

Муниципальное бюджетное общеобразовательное учреждение
«Каменская средняя общеобразовательная школа»

ПРИНЯТА
на заседании педагогического совета
протокол от 29.08.2022 №1

УТВЕРЖДЕНА
приказом директора
МБОУ «Каменская СОШ»
от 29.08.2022 №3

**Дополнительная общеобразовательная общеразвивающая программа
по технической направленности
«Азы программирования»**

Возраст обучающихся: 14-16 лет
Срок реализации: 1 год

Автор-составитель:
Стексова Светлана Алексеевна
педагог дополнительного образования

Раздел 1. Комплекс основных характеристик дополнительной общеобразовательной общеразвивающей программы

Пояснительная записка

Данная дополнительная общеразвивающая программа «Азы программирования» имеет техническую направленность и составлена в соответствии с действующим законодательством, Уставом и локальными актами МБОУ «Каменская СОШ».

Общая характеристика программы: программа «Азы программирования» направлена на то, чтобы углубить знания, умения и навыки, приобретаемые в общеобразовательной школе на уроках информатики, изучить основы нового языка программирования Python.

Направленность программы – техническая.

Уровень программы – базовый. Имеет место вариативность: учащиеся с достаточной степенью свободы и самостоятельности могут осваивать программу на своем уровне, исходя из скорости усвоения.

Актуальность: с развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Отличительной особенностью программы «Азы программирования» является то, что программа позволяет создавать собственные программы для решения конкретной задачи. Программа построена таким образом, чтобы помочь учащимся заинтересоваться программированием вообще и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации; при решении практических и жизненных задач. Программа реализуется с использованием элементов системы электронного обучения, а именно языка программирования Python.

Новизна заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие Интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Педагогическая целесообразность данной образовательной программы заключается в привлечении учащихся к занятиям техническим творчеством, что

способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования, полученные на уроках информатики, и на их основе, углубляя их, учатся составлять простые и сложные программы на новом для них языке программирования.

Адресат программы – учащиеся 8-9 классов. По данной программе предполагается обучение двух разновозрастных групп. В каждой группе 12-15 человек. Возраст учащихся, для обучения которых предназначена программа – 14-16 лет. Пол учащихся значения не имеет, т.к. к программированию бывают склонны как мальчики, так и девочки. В качестве предварительной подготовки к работе по данной программе требуется знание основных алгоритмических конструкций. Физическое здоровье учащихся не играет роли при обучении по данной программе. Группы набираются из числа учащихся, у которых сформирован интерес и мотивация к данной предметной области, то есть программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

Практическая значимость для целевой группы состоит в том, что данная программа способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения. Программа отражает требования и актуальные тенденции не только сегодняшнего, но и завтрашнего дня и имеет междисциплинарный характер, что полностью отражает современные тенденции построения как дополнительных общеобразовательных программ, так и образования в целом. Ознакомление с фундаментальными понятиями алгоритмизации и программирования на доступном уровне имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту обучающегося, охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний.

Преимственность программы состоит в том, что осваиваемые учащимися на уроках информатики основные алгоритмические конструкции изучаются и в рамках данной программы, но на уроках информатики они изучаются на синтаксисе языка Pascal, а в данной программе – на языке Python. Знания, получаемые на уроках, углубляются благодаря тому, что на изучение данной программы выделено в два раза больше времени, чем на урочное изучение информатики.

Объем программы – 72 часа для каждой группы.

Срок освоения программы – 1 год.

Особенности реализации образовательного процесса: занятия организуются в форме кружка, практическое содержание занятий в котором ориентировано не только на овладение обучающимися навыками программирования, но и на подготовку их как грамотных пользователей ПК. Кроме того, формируются навыки участия в дистанционных конкурсах и олимпиадах, умения успешно использовать навыки сетевого взаимодействия.

Формы организации образовательного процесса: индивидуальная, групповая и

фронтальная формы работы.

Виды занятий: лекции, практические занятия, выполнение самостоятельной работы.

Форма обучения: очная. При реализации программы частично применяется электронное обучение.

Режим занятий: 2 раза в неделю по 1 академическому часу. Во время практических занятий за компьютером проводится динамическая пауза, гимнастика для глаз.

Цель и задачи программы

Цель программы – способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи программы:

- формировать элементы ИТ-компетенций;
- изучить язык программирования Python и научить создавать программы на его основе;
- расширять кругозор обучающихся в области программирования.

Содержание программы

Учебный план (1 год обучения)

№ п.п.	название раздела, темы	количество часов				формы аттестации/контроля
		всего	теория	практика	электронное обучение	
1	Знакомство с языком Python					
1.1.	Общие сведения о языке	1	1	0	0	
1.2.	Режимы работы	2	1	1	0	
2	Переменные и выражения					
2.1.	Переменные	2	1	1	0	
2.2.	Выражения	2	1	1	0	
2.3.	Ввод и вывод	3	1	2	0	
2.4.	Задачи на элементарные действия с числами	2	0	1	1	
3	Условные предложения					
3.1.	Логические выражения и операторы	3	1	1	1	
3.2.	Условный оператор	3	1	2	0	
3.3.	Множественное ветвление	3	1	2	0	
3.4.	Реализация ветвления в языке Python	3	0	2	1	
3.5.	Составление программ с ветвлением	1	0	1	0	- зачетная работа.
4	Циклы					
4.1.	Оператор цикла с условием	2	1	1	0	
4.2.	Оператор цикла for	2	1	1	0	

№ п.п.	название раздела, темы	количество часов				формы аттестации/контроля
		всего	теория	практика	электронное обучение	
4.3.	Вложенные циклы	2	0	1	1	
4.4.	Случайные числа	2	1	1	0	
4.5.	Примеры решения задач с циклами	9	0	7	2	- зачетная работа.
5	Функции					
5.1.	Создание функций	3	1	2	0	
5.2.	Локальные и глобальные переменные	2	1	1	0	
5.3.	Примеры решения задач с функциями	3	0	2	1	
5.4.	Рекурсивные функции	2	1	1	0	
6	Строки – последовательности символов					
6.1.	Строки	2	1	1	0	
6.2.	Срезы строк	2	1	1	0	
6.3.	Примеры решения задач со строками	3	0	2	1	
7	Списки					
7.1.	Списки	2	1	1	0	
7.2.	Срезы списков	2	1	1	0	
7.3.	Примеры решения задач со списками	3	1	2	0	
8	Стиль программирования и отладка программы					
8.1.	Стиль программирования	2	1	1	0	
8.2.	Отладка программ	2	1	1	0	
8.3.	Итоговый зачет	2	0	2	0	- зачетная работа.
Итого часов		72	21	43	8	

Содержание учебного плана (1 год обучения)

1. Знакомство с языком Python

1.1. Общие сведения о языке – 1 час.

Теоретическая часть: Техника безопасности в компьютерном классе. Общие сведения о языке Python.

Практическая часть: Установка Python на компьютер.

1.2. Режимы работы – 2 часа.

Теоретическая часть: Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая часть: Написание простейших программ на Python. Написание комментариев к программе.

2. Переменные и выражения

2.1. Переменные – 2 часа.

Теоретическая часть: Типы данных. Целые и вещественные типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Практическая часть: Использование интерфейса среды программирования Python.

2.2. Выражения – 2 часа.

Теоретическая часть: Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Общая структура программы.

Практическая часть: Запись арифметических выражений.

2.3. Ввод и вывод – 3 часа.

Теоретическая часть: Операторы ввода и вывода. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных.

Практическая часть: Организовывают ввод и вывод данных.

2.4. Задачи на элементарные действия с числами – 2 часа.

Теоретическая часть: Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая часть: Решение задач на элементарные действия с числами в среде программирования Python.

3. Условные предложения

3.1. Логические выражения и операторы – 3 часа.

Теоретическая часть: Логический тип данных. Логические выражения и операторы.

Практическая часть: Составление логических выражений.

3.2. Условный оператор – 3 часа.

Теоретическая часть: Сложные условные выражения (логические операторы and, or, not). Условный оператор. Назначение условного оператора. Способ записи условного оператора.

Практическая часть: Использование условного оператора в программах. Создание сложных условий с помощью условных операторов.

3.3. Множественное ветвление – 3 часа.

Теоретическая часть: Множественное ветвление. Альтернативное выполнение.

Практическая часть: Реализация множественного ветвления в программах.

3.4. Реализация ветвления в языке Python – 3 часа.

Теоретическая часть: Реализация ветвления в языке Python.

Практическая часть: Решение задач с применением разного рода ветвлений.

3.5. Составление программ с ветвлением – 1 час.

Теоретическая часть: Примеры решения задач с условным оператором.

Практическая часть: Составление программ с применением разного рода ветвлений.

4. Циклы

4.1. Оператор цикла с условием – 2 часа.

Теоретическая часть: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Виды операторов цикла с условием. Правила записи циклов с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Практическая часть: Использование цикла с условием при решении задач.

4.2. Оператор цикла for – 2 часа.

Теоретическая часть: Оператор цикла с параметром for. Назначение и особенности использования цикла с параметром. Формат записи цикла с параметром. Операторы управления циклом. Пример задачи с использованием цикла for.

Практическая часть: Определение целесообразности применения и использования цикла с параметром для решения поставленной задачи. Решение задач с использованием цикла с параметром.

4.3. Вложенные циклы – 2 часа.

Теоретическая часть: Вложенные циклы. Циклы в циклах.

Практическая часть: Реализация множественного ветвления в программах. Решение задач с применением множественного ветвления.

4.4. Случайные числа – 2 часа.

Теоретическая часть: Случайные числа. Функция randrange. Функция random.

Практическая часть: Решение задач с применением циклов и случайных чисел.

4.5. Примеры решения задач с циклами – 9 часов.

Теоретическая часть: Примеры использования циклов различных типов при решении задач.

Практическая часть: Определение вида цикла, наиболее удобного для решения поставленной задачи. Составление программ с применением циклов различных типов.

5. Функции

5.1. Создание функций – 3 часа.

Теоретическая часть: Понятие функции. Создание функций. Способы описания функции. Параметры и аргументы. Принципы структурного программирования.

Практическая часть: Создание функций.

5.2. Локальные и глобальные переменные – 2 часа.

Теоретическая часть: Локальные и глобальные переменные. Понятие локальных переменных подпрограмм. Понятие формальных и фактических параметров подпрограмм. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Способ передачи параметров.

Практическая часть: Использование механизма параметров для передачи значений. Использование lambda-функций.

5.3. Примеры решения задач с функциями – 3 часа.

Теоретическая часть: Примеры решения задач с использованием функций.

Практическая часть: Решение задач с использованием различных функций.

5.4. Рекурсивные функции – 2 часа.

Теоретическая часть: Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая часть: Решение задач с использованием рекурсивных функций. Решение задачи по вычислению факториала числа.

6. Строки – последовательности символов

6.1. Строки – 2 часа.

Теоретическая часть: Составной тип данных – строка. Назначение строкового типа данных. Операторы для работы с строками. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Процедуры и функции для работы со строками. Операции со строками.

Практическая часть: Описание строк. Соединение строк. Нахождение длины строки.

6.2. Срезы строк – 2 часа.

Теоретическая часть: Срезы строк. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи).

Практическая часть: Решение задач со строками. Вырезание части строки. Нахождение подстроки в строке. Нахождение количества слов в строке.

6.3. Примеры решения задач со строками – 3 часа.

Теоретическая часть: Примеры решения задач со строками.

Практическая часть: Решение задач со строками.

7. Списки

7.1. Списки – 2 часа.

Теоретическая часть: Сложные типы данных. Списки. Тип список (list). Индексы. Способ описания списка. Способ доступа к элементам списка. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Операции, выполняемые со списками.

Практическая часть: Описание списков. Ввод и вывод элементов списка.

7.2. Срезы списков – 2 часа.

Теоретическая часть: Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range.

Практическая часть: Решение задач со срезами списков.

7.3. Примеры решения задач со списками – 3 часа.

Теоретическая часть: Примеры решения задач со списками.

Практическая часть: Решение задач со списками. Поиск элемента в списке. Поиск минимума и максимума. Нахождение суммы элементов списка. Использование вложенных списков.

8. Стиль программирования и отладка программы

8.1. Стиль программирования – 2 часа.

Теоретическая часть: Стиль программирования. Правила именования объектов. Основные рекомендации по написанию программ.

Практическая часть: Написание программ по полученным рекомендациям.

8.2. Отладка программы – 2 часа.

Теоретическая часть: Виды ошибок в программе.

Практическая часть: Определение вида ошибки. Нахождение ошибок в программе. Выполнение тестирования и отладки программы.

8.3. Итоговый зачет – 2 часа.

Теоретическая часть: Повторение пройденного материала.

Практическая часть: Решение различных задач в среде программирования Python.

Планируемые результаты реализации программы

Метапредметные:

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

- использование коммуникационных технологий в учебной деятельности и повседневной жизни;
- обучающиеся будут иметь сформированные элементы IT-компетенций.

Личностные:

- широкие познавательные интересы, инициатива и любознательность, мотивы познания и творчества; готовность и способность учащихся к саморазвитию и реализации творческого потенциала в духовной и предметно-продуктивной деятельности за счет развития их образного, алгоритмического и логического мышления;
- готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ;
- интерес к информатике и ИКТ, стремление использовать полученные знания в процессе обучения другим предметам и в жизни;
- способность связать учебное содержание с собственным жизненным опытом и личными смыслами, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества;
- готовность к самостоятельным поступкам и действиям, принятию ответственности за их результаты;
- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных
- способность к избирательному отношению к получаемой информации за счет умений ее анализа и критичного оценивания; ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ;
- готовность и способность учащихся к саморазвитию и личностному самоопределению.

Предметные:

- умение формально выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов;
- умение создавать и выполнять программы для решения алгоритмических задач в программе Python;
- навыки и умения безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете.

Раздел 2. Комплекс организационно-педагогических условий

Календарный учебный график

п.п.	тема занятия	кол-во часов	дата проведения занятия (план)	дата проведения занятия (факт)
	<i>1. Знакомство с языком Python</i>			
	<i>1.1. Общие сведения о языке</i>			
1	Техника безопасности. Общие сведения о языке Python	1	02.09.2022	
2	Режимы работы Python. Что такое программа	1	06.09.2022	
3	Структура программы на языке Python. Комментарии	1	09.09.2022	
	<i>2. Переменные и выражения</i>			
	<i>2.1. Переменные</i>			
4	Переменные. Типы данных	1	13.09.2022	
5	Оператор присваивания	1	16.09.2022	
	<i>2.2. Выражения</i>			
6	Выражения. Операции. Порядок выполнения операций	1	20.09.2022	
7	Математические функции. Композиция. Общая структура программы		23.09.2022	
	<i>2.3. Ввод и вывод</i>			
8	Операторы ввода и вывода. Ввод данных с клавиатуры. Вывод данных на экран	1	27.09.2022	
9	Пример скрипта, использующего ввод и вывод данных	1	30.09.2022	
10	Использование ввода и вывода данных	1	04.10.2022	
	<i>2.4. Задачи на элементарные действия с числами</i>			
11	Решение задач на элементарные действия с числами	1	07.10.2022	
12	Решение задач на элементарные действия с числами	1	11.10.2022	
	<i>3. Условные предложения</i>			
	<i>3.1. Логические выражения и операторы</i>			
13	Логический тип данных. Логические выражения	1	14.10.2022	
14	Логические операторы	1	18.10.2022	

п.п.	тема занятия	кол-во часов	дата проведения занятия (план)	дата проведения занятия (факт)
15	Решение задач с использованием логических выражений и операторов	1	21.10.2022	
	<i>3.2. Условный оператор</i>			
16	Сложные условные выражения	1	25.10.2022	
17	Условный оператор.	1	28.10.2022	
18	Использование условного оператора в программах	1	01.11.2022	
	<i>3.3. Множественное ветвление</i>			
19	Множественное ветвление	1	04.11.2022	
20	Альтернативное выполнение	1	08.11.2022	
21	Реализация множественного ветвления в программах	1	11.11.2022	
	<i>3.4. Реализация ветвления в языке Python</i>			
22	Решение задач с применением разного рода ветвлений	1	15.11.2022	
23	Решение задач с применением разного рода ветвлений	1	18.11.2022	
24	Решение задач с применением разного рода ветвлений	1	22.11.2022	
	<i>3.5. Составление программ с ветвлением</i>			
25	Составление программ с ветвлением	1	25.11.2022	
	<i>4. Циклы</i>			
	<i>4.1. Оператор цикла с условием</i>			
26	Понятие цикла. Оператор цикла с условием while	1	29.11.2022	
27	Примеры использования циклов	1	02.12.2022	
	<i>4.2. Оператор цикла for</i>			
28	Оператор цикла с параметром for	1	06.12.2022	
29	Решение задач с использованием цикла for	1	09.12.2022	
	<i>4.3. Вложенные циклы</i>			
30	Вложенные циклы	1	13.12.2022	
31	Реализация множественного ветвления в программах	1	16.12.2022	
	<i>4.4. Случайные числа</i>			
32	Случайные числа. Функции randrange и random	1	20.12.2022	
33	Решение задач с применением циклов и случайных чисел	1	23.12.2022	

п.п.	тема занятия	кол-во часов	дата проведения занятия (план)	дата проведения занятия (факт)
	<i>4.5. Примеры решения задач с циклами</i>			
34	Примеры использования циклов различных типов при решении задач.	1	27.12.2022	
35	Примеры использования циклов различных типов при решении задач.	1	30.12.2022	
36	Примеры использования циклов различных типов при решении задач.	1	10.01.2023	
37	Примеры использования циклов различных типов при решении задач.	1	13.01.2023	
38	Примеры использования циклов различных типов при решении задач.	1	17.01.2023	
39	Примеры использования циклов различных типов при решении задач.	1	20.01.2023	
40	Примеры использования циклов различных типов при решении задач.	1	23.01.2023	
41	Примеры использования циклов различных типов при решении задач.	1	27.01.2023	
42	Примеры использования циклов различных типов при решении задач.	1	31.01.2023	
	<i>5. Функции</i>			
	<i>5.1. Создание функций</i>			
43	Понятие функции. Создание функций	1	03.02.2023	
44	Принципы структурного программирования	1	07.02.2023	
45	Решение задач с использованием функций	1	10.02.2023	
	<i>5.2. Локальные и глобальные переменные</i>			
46	Локальные и глобальные переменные	1	14.02.2023	
47	Инструкция lambda	1	17.02.2023	
	<i>5.3. Примеры решения задач с функциями</i>			
48	Решение задач с использованием функций	1	21.02.2023	
49	Решение задач с использованием функций	1	28.02.2023	
50	Решение задач с использованием функций	1	03.03.2023	
	<i>5.4. Рекурсивные функции</i>			
51	Рекурсивные функции	1	07.03.2023	
52	Вычисление факториала. Числа Фибоначчи	1	10.03.2023	
	<i>6. Строки – последовательности символов</i>			
	<i>6.1. Строки</i>			
53	Составной тип данных – строка	1	14.03.2023	

п.п.	тема занятия	кол-во часов	дата проведения занятия (план)	дата проведения занятия (факт)
54	Операции со строками	1	17.03.2023	
	<i>6.2. Срезы строк</i>			
55	Срезы строк	1	21.03.2023	
56	Решение задач со строками	1	24.03.2023	
	<i>6.3. Примеры решения задач со строками</i>			
57	Решение задач со строками	1	28.03.2023	
58	Решение задач со строками	1	31.03.2023	
59	Решение задач со строками	1	04.04.2023	
	<i>7. Списки</i>			
	<i>7.1. Списки</i>			
60	Списки. Способы описания списка и доступа к его элементам	1	07.04.2023	
61	Операторы для списков. Операции, выполняемые со списками	1	11.04.2023	
	<i>7.2. Срезы списков</i>			
62	Срезы списков. Удаление списка. Клонирование списков	1	14.04.2023	
63	Списочные параметры. Функция range	1	18.04.2023	
	<i>7.3. Примеры решения задач со списками</i>			
64	Решение задач со списками	1	21.04.2023	
65	Решение задач со списками	1	25.04.2023	
66	Решение задач со списками	1	28.04.2023	
	<i>8. Стиль программирования и отладка программы</i>			
	<i>8.1. Стиль программирования</i>			
67	Стиль программирования. Правила именования объектов	1	02.05.2023	
68	Основные рекомендации по написанию программ	1	05.05.2023	
	<i>8.2. Отладка программы</i>			
69	Виды ошибок в программе	1	12.05.2023	
70	Тестирование и отладка программ	1	16.05.2023	
	<i>8.3. Итоговый зачет</i>			
71	Итоговый зачет	1	19.05.2023	

п.п.	тема занятия	кол-во часов	дата проведения занятия (план)	дата проведения занятия (факт)
72	Итоговый зачет	1	23.05.2023	

Условия реализации программы

Кадровые ресурсы: учитель информатики Стексова Светлана Алексеевна, образование высшее, соответствие занимаемой должности.

Материально-технические ресурсы: компьютерный класс МБОУ «Каменская СОШ» на 20 посадочных мест, ноутбук – 20 шт., МФУ, проектор, экран, магнитно-маркерная доска, электронная доска, ноутбук учителя, рабочее место учителя.

Информационные ресурсы: доступ к сети Интернет на всех ноутбуках.

Программное обеспечение: ОС Альт, Libre Office, Python 3.10.6.

Формы аттестации/контроля. Оценочные материалы

Формы аттестации/контроля: зачетная работа.

Зачетные работы позволяют выявить соответствие результатов образования сформулированным образовательным задачам, с их помощью можно понять, выработан ли новый навык или нет.

Оценочные материалы:

№	Критерий	Оценка (в баллах)
1	Новизна используемых методов решения задачи	2 – придуманы абсолютно новые методы решения задачи 1 – метод решения имеет элемент новизны 0 – задача решена давно известными методами
2	Верность решения задачи	2 – задача решена абсолютно верно для любых входных данных 1 – задача выдает верное решение лишь для некоторых входных данных 0 – решение неверное при любых входных данных
3	Скорость работы программы	1 – программа работает быстро 0 – программа работает с низкой скоростью
4	Объем памяти на программу	2 – программа работает и занимает небольшой объем памяти 1 – программа работает, но затрачивает большой объем памяти 0 – программа не работает
5	Структура программы	2 – программа написана по всем требованиям к структуре программы, имеются комментарии 1 – программа написана по всем требованиям к структуре программы, комментарии отсутствуют 0 – программа не имеет четкой структуры
6	Уровень самостоятельности при решении задачи	2 – задача решена полностью самостоятельно 1 – при решении использована минимальная помощь учителя 0 – решение выполнено со значительной помощью учителя
	Максимальное количество баллов	11 баллов

Оценка «зачет» выставляется при сумме баллов – от 6 и более.

Методические материалы

Для реализации программы используется программа Python 3.10.6.

Методические особенности организации образовательного процесса: наиболее часто используемая форма организации деятельности – индивидуальная, при этом отдельные вопросы и ошибки рассматриваются в группах или всем классом, исходя из особенностей обучающихся в усвоении пройденного материала.

Первая часть занятия предполагает получение обучающимися нового материала. Во время второй части занятия каждый обучающийся пытается самостоятельно реализовать полученную теоретическую базу в рамках собственной программы. Оценка результатов производится коллективно всей группой. Некоторые занятия полностью отведены на написание программ.

Общение на занятии ведётся в свободной форме — каждый обучающийся в любой момент может задать интересующий его вопрос без поднятия руки. Данный момент очень важен в процессе обучения, так как любой невыясненный вопрос, может превратиться в препятствие для получения обучающимся последующих знаний и реализации им собственных проектов.

Практическая часть работы – работа в среде программирования Python и проектирование информационных продуктов. Для наилучшего усвоения материала практические задания рекомендуется выполнять каждому за компьютером. При выполнении глобальных проектов рекомендуется объединять школьников в пары. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения рефлексивных упражнений и практических заданий. Итоговый контроль осуществляется по результатам итогового зачета. Формы подведения итогов: зачет, тест, практическая работа, зачетная работа.

Методы обучения:

- словесные методы (лекция, объяснение);
- демонстративно-наглядные (демонстрация работы в программе, схем, скриптов);
- исследовательские методы;
- работа в парах;
- работа в малых группах.

Методы воспитания: мотивация, поощрение.

Педагогические технологии: личностно-ориентированное обучение, здоровьесберегающая технология, технология группового обучения, ИКТ.

Дидактические материалы: обучающие материалы расположены на сайте <https://wiki.python.org/moin/BeginnersGuide/NonProgrammers>, инструкция по технике безопасности и правилам поведения в кабинете информатики.

Методические разработки находятся в приложении 1.

Рабочая программа воспитания, календарный план воспитательной работы

1. Характеристика кружка.

Кружок рассчитан на учащихся 8-9 классов (возраст 14-16 лет). Как форма внеклассной, внешкольной работы, кружок выполняет функции расширения, углубления, компенсации предметных знаний; приобщения детей к разнообразным социокультурным видам деятельности; расширения коммуникативного опыта; организации детского досуга и отдыха.

2. Цель, задачи и планируемые результаты воспитательной работы (в рамках реализации дополнительной общеобразовательной общеразвивающей программы).

Цель воспитательной работы состоит в формировании гармонично развитой личности, у которой развито логическое мышление и соответствующие компетенции. Для достижения данной цели необходимо решить задачу по выявлению и развитию технических способностей и наклонностей детей.

Планируемые результаты воспитательной работы:

- проявлять дисциплинированность, трудолюбие и упорство в достижении поставленных целей;
- оказывать помощь членам коллектива, находить с ними общий язык и общие интересы

3. Работа с коллективом учащихся (в рамках реализации дополнительной общеобразовательной общеразвивающей программы):

- инициирование и поддержка участия членов кружка в общих ключевых делах, оказание необходимой помощи детям в их подготовке, проведении и анализе;
- организация интересных и полезных для личностного развития ребенка совместных дел с обучающимися вверенного ему объединения профориентационной направленности, позволяющая с одной стороны, вовлечь в них детей с самыми разными потребностями, и тем самым дать им возможность самореализоваться в них, а с другой,
- установить и упрочить доверительные отношения с обучающимися объединения, стать для них значимым взрослым, задающим образцы поведения в обществе;
- сплочение коллектива кружка через: игры и тренинги на сплочение и командообразование;
- выработка совместно с обучающимися законов объединения, помогающих детям освоить нормы и правила общения, которым они должны следовать в учреждении.

4. Работа с родителями (в рамках реализации дополнительной общеобразовательной общеразвивающей программы):

- регулярное информирование родителей об успехах и проблемах их детей, о жизни объединения и учреждения в целом;
- помощь родителям обучающихся в регулировании отношений между ними, администрацией и педагогами учреждения;
- привлечение членов семей обучающихся к организации и проведению дел кружка.

5. Календарный план воспитательной работы.

№ п/п	Мероприятие	Воспитательные задачи, решаемые в ходе мероприятия	Сроки проведения	Примечание
1	День программиста	- расширение кругозора учащихся; - формирование навыков поведения в информационном обществе с целью обеспечения информационной безопасности..	сентябрь	
2	Олимпиада школьников по информатике	- самообразование учащихся; - побуждение к участию в кружках технического и научного творчества.	октябрь	
3	Встреча с представителем профессии	- расширение кругозора учащихся; - воспитание уважительного отношения к людям умственного труда.	декабрь	
4	Научно-практическая конференция	- получение навыков научно-исследовательской деятельности; - самообразование учащихся.	март	

Список литературы

Для педагога:

- 1) Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
- 2) Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
- 3) <https://python.org>

Для учащихся:

- 1) Голиков Д. В. Scratch для юных программистов. — СПб.: БХВ-Петербург, 2017. — 192 с.: ил Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
- 2) <https://wiki.python.org/moin/BeginnersGuide/NonProgrammers> rted

Тест №1

Внимание, возможно несколько вариантов ответа.

1) Python является:

- А. Машинно-ориентированным языком (низкого уровня)
- Б. Языком высокого уровня
- В. Объектно-ориентированным языком

2) Область применения Python:

- А. Робототехника и искусственный интеллект
- Б. Обучение
- В. Интернет

3) Год разработки Python:

- А. 1990
- Б. 1991
- В. 1993

4) Разработчик языка Python:

- А. Бил Гейтс
- Б. Никлаус Вирт
- В. Гвидо Ван Россум

5) В Python блоки кода выделяются:

- А. пробельными отступами
- Б. символом /
- В. символом ;

6) Минусы языка Python:

- А. высокая стоимость, низкая читабельность и сложный синтаксис;
- Б. более низкая скорость работы по сравнению с аналогичным кодом, написанным, например, на C++;
- В. более высокое потребление памяти по сравнению с аналогичным кодом, написанным, например, на C++.

7) Крупные компании, применяющие Python:

- А. Google
- Б. Facebook
- В. Spotify

Тест №2

Внимание, возможно несколько вариантов ответа.

1) В Python переменная должна быть объявлена до того, как ей будет присвоено значение:

- А. Да
- Б. Нет

2) Выберите правильное объявление переменной:

- А. `x := 100`
- Б. `x ← 100`
- В. `let x = 100`
- Г. `x = 100`

3) В Python переменная может иметь сначала один тип, а затем другой;

- А. Да
- Б. Нет

4) Ниже представлен список имен переменных. Какие из них корректны?

- А. `sum1.3`
- Б. `value`
- В. `4square`
- Г. `massa42`
- Д. `return`
- Е. `phone_number`

5) Вы читаете код. В нем присутствуют следующие строки:

```
c=52  
a=c
```

- А. Эти объекты ссылаются на одни и те же переменные
- Б. Эти объекты ссылаются на разные переменные

6) Ниже представлен список слов. Какие из них являются ключевыми?

- А. `goto`
- Б. `None`
- В. `class`
- Г. `default`
- Д. `and`
- Е. `or`

Тест №3

Внимание, возможно несколько вариантов ответа.

1) Что такое итерация?

- А. Такого понятия не существует.
- Б. Однократное выполнение какого-либо действия.
- В. Второе название переменной
- Г. Организация обработки данных, при которой действия повторяются многократно.

2) Какие операторы итерации существуют?

- А. for
- Б. from
- В. def
- Г. while

3) Выберите правильный синтаксис:

- А. for i in list:
- Б. from i in list:
- В. for I in list
- Г. i in for list:

4) В чем разница между циклами for и while?

- А. Нет разницы.
- Б. while быстрее, чем for.
- В. Оператор for выполняет итерацию через функцию коллекции или итерируемого объекта, или генератора.
- Г. Оператор while выполняет итерацию до тех пор, пока условие не будет False.

5) Какой результат выполнения этого кода?

```
for I in range(1,10):  
    i--5  
print (i)
```

- А. -5
- Б. 4
- В. 5
- Г. 10

6) Что делает break?

- А. Такого понятия не существует.
- Б. Останавливает цикл.
- В. Останавливает выполнение всего кода.
- Г. Продолжает цикл.

7) Какой результат выполнения этого кода?

```
i=0
while i<7:
    i+=1
i-=7
print(i)
```

- А. -7
- Б. 0
- В. 7
- Г. 14

8) Для чего нужен else в циклах?

- А. Не используется в циклах.
- Б. Проверяет, работает ли цикл.
- В. Создает условие для цикла.
- Г. Проверяет, был ли произведен выход из цикла инструкцией break, или же «естественным» способом.

9) Какой результат выполнения этого кода?

```
my_list=[1, 2, 3, 4, 5]
for i in my_list:
    if i==3:
        print("Item found!")
        break
    print(i)
else:
    print("Item not found!")
```

- А. 1 2 Item not found!
- Б. Item found!
- В. Item not found!
- Г. SyntaxError

Тест №4

Внимание, возможно несколько вариантов ответов.

1) Что такое функция?

А. Любой код в Python.

Б. Код, заключенный в круглые скобки.

В. Структура, определяющая поведение объекта.

Г. Объект, принимающий аргументы и возвращающий значение.

2) Какое ключевое слово используется для создания функции?

А. fun

Б. def

В. vod

Г. function

3) Сколько параметров может принимать функция?

А. 0

Б. 1

В. 2

Г. Бесконечно много

4) Какое ключевое слово используется для возврата значения из функции?

А. get

Б. post

В. return

Г. answer

5) Что выведет этот код?

```
def get_sum(a,b)
```

```
    pass
```

```
print(get_sum(5,3))
```

А. None

Б. 5

В. 3

Г. 8

6) Что такое рекурсивная функция?

А. Функция, которая курсирует между модулями.

Б. Функция, которая возвращает саму себя.

В. Функция, которая выполняется с последней строки по первую.

Г. Функция, написанная курсивом.