

Министерство образования и науки Российской Федерации  
ФГБОУ ВО «Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологии  
Кафедра

**Обучение программированию на языке Python учащихся старших классов  
в рамках элективного курса**

Выпускная квалификационная работа

Квалификационная работа  
допущена к защите

Исполнитель: Лысякова Ксения  
Павловна

Зав. Кафедрой

Подпись:

\_\_\_\_\_

дата

\_\_\_\_\_

подпись

Руководитель: Газейкина Анна  
Ивановна  
ученая степень,  
ученое звание/должность

Подпись:

Екатеринбург 2020



## Оглавление

Введение.....	4
<b>1. Теоретические особенности обучения старшекласников программированию в среде Python .....</b>	<b>9</b>
<b>1.1. Место и содержание элективных курсов: назначение, цели, виды и     создание и организация элективного курса .....</b>	<b>9</b>
<b>1.2. Подходы к обучению программированию на старшей ступени     современной школы.....</b>	<b>22</b>
<b>1.3. Анализ существующих элективных курсов по программированию     .....</b>	<b>32</b>
<b>1.4. Выводы по главе 1 .....</b>	<b>44</b>
<b>2. Разработка элективного курса по программированию на языке Python для учащихся старшей школы.....</b>	<b>47</b>
<b>2.1 Цели и содержание курса .....</b>	<b>47</b>
<b>2.2 Поурочное планирование .....</b>	<b>58</b>
<b>2.3. Практические особенности апробации курса .....</b>	<b>65</b>
<b>2.4. Выводы по главе 2.....</b>	<b>72</b>
Заключение .....	73
Список литературы.....	74
Приложение.....	Error! Bookmark not defined.

## Введение

В мире высоких технологий программирование является неотъемлемой частью. Компьютеры уже давно вошли в нашу жизнь, и любой человек, которому приходится хоть какое-то время проводить за компьютером, так или иначе, сталкивается с программированием. Форумы, социальные сети, интернет-магазины, интернет-газеты, различные приложения и др. – всё это основано на программировании и поддерживается специалистами данной области. В свою очередь программирование базируется на языке программирования, которых на сегодняшний день насчитывают огромное множество, но самыми распространенными и популярными по проведенной статистике компанией Tiobe Software в июне 2019г. являются Java, C, C++ и на 4 месте набирает обороты язык программирования Python.

Python – это универсальный язык программирования, созданный нидерландским программистом Гвидо Ван Россумом в 1992г., с помощью которого можно создавать любые приложения в диапазоне от интернетсайтов до роботов и системных сервисов. Отличается он простотой синтаксиса и быстрым написанием кода, быстро осваивается новичками в области программирования. Python не использовался так широко в прошлом, однако стал особенно популярным за последние годы благодаря инвестициям Google. А все потому, что спектр задач, решаемых с его помощью огромен, а простота написания кода дает ему преимущество перед многими другими языками. Питер Норвиг, директор по исследованиям в корпорации Google говорит о Python следующее: «Python был важной частью Google с самого начала, и останется таким до тех пор, пока система растет и развивается. На сегодняшний день десятки инженеров Google используют Python, и мы ищем большое количество людей с навыками в работе на этом языке» [2].

В настоящее время некоторые весьма известные и надежные сайты работают на Python, особенно [pinterest.com](https://www.pinterest.com), [instagram.com](https://www.instagram.com), [rdio.com](https://www.rdio.com) и [YouTube.com](https://www.youtube.com). Архитектор программного обеспечения в компании YouTube

Куонг До говорит о том, что их сайт на Python работает достаточно быстро и позволяет производить обслуживание в рекордное время с минимальным количеством разработчиков. Также Python можно использовать для разработки веб-приложений [2].

Помимо всего вышеперечисленного данный язык программирования используют для тестирования аппаратного обеспечения такие компании как Intel и IBM, также для шифрования и анализа данных, для прогнозирования финансового рынка. Компания Nasa использует Python для научных вычислений.

Опираясь на данные положения, можно сделать вывод о том, что Python является на сегодняшний день достаточно востребованным языком программирования, и потому создается большое количество дистанционных курсов для его изучения, как для школьников, так и для студентов и взрослых.

Основам программирования учат в школах ещё с 1985г., начиная с простого алгоритмического языка. Последние несколько лет обучающихся с 9 по 11 класс учат писать несложные программы на языке Паскаль, реже на Visual Basic и Delphi. В связи с набором популярности языка программирования Python с 2015 года во все варианты КИМ ЕГЭ по информатике примеры программ стали записывать на нём, так как практика показывает, что задания типа С4 проще записывать на языке Python, нежели на классических языках. Отсюда вытекает проблема, которая заключается в том, что на данный момент Python не преподают в школах, отсутствуют учебники и специалисты в общеобразовательных учебных заведениях.

Углубленному изучению темы «Понятие алгоритма. Программирование» предмета всегда сопутствует преподавание элективных курсов, которые способствуют достижению достаточного теоретического и практического уровня.

В связи с вышесказанным в данной работе была поставлена следующая цель: разработать элективный курс «Основы программирования на языке Python» для обучающихся 10–11 классов.

Проблема исследования заключается в том, что на данный момент определяется явный недостаток информационных ресурсов, посвященных программированию на Python, отсутствуют обучающие курсы по программированию на Python для школьников, несмотря на то, что данный язык набирает популярность в современной веб-разработке в целом и является наиболее оптимальным в применении.

Объектом исследования является преподавание программированию на языке Python.

Предмет исследования – элективный курс для учащихся 10-11 классов школы.

Целью работы является разработка элективного курса по программированию на языке Python для подготовки к ЕГЭ старших школьников. Для достижения поставленной цели необходимо выполнить ряд следующих задач:

- формы до образования (элективные курсы: назначение, цели, виды, перет к созданию и организации элективного курса);
- обучение программированию на старшей ступени современной школы;
- анализ существующих элективных курсов по программированию;
- разработка элективного курса по программированию на языке Python для учащихся старшей школы;
- поурочное планирование;
- апробация

При решении поставленных задач были использованы следующие теоретические и практические методы исследования: теоретический анализ и синтез данных психолого-педагогической литературы по теме исследования, анализ нормативно-правовой документации образовательной сферы,

требований к факультативным курсам, к электронным курсам и к образовательным блогам, изучение и анализ электронных курсов для старших школьников по схожей тематике, метод анкетирования, апробация разработанного курса, а также анализ результатов апробации

Предлагаемый элективный курс адресован учащимся 10-11 классов. Главная его идея - это реализация идеи предпрофильной подготовки учащихся, организация систематического и системного повторения, углубления и расширения знаний по теме «алгоритмы и элементы программирования» курса информатики за период изучения в основной школе.

Вопросы, рассматриваемые в курсе, выходят за рамки обязательного содержания. Вместе с тем, они тесно примыкают к основному курсу, поэтому данный элективный курс будет способствовать совершенствованию и развитию важнейших знаний и умений в области информатики, предусмотренных школьной программой. Данный курс является практикоориентированным, дает учащимся возможность познакомиться с основами программирования на языке Python и применить знания на практике при работе над проектами. Все это будет направлено на осмысленное изучение предмета, а значит и правильный выбор дальнейшего профиля обучения в старшей школе.

Выбор Python обусловлен тем, что это язык, обладающий рядом преимуществ перед другими языками: ясность кода, быстрота реализации.

Python — развивающийся язык, используемый в реальных проектах. Средства для работы с Python относятся к категории свободно распространяемого программного обеспечения. Python имеет обширную область применения. Так, на Python создаются расширения к графическому редактору GIMP, на Python можно программировать в офисном пакете OpenOffice.org, на Python пишутся сценарии для пакета 3D-моделирования Blender, Python активно используется при создании компьютерных игр и web-приложений.

Выбор данного направления облегчит обучающимся в дальнейшем изучение программирования, а также разовьёт творческий потенциал для решения нестандартных задач по программированию.

# **1. Теоретические особенности обучения старшекласников программированию в среде Python**

## ***1.1. Место и содержание элективных курсов: назначение, цели, виды и создание и организация элективного курса***

В настоящее время количество учебных часов на изучение дисциплин уменьшается, а количество часов отведенных на самостоятельную работу увеличивается и это прописано в образовательных стандартах [1]. Для более полного обеспечения прав обучающихся, о которых говорится в законе об образовании, а также обозначенных в федеральных образовательных стандартах высшего профессионального образования, в настоящее время школами реализуются возможности, которые данный закон и стандарты предоставляют. В связи с этим, часы, отведенные на самостоятельную работу, используются для предоставления обучающимся возможности принять непосредственное участие в формировании содержания своего профессионального образования в рамках каждой конкретной дисциплины [13].

Элективные курсы – курсы, входящие в состав профиля, способствующие углублению индивидуализации профильного обучения. Работа элективных курсов призвана удовлетворить образовательный запрос (интересы, склонности) ученика (его семьи). В информационном письме Минобрнауки РФ от 13 ноября 2003 г. №14-51-277/13 говорится о том, что «они по существу и являются важнейшим средством построения индивидуальных образовательных программ, так как в наибольшей степени связаны с выбором каждым школьником содержания образования в зависимости от его интересов, способностей, последующих жизненных планов» [27, с. 75].

Прилагательное «элективный» (electus) в переводе с латинского языка означает «избранный, отобранный». Отсюда следует, что любой курс, нареченный в учебном плане элективным должен выбираться. В Концепции

профильного обучения на старшей ступени общего образования (Приказ Минобробразования РФ от 18 июля 2002 г. 3 2783) дано следующее определение: «Элективные курсы – обязательные для посещения курсы по выбору учащихся, входящие в состав профиля обучения на старшей ступени школы» [27, с. 16]. Элективные курсы связаны с удовлетворением индивидуальных образовательных интересов, потребностей и склонностей каждого школьника, направленных на формирование компетенций [37, с. 49]. В отличие от факультативов и кружков курсы обязательны для посещения, именно они позволяют школьникам развить интерес к тому или иному предмету и четко определиться со своим дальнейшим выбором. «Элективные курсы принципиально отличаются от давно существующих у нас факультативов. Факультативный курс – это не обязательный, а только возможный для изучения <...> Идеология «изучаю – не изучаю» не предполагает, что выбор становится обязательным элементом общего образования. А введение элективных курсов направлено на решение именно этой задачи» [18, с. 9].

Элективные курсы реализуются за счет школьного компонента учебного плана, характеризуются минимальной численностью учебной группы (1-15 человек). Каждый учащийся в течение двух лет должен выбрать и изучить 5-6 элективов, при этом количество предлагаемых курсов должно быть значительно выше. Примерное соотношение объемов базовых общеобразовательных, профильных общеобразовательных предметов и элективных курсов определяется пропорцией 50: 30: 20.

Курс по выбору, согласно П.С. Лернеру [12], это — факультативный, элективный курс, направленный как на внутрипрофильную дифференциацию, так и на компенсацию профильной однонаправленности. Курс способствующий углублению индивидуализации профильного образования, расширению мировоззренческих представлений учащихся [12]. Исходя из данного определения, очевидно, что курсами по выбору могут

быть не только элективные курсы, но и факультативы, однако это не является одним и тем же.

В соответствии с Письмом Минобразования России от 13.11.2003 г. № 14-51-277/13 «Об элективных курсах в системе профильного обучения на старшей ступени общего образования» [5], элективные курсы принципиально отличаются от факультативных курсов. Они являются обязательными для обучающихся. Факультативные же занятия обучающиеся могут посещать по желанию [3]. Кроме того, из определения видно, что элективные курсы ориентированы не только на углубленное изучение учебного материала по профильным дисциплинам, но и на внутрiproфильную дифференциацию и индивидуализацию профильного образования, т.е. элективные курсы изначально ориентированы на то, чтобы предоставлять обучающимся возможность строить индивидуальный план обучения.

Концепция профильного обучения на старшей ступени общего образования предусматривает возможность введения пяти профилей. В федеральном базисном учебном плане предлагается вариант введения десяти профилей. Профильное обучение предусматривает возможность для каждого учащегося заниматься по индивидуальному плану, который строится на основе базисного и включает следующие основные блоки. Базовые общеобразовательные предметы – обязательные для всех учащихся во всех профилях обучения. Профильные общеобразовательные предметы – предметы повышенного уровня, определяющие направленность каждого конкретного профиля обучения, обязательные для учащихся, выбравших данный профиль.

Содержание указанных блоков учебных предметов определяется федеральным компонентом государственного стандарта общего образования и контролируется ЕГЭ.

Элективные курсы – обязательные курсы по выбору учащихся, определяемые профилем обучения. Примерное соотношение объемов базовых общеобразовательных, профильных общеобразовательных

предметов и элективных курсов определяется пропорцией 50 : 30 : 20. Элективы реализуются за счет времени, отводимого на школьный компонент образовательного учреждения. В отличие от факультативов, элективные курсы в профильном обучении обязательны. Количество элективных курсов, предлагаемых в составе профиля, должно быть избыточно по сравнению с числом курсов, которые обязан выбрать учащийся. По элективным курсам ЕГЭ не проводится [23].

Элективные курсы – относительно новый элемент учебного плана, дополняющий содержание профиля, что позволяет удовлетворять разнообразные познавательные интересы школьников. Элективные курсы могут касаться любой тематики, как лежащей в пределах общеобразовательной программы, так и вне её. С хорошо разработанной системой элективных курсов каждый ученик может получить образование с определенным желаемым уклоном в ту или иную область знаний.

Актуализация элективных курсов заключается в том, что они предоставляют обучаемым возможность за счет дифференциации и индивидуализации образования, самостоятельно создавать пространство индивидуальной познавательной деятельности, что является одним из приоритетных направлений федеральных государственных образовательных стандартов и отвечает запросам современного общества [23]

В 10-11 классах целью элективного курса является расширение, углубление знаний, выработка специфических умений и навыков, знакомство с новыми областями науки в рамках выбранного профиля [9].

Набор элективных курсов на основе базисного учебного плана определяется самой школой.

Элективные курсы имеют очень широкий спектр функций и задач:

- обеспечивают повышенный уровень освоения одного из профильных учебных предметов, его раздела;
- служат освоению смежных учебных предметов на междисциплинарной основе;

- обеспечивают более высокий уровень освоения одного (или нескольких) из базовых учебных предметов;
- служат формированию умений и способов деятельности для решения практически значимых задач;
- обеспечивают непрерывность профориентационной работы;
- служат осознанию возможностей и способов реализации выбранного жизненного пути;
- способствуют удовлетворению познавательных интересов, решению жизненно важных проблем;
- способствуют приобретению школьниками образовательных результатов для успешного продвижения на рынке труда

Цель преподавания элективных курсов – ориентация учащихся на индивидуализацию обучения и социализацию, на подготовку к осознанному и ответственному выбору сферы будущей профессиональной деятельности. Перед элективными курсами стоит ряд задач: расширить знания по изучаемым предметам; обеспечить более высокий уровень знаний, умений и навыков; способствовать активному самоопределению, в том числе и профессиональному; формировать и развивать познавательный интерес к предметам.

Исходя из этого, тематика и содержание элективных курсов должны отвечать следующим требованиям:

- иметь социальную и личностную значимость, актуальность, как с точки зрения подготовки профессиональных кадров, так и личностного развития учащихся;
- способствовать социализации и адаптации, представлять возможность для выбора индивидуальной образовательной траектории, осознанного профессионального самоопределения;
- поддерживать изучение базовых и профильных общеобразовательных предметов, а также обеспечивать условия для внутрипрофильной социализации обучения;

- обладать значительным развивающим потенциалом;
- вносить вклад в формирование целостной картины мира;
- способствовать развитию общеучебных, интеллектуальных и профессиональных умений и навыков, ключевых компетенций [10, с. 38-39].

Методы и формы обучения определяются уровнем развития и саморазвития учащихся, уровнем их индивидуальных способностей и требованиями профилизации обучения. Ведущие методы – проблемно-поисковые и исследовательские, стимулирующие познавательную активность учащихся. Виды элективных курсов определяются и задачами ступени школы, и видом учебного заведения, и местными условиями, и возможностями класса, группы, учителя, и так далее. Обсуждая вопрос о месте элективных курсов в старшей профильной школе, Каспржак А.Г. (2004) выделяет их виды, содержание и способы работы, образовательные задачи, решаемые курсами по выбору [17, с. 16-21].

По назначению выделены несколько типов элективных курсов: первый тип – это своеобразное дополнение базовых профильных курсов, обеспечивающее повышенный уровень изучения того или иного учебного предмета; второй тип – это курсы, направленные на интеграцию учебных предметов, они обеспечивают межпредметные связи; третий тип – курсы, направленные на подготовку школьников к сдаче ЕГЭ или на подготовку к сдаче вступительного экзамена в вуз по данному предмету, т.е. имеет «репетиторскую» направленность; четвертый тип – курсы, ориентированные на приобретение узкоспециализированных навыков, необходимых для определенной профессии или для успешной адаптации на рынке труда; пятый тип – курсы, носящие внепредметный характер, которые направлены на удовлетворение тех интересов учащихся, которые не связаны с рамками школьной программы, предназначены для расширения кругозора старшеклассника[27].

Основные приоритеты методики изучения элективных курсов:

- междисциплинарная интеграция, содействующая становлению целостного мировоззрения;
- обучение через опыт и сотрудничество;
- интерактивность;
- учет индивидуальных особенностей и потребностей учащихся;
- личностно-деятельностный и субъект-субъектный подход.

Сегодня многие педагоги, методисты и учителя-практики занимаются созданием различных элективных курсов для предпрофильной подготовки учащихся. Классификация предпрофильных элективных курсов, как и любая классификация, является относительной, но большинство авторов выделяют общеориентационные, предметно-ориентационные и межпредметные элективные курсы.

Общеориентационные элективные курсы призваны проинформировать ученика о различных профилях обучения в старшей школе, познакомить его с миром профессий и помочь выбрать профиль обучения с учетом своих индивидуальных особенностей.

Предметно-ориентационные элективные курсы направлены на осуществление предпрофильной подготовки по определенному учебному предмету. Учителя стремятся создать такой элективный курс, который вызовет интерес у ученика, привлечет его к дальнейшему изучению предмета в классе данного профиля. В результате ученик выбирает профиль для продолжения своего дальнейшего образования. Другой особенностью имеющихся сегодня предметно-ориентационных предпрофильных элективных курсов является стремление к углублению знаний учащихся. Такие элективные курсы предполагают углубленное изучение отдельных тем или разделов учебных курсов основной школы, выходящее за пределы школьной программы.

В свою очередь, предметные элективные курсы можно разделить на несколько групп.

1) Элективные курсы повышенного уровня, направленные на углубление того или иного учебного предмета, имеющие как тематическое, так и временное согласование с этим учебным предметом. Выбор такого элективного курса позволит изучить выбранный предмет не на профильном, а на углубленном уровне. В этом случае все разделы углубляются курса более или менее равномерно.

2) Элективные спецкурсы, в которых углубленно изучаются отдельные разделы основного курса, входящие в обязательную программу данного предмета.

3) Элективные спецкурсы, в которых углубленно изучаются отдельные разделы основного курса, не входящие в обязательную программу данного предмета.

4) Прикладные элективные курсы, цель которых - знакомство учащихся с важнейшими путями и методами применения знаний на практике, развитие интереса учащихся к современной технике и производству.

5) Элективные курсы, посвященные истории предмета, как входящего в учебный план школы, так и не входящего в него.

6) Элективные курсы, посвященные изучению методов решения задач.

Межпредметные элективные курсы в системе предпрофильной подготовки не только ориентируют учеников на изучение конкретного учебного предмета на профильном уровне, но и раскрывают специфику изучения этого предмета во взаимосвязи с другими профильными предметами. Такие элективные курсы следует называть профильно-ориентационными, поскольку именно они в полной мере реализуют саму идею предпрофильной подготовки. Они могут быть либо компенсирующими, либо обобщающими знания.

Для всех элективных курсов можно сказать, что они выполняют три основные функции:

1) Являются «надстройкой» профильного курса, и такой дополненный профильный курс становится в полной мере углубленным.

2) Развивают содержание базового курса, что позволяет поддерживать изучение смежных учебных предметов на профильном уровне или получить дополнительную подготовку для сдачи выпускного экзамена по предмету.

3) Способствуют удовлетворению познавательных интересов в различных областях деятельности человека.

Чтобы успешно выполнять названные выше функции, отобранное содержание должно соответствовать познавательным возможностям старшеклассников, предоставлять им возможность учения на уровне повышенных требований и развивать учебную мотивацию.

Особенности содержания любого элективного курса по информатике определяются большим количеством различных факторов.

Самыми важными из них являются следующие четыре фактора:

1. информатика тесно связана с большинством других учебных предметов;

2. изучение информатики необходимо для формирования ключевых компетенций выпускника современной школы, приобретения образовательных достижений, востребованных на рынке труда;

3. изучение информатики играет важную роль в формировании современной научной картины мира;

4. изучение информатики позволяет связать понятийный аппарат естественных, гуманитарных и филологических учебных дисциплин.

Содержание школьного образования обладает немалой инертностью, но необходимо, чтобы образование постоянно модернизировалось, для того чтобы отвечать новым потребностям общества [9].

Каждый новый элемент содержания должен в первую очередь апробироваться в вариативной части школьного образования, и только после этого входить в его инвариантную часть. Ранее новое содержание образования проверялось, как правило, в рамках факультативных курсов.

Ныне данную функцию на старшей ступени выполняют элективные курсы. Эта задача в первую очередь актуальна для развития школьного

образования по информатике. В этом заключается еще одна важная особенность элективных курсов по этому предмету [10].

Современное понимание функциональной грамотности человека все больше включает в себя элементы информационных технологий, информационной культуры. Исключительно велика роль изучения информатики в социализации школьников, подготовке их к труду, профессиональной деятельности, в профессиональном самоопределении молодежи. Это приводит к новому пониманию готовности выпускников учебных заведений к жизни и труду в информационном обществе, заставляет переосмыслить традиционные представления о содержании образования, путях его осуществления [11].

Существует несколько типов элективных курсов по информатике [12].

Классификация обычно приводится по значению:

- Курс, посвященный углубленному изучению учебного материала основного курса;

- Курс, посвященный изучению материала выходящего за рамки школьного курса, изучению дополнительных вопросов информатики (изучение языка программирования и пр.), - Подготовительный элективный курс, посвященный подготовке к различным олимпиадам по информатике, архитектуре компьютера, и т.д.;

- Курс, направленные на демонстрацию межпредметных связей (например, курсы по компьютерному моделированию в физике, математике, химии и т.д.);

- Познавательные курсы, ориентированные на учащихся, интересы которых выходят за рамки школьной программы (например, изучение различных графических и музыкальных редакторов).

Несомненно, организация каждого элективного курса, немислима без методического обеспечения. Подбор учебной литературы для проводимых элективных курсов занимает важное место в их организации. Литературой могут служить учебные пособия к факультативам, кружкам, научная

литература, различные справочники, авторские элективные курсы, Интернетресурсы.

Учебная литература по элективным курсам имеет статус дополнительных учебных материалов, и не требует специального разрешения на использование в учебном процессе от Министерства образования и науки России [3].

Программа любого элективного курса состоит из нескольких компонентов: пояснительной записки, учебно-тематического плана, содержания элективного курса, критериев оценок, перечня литературы для учителя и учащихся, методических рекомендации к курсу и библиографического списка.

Необходимо, чтобы программа элективного курса соответствовала особенностям профильной школы и имела практическую направленность.

Должна присутствовать определенная логика построения и подачи учебного материала. Важно, чтобы программа была хорошо структурированной и связной по содержанию, была реалистичной по времени и затраченным ресурсам, предполагала активные методы обучения, дающие учащимся осознанно и объективно сделать выбор для продолжения образования, иметь определенную степень новизны [13].

Алгоритм написания программы элективного курса:

- 1) Выбор привлекательного названия элективного курса.
- 2) Разработка пояснительной записки.

В ней должно быть отражено: тип элективного курса; общая продолжительность по времени и количество часов в неделю; цели курса; задачи курса; ожидаемые результаты освоения материалов курса; методы преподавания и учения; возможные критерии оценивания.

- 3) Разработка учебно-тематического планирования занятий.
- 4) Разработка содержания занятий элективного курса.
- 5) Набор методических рекомендации по проведению курса.

Основные требования предъявляемые к элективным курсам:

- доступность,
- вариативность,
- краткосрочность,
- оригинальность содержания,
- нестандартность,
- деятельностный подход в организации.

Необходимо применять методы работы, которые стимулируют познавательную деятельность учащихся.

Возможно, применять такие методы как метод проектов, вынос теоретического материала на самостоятельное изучение, подготовка докладов учащимися и так далее [14].

Также в процессе обучения полезно применение элементов обращенного обучения, суть которого состоит в следующем: учащиеся самостоятельно осуществляют первичное знакомство с теоретическим материалом, освобождая при этом время для коллективных обсуждений, анализа и рефлексии на занятиях. Однако подобный метод требует существенной подготовки со стороны учителя [15].

Как уже было отмечено ранее учебная литература по элективным курсам имеет статус дополнительных учебных материалов, и не требует специального разрешения на использование в учебном процессе.

Однако проблема обеспечения учебного процесса в школе всегда находится в центре внимания педагогов. Особенно это касается элективных курсов.

Методическое обеспечение – это необходимая информация, учебнометодические комплексы, т.е. разнообразные методические средства, оснащающие и способствующие более эффективной реализации программно-методической, научно-экспериментальной, воспитательной, организационно-массовой, досугово – развлекательной деятельности работников системы школьного образования.

Методическое обеспечение - это процесс, направленный на создание разнообразных видов методической продукции, на оказание методической помощи различным категориям педагогических работников, на выявление, изучение, обобщение, формирование и распространение положительного педагогического опыта.

Перед учителем встает вопрос выбора типа методического обеспечения к конкретному элективному курсу. На сегодняшний день наиболее актуальной является тема информатизации образования. В частности тема использования современных технологий на уроках. Поэтому, сегодня наибольшую популярность набирают электронные методические пособия, такие как электронные учебники и электронные тесты.

Также любой из видов методической продукции может быть представлен как электронное издание (CD, DVD, публикация в сети Интернет) или мультимедийной презентации [16].

Таким образом:

На основании рассмотренного теоретического материала по наукометодическим и теоретическим основам организации элективных курсов было установлено следующее.

Элективные курсы предназначены, прежде всего, для удовлетворения индивидуальных образовательных интересов, потребностей и склонностей каждого школьника. Именно они являются важнейшим средством построения индивидуальных образовательных программ, т. к. в наибольшей степени связаны с выбором каждым школьником содержания образования в зависимости от его интересов, способностей, последующих жизненных планов. Элективные курсы компенсируют возможности базовых и профильных курсов в удовлетворении образовательных потребностей старшеклассников. Тематика данных курсов зависит непосредственно от пожеланий конкретной школы.

При создании методического обеспечения к элективному курсу целесообразно использовать современные информационные технологии, в частности электронные учебно-методические пособия и презентации.

## ***1.2. Подходы к обучению программированию на старшей ступени современной школы***

Преподавание информатики в школах нашей страны фактически начиналось с преподавания программирования. В то время даже был провозглашен лозунг: «Программирование – это вторая грамотность». Заметим, что компьютеры в школах тогда практически отсутствовали. А та техника, которой оснащались школы в конце 80-х – начале 90-х годов прошлого века, практическую составляющую курса информатики все равно невольно сводила к программированию.

Одновременно с революционным развитием аппаратного и программного обеспечения и оснащением современной компьютерной техникой учебных заведений курс информатики претерпел существенные изменения. Наиболее яркая характеристика такого изменения – вымывание программирования из школьного курса информатики. Основное внимание в большинстве школ стало уделяться освоению современных информационных технологий. Эти тенденции отражены и в новом «Стандарте по информатике», в котором собственно обучению программированию отводится очень мало времени. Но, как заметил А.Г. Гейн, «...очевидно, что именно алгоритмизация с самого начала вытянула на школьную арену курс информатики и ныне во многих реально существующих курсах информатики позволяет уйти от умных, но пустоватых разговоров к конкретному делу».

Программирование является одной из самых сложных и динамично развивающихся дисциплин, преподаваемых в школах и вузах. Согласно приказам Минобрнауки России от 17 декабря 2010 г. № 1897 для 5-9 классов [0] и Минобрнауки России от 17 мая 2012 г. № 413 для 10-11 классов

программированию начинают учить в старших классах средней школы. Данный процесс затрагивает обучение основам алгоритмизации, понятие информации и принципы работы с ней. Стандарты, указанные в приказах (№ 413 п. 11.3, № 1897 п. 9.3) об общем образовании регламентируют начальный этап обучения информатике.

В школе затрагивают понятие программирование, но более углубленно занимаются этим в вузах, хотя, к сожалению, далеко не всегда успешно.

Анализ особенности обучения программированию в подростковом и юношеском периодах опирается на результаты исследований проблемы процессов обучения, развития и оценку степени зрелости психических функций к началу обучения, описанные Л.С. Выготским в книге «Мышление и речь» [16].

Язык программирования – это искусственно созданный человеком язык для описания алгоритма, предназначенного для исполнения компьютером. Язык программирования не используется в устной форме, так как не является языком общения, это язык инструкций для формального исполнителя, которые должны быть представлены в письменной форме. Проведем параллель между изучением естественной письменной речи для общения и искусственной письменной речи для управления, опираясь на результаты исследований развития речи Л.С. Выготского, сформулируем выводы об особенностях изучения ребенком языка программирования.

Исследования Л.С. Выготского показывают, что освоение письменной речи для ребенка сложнее, чем освоение устной речи, письменная речь «не есть простой перевод устной речи в письменные знаки», «письменная речь есть совершенно особая речевая функция..., которая требует для своего хотя бы минимального развития высокой степени абстракции» [16]. Ребенок с помощью звуковой речи уже достиг определенной степени абстракции по отношению к предметному миру, далее он должен перейти к абстракции, которая обозначается не словами, а представлением слов в знаковой форме. Особенность письменной речи в том, что это речь без собеседника, т.е. от

ребенка требуется «двойная абстракция: от звучащей стороны и от собеседника» [16], автор сравнивает письменную речь с «символизацией второго порядка».

Продолжая эту мысль, можно назвать изучение языка программирования «символизацией третьего порядка». Это уже речь с «собеседником», собеседник – компьютер требует еще более высокой степени абстракции, так как реакцию такого собеседника ребенок должен сам спрогнозировать, продумать, просчитать, ожидать, т.е. это собеседник, который думает мыслями самого программиста. Так при изучении программирования мышление, опираясь на уже сформированные степени абстракции при развитии устной речи и письменной речи, поднимается еще на одну ступень абстракции вверх.

В работе «Мышление и речь» Л.С. Выготский пишет: «Понятие невозможно без слов, мышление в понятиях невозможно вне речевого мышления; новым, существенным, центральным моментом всего этого процесса, имеющим все основания рассматриваться как производящая причина созревания понятий, является специфическое употребление слова, функциональное применение знака в качестве средства образования понятий». Таким образом, в речи слово – это знак, обозначающий понятие, действие, в программировании слово – это опосредованное активное действие, с помощью которого осуществляется деятельность.

Дальнейшие исследования Л.С. Выготского показывают, что сложность изучения письменной речи связана с отсутствием мотивов для обращения к письму. Необходимость устной речи есть результат эгоцентрического сознания ребенка [29], развитие устной речи мотивировано жизненными ситуациями. «К началу школьного обучения потребность в письменной речи является совершенно незрелой. Можно даже сказать на основании данных исследования, что школьник, приступающий к письму, не только не ощущает потребности в этой новой речевой функции, но он еще в высшей степени смутно представляет себе, для чего вообще эта функция нужна ему» [16].

Освоение устного языка осуществляется бессознательно, освоение письменного языка осуществляется сознательно.

Изучение языка программирования становится еще более сложным, чем изучение письменной речи, т.к. еще меньше мотивации у ребенка при изучении языка программирования, повседневная жизнь не создает естественной потребности в искусственном языке. Освоение языка программирования должно опираться на мотивированную деятельность и будет являться сознательностью как бы второго порядка, опирающуюся на сознательно изученную письменную речь.

Требуемой мотивацией при изучении языка программирования может быть интерес, содержащийся в деятельности по исследованию окружающей действительности, которая выполняется в процессе моделирования. Моделирование - деятельность, к которой учащийся готов, так как активно вовлечен в эту деятельность с момента рождения, является ее непосредственным участником при обучении в явном или неявном виде.

В чем состоит проблема понимания и обучения принципам программирования [9]? Как в школах, так и в вузах, учеников и студентов сложно заинтересовать чем-либо, если результат их трудов не виден сразу на экране компьютера. Эта сфера требует от человека усидчивости и терпения. При изучении нового (или первого) языка программирования нужна практика. Также необходимо прочитать немало обучающей литературы, чтобы изучить конкретный язык программирования или хотя бы понимать основы. Наибольший интерес у обучающихся вызывает графика (как в восьмых, так и в одиннадцатых классах), при работе с которой на экране почти сразу виден красочный результат выполнения программы [4].

Перед учителем поставлена задача не просто предоставить выпускнику набор знаний, а подготовить учащегося, умеющего самостоятельно решать возникающие перед ним проблемы, способного отвечать за собственное благополучие и общества в целом. Для этого требуется создать необходимые условия для подготовки ответственной, самостоятельной и активной

личности, владеющей основными компетентностями. Одна из этих основных компетентностей - это умение составлять и осуществлять программу своей деятельности. Такие умения у учащихся формируются на уроках информатики при выполнении различных заданий и, прежде всего, при изучении темы «Алгоритмизация и программирование». Таким образом, особое внимание в обучении информатике необходимо уделять вопросу системы обучения программированию.

Тема «Алгоритмизация и программирование» является одной из самых сложных тем при изучении курса информатики. В настоящее время существует большое количество проблем в данном направлении. В целом есть два пути: либо преподавать на уроках информатики лишь офисные средства и различные прикладные программы, исключив программирование, либо изучать программирование. На сегодняшний день выпускник должен иметь необходимый набор знаний по информатике, позволяющий сдать ЕГЭ на достаточном уровне, для этого необходимо изучать программирование на уроках информатики, то есть первый путь отмечается сразу. Проводящиеся соревнования (олимпиады, конкурсы, конференции) по программированию в нашей республике также являются подтверждением необходимости изучения темы «Алгоритмизация и программирование» в школьном курсе информатики.

Весь процесс обучения программированию в школе разбивается на несколько этапов. Перед началом обучения учитель сталкивается с проблемой выбора языка программирования для изучения. Одна группа учителей обучают теме «Алгоритмизация и программирование» на основе формальных алгоритмов, построив обучение учащихся на языке блок-схем. Другая группа учителей информатики обучают учащихся тому языку программирования, с помощью которого умеют решать задачи, знают основы соответствующего языка и используют методически правильное изучение выбранного ими языка программирования. Тем самым потеряно единое образовательное информационное пространство не только страны, но и отдельных регионов.

Таким образом, в начале обучения теме «Алгоритмизация и программирование» учителю необходимо выбрать язык программирования с учетом компетентности учителя, интересов учащихся, их направленности и структуры обучения информатике в школе. Большинство учителей в самом начале XXI века преподавали язык Basic, который стал известен множеству школьников.

Сегодня во многих школах проводится изучение языка Pascal, который является более подходящим с методической точки зрения для изучения основных принципов программирования. Язык Pascal является учебным структурным языком программирования, который предполагает не только изучение алгоритмических конструкций, формирование логического и алгоритмического мышления у учащихся, но и решение сложных технологических и производственных задач. И только учащиеся небольшого числа школ изучают языки программирования Visual Basic, C, C++, Visual C++, Delphi, Java, Python и другие.

На мой взгляд, язык Python лучше традиционных для школы языков программирования и оптимально подходит как для начального обучения программированию, так и для построения профильных курсов. Можно отметить следующие достоинства языка Python [9]:

1. Понятность языка выше, чем у Pascal и Basic. Простые программы записываются в несколько строк, не нужны инструкции, не имеющие непосредственного отношения к алгоритму (например, `int main()`).

2. Простой и лаконичный синтаксис. Как правило, программа на языке Python записывается короче, чем на C++, Pascal и Basic.
3. Свободная реализация.

4. Современность языка, наличие в нем высокоуровневых структур данных (списки, множества, ассоциативные массивы, длинная арифметика).

5. Наличие средств объектно-ориентированного программирования (ООП).

б. Наличие библиотеки, позволяющей легко разрабатывать графические приложения, web-приложения и т.д

Следующим важным этапом при построении системы изучения темы «Алгоритмизация и программирование» является организация самого процесса обучения. В качестве основных видов организации процесса обучения программированию можно выделить традиционные виды занятий: лекционные, лабораторные и практические занятия. Для проведения занятий необходимо выбрать соответствующее учебнометодическое, техническое и программное обеспечение.

Основным этапом построения обучения является технология обучения программированию. Учитель не должен забывать, что цель обучения - не изучение языка программирования и не подготовка высококвалифицированных программистов, а изучение способов, алгоритмов и методов программирования при решении задач, то есть привитие навыков алгоритмического мышления учащимся.

В настоящее время изучение темы «Алгоритмизация программирования» в школе проводится для разных категорий учащихся, главным образом это связано с появлением профильного обучения. Одна категория учащихся изучает программирование в соответствии с содержанием образования по информатике, обязательного для всех учащихся общеобразовательных школ. Для основной школы цель обучения определяется изучением основных алгоритмических конструкций с использованием простых типов данных и массивов. Другая категория учащихся изучает программирование на углубленном уровне, что соответствует обучению в таких профильных классах, как информационно-коммуникационное и физико-математическое направления. Кроме базовых знаний в области программирования, описанных выше, учащиеся данных направлений должны владеть одним из современных языков программирования, обладая умением проводить вычислительные процессы и оперировать всеми основными типами данных.

Изучение темы «Алгоритмизация и программирование» на базовом уровне предлагаем начинать с 8-го класса, если существует возможность построения такого процесса обучения информатике в школе. Сначала необходимо изучать алгоритмы на формальных языках и использовать их на формальных исполнителях.

В 9-м классе учащимся необходимо освоить понятие языка программирования, рассмотреть классификацию языков программирования, их необходимость и направленность. Затем рассматриваются основы структурного программирования на одном алгоритмическом языке программирования совместно с основными алгоритмическими конструкциями, для этого лучше всего выбрать язык Python.

Начинается изучение языка программирования со знакомства с основными компонентами данного языка (константы, идентификаторы, переменные, типы данных, принципы записи математических выражений, составной оператор, пустой оператор, комментарии), основными стандартными процедурами и функциями, структурой программы на языке Python, описывается каждый раздел описания языка. Далее последовательно изучаются основные алгоритмические конструкции. Сначала учащимся дается конструкция на языке блок-схем, а затем реализация в виде программ на выбранном языке программирования. Каждая конструкция закрепляется самостоятельным решением учащимися как общих, так и индивидуальных задач на практических занятиях.

В 10 классе необходимо повторить основные конструкции языка и перейти к изучению темы «Процедуры и функции». Далее рекомендуется рассмотреть понятие локальных и глобальных переменных, их отличия, способы реализации и использования на основе процедур и функций. При изучении процедур и функций на практических занятиях учащиеся могут вспомнить основные алгоритмические конструкции. Затем осуществляется переход на изучение темы «Модули», во время практических занятий учащиеся должны овладеть умением формирования и работой с модулями с

помощью процедур и функций. При изучении данной темы учащиеся закрепляют знания алгоритмических конструкций и их использование при решении задач различного уровня сложности.

В 11 -м классе на базовом уровне учащиеся знакомятся со строковым и символьным типом данных. Изучают принципы работы с символьными и строковыми переменными, также реализуя алгоритмические конструкции. Дополнительно рассматриваются такие структуры данных, как записи и множества. Изучение языка программирования заканчивается освоением понятия файловой переменной, повторением и решением задач сложного уровня, реализуя основные методы работы с данными.

Благодаря такому построению изучения языка программирования на базовом уровне дается возможность учащимся освоить основы алгоритмического языка Python, способствует развитию технологического, алгоритмического и творческого мышления учащихся, формирует основные навыки программирования.

В итоге изучения учащиеся должны знать:

- что такое алгоритм, какова роль алгоритма в системах управления;
- в чем состоят основные свойства алгоритма;
- способы записи алгоритмов: блок-схемы, учебный алгоритмический язык;
- основные алгоритмические конструкции: следование, ветвление, множественный выбор, цикл, структуры алгоритмов;
- назначение вспомогательных алгоритмов, технологии построения сложных алгоритмов: метод последовательной детализации и сборочный (библиотечный) метод;
- основные свойства величин в алгоритмах обработки информации: что такое имя, тип, значение величины; смысл присваивания;
- назначение языков программирования;
- в чем различие между языками программирования высокого уровня и машинно-ориентированными языками;

- правила представления данных на одном из языков программирования высокого уровня (например, на языке Python);

- правила записи основных операторов: ввода, вывода, присваивания, цикла, ветвления;

- правила записи программы;

- что такое трансляция;

- назначение систем программирования;

- содержание этапов разработки программы: алгоритмизация, кодирование, отладка и тестирование.

При этом учащиеся должны уметь:

- пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;

- выполнять трассировку алгоритма для известного исполнителя;

- составлять несложные линейные, ветвящиеся и циклические алгоритмы управления на одном из учебных исполнителей;

- выделять подзадачи, определять и использовать вспомогательные алгоритмы;

- составлять несложные программы решения вычислительных задач с числами;

- программировать простой диалог;

- работать в среде одной из систем программирования (например, Python);

- осуществлять отладку и тестирование программы и другое.

Таким образом, изучение языка программирования учащимся осложнено необходимостью сформированности мышления с высоким уровнем абстрагирования и отсутствием естественной мотивации у ребенка. Однако, верны и выводы А.П. Ершова, который видит в программе «...огромный запас операционного знания, накопленный человечеством и теперь лишь актуализируемый вычислительными машинами... Еще более огромный запас программ хранится в генофонде всего живого» [33]. Поэтому,

предлагая к изучению в школе язык программирования, А.П. Ершов отмечает, что «речь идет не о том, чтобы навязать детям новые, несвойственные им навыки и знания, а о том, чтобы проявить и сформулировать те стороны мышления и поведения, которые реально существуют, но формируются стихийно, неосознанно» [33].

### ***1.3. Анализ существующих элективных курсов по программированию***

В программе «Цифровая экономика Российской Федерации» определено, что система общего образования должна работать в интересах подготовки граждан к жизнедеятельности в условиях цифровой экономики. К 2024 году в образовательных организациях должны быть созданы условия для реализации обучающимися персональных образовательных маршрутов, для формирования базовых компетенций цифровой экономики.

Концептуальные основы формирования цифровых навыков в настоящее время находятся в стадии разработки [3-5]. Выделено пять групп цифровых навыков: общие ИТ-навыки, профессиональные ИТ-навыки, проблемно-ориентированные цифровые навыки, комплементарные ИТ-навыки (complementary skills), навыки использования сервисов цифровой экономики [5]. При этом, как отмечают авторы данной концепции, в процессе перехода к цифровой экономике ускоренными темпами возрастает спрос на общие и профессиональные ИТ-навыки.

Программа 10-11 класса базового уровня предусматривает 13 часов из 68 часов на обучение программированию, углубленного - 100 часов из 272. Кроме того, изучение разделов программирования не предусматривает непрерывной линии подготовки. Темы, связанные с программированием, появляются в рабочих программах фрагментарно, начиная с 8 класса. Таким образом, реализовать качественную, непрерывную, углубленную подготовку школьников по программированию только средствами школьной

дисциплины «Информатика» в настоящее время не представляется возможным.

Относительно недавно разработанный язык программирования Python имеет лаконичный и в то же время, довольно, нетрудный и понятный синтаксис в связи с этим все чаще предлагается к обучению авторами учебных пособий. Python поддерживает модули и пакеты, поощряя модульность и повторное использование программного кода.

Алгоритмический язык программирования может быть предложен для решения задачи учащимся не знакомыми с языками программирования.

Преподавание элективных курсов по программированию мы рекомендуем начинать с 9 класса, так как к этому моменту у школьников уже сформированы начальные представления об основах программирования, ими приобретены первые навыки написания небольших программ и они, вполне вероятно, сделали выбор языка программирования для дальнейшего изучения.

Основными задачами подобных элективных курсов мы считаем следующие:

- создание условий для персонифицированного обучения школьника программированию;
- создание условий для реализации интереса учащегося к освоению выбранного им языка программирования;
- развитие креативных и рефлексивных способностей учащихся в области программирования;
- формирование интереса к изучению профессий, связанных с программированием.

Предлагаемая нами методика персонифицированного обучения школьников программированию реализуется за несколько этапов.

На первом этапе учитель анализирует тенденции и востребованность различных языков программирования в современном обществе: в профессиональной сфере(взгляд ИТ-профессионалов), в сфере высшего и

среднего профессионального образования (успешное дальнейшее обучение выпускников), в олимпиадах и иных конкурсных мероприятиях. На основе данного анализа и, учитывая имеющиеся в распоряжении информационные ресурсы, он выбирает несколько языков программирования для дальнейшего освоения. Нами были отобраны три языка программирования: C++, Python и Pascal.net.

На втором этапе проектируется программа персонифицированного обучения школьников программированию (9-11 классы) на основе интеграции предмета «Информатика», элективных курсов по программированию и внеурочной деятельности. Определяются необходимые информационные ресурсы и формируются примерные персональные маршруты для учащихся.

Далее на третьем этапе с учетом спроектированной программы персонифицированного обучения школьников программированию и действующей программы предмета «Информатика» формируются программы элективных курсов по программированию.

Четвертый этап методики представляет собой реализацию предмета «Информатика» и элективных курсов по программированию на основе интеграции классно-урочной и внеурочной работы (рис. 1)



Рис. 1. Схема интеграции классно-урочной и внеурочной работы при персонифицированном обучении школьников программированию.

Цель данных занятий – дать возможность ученику определиться с выбором конкретного языка программирования, с использованием которого будет проходить его дальнейшее обучение по программе курса.

Цель учителя – понять мотив его выбора, интерес школьника к изучению того или иного языка, уровень его начальной подготовки, что позволит верно выстроить индивидуальный образовательный маршрут для каждого ученика. Для этого учитель после первой двух занятий в форме устной беседы проводит опрос учеников и возможно помогает ученику определиться с выбором языка программирования.

Дальнейшие теоретические занятия представляют собой уроки-лекции учителя об основных алгоритмических конструкциях, способах решения различных задач с использованием этих алгоритмических конструкций, типовых и продвинутых алгоритмах обработки различных структур данных с примерами на языке блок-схем и формальном алгоритмическом языке. При этом важно понимание, что цель обучения – не изучение конкретного языка программирования, не подготовка высококвалифицированных программистов, а изучение способов, алгоритмов, подходов и методов программирования при решении различных задач.

После каждого урока-лекции ученику во внеурочное время самостоятельно следует познакомиться с индивидуально отобранным для него материалом – лекцией-презентацией от учителя или видеоуроком, материалом с образовательных онлайн ресурсов, в которых рассказывается об особенностях реализации основных алгоритмических конструкций на одном из языков программирования C++, Python или Pascal.net. При этом школьники могут воспользоваться многочисленными образовательными онлайн ресурсами (презентации с сайта К.Ю.Полякова по основам программирования на языках C++, Python или Pascal.net <http://kpolyakov.spb.ru/school/ppt.htm>; образовательный ресурс Stepik,

содержащий многочисленные бесплатные курсы по программированию на различных языках C++, C#, Java, Python и др., PythonTutor – интерактивный учебник языка Python; Hexlet —бесплатные курсы по C, PHP, JavaScript, Bash; CodeHunt - игра, которая учит искать ошибку в коде на языке Java, C#; Школа программиста ascp.ru – образовательный интернет-ресурс по различным языкам программирования и олимпиадному программированию,

Дистанционная подготовка по информатике Informatisc – образовательная онлайн платформа с множеством теоретического материала и задач по соответствующим темам программирования с поддержкой различных языков программирования). Таким образом, при такой форме обучения ученик сам (или консультируясь с учителем) определяет для себя формы и темп работы, уровень трудности, самостоятельно изучает новый материал и применяет полученные знания, получает адекватную оценку своей работе.

На уроке-практике происходит закрепление изученного теоретического материала. Для этого школьникам предлагается воспользоваться полученными теоретическими знаниями при решении задач по программированию. Ученики делятся на небольшие группы. При этом отсутствует традиционное деление на сильных, средних и слабых учеников. Деление по группам происходит по интересам. Основным интересом в данном случае является изучаемый язык программирования. Группы имеют свойство мобильности, т.е. состав и численность участников группы может меняться и не один раз, чтобы добиться активной и заинтересованной работы всех членов группы.

На самых начальных этапах, как правило, группы малочисленны (2 человека), при этом один из участников группы уже имеет небольшой навык программирования и может выступить в роли наставника совместно с учителем для второго – начинающего участника группы. При этом создание атмосферы коллективного труда, когда сильный ученик имеет возможность помочь более слабому, дает возможность каждому участнику группы понять,

что работа в группе – это не поле для соперничества и борьбы за лидерство, а хороший стимул к активизации интереса, познавательной потребности.

Таким образом, реализуется и командный стиль работы. На уроках-практиках каждая группа школьников получает материалы к уроку, в которых есть краткая теоретическая часть с ориентацией на теперь уже конкретный язык программирования (C++, Python или Pascal.net) и набор практических заданий. Во время групповой работы учитель выполняет роль тьютора, который контролирует ход и порядок работы в группах, отвечает на вопросы, регулирует споры и в случае крайней необходимости оказывает помощь отдельным учащимся или группе в целом. При этом следит за активностью и заинтересованностью в работе каждого участника группы.

Выполнение самостоятельной внеклассной работы также происходит с помощью онлайн ресурсов. Как правило, учителем создается группа из учеников данного класса и руководителя группы – учителя. Выполненные домашние задания – программы, ученик загружает на соответствующий онлайн ресурс, а учитель имеет возможность контролировать ход выполнения задач в группе и консультировать при необходимости учеников. Например, это можно сделать с помощью образовательной онлайн платформы «Дистанционная подготовка по информатике» [informatics.mcsme.ru](https://informatics.mcsme.ru), Школа программиста <https://acmp.ru/>.

На этапе, когда школьники хорошо освоили основные типовые алгоритмы решения задач с помощью языка программирования, последующее обучение в рамках элективного курса состоит из командной или индивидуальной проектной деятельности. При этом на начальном этапе работы над проектом основная задача учителя состоит в том, чтобы верно распределить или помочь ученикам самостоятельно распределить роли в группе при работе над проектом, а также определить основные этапы и формы работы над проектом.

Темы проектов могут быть разного уровня сложности, но конечным результатом проекта является написанное готовое ИТ-решение, программный продукт.

Овладев на достаточно хорошем уровне основными и более продвинутыми алгоритмами решения задач средствами того или иного языка программирования, учащиеся имеют возможность принять участие в различных олимпиадах, конкурсах, турнирах, требующих хороших навыков программирования.

Анализ учебно-методических комплексов (УМК) по информатике и информационным технологиям авторов Л. Л. Босовой, И. Г. Семакина, А. Г. Гейн, К. Ю. Полякова на предмет выявления рекомендованных к изучению языков программирования позволяет сделать следующие выводы.

В УМК Л. Л. Босовой программирование рассматривается только на базовом уровне в 11 классе, во второй главе «Алгоритмы и элементы программирования». Авторский коллектив предлагает начать главу с повторения рассмотренных ранее сведений о понятии алгоритма, а затем приступают к введению определения массива и операций над ними, работе с подпрограммами и рекурсивными алгоритмами. В качестве основополагающего берется язык программирования Pascal. Однако УМК Босовой Л. Л. не включено в федеральный перечень учебников на 2017-2018 [1].

Аналогично, в учебниках И. Г. Семакина для 10-11 классов базового уровня система понятий алгоритмов плавно переходит к программированию на языке Pascal. Глава «Программирование обработки информации» изучается в 10 классе. Преимуществом УМК считаем рассмотрение символьных, комбинированных и строковых типов данных [4].

На углубленном уровне обучения информатике предлагается к использованию универсальный язык C, и допускается возможность самостоятельной теоретической подготовки на языке программирования Python [3].

В УМК авторского коллектива К. Ю. Полякова и Е. А. Ерёмкина рассматривается язык программирования Python, как на базовом, так и на углубленном уровне обучения. На базовом уровне вводятся представления о языке, типах данных, простейших операциях, переменных, вычислениях и т. д. В учебном пособии для углубленного изучения язык Python представлен более широко. Однако учебник для базового уровня в настоящее время не включен в федеральный перечень [2].

Таким образом, приходим к выводам, что в используемых УМК по информатике основам алгоритмизации и программирования предлагаются к рассмотрению на примере нескольких языков программирования, среди которых чаще встречаются Pascal и все более набирающий популярность Python.

Далее поговорим об анализе элективных курсов по программированию. Существует множество источников по написанию элективных курсов и учебных программ, которые помогают учителю как можно лучше подготовить ученика к основному государственному экзамену.

Элективные курсы (дополнительные занятия выбору) – это непосредственно занятия по выбору учащихся, входящие в состав профильного обучения на старшей ступени школы, позволяющие школьникам развивать интерес к определенным предметам и определять свои профессиональные склонности [6]. Их включают в образовательный процесс за счет школьного компонента учебного плана и они дополняют содержания профиля. Могут быть в качестве курсов для углубленного изучения отдельных учебных предметов, в том числе и информатики. Занятия, развивающие программу одного из базисных курсов, изучение, которого в данной школе преподается на минимальном общеобразовательном уровне. Это позволяет интересующимся школьникам удовлетворить свои познавательные потребности и получить дополнительную подготовку, например, для сдачи ОГЭ или ЕГЭ по информатике. Также курсы направлены на удовлетворение интересов

учащихся в областях деятельности, выходящих за рамки выбранного направления для обучения в школе. Например, если школьник, обучающийся в классах гуманитарного профиля, проявит интерес к курсу «Программирование».

Рассмотрим основные требования к элективным курсам:

Базисный размер каждого элективного курса составляет 34-70 ч. в год (1-2 часа в неделю).

Поскольку элективные курсы считаются курсами по выбору, в связи с этим формируются подгруппы в количестве от 10 до 20 человек.

Формы обучения имеют все шансы быть как академическими, так и нацеленными на инновационные педагогические технологии (проектные, исследовательские, игровые, тренинги и др.).

Способы изучения подразумевает присутствие важного оснащения и программного обеспечения.

Методическое обеспечение элективного курса включает в себя: программу курса, пособие для учащихся, материалы для учителя, справочную литературу.

Перечень учебно-методического обеспечения

- Ресурсы сайта Л. Шапошниковой (<http://younglinux.info>) ,  
содержащие

- авторские программы

- <http://younglinux.info/python.php>

(<http://window.edu.ru/resource/825/76825>)

- <http://younglinux.info/algorithm>

- учебные пособия

[http://younglinux.info/sites/default/files/python\\_structured\\_programming.pdf](http://younglinux.info/sites/default/files/python_structured_programming.pdf)

- Сайт Д.П.Кириенко. Московский институт открытого образования. Школа №179

- <http://www.179.ru/~dk/python.html>

Авторская программа Д. П. Кириенко. Программирование на python (школа 179 г. Москвы) <http://informatics.mccme.ru/course/view.php?id=156>

- Ресурсы сайта профессора, доктора технических наук, учителя информатики высшей категории, автора учебников по информатике К.Ю.

Полякова:

- [http://kpolyakov.spb.ru/loadstat.php?f=/download/ch10-8\\_python.pdf](http://kpolyakov.spb.ru/loadstat.php?f=/download/ch10-8_python.pdf)
- <http://kpolyakov.spb.ru/loadstat.php?f=/download/slides10-8py.zip>
- <http://kpolyakov.spb.ru/download/progr1011.doc>
- [http://kpolyakov.spb.ru/download/infobr\\_2013-6.pdf](http://kpolyakov.spb.ru/download/infobr_2013-6.pdf)

К.Ю. Поляков, Е.А. Еремин. Информатика. 10 класс. Углубленный уровень. - М.: Бином, 2015. (варианты глав по программированию для изучающих python <http://kpolyakov.spb.ru/school/probook/python.htm>)

- Сайт дистанционная подготовка по информатике

<http://informatics.mccme.ru/>

- <http://informatics.mccme.ru/course/view.php?id=95>

задачник: <http://informatics.mccme.ru/course/view.php?id=666>

- Изучаем Python Автор: Марк Лутц Издательство: Символ-Плюс ISBN 978-5-93286-159-2, 978-0-596-15806-4; 2011 г. Переводчик: А. Киселев

- Head First Python. Автор: Paul Barry. Издательство: O'Reilly, год: 2010, Язык: Английский, страниц: 494, формат: pdf, размер: 33 МБ (<http://forcoder.ru/python/head-first-python-1191>)

- Hello World! Computer Programming for Kids and Other Beginners Second edition, Warren D. Sande and Carter Sande <http://helloworldbookblog.com/>

- Python for Kids. A Playful Introduction to Programming by Jason R. Briggs 2012, 344 pp. <http://www.nostarch.com/pythonforkids>)

Учебная программа элективного курса – это нормативный документ, в котором отражены цели, содержание, особенности оценки эффективности результатов процесса обучения определенному учебному курсу [6]. Программа в обязательном порядке обязана быть утверждена Министерством

образования и науки РФ или же возможно использовать авторскую адаптированную программу, утвержденную на методическом объединении учителей информатики.

При наличии маленького размера учебного времени, программирование в базисном курсе имеет возможность изучаться только на уровне введения. Тема «Алгоритмизация и программирование» является одной из самых сложных тем при изучении курса информатики. В настоящее время существует большое количество проблем в данном направлении. В целом есть два пути: либо преподавать на уроках информатики лишь офисные средства и всевозможные прикладные программы, исключив программирование, либо изучать программирование.

На сегодняшний день выпускник обязан владеть материалом по информатике, позволяющий сдать ЕГЭ на достаточном уровне, для этого необходимо изучать программирование на уроках информатики, то есть первый путь откладывается незамедлительно [5].

Для того чтобы учителю эффективно подготовить ученика к ЕГЭ, необходима программа, которая поможет ученику успешно сдать экзамен, а учителю правильно и точно распределить время занятий, выделить на каждую тему определенное количество часов. В настоящее время разработано большое количество готовых программ.

В каждой разработанной программе должны быть выделены цели элективного курса, задачи, требования к уровню подготовки выпускников 11 класса в области информатики: что должны знать и что должны уметь.

Таким образом, при подготовке к ЕГЭ учителю важно разработать программу – методический комплекс, предназначенный для изучения курса материала. Он является интегрированным средством, содержащим теорию, практику, задачи и другие компоненты. Программа облегчит работу учителя и поможет успешно подготовиться ученикам к экзамену.

Рассмотрим несколько программ для подготовки к ЕГЭ по программированию, выделим их несомненные достоинства.

«Готовимся к ЕГЭ по информатике» для 11 класса – курс рассчитан на один учебный год (32 ч.), охватывает полностью все темы для подготовки к ЕГЭ. На блок «Алгоритмизация и программирование» всего отводится 7 часов из курса. Данный курс хорош по своему содержанию, но на программирование выделено недостаточно времени для усвоения необходимых тем для сдачи ЕГЭ, даже с учетом материала изученного в школьном курсе.

«Подготовка к ЕГЭ по информатике» – курс рассчитан на 102 часа по 3 раза в неделю. В курсе разработаны все блоки для подготовки к ЕГЭ. На блок «Алгоритмизация и программирование» отводится достаточно большое количество часов. Элективный курс был разработан по старым стандартам для ЕГЭ. Многие задания поменяли содержание и формулировку.

«Программирование на языке Pascal». Программа элективного курса для 10-11 классов – курс рассчитан на 35 часов, рассматриваются многие необходимые для сдачи ЕГЭ темы из программирования. Но курс направлен конкретно для освоения навыков программированию, а не на подготовку к ЕГЭ.

Программа «Программируем на языке Паскаль» рассчитана на 2 учебного года (68 ч.), охватываются абсолютно все элементы в программировании. Курс направлен на формирование интереса к изучению профессии, связанной с программированием [7]. Обширный и полезный курс, который очень хорошо может помочь подготовиться к ЕГЭ. На данном элективном курсе имеется возможность получить много теоретической информации для подготовки к экзамену, но отсутствуют типовые задания из ЕГЭ и не рассматриваются виды заданий.

Рассмотрев разработанные программы для элективных курсов, мы пришли к выводу, что конкретно направленной программы для подготовки к ЕГЭ на тему программирования нет. Для организации элективного курса будет удобным являться разработка своей собственной программы, которая

позволит разобрать один из самых важных блоков в ЕГЭ – блок «Алгоритмизация и программирование».

#### ***1.4. Выводы по главе 1***

На основании рассмотренного теоретического материала, включающего в себя особенности создания элективных курсов по программированию для профильных классов, было установлено следующее: Создание программы элективного курса является сложным процессом, требующим высокого уровня квалификации от автора-разработчика.

Недостаточная готовность педагогических кадров к разработке новых или модернизации уже имеющихся программ является причиной нарушения технологии создания программ, серьезных недочетов в них. Поэтому, разрабатывая программу элективного курса на этапе предпрофильной и профильной подготовки обучающихся, нужно исходить из конкретных условий: материально-технической базы учреждения, запросов обучающихся, региональной и местной специфики, рынка труда и рынка образовательных услуг. Необходимо помнить, что элективные курсы как наиболее дифференцированная, вариативная часть школьного образования требуют новых решений в их организации. Широкий спектр и разнообразный характер элективных курсов могут поставить отдельную школу в затруднительное положение, определяемое нехваткой педагогических кадров, отсутствием соответствующего учебно-методического обеспечения.

Все это в полной мере относится к элективным курсам по информатике. Более того, реализация целого ряда элективных курсов по информатике связана с использованием дорогостоящего аппаратного и программного обеспечения, которым большинство школ не обладает. В этих случаях особую роль приобретают сетевые формы взаимодействия образовательных учреждений. Сетевые формы предусматривают объединение, кооперацию образовательного потенциала нескольких

образовательных учреждений, включая учреждения НПО и СПО, дополнительного образования и вузов.

При разработке элективного курса по программированию важно учитывать индивидуальный запрос обучающихся, который может быть направлен на сферу веб-программирования. На данный момент существует огромное количество языков веб-программирования, каждый из которых имеет свои особенности и закономерности. Наиболее широко используемыми языками в веб-программировании являются PHP, Python и Ruby. Python — универсальный язык программирования, при помощи которого можно делать любые приложения в диапазоне от интернет-сайтов и десктоп-ных приложений до роботов и системных сервисов.

В ходе проведенного теоретического анализа по теме исследования было установлено, что элективные курсы современных образовательных технологий являются частью обучения, отличительной особенностью которого выделяют предоставление обучающимся возможности в самостоятельном получении требуемых знаний, используя развитыми информационные ресурсы, предоставляемые современными информационными технологиями. Также были выявлены психолого-педагогические особенности преподавания факультативных курсов, которые, если говорить в общем, заключаются в использовании личностного образовательного потенциала человека, развивающегося в тех областях, которые он сам для себя выбирает [16].

Изучив теоретическую, нормативно-правовую и психолого-педагогическую литературу, нам также удалось выяснить, что факультативные курсы играют важную роль в образовательном процессе средней школы, они индивидуализируют данный процесс, отвечают потребностям обучающихся, способствуют формированию профессиональных умений и навыков, а также профессиональному самоопределению.

Кроме того, в рамках главы 1 нам удалось провести обзор существующих элективных курсов для углубленного изучения темы «Алгоритмизация и программирование» в соответствии с предъявляемыми требованиями к электронным курсам для эффективного обучения, который показал, что данные курсы не выполняют в полном объеме предъявляемые к ним требования, в результате чего появилась необходимость в создании нового элективного курса, который бы охватывал все вышеописанные требования для качественного изучения курса «Программирование на языке Python», структура, содержание и результаты апробации которого будут описаны в следующей главе.

## **2. Разработка элективного курса по программированию на языке Python для учащихся старшей школы**

В данной главе предлагаются к рассмотрению особенности организации обучения программированию обучающихся старшей школы с помощью элективного курса «Программирование на Python». Содержание курса рекомендовано к изучению в том случае, если цепи обучения в процессе изучения элективного курса совпадают с профессиональными интересами обучающихся. Другими словами для изучения курса обучающимся необходимо знание основ программирования, наличие опыта разработки приложений на любом другом языке программирования и понимание основ веб-разработки, так как курс ориентирован на предпрофессиональную подготовку старшеклассников в сфере веб-программирования.

### ***2.1 Цели и содержание курса***

Элективный курс «Программирование на Python» разработан в рамках реализации Концепции профильного обучения на старшей ступени общего образования и соответствует Государственному стандарту среднего образования по информатике и ИКТ

При разработке данного курса учитывалось то, что элективный курс как компонент образования должен быть направлен на удовлетворение индивидуальных потребностей и профессиональных интересов старшеклассников, на формирование у них новых видов познавательной и практической деятельности, которые не характерны для традиционных учебных курсов, для предпрофессиональной подготовки. В качестве компонентов структуры программы выступают: пояснительная записка, описание результативно-целевой направленности (цель, планируемые

результаты обучения, организация учебного процесса, формы контроля и критерии оценки планируемых образовательных результатов, критерии отбора содержания элективного курса, методические особенности УМК), содержание обучения с описанием теоретической и практической частей, тематическое планирование с описанием основных понятий и практикума по решению задач, учебно-тематическое планирование, список рекомендованной литературы.

Программа рассчитана на 34 часа (1 час в неделю) для обучающихся информационно-технологического профиля 10-11 классов, ранее изучавших как минимум один язык программирования из следующих: Pascal, C, C++ - на базовом уровне. Программа содержит все необходимые темы, отвечающие требованиям качественной предпрофессиональной подготовки обучающихся по предмету.

Целями данного курса является развитие алгоритмического мышления обучающихся, обучение основам веб-программирования с использованием системы программирования Python, освоение всевозможных методов решения задач, реализуемых на Python, углубление знаний, знакомство с конструированием объектов и технологией проектирования web-страниц, формирование профессионального интереса к особенностям веб-программирования на Python.

Python – это интерпретируемый, объектно-ориентированный язык программирования высокого уровня с динамической типизацией, автоматическим управлением памятью и удобными высокоуровневыми структурами данных, такими как списки, кортежи и словари. Поддерживает классы, модули, обработку исключений, а также многопоточные вычисления. Достоинством языка является то, что он обладает простым и выразительным синтаксисом, поддерживает структурное, объектно-ориентированное, функциональное и аспектно-ориентированное программирование [2].

Язык программирования Python эффективен с точки зрения разработчика: программы делают больше, чем на многих других языках, в

гораздо меньшем объёме кода. Значительно простой синтаксис позволяет легко читать код программы, возникает меньше проблем с отладкой и расширением программ. Он же является преимуществом при выборе языка начинающим программистом.

Одним из начальных понятий при изучении программирования является понятие переменной. Для успешного обучения написания оптимального программного кода очень важно понять, что такое переменная, научиться хранить в ней данные и использовать их в программах.

Понятие переменной в языке программирования C++ как и в языке программирования Паскаль тождественно: под переменной понимают поименованный участок памяти, в котором хранится значение определенного типа. Другими словами переменная – это именованная область памяти для хранения данных, которые могут изменяться в процессе выполнения программы. Исходя из определения, отличительными характеристиками данного понятия являются тип, идентификатор и значение. Имена и типы переменных указываются при описании и не могут изменяться в процессе выполнения программы. В Паскале тип переменной объявляется после служебного слова `var`, в разделе описания переменных.

В языке C++ переменная может быть объявлена в любом месте программы. Минусом является то, что созданная переменная, пусть в ней даже ничего не хранится, в Паскаль и C++ уже занимает определенный объем оперативной памяти компьютера. И этот объем, в свою очередь, зависит от типа переменной.

Синтаксис языка программирования C++ упрощает синтаксис языка программирования Паскаль. А Python, в свою очередь, упрощает синтаксис языка C++. Поэтому Python имеет много схожего с C++, но также решает более широкий спектр вопросов, чем его предшественники.

В языке программирования Python переменная является простейшим указателем на значение определенного типа. Поэтому она инициализируется присваиванием некоторому идентификатору значения при помощи оператора

присваивания. Таким образом, еще одно преимущество языка Python состоит в том, что переменные не занимают столько места в оперативной памяти компьютера. Тип переменных в данном языке не объявляется, он определяется автоматически при выполнении оператора присваивания.

Существуют различные классификации типов данных. Например, типы данных подразделяются на стандартные и заданные пользователем. К стандартным типам данных относятся `int` (целый), `float` (вещественный с плавающей точкой), `char` (символьный), `string` (строковый). К типам данных, заданным пользователем, относятся массивы и структуры.

Самой распространенной упрощенной структурой, реализованной практически во всех языках программирования, является массив. Рассмотрим подробнее использование массивов в языках программирования Паскаль, C++ и Python.

Массивы состоят из ограниченного числа компонент одного и того же типа. Структура массива однородна, к любой его компоненте можно обращаться произвольным образом и получить значение нужного элемента по его индексу.

В языке Паскаль тип массива задается с использованием специального слова `array` (англ. – массив), после чего указывается тип индекса массива и тип элементов массива. С помощью индекса можно обращаться с отдельным элементом любого массива, как к обычной переменной заданного типа: можно получать значение этого элемента, отдельно присваивать ему значение, использовать его в выражениях. Недостатком языка Паскаль является то, что переменные в массиве можно перебирать с интервалом равным 1 в прямом или обратном порядке с помощью стандартного цикла обхода (цикл `for` – цикл с параметром).

Доработка языка программирования C++ позволила шире использовать все возможности работы с массивами. При инициализации массива можно задать его размер, но не описывать его значения, можно задать размер и тут

же описать этот массив, а можно описать массив, не задавая при этом его размера.

Важным отличием является также то, что можно перебирать элементы массива не только в прямом и обратном порядке, а так же устанавливать интервал перебора элементов с помощью того же стандартного цикла обхода. Это является значительным плюсом языка C++ в работе с массивами, а также преимущества данной циклической конструкции `for`.

Массивы в Паскаль и C++ похожи тем, что требуют указывать точный диапазон использования элементов массива, не позволяя расширять его динамически.

Для хранения большого количества данных в Python вместо массивов используют такие структуры данных как кортежи и списки. Таким образом, понятие массивов претерпевает изменения. Классифицируются типы данных, заданных пользователем, на изменяемые и неизменяемые. К неизменяемым относятся кортежи, к изменяемым – списки и словари.

Кортеж используется для представления неизменяемой последовательности разнородных объектов. Его элементы записываются в круглых скобках. Так как кортеж – один из типов последовательностей, то всё, что можно применять к строкам в Python, верно и по отношению к кортежам. Можно найти длину кортежа, вывести все его элементы на экран с помощью цикла `for`, а также, пользуясь оператором `in`, проверить кортеж на вхождение какого-либо элемента. Кортежи можно сцеплять, создавать на них срезы.

Кортежи – хороший способ манипулировать элементами разных типов в составе одной последовательности. Но в силу того, что кортеж неизменяем, иногда возникают небольшие проблемы. Однако существуют последовательности другого типа, называемые списками. Главным преимуществом списка является то, что он изменяем. С ним можно проводить те же самые операции, что и с кортежем и даже больше: его элементы можно удалять, а также добавлять новые. Список, как и кортеж,

представляет собой последовательность элементов, пронумерованных от 0, как символы в строке. Список можно задать перечислением элементов в квадратных скобках.

Кортежи и списки позволяют упорядочить данные в виде последовательностей. Со словарями дело обстоит немного иначе. В них информация представлена не как последовательность элементов, а как набор пар. Заметно сходство словарей с такой структурой данных как вложенный массив. Элементы словаря заключаются в фигурные скобки.

При работе с кортежами, словарями и списками удобно пользоваться функцией `len(s)`, где `s` последовательность. Данная функция определяет длину последовательности. Необходимо заметить, что в Паскаль и C++ такой функции нет! Чтобы в Паскале и C++ определить количество элементов в массиве, необходимо инициализировать новую переменную, присвоить ей значение 0, а затем с помощью цикла `for` перебрать все элементы массива, увеличивая созданную переменную с каждым элементом на 1 – стандартный цикл обхода массива.

Большим достоинством языка программирования Python является то, что к его изменяемым структурам данных, то есть к спискам и словарям, применяется определенный ряд встроенных методов, значительно упрощающих работу с этими типами данных.

Еще одним важным элементом сравнения для анализируемых языков программирования является запись блоков или составных операторов.

Для логически верного использования конструкций языка важно помнить, что в Паскаль блоки выделяются с помощью служебных слов `begin` – `end`, в C++ – с помощью фигурных скобок `{ }`. В языке программирования Python дело обстоит гораздо иначе. Здесь большую роль играют отступы. Вложенные инструкции объединяются в блоки по величине отступов. Отступ может быть любым, главное, чтобы в пределах одного вложенного блока отступ был одинаков. Но не стоит забывать про читаемость кода! Именно эта

особенность Python и повышает читаемость кода только лишь спецификацией языка.

Например, при записи условного оператора в первой строке размещается условие (логическое выражение), а в блоке с отступом – практически любое действие. Если условие истинно, то выполняется код в блоке после команды `if`, а если ложно, этот код игнорируется.

Нередко в программе требуется проверять более двух возможных ситуаций. Для этого в Python предусмотрен синтаксис `if – elif – else`. Python выполняет только один блок в этой цепочке. Все условия проверяются по порядку до тех пор, пока одно из них не окажется истинным. Далее выполняется код, следующий за этим условием, а все остальные проверки Python пропускает. Для языков C++ и Паскаль реализация подобных ситуаций в коде требует использование вложенных конструкций, что усложняет читабельность кода и требует усиления внимания программиста к выполнению правил их использования («else относится к ближайшему if»).

Использование в Python конструкции `if – elif – else` похоже на использование операторов выбора `case of` и `switch()` в Паскале и C++ соответственно.

Циклы `while` с предусловием и постусловием в данных языках программирования особого отличия не имеют.

Цикл со счетчиком или, как он еще называется — цикл с параметром в Паскале служит для организации повторений с заранее известным их количеством. Оператор в теле конструкции может быть одиночным или составным оператором. Счетчик цикла или параметр цикла – это обычная переменная, которая может быть только порядкового (целочисленного, символьного, логического) или перечислимого типа. При прямом направлении цикла счетчик в каждой итерации всегда увеличивается на единицу, при обратном направлении – уменьшается на единицу. Важно знать: параметр цикла нельзя изменять внутри тела цикла! Также нельзя менять его начальное и конечное значения.

В C++ дело обстоит немного иначе. Сначала инициализируется параметр цикла, после чего ставится точка с запятой. Затем задается конечное значение счетчика цикла. Снова ставим точку с запятой. Задаем шаг цикла. Шаг цикла – это значение, на которое будет увеличиваться или уменьшаться счетчик цикла при каждом проходе.

Чтобы в Python использовать цикл `for` необходимо сначала написать служебное слово `for`, потом имя переменной, в которой предполагается сохранять значения элементов, потом `in`, затем имя перебираемой последовательности, двоеточие и, наконец, тело цикла. Вот и всё.

В проведенном выше анализе языков программирования высокого уровня: Паскаль, C++ и Python были выделены основные преимущества последнего и одновременно представлена линейка синтаксических усовершенствований, которые дают представления о направлениях эволюции языков программирования.

Актуальность данного курса состоит в общей недостаточности ресурсов на русском языке, посвященных изучению основ веб-программирования на Python старшими школьниками при том, что язык веб-программирования Python является наиболее высокоуровневым, и интерпретируемым, динамическим языком, максимально ориентированным на разработку веб-приложений. Для обучающихся, выбравших для углубленного изучения основы программирования и веб-программирование, данный курс поможет в подготовке к профессиональной деятельности.

В пояснительной записке к программе представлено обоснование необходимости освоения способов учебной деятельности, направленных на изучение программирования, что связано с целым рядом умений и навыков (организация деятельности, ее планирование, контроль осуществления), которые по праву носят общепедагогический характер и формирование которых - одна из приоритетных задач современной школы. Содержание программы структурировано, имеет практическую направленность, предполагает использование широкого спектра активных методов обучения.

Целевая аудитория: 10-11 классы.

Цель курса: подготовка обучающихся к решению заданий на языке Python на ЕГЭ;

Задачи курса:

- формирование базовых понятий структурного программирования;
- показать практическую значимость использования программирования для наглядного представления решения задач в различных областях жизнедеятельности человека;
- научить учащихся программированию на языке Python;
- научить составлению и оформлению программ в соответствии с требованиями к синтаксису Python;
- содействовать развитию логического и аналитического мышления учащихся.

Требования к начальной подготовке, необходимые для успешного усвоения курса:

- Знать/понимать понятие алгоритма как организованной последовательности действий, допустимых для некоторого исполнителя и записанной на естественном языке; свойства алгоритма, основные алгоритмические структуры, правила записи арифметических выражений.
- Уметь составлять алгоритмы при решении задач на естественном языке.

Требования к уровню усвоения учебного материала к концу курса:

- владеть навыками алгоритмического мышления и понимание необходимости формального описания алгоритмов;
- владеть стандартными приёмами написания программы для решения стандартной задачи с использованием основных конструкций программирования и отладки таких программ на языке Python;
- знать место языка Python среди языков программирования высокого уровня,

- знать особенности структуры программы, представленной на языке Python,

- иметь представление о модулях, входящих в состав среды Python

- знать возможности и ограничения использования готовых модулей,

- иметь представление о величине, ее характеристиках,

- знать что такое операция, операнд и их характеристики,

- знать принципиальные отличия величин структурированных и не структурированных,

- иметь представление о таких структурах данных, как число, текст, кортеж, список, словарь,

- иметь представление о составе арифметического выражения;

- знать математические функции, входящие в Python, иметь представление о логических выражениях и входящих в них операндах, операциях и функциях,

- уметь записывать примеры арифметических и логических выражений всех атрибутов, которые могут в них входить,

- знать основные операторы языка Python, их синтаксис,

- иметь представление о процессе исполнения каждого из операторов,

- уметь разрабатывать программы обработки числовой и символьной информации,

- уметь разрабатывать программы (линейные, разветвляющиеся и с циклами),

- иметь представление о значении полноценных процедур и функций для структурно-ориентированного языка высокого уровня,

- знать правила описания функций в Python и построение вызова,

- знать принципиальные отличия между формальными, локальными и глобальными переменными,

- знать область действия описаний в функциях,

- иметь представление о рекурсии, знать ее реализацию на Python,

- владеть основными приемами формирования процедуры и функции,

- знать свойства данных типа «массив», «матрица»
- уметь воспроизводить алгоритмы сортировки массивов и двумерных массивов, поиска в упорядоченном массиве, распространять эти алгоритмы на сортировку и поиск в нечисловых массивах
- уметь читать и записывать текстовые файлы в заданном формате
- уметь определять типы данных, записывать логические выражения, использовать в задачах такие алгоритмические конструкции как цикл и ветвление, решать тематические задачи, решать задачи уровня ЕГЭ.
- применять полученные знания в решении заданий уровня ЕГЭ, а также в изучении других языков программирования.

Личностные результаты: готовность и способность обучающихся к саморазвитию и личностному самоопределению, сформированность их мотивации к обучению и целенаправленной познавательной деятельности, системы значимых социальных и межличностных отношений, способность ставить цели и строить жизненные планы, сформированность мировоззрения, соответствующего современному уровню развития науки и техники; осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов

Метапредметные результаты: самостоятельность в планировании и осуществлении учебной деятельности и организации учебного сотрудничества с педагогами и сверстниками, способность к построению индивидуальной образовательной траектории, владение навыками учебно-исследовательской, проектной и социальной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания.

Благодаря тому, что рекомендуемые источники содержат большое количество заданий разного уровня сложности, можно составлять для каждого учащегося индивидуальное задание по каждой изучаемой теме,

которое будет учитывать индивидуальные интересы ученика, уровень освоения учебного материала, особенности освоения учебного материала.

Текущий контроль уровня усвоения материала осуществляется по результатам выполнения практических работ, зачетных практических работ по основным темам данного элективного курса, а также по результатам выполнения мини-проектов.

Итоговый контроль реализуется в форме защиты итоговых проектов.

Защита итогового проекта предполагает публичное выступление учащегося с демонстрацией результатов работы программы и презентации, содержащей аргументы в пользу практической ценности работы, анализ полученных результатов, оценку достигнутого прогресса в знаниях. Также обязательной частью отчёта является ответы на вопросы одноклассников.

В завершении данного элективного курса учащиеся могут представить результаты работы над проектами на классном или школьном конкурсе творческих работ.

Преобладающий тип занятий - практикум. Все задания курса выполняются с помощью персонального компьютера. Форма занятий направлена на активизацию познавательной деятельности, на развитие алгоритмического, операционного мышления учащихся.

Уроки строятся в соответствии с требованием санитарных норм, теоретические и практические части занятий чередуются, во время работы за компьютером используются упражнения для глаз.

## ***2.2 Поурочное планирование***

Содержание курса

Модуль 1. Синтаксис языка программирования Python (2 часа) Понятие о языке Python. Где применяется. Технология разработки программного обеспечения. Стиль программирования. Структура простейшей программы. Переменные и константы. Решение задач.

Модуль 2. Основные управляющие конструкции линейного алгоритма (4 ч.) Ввод-вывод. Концепция присваивания. Арифметические и логические выражения. Программы с линейной структурой. Решение задач.

Модуль 3. Основные управляющие конструкции ветвления (2 ч.) Логический тип. Условная инструкция. Решение задач на циклы и условия.

Модуль 4. Основные управляющие конструкции циклического алгоритма (3 ч.) Знакомство с исполнителем PyRobot. Цикл for. Цикл while. Вложенные циклы. Решение задач.

Модуль 5. Элементы структуризации программы (2 ч.)

Исполнитель PyRobot. Функции в программировании. Функции с аргументами. Функции с результатами. Задачи с исполнителем. Парадигма структурного программирования.

Модуль 6. Структура данных - список, кортеж, множество (2 ч.)

Графический модуль turtle. Списки. Решение задач со списками. Срезы в списках. Генераторы списков. Решение задач со списками и срезами. Кортежи, множества и диапазоны.

Модуль 7. Модуль tkinter (2 ч.)

Событийная модель построения приложения. Виджеты. Модуль tkinter. Миниигра

Модуль 8. Обработка массивов (4 ч.)

Операции со списками. Сортировка массива. Двоичный поиск. Двумерные массивы. Словари (ассоциативные массивы) Разработка мини-игры

Модуль 9. Объектно-ориентированное программирование (3 ч.)

Объектно-ориентированное-программирование. Классы в PyShop. Особенности ООП в Python. Разработка мини-игр.

Модуль 10. Обработка текстов (2 ч.)

Строки. Срезы в строках. Методы строк. Решение задач.

Модуль 11. Обработка чисел (2 ч.)

Анализ цифр числа. Сумма и произведение последовательности чисел,  
поиск максимального и минимального в потоке, проверка простоты.

Модуль 12.Разработка web-приложений (2 ч.)

Основы разработки web-приложений в Python.

Модуль 13.Проект(4 ч.)

Выполнение и защита проекта.

Учебно-тематическое планирование

Содержание учебного материала	Колво часов	В т.ч.		Формы занятий	Формы контроля
		лекции	практикум		
Модуль 1. Синтаксис языка программирования Python	4	1	3	Мини-лекция, практикум	Практическая работа 1
Модуль 2.Основные управляющие конструкции линейного алгоритма	8	1	7	Мини-лекция, практикум	Практическая работа 2
Модуль 3. Основные управляющие конструкции ветвления	6	1	5	Мини-лекция, практикум	Практическая работа 3-5
Модуль 4. Основные управляющие конструкции циклического алгоритма	4	1	3	Мини-лекция, практикум	Практическая работа 6 Зачетная практическая
Модуль 5.Элементы структуризации программы	4	1	3	Мини-лекция, практикум	Практическая работа 7-8
Модуль 6. Структура данных - список, кортеж, множество	4	1	3	Мини-лекция, практикум	Практическая работа 9-10
Модуль 13.Проект	4		4	Работа над проектом, практикум. Отчетная	Защита проектов
Итого	34	6	28		

Таблица 2 Программа курса

Тема	Содержание	Вид деятельности	Задания для учащихся	Оценка
Первая программа «Привет, Мир!»	Функции ввода и вывода, написание первой программы «Привет, мир!», Выполнение арифметических действий	Лекция, вопросы для обсуждения, задачи	Приведите примеры по изучаемым функциям из реальной жизни. Например, из колонок выводится звук. Это пример функции вывода. Задача 1. Выведите решение примеров: 1) 3+5; 2) 54*2; 3) 4/2; 4) 11-6. Задача 2. Используя функцию print, выведите в интерпретатор своё имя, возраст и класс	Оценивается участие в обсуждениях. Каждая задача оценивается в 0,5 баллов
Условный оператор if	Оператор ветвления (if-elif-else), проверка истинности, логические операторы, операторы сравнения	Лекция, задание, задача	Постройте логическую цепочку из чисел 3 12 6 8,2 0 0 и опишите принцип её построения. Задача 3. Даны три целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 3 (если все совпадают), 2 (если два совпадают) или 0 (если все числа различны).	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 3 балла
Вычисления	Целочисленная арифметика, действительные числа, операция деления, операция взятия остатка от деления, числа с плавающей точкой, целое число int, результат операции деления float	Лекция, задание, задача	Преобразуйте ряд действительных чисел к целому: 2.99·10-23 Задача 4. Пирожок в столовой стоит a рублей и b копеек. Определите, сколько рублей и копеек нужно заплатить за n пирожков. Программа получает на вход три числа: a, b, n, и должна вывести два числа: стоимость покупки в рублях и копейках. Входные данные: 10 15 2	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 2
Цикл for	Цикл for, синтаксис, функция range, переменная color, значения типа int и str,	Лекция, задание, задача	Задайте множество значений для сказки «Репка». Задача 5. Даны два целых числа A и B. Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае. Входные данные: вводятся два целых числа. Выходные данные: выводится ответ на задачу	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 2
Строки	Функция len, срезы (slices), методы find, rfind, replace, count, параметры old, new	Лекция, задание, задача	Метод вызывается в виде Имя_объекта.Имя_метода(параметры) – по данной конструкции опишите поход красной шапочки к бабушке. Задача 6. Дана строка. Сначала выведите третий символ этой строки. Во второй строке выведите предпоследний символ этой строки. В третьей строке выведите первые пять символов этой строки. В четвертой строке выведите всю строку, кроме последних двух символов. В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся начиная с первого). В шестой строке выведите все символы с нечетными индексами, то	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 4

			<p>есть начиная со второго символа строки.  В седьмой строке выведите все символы в обратном порядке.  В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.  В девятой строке выведите длину данной строки.  Входные данные: Abrakadabra  Выходные данные: r r Abrak  Abrakadab Arkdba baaar arbadakarbA abdkrA</p>	
Цикл while	Цикл while, синтаксис, переменная i, переменная length, инструкции управления циклом	Лекция, задание, задача	<p>Составьте алгоритм действия цикла пока из реальной жизни.  Например, стирка в стиральной машине.  Задача 7. По данному целому числу N распечатайте все квадраты натуральных чисел, не превосходящие N, в порядке возрастания.  Входные данные: 50  Выходные данные: ответ на задачу.</p>	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 2
Списки	Список Primes, список Rainbow, операции конкатенация и повторение, функции map и list, методы split и join, генераторы списков	Лекция, задание, задача	<p>Составьте список животных из сказки «Колобок».  Задача 8. Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет — не выводите ничего. Если таких пар соседей несколько — выведите первую пару.  Входные данные: -1 2 3 -1 -2  Выходные данные: ответ на задачу</p>	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 3
Функция и рекурсия	Функции def, factorial, binominal, max, локальные и глобальные переменные, рекурсия	Лекция, задание, задача	<p>При помощи рекурсии напишите эпитафия «У попа была собака, он её любил, она съела кусок мяса, он её убил».  Задача 9. Напишите функцию min4(a, b, c, d), вычисляющую минимум четырех чисел, которая не содержит инструкции if, а использует стандартную функцию min.  Считайте четыре целых числа и выведите их минимум. Входные данные: 4 5 6 7  Выходные данные: 4</p>	Оценивается правильность ответа к заданию. Максимальный балл за задачу – 3
Двумерные массивы	Обработка и вывод вложенных списков, создание вложенных списков, вывод двумерного массива, пример обработки двумерного массива, вложенных генераторы двумерных массивов	Лекция, вопрос на обсуждение, задача	<p>С помощью чего удобнее выводить массивы, с помощью цикла for или метода join?  Задача 10. Найдите индексы первого вхождения максимального элемента. Входные данные: Программа получает на вход размеры массива n и m, затем n строк по m целых чисел в каждой. Например: 3 4 0 3 2 4 2 3 5 5 5 1 2 3  Выходные данные: Выведите два числа: номер строки и номер столбца, в которых стоит наибольший элемент в двумерном массиве. Если таких</p>	Оценивается активное участие в обсуждении вопроса. Максимальный балл за задачу – 2

			элементов несколько, то выводится тот, у которого меньше номер строки, а если номера строк равны то тот, у которого меньше номер столбца. Например: 1 2	
Множества	Случаи использования словарей, создание словаря, работа с элементами словаря, перебор элементов словаря	Лекция, вопрос на обсуждение, задача	Напишите ключ и значение телефонного справочника. Задача 12. Вам дан словарь, состоящий из пар слов. Каждое слово является синонимом к парному ему слову. Все слова в словаре различны. Для слова из словаря, записанного в последней строке, определите его синоним. Входные данные: 3 Hello Hi Bye Goodbye List Array Goodbye Выходные данные: ответ на задачу.	Оценивается активное участие в обсуждении вопроса. Максимальный балл за задачу – 1
Контрольная работа	Контрольные задачи, контрольный тест, групповая работа в виде задачи повышенной сложности	Решение задач, тестирование, групповое решение задачи и обсуждение	Тест, состоящий из 25 вопросов 5 задач Групповая задача	Тест оценивается в 25 баллов: 1 задание – 1 балл. Задача 1 – 3 балла; Задача 2 – 2 балла; Задачи 3-5 – 1 балл; Групповая задача – 6 баллов

Разработка тематических заданий осуществлялась с упором на тему урока. Задания разрабатывались таким образом, чтобы попытаться вызвать заинтересованность у учащихся и спровоцировать их на обсуждение.

Практические же задачи в основном взяты из КИМов ЕГЭ по информатике и из вышеперечисленных учебников по программированию на Python. Такие задачи направлены на проверку сформированности важнейших умений записи и анализа алгоритмов на языке Python, необходимых для решения задач уровня ЕГЭ. Критерии оценивания задач разрабатывались в соответствии с условием задачи.

Программа данного элективного курса носит примерный характер и является открытой для внесения изменений, позволяя учителю, в зависимости от использования курса в рамках того или иного профиля конкретного образовательного учреждения и исходя из особенностей состава учащихся, осуществлять доработку факультативного курса.

Таким образом, мы можем отметить, что одной из особенностей разработанного курса является открытость для внесения изменений при реализации в конкретных образовательных условиях.

Таким образом, используя методические рекомендации к разработке элективных курсов, требования к электронным курсам, а также требованиям к образовательному блогу, мы описали все компоненты программы разрабатываемого курса «Программирование на языке Python».

Результаты оценочного этапа проектирования представлены в следующем параграфе.

### ***2.3. Практические особенности апробации курса***

Апробация разработанного элективного курса осуществлялась в соответствии с теоретической концепцией исследования, основой которой является выдвинутое в форме гипотезы утверждение о том, что обучение с помощью элективного курса по программированию на языке Python для подготовки к ЕГЭ по информатике старших школьников будет эффективным, если:

- он будет включать в себя теоретический материал по теме курса;
- промежуточный контроль в виде тестов и тематических задач;
- контрольные и групповые работы;
- дополнительные материалы, соответствующие требованиям к электронным курсам и образовательному блогу;
- обратную связь.

Педагогический эксперимент проводился в три этапа. Первый этап был мотивирование учащихся на прохождение элективного курса. При очной встрече учащимся 11А и 11Б, выбравшие в качестве дополнительного предмета для сдачи ЕГЭ информатику, было рассказано о том, что такое Python, его преимущества, особенности, перспективность. Важно отметить, что вся эта информация также размещена на курсе. Учащимся было предложено зайти на курс и пройти предварительное анкетирование. В анкетировании все обучающиеся приняли участие, однако, приступили к обучению 6 человек из 13. Результаты анкетирования представлены на рисунках 2-4.

13 ответов

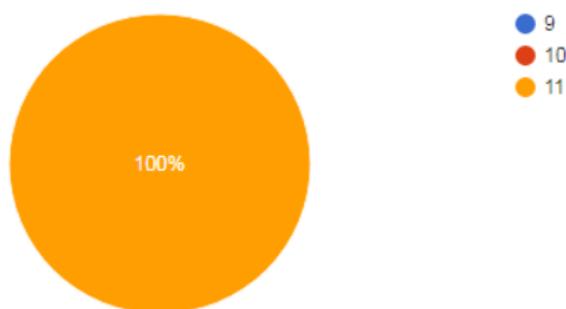


Рисунок 2 - Укажите класс, в котором вы обучаетесь на данный момент

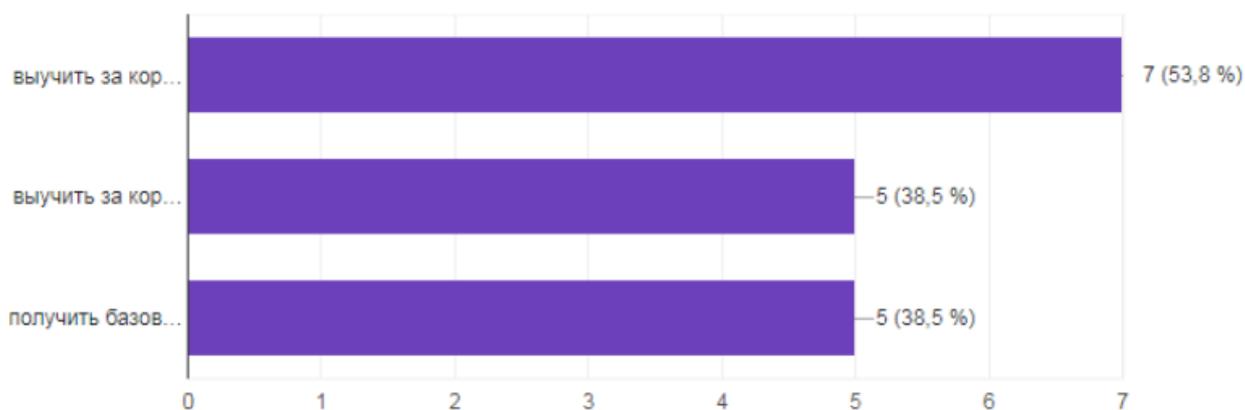


Рисунок 3 - Цели прохождения курса



Рисунок 4 - На каком языке программирования вы умеете решать задачи из ЕГЭ по информатике

Сводка результатов показала, что все учащиеся являются выпускниками, 53,8% в качестве цели прохождения курса выделяют «Выучить за короткий срок язык Python для успешной сдачи ЕГЭ, ибо не знаю ни одного языка программирования», 38,5% учащихся желают выучить язык Python в качестве запасного языка, 38,5% учащихся имеют цель освоить

базу программирования на Python. Также результаты анкетирования показали, что 46,2% учащихся не умеют программировать ни на одном языке программирования, 38,5% умеют писать программы на алгоритмическом языке, 38,5% умеют решать задачи на языке Паскаль. Предположительно, курс проходили те 46,2% обучающихся, которые не знают ни одного языка программирования.

Второй этап педагогического эксперимента проходил в виртуальной среде. Учащиеся записались на курс, ознакомились с правилами поведения на курсе, руководствами, процедурой оценки успеваемости, а также с лекциями, представленными на курсе. На данном этапе главным было поддерживать мотивацию учащихся. Мотивация поддерживалась при помощи обратной связи, обновлением интересных новостей по теме курса, ежедневными опросами, которые поддерживали дружескую атмосферу в рамках курса, а также заданиями по темам лекций, в которых прослеживалась связь с художественной литературой и с реальной жизнью. Учащиеся принимали активное обсуждение при выполнении предложенных заданий. Здесь же учащимся необходимо было решать задачи, соответствующие темам уроков. Результаты представлены в таблице .

Таблица 4 – Результаты выполнения практической части

Ученик	Номер задачи	Количество баллов	Замечание	Общее количество баллов
Надежда Д	1	0,5		19
	2	0,5		
	3	2	Выполнены два условия их трех.Имеется ошибка в синтаксисе	
	4	2		
	5	1	Имеется несколько ошибок в синтаксисе	
	6	2	Выполнены только 7 из 9 пункты. Ошибки в синтаксисе	
	7	2		
	8	1	Выполнено только одно условие задачи. Ошибки в синтаксисе	
	9	3		
	10	2		
	11	2		
	12	1		
Глеб А	1	0,5		19
	2	0,5		
	3	2	Выполнено 2 условия из 3	
	4	1	Программа выполняет одно из двух условий	

	5	2		
	6	1	Выполнено только 6 условий из 9	
	7	2	Есть ошибки в синтаксисе	
	8	3		
	9	2	В целом программа соответствует условию задачи	
	10	2		
	11	2		
	12	1		
Анастасия Н.	1	0,5		17
	2	0,5		
	3	1	Выполнено одно условие из трех.	
	4	1	Программа выполняет одно из двух условий	
	5	2		
	6	2	Выполнено только 7 условий из 9	
	7	2	Есть ошибки в синтаксисе	
	8	2	Выполнено только 2 условия из 3	
	9	1	Выходные данные отличаются от ожидаемых	
	10	2		
	11	2		
	12	1		
Егор К	1	0,5		21
	2	0,5		
	3	2		
	4	2		
	5	2		
	6	3	Выполнено только 8 условий из 9	
	7	1	Много ошибок в синтаксисе	
	8	2	Выполнено 2 условия из 3	
	9	3		
	10	2		
	11	2		
	12	1		
Егор Ф	1	0,5		17
	2	0,5		
	3	2	Выполнено только 2 условия из 3	
	4	1	Программа выполняет одно из двух условий	
	5	2	Есть ошибки в синтаксисе	
	6	3	Выполнено только 8 условий из 9	
	7	2		
	8	1	Выполнено только 1 условия из 3	
	9	1	Выходные данные отличаются от ожидаемых	
	10	1	Ошибки в программе	
	11	2		
	12	1		
Виктор Г.	1	0,5		19
	2	0,5		
	3	1		
	4	2		
	5	2		
	6	1	Выполнено только 6 условий из 9	
	7	2	Есть ошибки в синтаксисе	
	8	2	Есть ошибки в синтаксисе	
	9	3		
	10	2	Есть ошибки в синтаксисе	
	11	2	Есть ошибки в синтаксисе	
	12	1		

Для наглядности общее количество баллов за практическую часть представлено на Рисунке 5.

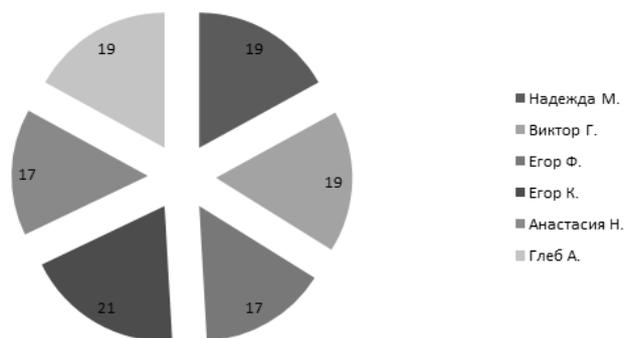


Рисунок 5 - Общее количество баллов за практическую часть

На третьем этапе педагогического эксперимента учащиеся выполняли контрольную работу в виде контрольного теста (Рис. 6), задач (Таблица 21) и группового решения задачи повышенной сложности

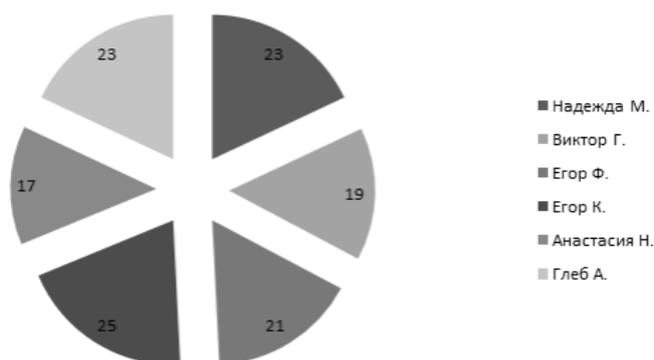


Рисунок 6 - Результаты тестирования

Таблица 5 – Результаты выполнения контрольных задач

Ученик	Номер задачи	Количество баллов	Замечание	Общее количество баллов
Надежда Д	1	3		7
	2	1	Используется переменная, не объявленная в разделе описания переменных	
	3	1		
	4	1		
	5	1		
Глеб А	1	3		7
	2	1	В цикле происходит выход за границу массива	
	3	1		
	4	1		
	5	1		
Анастасия Н.	1	2	Выполнено три из четырех необходимых действий. Одна строка является ошибочной.	6
	2	1	Отсутствует вывод ответа	
	3	1		
	4	1		

	5	1		
Егор К	1	3		7
	2	1	Используется переменная, не объявленная в разделе описания переменных	
	3	1		
	4	1		
	5	1		
Егор Ф	1	2	Одна строка является ошибочной	6
	2	1	Неверно расставлены операторные скобки.	
	3	1		
	4	1		
	5	1		
Виктор Г.	1	2	Выполнено три из четырех необходимых действий. Одна строка является ошибочной.	6
	2	1	В цикле происходит выход за границу массива	
	3	1		
	4	1		
	5	1		

Для наглядности общее количество баллов за выполнение контрольных задач представлено на Рисунке 7

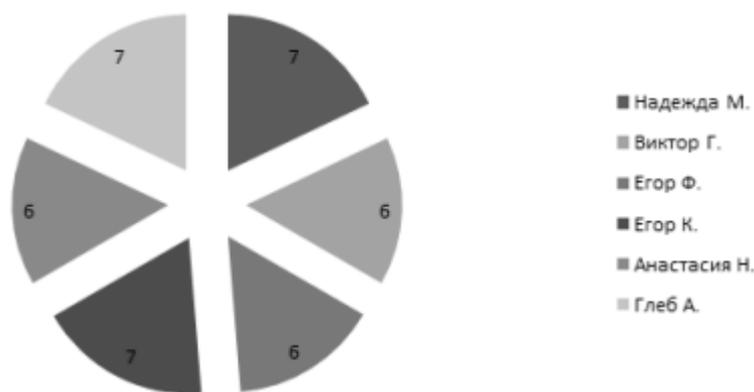


Рисунок 7 - Результаты выполнения контрольных задач

Групповая работа осуществлялась в виде совместного решения задачи повышенной сложности. Здесь оценивалось активное участие в обсуждении, за это учащимся добавлялось дополнительно 5 баллов, а также балл за правильность выполнения задачи. Если задача выполнена правильно, все учащиеся, принявшие участие в её решении, получают 6 баллов. Таким образом, мы можем предоставить отчет успеваемости по каждому учащемуся. Максимальный балл, который можно получить за прохождение курса – 80. Отчет представлен в Таблице 6

Таблица 6– Отчет об успеваемости учащихся

	Надежда М	Глеб А	Анастасия Н	Егор К	Егор Ф	Виктор Г
Активное участие в обсуждениях	4	5	2	5	5	3
Выполнение тематических заданий	11	9	7	11	10	8
Выполнение тематических задач	19	19	17	21	17	19
Контрольный тест	23	23	21	25	17	19
Выполнение контрольных задач	7	7	6	7	6	6
Групповая работа	6	6	6	6	6	6
Общий балл	70	69	59	75	61	61

Для наглядности результаты представлены в виде диаграммы (Рис. 8)

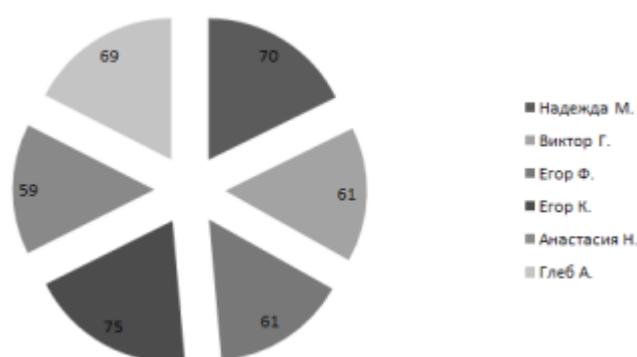


Рисунок 8 - Отчет об успеваемости учащихся

Опираясь на полученные результаты, мы можем сказать о том, что все учащиеся получают зачет по курсу. Отсюда мы можем сделать вывод о том, что цели и задачи курса были выполнены. Однако задачи были выполнены не все. Практически у всех учащихся имеются проблемы в написании синтаксиса языка Python, а также у некоторых учащихся имеются ошибки в использовании основных алгоритмических конструкций. Помимо этого, в ходе контрольного теста было обнаружено, что не все учащиеся умеют искать информацию, но в целом, несмотря на данные пробелы в знаниях, все обучающиеся получили по курсу балл выше среднего, это говорит о том, что курс можно использовать для обучения, но необходимо внести некоторые коррективы в его содержание.

## **2.4. Выводы по главе 2**

В рамках главы 2 мы выполнили две основные задачи: разработали структуру и содержание элективного курса на базе образовательного блога и апробировали его на практике.

На предпроектном этапе курса мы выявили преимущества образовательных блогов, в связи с чем был связан выбор среды для создания курса. При проектировании курса мы опирались на требования ФГОС СОО, требования к электронным курсам и требования к образовательному блогу, которые описаны в теоретической части исследования. Несмотря на то, что нами были учтены все перечисленные требования, в ходе апробации выяснились следующие недостатки курса: недостаточное количество теоретических заданий и недостаточное выделение главных теоретических моментов, в связи с чем учащиеся не в полной мере смогли ориентироваться по курсу, а также недостаточное количество часов на одно занятие.

Но всё же, не смотря на это, судя по достаточно высокой успеваемости учащихся, нам удалось выполнить цель курса. Помимо результатов нам об этом говорят ответы на ежедневные опросы, в ходе которых выяснилось, что учащиеся, в первую очередь, довольны своими результатами и большинство обучающихся планируют заняться углубленным изучением программирования на языке Python. Также обучающиеся оставили положительные отзывы о курсе, которые также подтверждают нашу гипотезу.

## **Заключение**

Разработанный нами элективный курс посвящен одному из бурно развивающихся и популярных в настоящее время сценарных языков программирования – Python.

В процессе проведения исследования нами была достигнута цель, подтверждена выдвинутая гипотеза, получены положительные результаты в решении поставленных задач:

- В результате изучения теоретических основ элективных курсов были изучены и проанализированы теоретические источники, психологопедагогическая литература, нормативно-правовые документы и выявлены психолого-педагогические основы преподавания элективных курсов и их место в средней школе, и методические требования к разработке и ведению элективных курсов;

- Проведен обзор существующих элективных курсов с применением ДОТ по программированию на языке Python, выявлены их достоинства и недостатки, которые позволили нам правильным образом разработать собственный элективный курс по данной тематике;

- На основе проведенного анализа теоретической составной исследования было разработано учебно-методическое обеспечение элективного курса «Программирование на языке Python»;

- Проведена апробация курса в рамках педагогической практики и проанализированы полученные результаты эксперимента.

Анализ результатов апробации подтвердил эффективность применения разработанного элективного курса. Полученные результаты показали, что данный курс способствовал формированию предъявляемых к нему результатов, в том числе личностных и межпредметных.

## Список литературы

1. Акимова И.В., Губанова О.М., Титова Н.В. Изучение элементов логического программирования в рамках факультатива по информатике // Современные проблемы науки и образования. 2018. № 4. [Электронный ресурс]. URL: <https://www.scienceeducation.ru/ru/article/view?id=27816> (дата обращения: 15.01.2020).
2. Босова Л.Л., Босова А.Ю. Информатика и ИКТ: учебник для 9 класса. М.: Бином, 2013.182 с
3. Жуковская, Е. П. Дидактические аспекты организации факультативов [Электронный ресурс] / Е. П. Жуковская // Открытый урок. Первое сентября. – Режим доступа: <http://festival.1september.ru/articles/594252/>
4. Зайцева, С. А. Современные информационные технологии в образовании [Электронный ресурс] / С. А. Зайцева, В. В. Иванов // ИНФОТЕК. – Режим доступа: <http://sgpu2004.narod.ru/infotek/infotek2.htm>
5. Закон РФ «Об образовании» [Электронный ресурс] : федер. Закон от 10.01.2003 №11-ФЗ утратил силу с 1.09.2013. // Справочная правовая система «Электронный фонд правовой и нормативно-технической документации». – Режим доступа: <http://docs.cntd.ru/document/9003751>
6. Запрудский, Н. И. Моделирование и проектирование авторских дидактических систем: пособие для учителя / Н. И. Запрудский. – Минск, 2008. – 288 с.
7. Кочеткова О.А., Пудовкина Ю.Н. Методические аспекты к изучению программирования на языке ассемблера в подготовке бакалавров направления «Педагогическое образование» по профилю «Информатика» // Современное образование: научные подходы, опыт, проблемы, перспективы: сборник статей XIV Международной научно-практической конференции. (Пенза, 18-19 апреля 2018 г.). Пенза: Издательство Пензенского государственного университета, 2018. С. 243-245

8. Кочеткова О.А., Долгополов И.В. Разработка электронных средств учебного назначения по курсу «Программирование на языке Python» // Университетское образование (МКУО-2016): сборник статей XX Международной научно-методической конференции (Пенза, 7-8апреля 2016 г.). Пенза: Издательство Пензенского государственного университета, 2016. С.104-105
9. Кириенко Д.П. Язык программирования Python – современный язык для обучения // Всероссийский съезд учителей информатики. М.: МГУ имени М.В. Ломоносова, 2011. С.358-359
10. Лесневский А.С. Объектно-ориентированное программирование для начинающих. М.: Бином. Лаборатория знаний, 2010. 232 с
11. Лутц, М. Изучаем Python, 4-е издание. – Пер. с англ. : учебник / М. Лутц. – Спб. : Символ-Плюс, 2011. – 1280 с.
12. Марк Саммерфилд. Программирование на Python 3. - М.: ДМК Пресс, 2014
13. Мэтиз, Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – Пер. с англ. : учебник / Э. Мэтиз. – Спб. : Питер, 2017. – 496 с.
14. Об утверждении Концепции профильного обучения на старшей ступени общего образования [Электронный ресурс] : Приказ Минобразования РФ от 18.07.2002 № 2783 // Справочная правовая система «Гарант». – Режим доступа: <http://www.garant.ru/>
15. Павленко, В. Интерактивный учебник языка Питон [Электронный ресурс] / В. Павленко, В. Соломатин // ПИТОНТЮТОР. – 2014. – Режим доступа: <http://pythontutor.ru/>
16. Постановление ЦК КПСС, Совмина СССР "О мерах дальнейшего улучшения работы средней общеобразовательной школы" [Электронный ресурс] : от 10.11.1966 N 874. // Справочная правовая система «КонсультантПлюс». – Режим доступа: <http://www.consultant.ru>

17. «Программирование: типовые задачи, алгоритмы, методы» / Д. М. Златопольский - М.: БИНОМ. Лаборатория знаний, 2012.
18. Семакин И.Г., Шеина Т.Ю., Шестакова И.Г. Информатика. Углубленный уровень: учебник для 10 кл.: в 2 ч. М.: Бином, 2014. 184 с
19. Сборник задач по программированию» / Д. М. Златопольский - СПб.: БХВ-Петербург, 2011.
20. «Задачи по программированию» / под ред. С. М. Окулова - М.: БИНОМ. Лаборатория знаний, 2014.
21. Окулов С.М. Основы программирования. М.: Бином, 2012. 336 с
22. Профильное обучение: педагогическая система и управление. Книга 1. Система профильного обучения старшеклассников / Т. П. Афанасьева, Н. В. Немова ; под общ. ред. Н. В. Немовой. – М. : АПК и ПРО, 2004. – 73 с.
23. Российская педагогическая энциклопедия : в 2 т. / редкол. : В. В. Давыдов. – М. : Научное издательство «Большая российская энциклопедия», 1994.
24. Создание образовательного блога [Электронный ресурс] / Adu.by. – Режим доступа: <http://e-asveta.adu.by/index.php/sozdanie-obrazovatelnogobloga>
25. Угринович Н.Д. Информатика и ИКТ: учебник для 9 класса. М.: Бином, 2012. 295 с.
26. Федеральный государственный образовательный стандарт
27. среднего (полного) общего образования [Электронный ресурс] : от 17 мая 2012 г. № 413 // Справочная правовая система «Минобрнауки.рф». – Режим доступа: <http://xn--80abucjiibhv9a.xn--p1ai/>
28. Федеральный закон «Об образовании в Российской Федерации» [Электронный ресурс] : федер. закон от 29.12.2012 N 273-ФЗ. // Справочная правовая система «КонсультантПлюс». – Режим доступа: <http://www.consultant.ru/>
29. Самоучитель для начинающих python 3 <https://pythonworld.ru/samouchitel-pvthon>(дата обращения 24.01.2020)

30. Интерактивный учебник <http://pythontutor.ru/>(дата обращения 24.01.2020)
31. Сайт олимпиад по информатике в Санкт-Петербурге <http://neerc.ifmo.ru/school/io/index.html> (дата обращения 24.01.2020)
32. Hlomborg, B. The Evolution, Principles and Practices of Distance Education. Oldenburg / B. Hlomborg. – Bibliotheks-und Informationssystem der Universitat Oldenburg, 2015. – 171 p.
33. Quotes about Python [Electronic resource] / Python.org. – URL: <https://www.python.org/>