоритм, в том числе пошагово.

Системный (средний) уровень характеризуется тем, что обучающиеся владеют способами сочетания операций алгоритмических конструкции, а также составления сложных алгоритмов, в том числе с использованием подалгоритмов. Они способны анализировать множество входных и выходных данных, знают способы решения базовых классов алгоритмических задач, владеют операциями классификации. Учащиеся уже выдвигают гипотезы по решению задач, умеют проводить анализ задачи, детализацию алгоритма для ее решения. Способны анализировать и учитывать различные сценарии выполнения алгоритма, умеют

видеть логические ошибки в алгоритме и планировать структуру действий для построения алгоритма.

Методологический (высокий) уровень отличается тем, что обучающиеся умеют использовать мыслительные схемы решения некоторых задач, преобразовывать известные схемы решения в нестандартных ситуациях, разрабатывать стратегию построения алгоритма, выдвигать и доказывать гипотезы опытным путем. Они прогнозируют результаты выполнения алгоритма, видят возможные проблемы, которые могут возникнуть при реализации алгоритма, способны оптимизировать алгоритм. Умеют оценивать сложность алгоритма и выбирать эффективный среди возможных, а также представляют алгоритм на формальном языке.

На основе этих уровней выделяются требования к развитию алгоритмического мышления:

Операционный уровень характеризуется тем, что ученик имеет представление об алгоритме.

Системный уровень характеризуется тем, что ученик имеет представления об алгоритме, его свойствах, составляет небольшие линейные алгоритмы или с простейшими ветвлениями и циклом; владеет конкретными операциями классификации, сериации; знает способы решения некоторого класса алгоритмических задач; имеет представление об исполнителе и системе команд исполнителя.

Методологический уровень характеризуется тем, что ученик имеет представления об алгоритме, знает его свойства, умеет составлять и записывать формальные и неформальные алгоритмы линейной структуры, с простейшими ветвлениями и циклами; владеет операциями классификации, сериации и взаимно однозначного соответствия; легко справляется с задачами алгоритмического характера; имеет представление об исполнителе, системе команд исполнителя.

Проанализировав вышперечисленное важно выделить основные принципы построения обучения, направленного на развитие алгоритмического мышления, которые сводятся к следующим: систематичность работы, направленной на

развитие алгоритмического мышления; системность, полнота и всесторонность рассмотрения отдельных действий, входящих в структуру алгоритмического мышления; возможность соотнесения полученных результатов с эталоном. Для выполнения всех этих условий целесообразно и необходимо использование ПК.

1.2. Особенности когнитивной деятельности обучающихся 8 классов

В общеобразовательных учреждениях изучение информатики начинается с 7 класса, то есть когда школьник достигает 13-14 лет. В психологии, согласно периодизации психического развития Д.Б. Эльконина, данный возраст называют младший подростковый возраст [7]. В 7 классе обучающиеся изучают правила работы с компьютером, текстовыми документами и т.д. Углубление в основы алгоритмизации и программирования начинается в 8 классе, так как обучающиеся уже отличаются своими особенностями когнитивной деятельности по сравнению с младшими школьниками. В последние годы пристальное внимание, в процессе обучения, выделяется процесс мотивации обучающихся, так как для подростков учебная деятельность не является приоритетом. Это связано с тем, что подросток занят самоопределением, самопознанием и саморазвитием. Для рациональной организации когнитивной деятельности в процессе обучения, необходимо рассмотреть возрастные особенности подростков в социально-когнитивном контексте.

Когнитивная деятельность в более широком значении – это деятельность, в результате которой человек приходит к определенному решению или знанию, т.е. мыслительная деятельность, приводящая к пониманию чего-либо. В когнитивной деятельности участвуют разные системы переработки информации, а потому образуемые структуры сознания не тождественны и зависят от того, по какому каналу пришла к человеку информация [21].

Уровень развития когнитивной деятельности состоит из определенных взаимосвязанных психических процессов: восприятие, память, мышление, внимание, воображение и эмоционально-волевые процессы. Отмечается, что чем

больше эти процессы соответствуют возрасту, тем выше эффективность когнитивной деятельности.

Когнитивная деятельность подростков существенно отличается от младшего школьного возраста, она обретает новые качества. Изменения характеризуются повышением возможностей памяти, регулированием внимания, активизацией и совершенствованием речи, развитием мышления. Подростки уже способны усваивать широкий объем информации, выдвигать и доказывать гипотезы, участвовать в рассуждениях, планировать и осуществлять исследовательскую деятельность.

А.А. Реан связывает когнитивную деятельность подростков с количеством и качеством познавательных процессов. «Развитие познавательных процессов и особенности интеллекта в подростковом возрасте имеют две стороны — количественную и качественную. Количественные изменения проявляются в том, что подросток решает интеллектуальные задачи значительно легче, быстрее и эффективнее, чем ребенок младшего школьного возраста. Качественные изменения, прежде всего, характеризуют сдвиги в структуре мыслительных процессов: важно не то, какие задачи решает человек, а каким образом он это делает» [2].

Восприятие — процесс, посредством которого формируются образы окружающего мира, отражение реальности в психике. Развитие восприятия младших подростков зависит от процесса обучения, в котором происходит усложнение учебного материала по мере взросления обучающихся.

Т.В. Алейникова отличает восприятие подростков от детей младшего школьного возраста. «В начале младшего школьного возраста (7-11 лет) восприятие имеет еще черты дошкольного возраста. У подростков (11-15 лет) продолжается интеллектуализация восприятия. Это связано с усложнением школьного материала. Геометрия и черчение способствуют развитию объёмного восприятия. На базе интеллектуализированного восприятия развиваются фантазирование и воображение, в том числе и творческое. Восприятие себя

(своего «Я») с разных сторон и в связи с этим развивающийся самоанализ способствуют формированию «Я-концепции» [29].

Память занимает важное место среди психических познавательных процессов, ее характеризуют как сквозной процесс, обеспечивающий преемственность психических процессов и объединяющих их в единое целое.

Память – психический процесс, который заключается в запечатлении, сохранении, воспроизведении и утрате информации.

Память по своему количеству и качеству также отличается от памяти младших школьников. Усложнение и увеличение объема изучаемого материала способствует развитию памяти. Подростки анализируют текст, выделяют в нем главные компоненты, то есть трансформируют текст лично под свое восприятие, что помогает им лучше понять материал и закрепить его. Для сравнения, когнитивная деятельность младших школьников направлена не на осознание и понимание, а на запоминание учебного материала.

Существенные изменения в младшем подростковом возрасте претерпевает и внимание. **Внимание** – это направленность сознания психической деятельности человека на определенные объекты с одновременным отвлечением от других. Внимание становится организованным и регулируемым процессом, у обучающихся развивается умение контролировать и управлять своим вниманием.

Особые изменения происходят и в **мыслительной деятельности** обучающихся. В подростковом возрасте, прежде всего продолжает развиваться теоретическое мышление. Используемыми в младшем школьном возрасте операциями становятся формально-логические. Подросток уже может абстрагироваться от конкретного, наглядного материала и рассуждать словесно.

Ж. Пиаже мыслительную деятельность, происходящую на фоне этих перемен, называет «формальные операции (11–15 лет)». «Учащиеся-подростки имеют индивидуальные различия и в характере мнемической деятельности: если ученики 5-го класса используют больше внешние приемы запоминания (ассоциации, смысловую группировку), ученики 8-го класса больше используют приемы запоминания и поиск для каждого материала специфических приемов.

Центральное место занимает анализ содержания материала, его своеобразия и внутренней логики» [7].

В отличие от младших школьников, обучающие младшего подросткового возраста при решении интеллектуальных задач проявляют способность оперировать гипотезами, производят поиск более эффективных путей решения задачи. Данные способности формируются и развиваются в процессе обучения.

Таким образом, одной из наиболее явных особенностей психических процессов младших подростков является то, что обучающиеся этого возраста осваивают на логическом уровне все мыслительные процессы. В общем и целом все умственные, отдельные мыслительные операции формируют единую структуру мышления.

Воображение – процесс преобразования представлений, отражающих реальную действительность, и создание на этой основе новых представлений. Воображение тесно связано с развитием мышления и памяти, поэтому чем выше уровень развития мышления и памяти, тем более сложные формы воображения могут проявляться у человека.

В подростковом возрасте ребенок чаще обращается к творчеству. Подростки начинают писать стихи, играть на музыкальных инструментах, занимаются изобразительным искусством и другими видами творчества. Занятие творчеством способствует полноценному развитию психических процессов.

Проведенный анализ позволяет утверждать, что когнитивная деятельность подростков обусловлена рядом особенностей. Мышление, память, воображение, восприятие – психические процессы, которые в ходе взросления ребенка претерпевают определенные изменения. Обучающиеся младшего подросткового возраста лучше концентрируются, рассуждают, ищут разные пути решения поставленной задачи, в сравнении с обучающимися, которые их младше. Вышеперечисленное подтверждает то, что обучающиеся младшего подросткового возраста способны анализировать и интерпретировать более сложную информацию.

1.3. Анализ языков программирования в контексте развития алгоритмического мышления

Прогресс, который совершило человечество в создании современных компьютерных технологий, был бы невозможен без языков программирования. Именно благодаря языку программирования пишутся коды программ, которые и управляют всей системой современных технологий. Языки программирования настолько разнообразны, что в мире их уже насчитывается более 700 [19]. Вместе с развитием языков программирования развиваются и современные технологии. Среди многообразия языков, необходимо определиться какой из них наиболее подходящий для обучения и развития алгоритмического мышления школьников, рассмотреть особенности разных языков программирования, определить их преимущества и недостатки.

Исследование IEEE (Институт инженеров электротехники и электроники), крупнейшей в мире профессиональной организации, занимающейся инженерными и прикладными науками, показало, что самыми популярными языками программирования за 2022 год стали Python, C, C++ и Java [16].

Популярность языка программирования определяется по разным характеристикам: от количества кода, написанного на нем, до объема вакансий и программ по изучению. Оценка популярности представляется рейтингами, которые опираются на статистику собранную за месяцы, кварталы, годы. Ни один из рейтингов не определяет язык самым удобным и лучшим, так как каждый язык программирования предназначен для решения определенных задач, однако с помощью рейтингов можно проследить динамику развития и востребованности языка на рынке.

Для сравнения языков необходимо выделить ряд характеристик, которым будет соответствовать язык программирования, так как в данной работе языки рассматриваются с точки зрения развития алгоритмического мышления у школьников.

Выбирая базовый язык программирования для массовой школы, нужно учесть следующие факторы:

- соответствие выбора достижению целей государственной политики РФ в области образования;
- методическая поддержка обучения и использования языка программирования в школе;
- сложность изучения языка;
- универсальность языка;
- рынок труда (востребованность и долгосрочные перспективы языка) [10].

Для анализа выбраны следующие языки программирования: *Pascal*, *Python*, *C*, *C++* и *Java*.

Далее рассмотрены достоинства и недостатки каждого из представленных языков программирования.

Pascal

Начать стоит с самого известного языка программирования *Pascal*, который был разработан швейцарским ученым Никлаусом Виртом в 1970 году на базе языка Алгол-68, созданного им для обучения студентов структурному программированию [27]. *Pascal* – это универсальный язык программирования, отличающийся строгой структурой и типизацией переменных, а также понятным синтаксисом.

Pascal является императивным языком, то есть программа написанная на этом языке выглядит как набор последовательных команд. Соблюдая следующие условия:

1. команды выполняются в строгой последовательности;
2. результат выполнения записывается в память;
3. каждая последующая команда использует данные, полученные при исполнении предыдущей [24].

Важной характеристикой данного языка программирования является строгая типизация данных, а это значит, что каждой конструкции языка (переменные, модули, функции) присвоены определенные свойства. Строгая

типизация позволяет избежать многих ошибок при написании кода, а также согласовать работу отдельных частей алгоритма.

Синтаксис является важной характеристикой любого языка программирования, на которую необходимо обращать внимание. *Pascal* отличается простым и интуитивным синтаксисом, однако его использование также требует знания определенных правил.

Для написания программ на языке *Pascal* используются следующие категории символов:

- прописные и строчные латинские буквы;
- арабские цифры;
- специальные символы – подчеркивания, квадратные и фигурные скобки, знаки препинания, знаки арифметических операций.

В *Pascal* предусмотрены комбинированные неделимые символы:

- := (операция присваивания);
- \geq и \leq (операция сравнения);
- (* и *) (начало и конец комментария).

Для присвоения имен константам и переменным используются любые комбинации символов из перечисленных ранее, но важно чтобы они отличались от названия служебных слов. Главное правило – начинаться они должны с буквы или символа подчеркивания. Регистр букв в *Pascal* значения не имеет.

Структура программы на языке *Pascal* состоит из трех составляющих:

1. заголовок – служебное слово *program*, имени программы и точки с запятой;
2. секции описания используемых данных — констант, типов, функций, переменных, процедур;
3. блока исполнения программы, в котором содержатся операторы, преобразующие данные.

Проанализировав вышеперечисленное можно вывести определенные преимущества и недостатки языка программирования *Pascal*. Преимуществами являются: простой и понятный синтаксис; компилируемость (перевод в машинный код); лояльность к ошибкам.

В свою очередь недостатками являются: ограниченное взаимодействие с компонентами написанными на других языках; ограниченные возможности работы с динамической памятью, массивами переменной длины, отсутствие отдельной компиляции.

Pascal – язык программирования, который был создан с целью обучения структурному программированию. Конечно, *Pascal* можно использовать как основной язык при обучении школьников, а также писать на нем программы, которые способствуют развитию алгоритмического мышления. В образовательной программе по информатике в общеобразовательных учреждениях многие годы обучающиеся изучали основы программирования именно на языке *Pascal*. Однако, современные технологии прогрессируют и язык *Pascal* теперь редко используется в прикладном программировании. Для того, чтобы в будущем обучающиеся смогли применять навыки, полученные в процессе обучения, необходимо идти в ногу со временем и выбирать актуальные формы и средства обучения. Язык программирования *Pascal* уже не является современным, поэтому в процессе обучения его использование не целесообразно.

Python

Далее рассмотрим язык программирования Python, который стал популярен и в общеобразовательных учреждениях, и в сфере прикладного программирования, который используют многие специалисты.

Python был разработан нидерландским программистом Гвидо ван Россумом в 1991 году, с тех пор язык активно развивался и обрел мировую популярность. Его используют сотни тысяч программистов по всему миру, в том числе сотрудники таких гигантов индустрии, как *Google* и *Yandex*, и число поклонников *Python* постоянно растет, особенно в последние годы можно наблюдать динамику неуклонного роста популярности *Python*.

Python – это объектно-ориентированный язык программирования высокого уровня, отличающиеся простотой синтаксиса и универсальностью. На сегодняшний день Python широко используется для создания приложений, веб-сайтов, в разработке прикладного программного обеспечения и обработке

больших данных, а также является одним из самых распространенных языков для обучения программированию.

Изучением и развитием языка программирования Python занимаются большие сообщества разработчиков, благодаря этому язык имеет обширный репозиторий, хранящий колоссальное количество подключаемых библиотек для решения широкого круга задач от создания простейших программ до разработок игр и приложений.

Синтаксис Python интуитивный и понятный, по сравнению с Pascal, написание программы стало еще проще.

Очевидно, видна разница и упрощение программы до одной строки, написанной на языке Python. Среди особенностей синтаксиса можно выделить необходимые отступы, которые являются обязательными при написании кода. Если в других языках программирования отступы являются необязательными, то при использовании Python это важная часть программы, иначе не избежать синтаксических ошибок. Отступы позволяют выстроить код грамотно, что в свою очередь упрощает его чтение.

Важным достоинством языка Python является его кроссплатформенность, то есть язык адаптирован под все существующие аппаратные и программные платформы, следовательно, открывает широкие возможности разработки программных продуктов различного назначения [24].

Python имеет динамическую типизацию данных, это взаимное автоматическое связывание переменной и типа в момент, когда ей присваивается определенное значение. Такой механизм ускоряет написание программы в различных ситуациях, например, при работе с переменными данными [31].

Бесспорно, язык программирования Python имеет много достоинств: интерпретируемость, понятный синтаксис, обширная сфера использования, востребованность, большое количество библиотек, многообразие учебных материалов, динамическая типизация, кроссплатформенность.

Однако недостатки у данного языка тоже есть: низкая скорость выполнения программ, так как в динамически типизированном языке код выполняется

построчно, это усложняет разработку ПО; программы написанные на *Python* требует большого объема памяти.

Python – понятный, простой в изучении, универсальный, востребованный и перспективный язык. Несмотря на наличие определенных недостатков, его преимущества существенно перевешивают все минусы. *Python* уже несколько лет изучается в общеобразовательных учреждениях, его внедряют в задания ЕГЭ и ГИА, а также олимпиады. Данный язык подходит для обучения школьников и является отличным средством для развития алгоритмического мышления. Писать программы на языке *Python* актуально, интересно и необходимо в процессе обучения.

C (Си)

Следующим не менее популярным языком программирования является C (Си). Этот язык программирования был создан в начале 70-х годов, когда Кен Томпсон и Дэннис Ритчи из *Bell Labs* разрабатывали операционную систему *UNDC*. Сначала они создали часть компилятора C, затем использовали ее для компиляции остальной части компилятора C и, наконец, применили полученный в результате компилятор для компиляции *UNIX*. Распространение исходного кода сделало операционную систему *UNIX* уникальной; программист мог изменить операционную систему, а исходный код мог быть перенесен с одной аппаратной платформы на другую [11].

C (Си) – компилируемый, процедурный, статически типизированный язык программирования общего назначения, который был создан программистами для разработки операционной системы. Так как многие языки программирования были созданы для обучения программированию, язык C был создан для прикладного программирования, следовательно он имеет ряд своих особенностей: возможность применять язык высокого и низкого уровня; возможность разрабатывать высокоэффективные программы не требующие большого объема памяти [32].

Данный язык программирования является мобильным, так как компиляторы с данного языка реализованы примерно на 40 типах вычислительных систем.

Язык *C* мощный и гибкий, поэтому применяется для решения физических и технических проблем и даже для производства мультипликационных фильмов.

Конечно как и все языки программирования *C* имеет свои недостатки: отсутствие средств работы с мультимедиа, с сетью и поддержки многозадачности; ослабленный контроль за типами данных; сложный для пониманий синтаксис.

Язык программирования *C*, придуманный еще в 70-х годах, до сих пор остается одним из самых популярных языков в мире. Он имеет широкий круг достоинств, но и недостатками не обделен. Язык *C* не совсем подходит для изучения в общеобразовательных учреждениях, так как имеет сложный для понимания синтаксис, требует серьезной и долгой подготовки перед процессом программирования. Безусловно, программирование развивает алгоритмическое мышление, но очень важно какими способами и средствами, язык программирования *C* является непростым, поэтому он и не изучается в общеобразовательных учреждениях, при использовании языка *C* процесс изучения алгоритмизации становится намного сложнее.

C++ (Си плюс плюс)

Далее будет рассмотрен язык похожий названием на предыдущий – *C++*. Язык был разработан на основе языка *C* (Си), который был создан в 1969-1973 годах в компании *Bell Labs* программистом Деннисом Ритчи. В 1980-х, другой сотрудник компании, Бьерн Страуструп, разработал *C++*. Сначала язык создавался как дополнение основному Си, в котором было возможно объектно-ориентированное программирование. Страуструп назвал свое творение «Си с классами». Затем *C++* становился все более востребованным и постепенно стал отдельным языком программирования. Теперь два языка развиваются сами по себе и имеют ряд отличий друг от друга:

- *C* (Си) нацелен на структурное и процедурное программирование, тогда как *C++* на объектно-ориентированное;
- *C* (Си) имеет переменные и структуры, а *C++* классы и объекты;

- В C (Си) нет функции перезапуска оператора, в C++ такая возможность есть [18].

Язык C++ отличается высокой скоростью и производительностью, это значит, что приложения написанные с его использованием, могут запускаться даже на устройствах с невысокой характеристикой.

Преимуществами языка C++ являются: кроссплатформенность; возможность работы в низком уровне с памятью, адресами, портами; создание обобщенных алгоритмов для разных типов данных.

Недостатки C++ схожи с недостатками языка C (Си), ими выступают: плохая поддержка модульности; недостаток информации о типах данных во время компиляции; сложный синтаксис.

Данный язык программирования также, как и предыдущий, не подходит для изучения программирования и развития алгоритмического мышления у школьников в общеобразовательных учреждениях, требует определенных навыков и знаний, которые обучающиеся только начинают приобретать.

Java

Java – консервативный и надежный язык для разработки под любые платформы. Применение языка в следующих областях: *desktop*-приложения, приложения для *Android*, серверные программы, сложное ПО (промышленные и банковские системы) [20].

В 1995 году широкое распространение получил новый объектно-ориентированный язык программирования *Java*, ориентированный на сети компьютеров и в первую очередь на *Internet*. Синтаксис данного языка похож на синтаксис C++, однако, несмотря на это, в остальном языки отличаются. *Java* является интерпретируемым на конечной стадии языком: для него определены внутреннее представление, то есть прекомпиляция в байткод (*bytecode*), и постинтерпретатор этого представления на целевой машине, реализованные уже сегодня на большинстве платформ [1].

У языка *Java* строгая типизация, что не позволяет смешивать в выражениях разные типы. Так как язык является объектно-ориентированным, в нем

существуют классы – типы данных и объекты – представители этих классов. Разработчик сам их создает, дает названия и присваивает им свойства и операции, которые с ними можно выполнять.

Преимуществами языка программирования *Java* являются – работает на всех операционных системах; возможность писать программное обеспечение любой сложности, от маленьких утилит до огромных программных комплексов; единственный язык для создания приложения под *Android* [9].

Недостатками языка является – большая нагрузка на оперативную память оборудования; время решения одних и тех же задач ниже, по сравнению с языком *C*; отсутствие поддержки низкоуровневого программирования.

Java непростой язык для первоначального освоения, это зрелый язык с массой дополнительных инструментов, также его синтаксис сложен для чтения и понимания. Этот язык будет трудным для начинающего в целях изучения основ алгоритмизации, поэтому в общеобразовательных учреждениях *Java* не распространен.

Проанализировав вышеперечисленное можно сделать вывод, что каждый язык программирования имеет свои особенности, преимущества и недостатки, каждый из них предназначен для решения самых различных задач. Какие-то языки устарели, какие-то удерживают в мире лидерские позиции уже много лет. В процессе обучения программированию, язык играет важную роль, так как с помощью языка разработчик познает основы алгоритмизации. Первый язык программирования должен быть интуитивно понятным, доступным и конечно же современным.

Алгоритмическое мышление развивается посредством создания правильно выстроенного алгоритма, последовательных шагов. Если для начального этапа выбирать язык со сложным синтаксисом, то от разработчика требуется серьезная подготовка, чтобы не отвлекаться на грамматические ошибки функций, расстановки знаков препинания, что влечет за собой уход от основной задачи – выстроить грамотный алгоритм, который приведет в нужному результату. Далее

приведен пример простейшей программы “Hello World!”, написанной на рассмотренных языках.

Таблица 1

Сравнение синтаксиса языков программирования

Pascal	<pre> program hello; begin writeln('Hello, World!'); end. </pre>
Python	<pre> print "Hello World" </pre>
C	<pre> /* Программа приветствия */ #include <stdio.h> int main() { printf ("Hello World!\n"); return 0; } </pre>
C++	<pre> #include <iostream> int main() { std::cout << "Hello, World!" << std::endl; return 0; } </pre>
Java	<pre> public class Hello { public static void main(String[] args) { System.out.println("Hello, World!"); } } </pre>

В таблице четко видно, какие языки обладают простым и понятным синтаксисом, это *Pascal* и *Python*. Как было упомянуто ранее, язык программирования *Pascal* устарел, так как ему на смену пришли более универсальные и кроссплатформенные языки. Язык программирования *Python* с точки зрения развития алгоритмического мышления отвечает всем необходимым требованиям, чтобы внедрить его в процесс обучения общеобразовательных учреждений. Благодаря простому синтаксису, многозадачности, репозиторию с подключаемыми библиотеками, обучающимся будет проще и интереснее изучить основы алгоритмизации на языке *Python*.

Выводы по главе 1

На основе теоретического анализа сущности понятия алгоритма и алгоритмического мышления, особенностей когнитивной деятельности обучающихся 8 класса и анализа языков программирования на предмет развития алгоритмического мышления было выявлено следующее: алгоритмический тип мышления необходимо развивать в процессе обучения в школе. Разработка логически выстроенного алгоритма требует определенных навыков и умений от обучающихся. Особенности когнитивной деятельности, которые характерны для младших подростков, а именно – память, мышление, воображение, восприятие, внимание, в полной мере определяют обучающегося как индивида, способного анализировать и интерпретировать сложную информацию, искать наиболее подходящие пути решения задачи, они лучше концентрируются и рассуждают. Эти особенности младших подростков дают понимание того, что именно обучающиеся данного возраста готовы к изучению области программирования на более глубоком уровне.

Программирование – это именно та область, в которой полноценно возможно развитие алгоритмического мышления, так как основой данной области является создание пошагового алгоритма. Обучающиеся учатся писать код программы, разбивая его на элементарные действия для решения поставленной задачи. Для написания кода необходимо знать язык программирования, поэтому был проведен анализ языков программирования, в ходе которого был определен язык Python как самый подходящий язык для обучения и развития алгоритмического мышления школьников.

Для создания собственного элективного курса по программированию на Python, необходимо проанализировать уже существующие онлайн-курсы по программированию для выявления их преимуществ и недостатков. В следующей главе будет представлен анализ онлайн-курсов по программированию на Python, а также программа собственного элективного курса – структура, содержание и методические рекомендации по использованию курса.

ГЛАВА 2. РАЗРАБОТКА КУРСА ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON

2.1. Анализ онлайн-курсов на языке программирования Python

С развитием интернета и информационных технологий, контента в социальных сетях становится с каждым днем только больше. В 2020 году из-за пандемии процесс обучения перешел в онлайн-формат, что в свою очередь помогло пережить вспышку вируса и вернуться к прежней жизни. Пандемия закончилась, а тенденция обучаться в домашних условиях перед компьютером осталась. На сегодняшний день в интернете возможно найти онлайн-обучение на многие профессии, в том числе и выучиться на программиста. Разработчики подготовили ряд курсов и для школьников, которые хотят изучить алгоритмизацию. Онлайн-курсов программирования на языке Python настолько много, что требуется немало времени для выбора подходящего курса.

При создании собственного курса программирования для внеурочных занятий в общеобразовательных учреждениях, необходимо проанализировать известные онлайн-курсы программирования на языке Python для школьников, выявить их преимущества и недостатки.

Рейтинг лучших онлайн-курсов программирования на языке *Python*:

1. «Minecraft: программирование на Python» от *GeekBrains*;
2. «Углубленный курс по Python» от *GeekBrains*;
3. «Курсы Python для детей» от *Skysmart*;
4. «Программирование на Python для детей и подростков» от *Rebotica*;
5. «Программирование на Python для детей» от *PIXEL* [33].

«Minecraft: программирование на Python» от *GeekBrains* [17].

Длительность курса: 8 месяцев (32 онлайн-занятия)

Возраст: 10-12 лет

Цена курса: 44 000 рублей (есть рассрочка)

Программа курса состоит из 5 модулей:

1. Введение в *Python* – особенности языка, условные операторы, типы данных, разработка первых программ.
2. Циклы в *Python* – виды, списки и коллекции.
3. Функции в *Python* – виды функций, аргументы, разработка собственных функций.
4. Основы объектно-ориентированного программирования (ООП) – разработка программ с использованием ООП.
5. Секретные возможности *Python* – работа с файлами и потоками, создание собственной мини-игры.

В рамках курса обучающиеся выполняют домашние задания, также предусмотрена поддержка куратора в чате. По окончании курса каждому обучающемуся выдается электронный сертификат.

Преимуществом курса является развлекательный процесс обучения, так как большую часть времени обучающиеся создают мини-игры.

Недостатками являются платное прохождение курса; недостаточная обратная связь от куратора; программа курса, которая направлена не на понимание процесса алгоритмизации.

«Углублённый курс по Python» от *GeekBrains* [28].

Длительность курса: 9 месяцев

Возраст: 11-14 лет

Цена курса: 45 000 рублей (5 000/месяц)

Программа курса состоит из 3 модулей:

1. 2D-игры на *Python* – изучение синтаксиса *Python*, решение математических задач с помощью кода, работа с библиотеками *Turtle* и *Pygame*, разработка игр в команде.
2. Сайты на *Python* – настройка внешнего облика и разработка внутренней логики сайта, разработка сайта в команде.
3. Чат-боты на *Python*.

Занятия в рамках курса проходят 1 раз в неделю в прямом эфире (*Zoom*), обучающиеся могут задавать вопросы преподавателю. Предусмотрена возможность посмотреть трансляцию в записи, если у ученика не было возможности посетить занятие во время. По окончании курса у каждый обучающийся будет иметь сертификат и портфолио со следующими работами: 6 2D-игр, web-сайт и 3 чат-бота.

Программа данного курса намного интереснее и фундаментальнее предыдущего, она рассчитана на погружение в процессы программирования. Также преимуществом является выполнение разного типа заданий, что в свою очередь позволяет обучающемуся научиться работать с разными функциями и типами данных.

Недостатками является высокая цена курса, а также недостаточная обратная связь от преподавателей.

«Курсы Python для детей» от *Skysmart* [12].

Длительность курса: от 4 до 128 занятий

Возраст: 10-18 лет

Цена курса: от 849 до 1599 за одно занятие

Программа курса создается лично преподавателем, программа зависит от количества занятий, которые выбирает ученик. Урок включает три шага обучения: теория, ответы на вопросы, самостоятельная работа. Урок длится 50 минут.

Преимуществом данного курса является индивидуальный подход к ученикам, у каждого индивидуальная образовательная траектория.

Недостатком является вариативность выбора количества занятий, от 4 до 128 занятий, данный интервал достаточно большой. Также недостатком является цена занятий, при выборе 128 занятий оплата курса составит 108 672 рубля.

«Программирование на Python для детей и подростков» от *Rebotica* [23].

Длительность курса: 40 уроков

Возраст: 8-16 лет

Цена курса: 78 480 рублей

Программа курса состоит из 5 модулей:

1. Базовое программирование;
2. Продвинутое программирование;
3. Модуль Tkinter;;
4. Разработка игр «Прыг-скок», «Человек спешит к выходу»;
5. Разработка чат-бота в Telegram.

Преимуществом курса является индивидуальный подход к ученикам, для каждого подбирается индивидуальная программа обучения и преподаватель.

Недостатком курса является высокая стоимость занятий.

«Программирование на Python для детей» от *PIXEL* [22].

Длительность курса: 12 занятий

Возраст: 10-14 лет

Цена курса: от 9500 до 16500 за курс

Программа курса состоит из 5 модулей:

1. Знакомство с языком программирования Python;
2. Условные конструкции и математические функции;
3. Функции, циклы while и for;
4. Списки, словари, множества и модули;
5. Создание собственных модулей под разные задачи.

Онлайн-занятия проходят в индивидуальном или групповом формате. Длительность одного занятия составляет 90 минут. Процесс обучения геймифицирован – за каждый успешный результат ученики получают призы.

Данный курс имеет ряд преимуществ – индивидуальный подход, вознаграждение за успешный результат, вариативность заданий.

В ходе анализа курсов было выявлено, что каждый из них имеет многофункциональную платформу – сайт, где каждый обучающийся имеет возможность актуализировать теоретические знания и просмотреть практические

задания. Большинство курсов предусмотрено, в конце обучения, выполнение обучающимися итогового проекта, что способствует проследить прогресс учеников. Рассмотренные курсы имеют свою программу обучения, преподавателей и систему оценивания. Необходимо отметить, что каждый из курсов проходит на платной основе, и не каждый родитель будет готов отдать немалую сумму именно за онлайн-обучение.

Из всех ранее проанализированных курсов, стоит выделить «Программирование на Python для детей и подростков» от *Rebotica*. Так как в рамках курса подбор материала подбирается конкретно под ученика, а также присутствует, немало важное в процессе обучения, вознаграждение за успешный результат, что в свою очередь, мотивирует обучающихся выполнять качественно и осмысленно каждое задание.

2.2. Разработка курса по программированию на языке Python

Пояснительная записка

Рабочая программа курса внеурочной деятельности «Python-программирование» для обучающихся 8 класса составлена на основе требований Федерального государственного образовательного стандарта основного общего образования к результатам освоения основной программы основного общего образования.

Рабочая программа курса дает представление о цели, задачах, общей стратегии обучения, воспитания и развития обучающихся средствами курса внеурочной деятельности, устанавливает содержание курса, предусматривает структурирование курса по разделам и темам; предлагает распределение учебных часов по разделам и темам курса и последовательность их изучения с учетом логики учебного процесса, возрастных особенностей обучающихся, включает описание форм организации занятий и учебно-методического обеспечения образовательного процесса.

Общая характеристика курса

Целесообразность изучения алгоритмизации, помимо необходимости в условиях информатизации школьного образования широкого использования знаний и умений по информатике в других учебных предметах, обусловлена также следующими факторами. Во-первых, положительным опытом обучения алгоритмизации обучающихся, во-вторых, существенной ролью изучения информатики в развитии мышления, формировании научного мировоззрения школьников, в-третьих, недостаточным количеством учебных часов по программе на изучение данных тем.

Цели курса:

- создание условий для достижения обучающимися результатов развития в личностном, предметном, метапредметном направлениях;
- развитие алгоритмического мышления;
- обучение основам веб-программирования с использованием системы программирования на языке *Python*.

Задачи курса:

Обучающие:

- познакомить обучающихся с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных;
- научить учащихся составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций;
- научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач;
- научить учащихся разрабатывать алгоритмы и реализовывать их на языке *Python*;
- научить обучающихся осуществлять отладку и тестирование программы.

Развивающие:

- формировать новый тип мышления – алгоритмический, который направлен на выбор оптимальных решений задачи;
- предоставление возможности узнать новое в области программирования;

- формирование представления о роли компьютерного программирования в развитии общества.

Воспитательные:

- повышение общекультурного уровня обучающихся;
- выделение и раскрытие роли информационных технологий и компьютеров в современном обществе;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и в профессиональной деятельности;
- формирование эмоционально-ценностного отношения к миру и к себе;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

Формы занятий

Наиболее подходящей формой организации занятий по программированию являются комбинированные занятия. Занятия состоят из теоретической и практической частей, причем большее количество времени занимает практическая часть. В теоретической части рассматриваются основные понятия языка программирования Python. В практической части предлагаются практические работы, направленные на отработку основных алгоритмических конструкций, на развитие алгоритмического мышления, на разработку.

При проведении занятий используются следующие формы работы:

- демонстрационная (обучающиеся следят за действиями педагога);
- фронтальная (весь класс работает вместе с педагогом);
- самостоятельная (обучающиеся выполняют индивидуальные задания);
- групповая (учащиеся совместно работают над проектом).

Виды контроля

Начальный (входной) контроль проводится с целью определения уровня развития обучающихся. Текущий контроль с целью определения степени усвоения обучающимися учебного материала. Промежуточный контроль с целью

определения результатов обучения. Итоговый контроль с целью определения изменения уровня развития учеников, их способностей.

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения практических работ, зачетных практических работ по основным темам элективного курса.

Итоговый контроль реализуется в форме защиты итоговых проектов. Защита предполагает публичное выступление перед классом, обучающиеся демонстрируют работу программы, объясняют принцип ее работы, аргументируют значимость своего проекта, а также отвечают на вопросы одноклассников.

Место курса в образовательном процессе

Программа курса предназначена для организации внеурочной деятельности, рассчитана на 46 часов, по 2 ч. в неделю в 8 классах.

Материально-техническая база курса

- персональный компьютер для учителя;
- персональные компьютеры для обучающихся;
- проектор;
- источник бесперебойного питания.

Планируемые результаты освоения курса

В ходе прохождения курса обучающиеся овладевают следующими знаниями и умениями:

Предметные:

- знают основные алгоритмические конструкции и правила их записи, знакомы с основными способами организации данных;
- владеют приемами написания программ для решения задач с использованием основных конструкций программирования и отладки программ на Python;
- приобретение опыта использования информационных технологий в индивидуальной и групповой учебной и познавательной, в том числе проектной деятельности;

- владение универсальным языком программирования Python, представлениями о базовых типах данных и структурах данных;
- формирование умения работать с библиотеками программ.

Личностные:

- сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
- готовность оценивать свои поступки и поступки других с позиции нравственных и правовых норм;
- формирование навыков соблюдения правил безопасности, в том числе в Интернет-среде.

Метапредметные:

- самостоятельность в планировании и осуществлении учебной деятельности и организации учебного сотрудничества с педагогами и сверстниками;
- владение навыками учебно-исследовательской, проектной и социальной деятельности, способность и готовность к самостоятельному поиску методов решения практических задач.

Учебно-тематический план элективного курса

№	Тема	Количество часов
1	Знакомство с языком Python	6 ч.
	Тема 1. История языка Python. Знакомство с интерфейсом программы Python. Основы языка. <i>Практическая работа №1.</i> Язык программирования Python. Установка. Текстовый редактор языка.	2
	Тема 2. Структура программы. <i>Практическая работа № 2.</i> Изучение структуры программы. Разделы описания.	2
	Тема 3. Основные математические функции. <i>Практическая работа №3.</i> Основные математические функции. Первая программа.	2
2	Типы данных. Ввод и вывод данных. Линейный алгоритм.	6 ч.
	Тема 4. Типы данных. <i>Практическая работа №4.</i> Составление линейных алгоритмов.	2

	Комментарии в программе.	
	Тема 5. Форматы вывода. <i>Практическая работа №5.</i> Составление линейных алгоритмов с использованием арифметических операций.	2
	Тема 6. Основные функции. <i>Практическая работа №6.</i> Составление линейных алгоритмов с использованием основных функций.	2
3	Условный оператор	8 ч.
	Тема 7. Определение и структура условного оператора. <i>Практическая работа №7.</i> Условный оператор. Структура условного оператора.	2
	Тема 8. Простые условия в алгоритме. <i>Практическая работа №8.</i> Условный оператор. Простые условия.	2
	Тема 9. Составные условия в алгоритме. <i>Практическая работа №9.</i> Условный оператор. Составные условия.	2
	Тема 10. Операторные скобки. <i>Практическая работа №10.</i> Написание алгоритма с использованием операторных скобок. Оператор выбора.	2
4	Алгоритмы с повторениями. Циклы	6 ч.
	Тема 11. Цикл с параметром For. <i>Практическая работа №11.</i> Решение задач с использованием цикла for.	2
	Тема 12. Цикл с предусловием и постусловием. <i>Практическая работа №12.</i> Решение задач с использованием цикла while.	2
	Тема 13. Вложенные циклы. <i>Практическая работа №13.</i> Решение задач с применением вложенных циклов.	2
5	Функции	4 ч.
	Тема 14. Определение функции. Оператор def и return <i>Практическая работа №14.</i> Решение задач с операторами def и return	2
	Тема 15. Параметры и аргументы функции. <i>Практическая работа №15.</i> Программа для обращения строки.	2
	Создание собственного проекта	
6	Строки	4 ч.
	Тема 16. Литеры строк. Кодировка строк. <i>Практическая работа №16.</i> Программа с кодировкой строк.	2

	Тема 17. Перенос и сравнение строк. <i>Практическая работа №17.</i> Отработка навыков переноса и сравнения строк.	2
7	Сложные типы данных	6 ч.
	Определение списков, словарей, кортежей.	2
	Тема 18. Определение списков, кортежей. <i>Практическая работа №18.</i> Операции со списками и кортежами.	2
	<i>Практическая работа №19.</i> Создание словарей.	2
	Итоговый проект: создание чат-бота	6 ч.
	Итого	46 ч.

Методические рекомендации о реализации курса

Курс «Python-программирование» для 8 классов направлен на формирование базовых понятий структурного программирования, развитие алгоритмического мышления обучающихся, использование информационных систем для решения практических задач, владение базовыми нормами информационной этики и права, основами информационной безопасности.

Курс предполагает работу с кодом и сервисами облачного хранения. Практическое освоение языка программирования Python, перспектива увидеть результаты своего труда в сети обеспечивает возможность создание коллективных проектов в области информационных технологий.

При разработке курса учитывались возрастные особенности восприятия, мышления, внимания, познания, памяти обучающихся.

Курс «Python-программирование» для 8 классов носит фундаментальный характер и опирается на базовые научные представления предметной области, такие как информация, информационные процессы, информационные модели. Основными модулями программы курса являются модули алгоритмизации и программирования на языке Python. Содержание учебных модулей по программированию охватывают как алгоритмическое направление, так и вопросы

практического применения полученных знаний при решении задач из различных областей знаний.

Вывод по главе 2

В 21 веке широко пропагандируется возможность обучаться в формате онлайн. Разработчики и маркетологи предлагают, в сети Интернет, широкий круг всевозможных онлайн-курсов по многим профессиям. В рамках исследования были рассмотрены курсы по программированию для школьников. В рамках каждого онлайн-курса разрабатывается программа обучения, выстраивается план проведения занятий, что является важной частью любого процесса обучения. Для проведения занятий подбираются педагоги с образованием и опытом в сфере программирования.

Самым главным недостатком всех рассмотренных онлайн-курсов является высокая цена за занятия. Создатели онлайн-курсов стараются создавать тарифы с разным ценовым сегментом, дают родителям возможность взять рассрочку. Однако, из-за денежной составляющей, ухудшается процесс обучения или вовсе родители не готовы отдавать большие деньги именно за онлайн-обучение.

Так как программирование уже давно изучается в школах на уроках информатики, решением вышеописанной проблемы является создание элективного курса по программированию.

В рамках данной работы был разработан курс по Python-программированию для обучающихся 8 классов, целями которого является создание условий для достижения обучающимися результатов развития в личностном, предметном, метапредметном направлениях; развитие алгоритмического мышления; обучение основам веб-программирования с использованием системы программирования на языке Python.

Курс соответствует тем требованиям и образовательным результатам, которые прописаны в федеральном государственном образовательном стандарте. Курс рассчитан на 46 часов обучения, большую часть которых занимают практические занятия. Для проведения курса был разработан сайт, который

содержит теоретический материал, а также примеры решения задач, которые обучающиеся будут решать. В рамках курса изучаются 7 основных разделов по Python-программированию: знакомство с языком Python, типы данных, условный оператор, алгоритмы с повторениями, функции, строки, сложные типы данных.

Практические занятия включают решение задач посредством языка программирования Python: математические операции с числами, создание игры – угадай число, поиск элементов в длинных списках, выполнение операции со словами (перевернуть слово, изменить кегль букв и тд.). Итоговой работой для обучающихся является разработка чат-бота для модернизации процесса обучения.

ЗАКЛЮЧЕНИЕ

Целью выполнения исследовательской работы была выявление педагогических условий способствующих развитию алгоритмического мышления восьмиклассников во внеурочной деятельности посредством программирования на языке Python.

Для достижения поставленной цели, в ходе написания выпускной квалификационной работы, была выявлена сущность понятия алгоритмического мышления, так как именно данный тип мышления позволяет развивать область программирования. В рамках исследования был проведен анализ научно-педагогической и учебно-методической литературы по программированию для основной школы. Результатом этого анализа были определены цели и задачи внеурочных занятий по программированию, на основе которых формировалось содержание курса и образовательные результаты обучающихся.

Изучение преимуществ и недостатков популярных языков программирования позволило определить самый подходящий язык программирования для развития алгоритмического мышления школьников, итогом анализа таким языком был выбран язык программирования Python.

Также был произведен, необходимый в рамках исследования, анализ уже существующих курсов по программированию на языке Python. Выявление преимуществ и недостатков данных курсов помогло более грамотно и качественно разработать программу курса по программированию на языке Python.

На основе методического исследования произведена разработка элективного курса для восьмиклассников по программированию на языке Python. Результатом выпускной квалификационной работы является элективный курс «Python-программирование» для 8 классов. Для данного курса был разработан сайт с теоретическим и практическим материалом.

Исходя из вышесказанного, цель исследовательской работы достигнута в полном объеме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Авдиль, С. Л. Анализ объектно-ориентированных языков программирования / С. Л. Авдиль, Э. А. Бекирова // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. – 2018. – № 4(22). – С. 107-113. – EDN VTSIGB.
2. Алейникова Т.В. Возрастная психофизиология: учеб. пособие для студентов высших учебных заведений / Т.В. Алейникова. — Ростов н/Д.: УНИИ, 2002. — 150 с.
3. Бобров, А. Н. Проблемы выбора языка программирования в школьном курсе информатики / А. Н. Бобров. — Текст : непосредственный // Молодой ученый. — 2015. — № 24 (104). — С. 61-64. — URL: <https://moluch.ru/archive/104/24471/> (дата обращения: 05.05.2023).
4. Богоутдинов Д. Г. Алгоритмы: понятие, свойства, виды. Словесная форма представления алгоритмов //МИФ-2. – 2005. – №. 3.
5. Большой энциклопедический словарь / Под ред. Б. М. Бим-Бада и др. — М.: Большая Российская энциклопедия, 2003. — 528 с.
6. Бочкин, А.И. Методика преподавания информатики [Текст]. – Минск: Высшая школа, 1998. – 431 с.
7. Детская психология: учеб. пособие для студ. высш. учеб. заведений / Д. Б. Эльконин; ред.-сост. Б. Д. Эльконин. — 4-е изд., стер. — М.: Издательский центр «Академия», 2007. — 384 с.
8. Ершов, А. П. Школьная информатика: концепции, состояние, перспективы /А. П. Ершов, Г. А. Звенигородский, Ю. А. Первин. – Новосибирск, 1979. – 51 с. – (Препринт / АН СССР. Сиб. отд-ние ; № 152).
9. История языка Си // Хабр URL: <https://habr.com/ru/articles/114588/> (дата обращения: 23.05.2023).
10. Каракозов С.Д., Маняхина В.Г. Python как базовый язык обучения программированию в школе. Информатика в школе. 2020;(1):26-30. <https://doi.org/10.32517/2221-1993-2020-19-1-26-30>

11. Кому и для чего нужен Python? // skillfactory media URL: <https://blog.skillfactory.ru/komu-i-dlya-chego-nuzhen-python/> (дата обращения: 20.05.2023).
12. Курсы Python для детей // skysmart URL: <https://skysmart.ru/programmirovanie-dlya-detej/python> (дата обращения: 03.06.2023).
13. Кушниренко, А. Г. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать / А. Г. Кушниренко, Г. В. Лебедев // Информатика. – 1999. – № 1. – С. 2–15.
14. Лебедева Т.Н. Формирование алгоритмического мышления школьников в процессе обучения рекурсивным алгоритмам в профильных классах средней общеобразовательной школы: дис.канд. пед. наук. Екатеринбург, 2005.
15. Лучко, Л.Г. Решение задач школьного курса информатики [Текст] : учебно-методическое пособие. – Омск: ОмГПУ, 2001. – 80 с.
16. Лучшие языки программирования 2022 // IEEE Spectrum URL: <https://spectrum.ieee.org/top-programming-languages-2022> (дата обращения: 12.02.2023).
17. Майнкрафт: программирование на Python // GeekBrains URL: <https://gb.ru/courses/geek-school/minecraft-python> (дата обращения: 02.06.2023).
18. Маслова, А. С. Обзор и сравнение популярных языков программирования / А. С. Маслова // Современные материалы, техника и технология : Сборник научных статей 9-й Международной научно-практической конференции. В 2-х томах, Курск, 28 декабря 2019 года / Ответственный редактор А.А. Горохов. Том 2. – Курск: Юго-Западный государственный университет, 2019. – С. 41-45. – EDN FSRSNS.
19. Новые языки программирования, которые стоит изучать в 2022 году // Bubble URL: <https://bbbl.dev/articles/languages-2022> (дата обращения: 15.04.2023).
20. Особенности объектно-ориентированных языков программирования [Электронный ресурс] // Kopilkaurokov.ru: сайт для учителей. – Режим доступа: <https://kopilkaurokov.ru/informatika/prochee/page=3?class=&count=20>

21. От рождения до смерти; под ред. А. А. Реана. — М.: Алма-Пресс, 2002. — 652 с.
22. Программирование на Python для детей // PIXEL URL: https://clubpixel.ru/python/?utm_term=4da6e2b1f2cc5980716519f037f5c436&utm_source=5 (дата обращения: 03.06.2023).
23. Программирование на Python для детей и подростков // rebotika URL: <https://goo.su/dM269zx> (дата обращения: 03.06.2023).
24. Солсо Р. Когнитивная психология / Р. Солсо. — СПб.: Питер, 2002. — 590 с.
25. Степанова, Т.А. Методические условия развития алгоритмического мышления школьников на уроках информатики /Т.А. Степанова // 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.202-205
26. Теория алгоритмов: Учебное пособие / М.С. Мирзоев, В.Л. Матросов — М.: Прометей, 2019 — 200 с.
27. Тюгашев А. А. Основы программирования //Часть I.—СПб.: Уни. – 2016.
28. Углубленный курс по Python // GeekBrains URL: <https://gb.ru/courses/geek-school/python-pro> (дата обращения: 02.06.2023).
29. Флейфелл Х.Д. Генетическая психология Жана Пиаже / Х.Д. Флейфелл. — М.: Просвещение, 1967. — 622 с.
30. Чебурина О.В. Формирование алгоритмического мышления в обучении программированию игр // Наука и перспективы. 2017. №2. URL: <https://cyberleninka.ru/article/n/formirovanie-algoritmicheskogo-myshleniya-v-obuchanii-programmirovaniyu-igr> (дата обращения: 28.04.2023).
31. Шило А. В., Железко Б. А. История развития языков программирования. – 2002.

32. Язык программирования C++ // сравни URL:
<https://www.sravni.ru/kursy/info/yazyk-programmirovaniya-c/> (дата обращения:
15.04.2023).

33. 10 лучших Python курсов для школьников // pythonchik.ru URL:
<https://pythonchik.ru/kursy/python-kursy-dlya-shkolnikov> (дата обращения:
02.06.2023).