

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования и науки Удмуртской Республики

Муниципальное образование

"Муниципальный округ Камбарский район Удмуртской Республики"

МБОУ "КООШ"

РАССМОТРЕНО
на педсовете протокол
№7 от «30» августа 2023 г.

УТВЕРЖДЕНО
приказом директора №46 от
«31» августа 2023 г.

РАБОЧАЯ ПРОГРАММА

учебного предмета «Основы программирования»

для обучающихся 8–9 классов

с. Камское 2023г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа курса внеурочной деятельности «Основы программирования» разработана в соответствии с требованиями Федерального государственного образовательного стандарта общего образования (приказ министерства образования и науки от 17 декабря 2010 г. №1897), Законом

Российской Федерации от 29.12.2012 года № 273-ФЗ «Об образовании в Российской Федерации» (с последующими изменениями и дополнениями), основной образовательной программы основного общего образования и положением о рабочей программе учебного предмета, курсов, дисциплин (модулей) в Камской основной общеобразовательной школы.

Рабочая программа конкретизирует содержание предметных тем, дает примерное распределение учебных часов по разделам курса и рекомендуемую последовательность изучения тем и разделов учебного курса с учетом межпредметных и внутрипредметных связей, логики учебного процесса, возрастных особенностей учащихся.

Программа определяет цели, планируемые результаты, содержание и организацию процесса формирования универсальных учебных действий через проектную деятельность.

Цели изучения курса «Основы программирования»:

- формирование интереса и положительной мотивации школьников к технологическому направлению обучения
- знакомство учащихся с историей программирования
- изучение обучающимися языка программирования Python
- формирование у учащихся теоретических и практических знаний в области программирования
- формирование информационной основы и персонального опыта, необходимых для определения обучающимся направлений своего дальнейшего образования в контексте построения жизненных планов, в первую очередь касающихся сферы и содержания будущей профессиональной деятельности.
- формирование умений фиксировать информацию об окружающем мире;
- формирование умений искать, анализировать, критически оценивать, отбирать информацию;
- формирование умения проектировать объекты и процессы, планировать свои действия; создавать, реализовывать и корректировать планы.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА

В соответствии с учебным планом школы на 2023-2024 учебный год в 8-9 классах на изучение курса «Основы программирования» отводится 51 час (8 класс-1 час, 9 класс-0,5 часа).

Планируемые результаты освоения обучающимися основной образовательной программы основного общего образования уточняют и конкретизируют общее понимание личностных, метапредметных и предметных результатов как с позиции организации их достижения в образовательном процессе, так и с позиции оценки достижения этих результатов.

В результате освоения курса информатики в 8-9 классах *учащиеся получают представление:*

- об алгоритмах обработки информации, их свойствах, основных алгоритмических конструкциях; о способах разработки и программной реализации алгоритмов;
- о программном принципе работы компьютера – универсального устройства обработки информации; о направлениях развития компьютерной техники;
- о принципах организации файловой системы, основных возможностях графического интерфейса и правилах организации индивидуального информационного пространства;
- о назначении и функциях программного обеспечения компьютера; об основных средствах и методах обработки числовой, текстовой, графической и мультимедийной информации; о технологиях обработки информационных массивов с использованием электронной таблицы или базы данных;
- о компьютерных сетях распространения и обмена информацией, об использовании информационных ресурсов общества с соблюдением соответствующих правовых и этических норм;

- о требованиях техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий.

Учащиеся будут уметь:

- формально исполнять алгоритмы для конкретного исполнителя с фиксированным набором команд, обрабатывающие цепочки символов или списки, записанные на естественном и алгоритмическом языках;
- формально исполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов, простых и табличных величин;
- использовать стандартные алгоритмические конструкции для построения алгоритмов для формальных исполнителей;
- составлять линейные алгоритмы управления исполнителями и записывать их на выбранном алгоритмическом языке (языке программирования);
- создавать алгоритмы для решения несложных задач, используя конструкции ветвления (в том числе с логическими связками при задании условий) и повторения, вспомогательные алгоритмы и простые величины;
- создавать и выполнять программы для решения несложных алгоритмических задач в выбранной среде программирования;
- оперировать информационными объектами, используя графический интерфейс: открывать, именовать, сохранять объекты, архивировать и разархивировать информацию, пользоваться меню и окнами, справочной системой; предпринимать меры антивирусной безопасности;
- составлять и записывать программы, используя известные алгоритмические конструкции;
- распознавать необходимость применения той или иной алгоритмической конструкции при решении задачи;
- использовать процедуры и функции при решении задач;
- проводить отладку и тестирование программ.

Личностные результаты – это сформировавшаяся в образовательном процессе система ценностных отношений учащихся к себе, другим участникам образовательного процесса, самому образовательному процессу, объектам познания, результатам образовательной деятельности. Основными личностными результатами, формируемыми при изучении программирования в средней школе, являются:

1. готовность и способность учащихся к саморазвитию и реализации творческого потенциала в предметно-продуктивной деятельности за счет развития их образного, алгоритмического и логического мышления;
2. способность связать учебное содержание с собственным жизненным опытом и личными смыслами, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества;
3. готовность к самостоятельным поступкам и действиям, принятию ответственности за их результаты; готовность к осуществлению индивидуальной и коллективной информационной деятельности;
4. способность к профессиональному и личностному самоопределению;
5. способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ.

Метапредметные результаты – освоенные обучающимися на базе одного, нескольких или всех учебных предметов способы деятельности, применимые как в рамках образовательного процесса, так и в других жизненных ситуациях. Основными метапредметными результатами, формируемыми при изучении программирования в основной средней школе, являются:

1. владение основными универсальными умениями информационного характера: постановка и формулирование проблемы;

2. поиск и выделение необходимой информации, применение методов информационного поиска;
3. структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий;
4. самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера;
5. владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
6. владение основами продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми: умение правильно, четко и однозначно сформулировать мысль в понятной собеседнику форме;
7. умение осуществлять в коллективе совместную информационную деятельность, в частности при выполнении проекта;
8. умение выступать перед аудиторией, представляя ей результаты своей работы с помощью средств ИКТ;

Предметные результаты включают в себя: освоенные обучающимися в ходе изучения учебного предмета умения специфические для данной предметной области, виды деятельности по получению нового знания в рамках учебного предмета, его преобразованию и применению в учебных, учебно-проектных и социально-проектных ситуациях, формирование научного типа мышления, научных представлений о ключевых теориях, типах и видах отношений, владение научной терминологией, ключевыми понятиями, методами и приемами. В соответствии с федеральным государственным образовательным стандартом общего образования основные предметные результаты изучения информатики в основной школе отражают:

8 класс	<ul style="list-style-type: none"> • основам программирования на языке Python; • общей структуре языка программирования Python и его синтаксису; • выводить на экран текст; • составлять и решать арифметические выражения в Python; • запускать генератор случайных чисел в Python; • понимать смысл понятия «алгоритм» и широту сферы его применения; анализировать предлагаемые последовательности команд на предмет наличия у них таких свойств алгоритма как дискретность, детерминированность, понятность, результативность, массовость; • оперировать алгоритмическими конструкциями «следование», «ветвление», «цикл» в Python (подбирать алгоритмическую конструкцию, соответствующую той или иной ситуации; переходить от записи алгоритмической конструкции на алгоритмическом языке к блок-схеме и обратно); • понимать термины в среде Python «исполнитель», «формальный исполнитель», 	<ul style="list-style-type: none"> • исполнять алгоритмы, содержащие ветвления и повторения, для формального исполнителя с заданной системой команд; • составлять все возможные алгоритмы фиксированной длины для формального исполнителя с заданной системой команд; • определять количество линейных алгоритмов, обеспечивающих решение поставленной задачи, которые могут быть составлены для формального исполнителя с заданной системой команд; • подсчитывать количество тех или иных символов в цепочке символов, являющейся результатом работы алгоритма; • импортировать различные виды библиотек в Python; • запускать графическое окно в Python, и указывать размеры окна; • рисовать по пикселям в графическом окне, с помощью системы координат; • рисовать цветные фигуры с помощью
---------	---	---

	<p>«среда исполнителя», «система команд исполнителя» и др.; понимать ограничения, накладываемые средой исполнителя и системой команд, на круг задач, решаемых исполнителем;</p> <ul style="list-style-type: none"> • составлять линейные алгоритмы в Python, число команд в которых не превышает заданное; • ученик научится исполнять записанный на естественном языке алгоритм, обрабатывающий цепочки символов. • рисовать геометрические фигуры разными цветами; • исполнять линейные алгоритмы, записанные на алгоритмическом языке. • исполнять алгоритмы с ветвлениями, записанные на алгоритмическом языке; • понимать правила записи и выполнения алгоритмов, содержащих цикл с параметром или цикл с условием в Python; • составлять бесконечные циклы; • определять значения переменных после исполнения простейших циклических алгоритмов, записанных на алгоритмическом языке; • разрабатывать и записывать на языке программирования Python короткие алгоритмы, содержащие базовые алгоритмические конструкции; • правила определения типа переменной и ее описания в Python; • правила записи операторов языка, порядок выполнения операций, стандартные и пользовательские функции, записи и выполнения арифметических и логических функций; • строить циклы с помощью компьютерной графики в Python; • правила записи на языке программирования Python условного оператора, оператора выбора, цикла с предусловием, цикла с постусловием; • составлять и выводить на экран простейшую анимацию; • основные приёмы отладки и тестирования программ в Python; • формализовать поставленную задачу; • применять полученные знания к различным предметным областям; 	<p>графики в Python;</p> <ul style="list-style-type: none"> • по данному алгоритму определять, для решения какой задачи он предназначен; • исполнять записанные на алгоритмическом языке циклические алгоритмы обработки одномерного массива чисел (суммирование всех элементов массива; суммирование элементов массива с определёнными индексами; суммирование элементов массива, с заданными свойствами; определение количества элементов массива с заданными свойствами; поиск наибольшего наименьшего элементов массива и др.); • разрабатывать в Python короткие алгоритмы, содержащие базовые алгоритмические конструкции; • разрабатывать и записывать на языке программирования эффективные алгоритмы, содержащие базовые алгоритмические конструкции. • составлять и записывать программы в Python, используя известные алгоритмические конструкции; • распознавать необходимость применения той или иной алгоритмической конструкции при решении задачи; • использовать процедуры и функции при решении задач; • правильно записывать символы и ключевые слова языка программирования; • записывать числовые и логические константы, задавать и описывать переменные; • описывать тип и размерность массивов; • составлять математическую модель, алгоритм и программу для решения простых задач; • понимать листинг, корректировать программу; • выводить результаты на дисплей; • решать олимпиадные задачи среднего уровня. • общие принципы построения и использования языка программирования Python, их классификацию; • современные интегрированные среды
--	---	---

	<ul style="list-style-type: none"> • составлять и оформлять программы на языке программирования Python; • тестировать и отлаживать программы в Python; • определять результат выполнения алгоритма при заданных исходных данных; • узнавать изученные алгоритмы обработки чисел и числовых последовательностей; • создавать на их основе несложные программы анализа данных; • читать и понимать несложные программы, написанные в Python; • выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных; • работать с таблицами, обрабатывать большие массивы данных и проводить математические операции больших объемов в Python; • разрабатывать программы, составляя этапы решения задач и проектирования их каркаса и подпрограмм; 	<p>разработки программ;</p> <ul style="list-style-type: none"> • понимать важность дискретизации данных; • выполнять сложные сортировки данных по возрастанию и убыванию; • использовать знания о постановках задач поиска и сортировки; их роли при решении задач анализа данных; • использовать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; • использовать основные управляющие конструкции последовательного программирования и библиотеки прикладных программ; • выполнять свои созданные программы в Python;
9 класс	<ul style="list-style-type: none"> • использовать процедуры и функции при решении задач; • различать локальные и глобальные переменные; • составлять задачи используя процедуры с параметрами; • понимать правила записи и выполнения алгоритмов, содержащих цикл с параметром или цикл с условием в Python; • определять значения переменных после исполнения сложных циклических алгоритмов, записанных на языке программирования Python; • разрабатывать и записывать на языке программирования Python большие алгоритмы, содержащие сложные алгоритмические конструкции. • основные приёмы отладки и тестирования программ. • формализовать поставленную задачу; • применять полученные знания к различным предметным областям; • составлять и оформлять программы на языках программирования; • тестировать и отлаживать программы; • определять результат выполнения алгоритма при заданных исходных данных Python; 	<ul style="list-style-type: none"> • исполнять алгоритмы, содержащие ветвления и повторения, для формального исполнителя с заданной системой команд; • составлять все возможные алгоритмы фиксированной длины для формального исполнителя с заданной системой команд; • по данному алгоритму определять, для решения какой задачи он предназначен; • исполнять записанные на алгоритмическом языке циклические алгоритмы обработки одномерного массива чисел (суммирование всех элементов массива; суммирование элементов массива с определёнными индексами; суммирование элементов массива, с заданными свойствами; определение количества элементов массива с заданными свойствами; поиск наибольшего наименьшего элементов массива и др.); • разрабатывать в среде формального исполнителя Python алгоритмы, содержащие алгоритмические конструкции;

<ul style="list-style-type: none"> • узнавать изученные алгоритмы обработки чисел и числовых последовательностей; • создавать на их основе сложные программы анализа данных; • читать и понимать сложные программы, написанные на языке Python высокого уровня; • выполнять пошагово сложные алгоритмы управления исполнителями и анализа числовых и текстовых данных; • создавать на алгоритмическом языке Python программы для решения типовых задач базового уровня из различных предметных областей с использованием основных алгоритмических конструкций; • использовать готовые прикладные компьютерные программы в соответствии с типом решаемых задач и по выбранной специализации; • использовать компьютерно-математические модели для анализа соответствующих объектов и процессов, в том числе оценивать числовые параметры моделируемых объектов и процессов, а также интерпретировать результаты, получаемые в ходе моделирования реальных процессов; • представлять результаты математического моделирования в наглядном виде, готовить полученные данные для публикации; • правильно составлять текстовые документы в соответствии с эстетическими нормами и оптимальным количеством необходимого текста; • работать с таблицами, обрабатывать большие массивы данных и проводить математические операции больших объемов; • выполнять поиск в символьных строках, добавлять или удалять строки; • составлять матрицы; • выводить на экран случайные и квадратные матрицы; • перебирать элементы матрицы и сортировать по столбцам и строкам в Python; • презентовать работу, используя соответствующие редакторы, не перегружать лишней информацией и правильно составлять структуру материала; • разрабатывать программы, составляя этапы решения задач и проектирования их каркаса и подпрограмм; 	<ul style="list-style-type: none"> • разрабатывать и записывать на языке программирования эффективные алгоритмы, содержащие сложные алгоритмические конструкции. • составлять и записывать программы, используя известные алгоритмические конструкции в Python; • распознавать необходимость применения той или иной алгоритмической конструкции при решении задачи; • использовать процедуры и функции при решении задач; • правильно записывать символы и ключевые слова языка программирования; • описывать тип и размерность массивов; • составлять математическую модель, алгоритм и программу для решения простых задач; • понимать листинг, корректировать программу; • решать олимпиадные задачи сложнейшего уровня. • проводить отладку и тестирование программ. • общие принципы построения и использования языков программирования, их классификацию; • современные интегрированные среды разработки программ; • понимать важность дискретизации данных; • использовать знания о постановках задач поиска и сортировки; их роли при решении задач анализа данных; • использовать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; • использовать основные управляющие конструкции последовательного программирования и библиотеки прикладных программ; • выполнять созданные программы; • разрабатывать и использовать компьютерно-математические модели; оценивать числовые параметры моделируемых объектов и процессов; интерпретировать результаты,
--	---

	<ul style="list-style-type: none"> • находить и исправлять ошибки в программах, написанных на языке Python; 	<p>получаемые в ходе моделирования реальных процессов; анализировать готовые модели на предмет соответствия реальному объекту или процессу; – применять базы данных и справочные системы при решении задач, возникающих в ходе учебной деятельности и вне ее; создавать учебные многотабличные базы данных; – критически оценивать информацию, полученную из сети Интернет.</p>
--	--	---

1. Система оценки достижения планируемых результатов освоения основной образовательной программы среднего общего образования

Контроль и оценка результатов освоения учебной дисциплины осуществляется учителем в процессе проведения практических работ за компьютером, тестирования, а также выполнения обучающимися индивидуальных заданий, проектов, исследований.

Критерии и нормы оценки практического задания

Оценка «5»:

а) выполнил работу в полном объеме с соблюдением необходимой последовательности ее проведения;

б) самостоятельно и рационально выбрал и загрузил необходимое программное обеспечение, все задания выполнил в условиях и режимах, обеспечивающих получение результатов и выводов с наибольшей точностью;

в) в представленном отчете правильно и аккуратно выполнил все записи, таблицы, рисунки, чертежи, графики, вычисления и сделал выводы;

Оценка «4»: работа выполнена правильно с учетом 2-3 несущественных ошибок, исправленных самостоятельно по требованию учителя.

Оценка «3»: работа выполнена правильно не менее чем на половину или допущена существенная ошибка.

Оценка «2»: допущены две (и более) существенные ошибки в ходе работы, которые учащийся не может исправить даже по требованию учителя.

Критерии и нормы оценки тестовых работ

Оценка 5 - 66-100% правильных ответов на вопросы;

Оценка 4 - 31-65% правильных ответов на вопросы;

Оценка 3 - 21-30% правильных ответов на вопросы;

Оценка 2 - 0-20% правильных ответов на вопросы.

Критерии оценки проектной и исследовательской работы

Разрабатываются с учётом целей и задач проектной деятельности на данном этапе образования. Индивидуальный проект целесообразно оценивать по следующим критериям:

1. Способность к самостоятельному приобретению знаний и решению проблем, проявляющаяся в умении поставить проблему и выбрать адекватные способы её решения, включая поиск и обработку информации, формулировку выводов и/или обоснование и реализацию/апробацию принятого решения, обоснование и создание прогноза, модели, макета, объекта, творческого решения и т. п. Данный критерий в целом включает оценку сформированности познавательных учебных действий.

2. Сформированность предметных знаний и способов действий, проявляющаяся в умении раскрыть содержание работы, грамотно и обоснованно в соответствии с рассматриваемой проблемой/темой использовать имеющиеся знания и способы действий.

3. Сформированность регулятивных действий, проявляющаяся в умении самостоятельно планировать и управлять своей познавательной деятельностью во времени, использовать ресурсные возможности для достижения целей, осуществлять выбор конструктивных стратегий в трудных ситуациях.

4. Сформированность коммуникативных действий, проявляющаяся в умении ясно изложить и оформить выполненную работу, представить её результаты, аргументированно ответить на вопросы.

Максимальная оценка по каждому критерию не должна превышать 3 баллов. При таком подходе достижение базового уровня (отметка «удовлетворительно») соответствует получению 4 первичных баллов (по одному баллу за каждый из четырёх критериев), а достижение повышенных уровней соответствует получению 7—9 первичных баллов (отметка «хорошо») или 10—12 первичных баллов (отметка «отлично»).

Перечень ошибок

Грубые ошибки

1. Незнание определений основных понятий, правил, основных положений теории, приёмов составления алгоритмов.

2. Неумение применять знания для решения задач и объяснения блок-схем алгоритмов, неправильно сформулированные вопросы задачи или неверное объяснение хода её решения, незнание приёмов решения задач, аналогичных ранее решённых в классе; ошибки, показывающие неправильное понимание условия задачи или неправильное истолкование решения, неверное применение операторов в программах, их незнание.

3. Неумение читать программы, алгоритмы, блок-схемы.

4. Неумение подготовить к работе ПК, запустить программу, отладить её, получить результаты и объяснить их.

5. Небрежное отношение к ПК.

6. Нарушение требований правил безопасного труда при работе на ПК.

Негрубые ошибки

1. Неточность формулировок, определений, понятий, вызванные неполнотой охвата основных признаков определяемого понятия; ошибки синтаксического характера.

2. Пропуск или неточное написание тестов в операторах ввода-вывода.

3. Нерациональный выбор решения задачи.

Недочёты

1. Нерациональные записи в алгоритмах, преобразований и решений задач.

2. Арифметические ошибки в вычислениях, если эти ошибки грубо не искажают реальность полученного результата.

3. Небрежное выполнение записей, блок-схем.

4. Пунктуационные ошибки.

ОСНОВНОЕ СОДЕРЖАНИЕ КУРСА

Курс по информатике «Основы программирования» для обучающихся 8,9 класса рассчитан на 64 часов (1 час в неделю). Курс ориентирован на предпрофильную подготовку и направлен на изучение языка программирования в 8 и 9 классе язык программирования Python. Программа построена с учетом принципов системности, научности и доступности, позволяет получить необходимые знания по основам программирования на этих языках. Изучение данного курса имеет важное значение для развития мышления учащихся. В современной психологии отмечается значительное влияние изучения информатики и использования компьютеров в обучении на развитие теоретического, творческого мышления, а также формирование нового типа мышления, так называемого операционного мышления, направленного на выбор оптимальных решений. Алгоритмические знания и умения необходимы для изучения других школьных предметов: математики, физики, химии и даже отдельных аспектов гуманитарных и естественных предметов.

8-й класс

Глава 1. Первые программы

Что такое программа? Самая простая программа. Вывод текста на экран. Ввод данных в одной строке. Диалоговые программы. Переменные. Сумма чисел.

Глава 2. Обработка целых и вещественных чисел

Арифметические выражения. Деление нацело и остаток от деления. Что такое вещественное число? Операции с вещественными числами.

Глава 3. Случайные и псевдослучайные числа

Пишем свой генератор случайных чисел. Генератор случайных чисел в Python.

Глава 4. Ветвления

Условный оператор. Неполная форма условного оператора.

Глава 5. Сложные условия

Операции «И», «ИЛИ», «НЕ». Порядок выполнения операций.

Глава 6. Циклы с условием

Как организовать цикл? Циклы с предусловием. Алгоритм Евклида. Обработка потока данных. Бесконечные циклы.

Глава 7. Компьютерная графика

Что такое компьютерная графика? Графика в Python. Система координат. Управляем пикселями. Рисуем линии, прямоугольники, окружность. Изменение координат.

Глава 8. Процедуры

Зачем нужны процедуры? Процедуры с параметрами.

Глава 9. Циклы по переменной

Делай N раз. Шаг изменения переменной цикла. Циклы в компьютерной графике. Узоры. Вложенные циклы. Рефакторинг. Штриховка.

Глава 10. Анимация

Принципы анимации. Начальное положение. Анимация движения. Обработка нажатия клавиши

Глава 11. Функции

Что такое функция? Примеры функций. Рекурсивные функции.

9-й класс

Глава 1. Процедуры

Программы: процедуры и функции. Простая процедура. Процедуры с параметрами. Локальные и глобальные переменные.

Глава 2. Массивы

Что такое массив? Массивы в языке Python. Создание массива. Обращение к элементу массива. Перебор элементов массива. Генераторы. Вывод массива. Вывод массива с

клавиатуры. Заполнение массива случайными числами. Максимальный элемент, удовлетворяющий условию. Особенности копирования списков в Python. Линейный поиск. Поиск максимального элемента в массиве.

Сумма элементов массива. Подсчёт элементов массива, удовлетворяющих условию.

Глава 3. Символьные строки

Что такое символьная строка? Сравнение строк. Сложение и умножение. Обращение к символам. Перебор всех символов. Срезы. Удаление и вставка. Встроенные методы. Символьные строки в функциях. Поиск в символьных строках. Замена.

2. ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ ПРЕДМЕТА (8-9 классы «Основы программирования»)

Классы – 8 класс

Количество часов – 34 часов в год

Количество часов в неделю – 1 час

Основное содержание по темам	Количество часов	Характеристика деятельности ученика
Глава 1. Первые программы		
Что такое программа? Самая простая программа. Вывод текста на экран. Ввод данных в одной строке. Диалоговые программы. Переменные. Сумма чисел.	3 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • знать историю возникновения языка; структуру программы на Python; • операторы ввода, вывода, правила записи выражений, линейный вычислительный алгоритм; • перечень основных операторов языка Python; • синтаксис этих операторов; • детали процесса исполнения каждого из операторов; • описывать словесно работу каждого из рассмотренных операторов. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • составлять простейшие алгоритмы; проверять и доказывать выполнение свойств конкретного алгоритма; различать алгоритмические конструкции; запускать программу; • находить нужные кнопки и пункты меню; • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 2. Обработка целых и вещественных чисел		
Арифметические выражения. Деление нацело и остаток от деления. Что такое вещественное число? Операции с вещественными числами.	5 часов	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • понимать программы обработки числовой и символьной информации, требующие вложения

		<p>одного и более основных операторов;</p> <ul style="list-style-type: none"> • работа со встроенным редактором системы Python; • компиляция, отладка программы; • исполнение программы, просмотр результата. • определять остаток от деления на 10, 100, 100 <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • применять режимы выполнения программы: пошаговый, до курсора, полностью; • исполнять готовые алгоритмы для конкретных исходных данных; • выполнять деление нацело и с остатком; • применять div и mod; • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 3. Случайные и псевдослучайные числа		
Пишем свой генератор случайных чисел. Генератор случайных чисел в Python.	2 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • знать понятия «случайные числа», «псевдослучайные числа», «генератор случайных чисел» <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • написать программу с псевдослучайными и случайными числами • написать программу, которая выводит наугад первые 5 выигрышных номеров. • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 4. Ветвления		
Условный оператор. Неполная форма условного оператора.	2 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • знать и понимать «условный оператор: полная и неполная форма», «вложенный условный оператор», «логические переменные», «экспертная система» <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • написать программы с условным оператором полной и неполной формы

		<ul style="list-style-type: none"> использовать основные алгоритмические приемы при решении математических задач решать простые, сложные и нестандартные задачи
Глава 5. Сложные условия		
Операции «И», «ИЛИ», «НЕ». Порядок выполнения операций.	2 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> понимать логические операции И,ИЛИ,НЕ <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> написать программы с равносильными условиями применять на практике порядок выполнения логических операций использовать основные алгоритмические приемы при решении математических задач решать простые, сложные и нестандартные задачи
Глава 6. Циклы с условием		
Как организовать цикл? Циклы с предусловием. Алгоритм Евклида. Обработка потока данных. Бесконечные циклы.	3 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> разрабатывать программы обработки числовой и символьной информации, требующие вложения одного и более основных операторов; работа со встроенным редактором системы Python; компиляция, отладка программы; исполнение программы, просмотр результата. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя арифметических действий; строить цепочки команд, дающих нужный результат при конкретных исходных данных для исполнителя, преобразующего строки символов; строить арифметические, строковые, логические выражения и вычислять их значения; использовать основные алгоритмические приемы при решении математических задач решать простые, сложные и нестандартные задачи
Глава 7. Компьютерная графика		
Что такое компьютерная графика? Графика в Python. Система координат. Управляем пикселями.	5 часов	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> знать определения «графический

Рисуем линии, прямоугольники, окружность. Изменение координат.		<p>режим», «холст», «пиксель», «координаты», «модуль»,</p> <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • писать программы которые строят рисунки : линия, пунктирная линия, прямоугольник, многоугольник, круг, треугольник, домик, корона и другие • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 8. Процедуры		
Зачем нужны процедуры? Процедуры с параметрами.	2 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • определять тип подпрограммы • различать понятия «включать», «вызвать» процедуру • понимать, различать, объявлять локальные и глобальные переменные <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • уметь создавать программы с процедурами • уметь проводить отладку программы, находить в них алгоритмические ошибки • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 9. Циклы по переменной		
Делай N раз. Шаг изменения переменной цикла. Циклы в компьютерной графике. Узоры. Вложенные циклы. Рефакторинг. Штриховка.	5 часов	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • знать и различать понятия «цикл по переменной» и «переменная цикла» <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • выполнять цикл по переменной • сравнивать цикл по переменной и цикл с условием • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 10. Анимация		
Принципы анимации. Начальное положение. Анимация движения.	2 часа	<p><i>Аналитическая деятельность:</i></p>

Обработка нажатия клавиши.		<ul style="list-style-type: none"> • знать принципы анимации • различать понятия «кадр» и «смена кадров», «событие» и «обработчик события» <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • написать программу, в которой через поле движутся два квадрата навстречу друг другу • доработать предыдущую программу так, чтобы квадраты отталкивались от границ поля и начинали движение в противоположном направлении • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи
Глава 11. Функции		
Что такое функция? Примеры функций. Рекурсивные функции.	4 часа	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • понимать различия функции и процедуры; • описывать формальные параметры; • понимать правила назначения имени для переменной, которое должно совпадать с именем функции; <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • уметь вызывать функцию; • внутри функции объявлять и вызывать переменные; • выполнять одинаковые расчеты в различных местах программы; • создавать общедоступные библиотеки функций; • создавать функции, которые будут выполнять простые арифметические действия. • использовать основные алгоритмические приемы при решении математических задач • решать простые, сложные и нестандартные задачи

Классы – 9 класс

Количество часов – 17 часа в год

Количество часов в неделю – 0,5 часа

Основное содержание по темам	Количество часов	Характеристика деятельности ученика
Глава 1. Процедуры		
Программы: процедуры и функции. Простая процедура. Процедуры с параметрами. Локальные и глобальные переменные.	(5 часов)	<i>Аналитическая деятельность:</i> <ul style="list-style-type: none">• определять тип подпрограммы• различать понятия «включать», «вызвать» процедуру• понимать, различать, объявлять локальные и глобальные переменные <i>Практическая деятельность:</i> <ul style="list-style-type: none">• уметь создавать программы с процедурами• уметь проводить отладку программы, находить в них алгоритмические ошибки• использовать основные алгоритмические приемы при решении математических задач• решать простые, сложные и нестандартные задачи
Глава 2. Массивы		
Что такое массив? Массивы в языке Python.Создание массива. Обращение к элементу массива.Перебор элементов массива. Генераторы.Вывод массива. Вывод массива с клавиатуры.Заполнение массива случайными числами.Максимальный элемент, удовлетворяющий условию.Особенности копирования списков в Python.Линейный поиск. Поиск максимального элемента в массиве. Сумма элементов массива. Подсчёт элементов массива, удовлетворяющих условию.	9 (часов)	<i>Аналитическая деятельность:</i> <ul style="list-style-type: none">• понимать, что такое массив,• как он составляется и заполняется;• знать команды вывода массива с клавиатуры;• определять максимальный и минимальный элемент массива;• знать основные принципы сортировки массива по возрастанию и убыванию. <i>Практическая деятельность:</i> <ul style="list-style-type: none">• разрабатывать программы для обработки одномерного массива: нахождение минимального (максимального) значения в данном массиве;• подсчёт количества элементов массива, удовлетворяющих некоторому условию; нахождение суммы всех элементов массива;• нахождение количества и суммы всех четных элементов в массиве;• сортировка элементов массива методом «Пузырька» и пр.

Сроки изучения учебного материала	№ урока	Раздел, тема, урок. Практические, контрольные работы и другие виды работ	Количество часов
		<ul style="list-style-type: none"> использовать основные алгоритмические приемы при решении математических задач решать простые, сложные и нестандартные задачи 	
Глава 3. Символьные строки			
Что такое символьная строка? Сравнение строк. Сложение и умножение. Обращение к символам.	3 часов)	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> знать последовательность символов <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> сравнивать, складывать, умножать, удалять, вставлять, осуществлять поиск и перебор всех символов уметь создавать простые программы линейной, разветвляющейся и циклической структуры уметь проводить отладку программы, находить в них алгоритмические ошибки знать составные структуры данных (строки, списки, кортежи, словари) знать основные методы решения практических задач. использовать основные алгоритмические приемы при решении математических задач решать простые, сложные и нестандартные задачи 	

Примерные темы проектной и исследовательской деятельности в 8 классах.

1. Программа PyInstaller для преобразования скриптов на языке Python.
2. Модуль graph для создания простых графических программ на языке Python.
3. Стандартная графическая библиотека языка Python.
4. Представление вещественных чисел в памяти компьютера.
5. Программирование А. Шеня.
6. Библиотека PyGame для программирования игр на языке Python.

Примерные темы проектной и исследовательской деятельности в 9 классах.

1. Алгоритмы и структуры данных с использованием Python.
2. Описание встроенных функций Python для обработки строк.
3. Пакет для научных исследований в Python, содержащий быстрые алгоритмы обработки матриц.
4. Что такое асимптотическая сложность?
5. Критерии для оценки качества алгоритмов.
6. Агрегатор контента.
7. GUI-проекты на Python
8. Где применяется Python

Тема 1. Первые программы		3
1	Что такое программа? Самая простая программа.	1
2	Вывод текста на экран. Ввод данных в одной строке.	1
3	Диалоговые программы. Переменные. Сумма чисел.	1
Тема 2. Обработка целых и вещественных чисел		5
4	Арифметические выражения.	1
5	Деление нацело и остаток от деления.	1
6	Что такое вещественное число?	1
7	Операции с вещественными числами.	1
8	Практикум «Арифметические выражения».	1
Тема 3. Случайные и псевдослучайные числа		2
9	Пишем свой генератор случайных чисел.	1
10	Генератор случайных чисел в Python.	1
Тема 4. Ветвления		2
11	Условный оператор.	1
12	Неполная форма условного оператора.	1
Тема 5. Сложные условия		2
13	Операции «И», «ИЛИ», «НЕ».	1
14	Порядок выполнения операций.	1
Тема 6. Циклы с условием		3
15	Как организовать цикл?	1
16	Циклы с предусловием. Алгоритм Евклида.	1
17	Обработка потока данных. Бесконечные циклы.	1
Тема 7. Компьютерная графика		5
18	Что такое компьютерная графика? Графика в Python.	1
19	Система координат. Управляем пикселями.	1
20	Рисуем линии, прямоугольники, окружность.	1
21	Практикум «Условия»	1
22	Изменение координат.	1
Тема 8. Процедуры		2
23	Зачем нужны процедуры?	1
24	Процедуры с параметрами.	1
Тема 9. Циклы по переменной		1
25	Делай N раз.	1
26	Шаг изменения переменной цикла.	1
27	Циклы в компьютерной графике. Узоры.	1
28	Вложенные циклы.	1
29	Рефакторинг.	1
30	Штриховка.	1
Тема 10. Анимация		2
31	Принципы анимации. Начальное положение.	1
32	Анимация движения. Обработка нажатия клавиши.	1
Тема 11. Функции		2
33	Что такое функция? Примеры функций.	1
34	Практикум «Программирование на языке Python»	1
Всего:		34

Сроки изучения учебного материала	№ урока	Раздел, тема, урок. Практические, контрольные работы и другие виды работ	Количество часов
-----------------------------------	---------	---	------------------

Тема 1. Процедуры		5
1	Программы: процедуры и функции.	1
2	Простая процедура.	1
3	Процедуры с параметрами.	1
4	Практикум «Процедуры и функции»	1
5	Локальные и глобальные переменные.	1
Тема 2. Массивы		9
6	Что такое массив? Массивы в языке Python.	1
7	Создание массива. Обращение к элементу массива.	1
8	Перебор элементов массива. Генераторы.	1
9	Вывод массива. Вывод массива с клавиатуры.	1
10	Заполнение массива случайными числами.	1
11	Сумма элементов массива. Подсчёт элементов массива, удовлетворяющих условию	1
12	Особенности копирования списков в Python.	1
13	Линейный поиск. Поиск максимального элемента в массиве.	1
14	Максимальный элемент, удовлетворяющий условию.	1
Тема 3. Символьные строки		3
15	Что такое символьная строка? Сравнение строк.	1
16	Сложение и умножение.	1
17	Обращение к символам. Перебор всех символов.	1
Всего:		17

Данная программа может быть реализована с применением дистанционных образовательных технологий.

Интернет-ресурсы

1. Российская электронная школа (<https://resh.edu.ru/>)
2. Московская электронная школа (<https://uchebnik.mos.ru/catalogue>)
3. Образовательная платформа «Учи.ру» (<https://uchi.ru/>)
4. Платформа Новой школы (<https://sberclass.ru/>)
5. Издательство «БИНОМ» (<https://lbz.ru/>)
6. Маркетплейс образовательных услуг (<https://elducation.ru/>)