

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ
МУНИЦИПАЛЬНОГО ОБРАЗОВАНИЯ АБИНСКИЙ РАЙОН

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ СТАНЦИЯ ЮНЫХ ТЕХНИКОВ
МУНИЦИПАЛЬНОГО ОБРАЗОВАНИЯ АБИНСКИЙ РАЙОН

Принята на заседании
педагогического совета
от 25 марта 2024 г.
протокол № 3

Утверждаю
Директор МБУ ДО СЮТ
_____ Н. А. Саянова
приказ от 13 апреля 2024 г. № 124

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«Программирование на языке Python»

Уровень программы: ознакомительный

Срок реализации программы: 1 год, 72 ч.

Возрастная категория: от 12 до 18 лет

Состав группы: от 8 человек

Форма обучения очная

Вид программы: модифицированная

Программа реализуется: на бюджетной основе

ID-номер Программы в Навигаторе:46559

Автор-составитель
Шельпякова Елена Викторовна
педагог дополнительного образования

п. Ахтырский, 2024 г.

ПАСПОРТ ПРОГРАММЫ

Наименование муниципалитета	Абинский район
Наименование организации	МБУ ДО СЮТ
ID-номер программы в АИС «Навигатор»	46559
Полное наименование программы	Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python»
Механизм финансирования (ПФДО, муниципальное задание, внебюджет)	На бюджетной основе
ФИО авторов составителей	Шельпякова Елена Викторовна
Краткое описание программы	данная программа имеет выраженную техническую направленность, личностно-ориентированная, личностно-значимая, направлена на развитие навыков программирования.
Форма обучения	очная
Уровень содержания	ознакомительный
Продолжительность освоения (объем)	1 год, 72 часа
Возрастная категория	12– 18 лет
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	<p>1. Образовательные:</p> <ul style="list-style-type: none"> – познакомить с понятиями алгоритма, вычислимой функции, языка программирования; – научить составлять и читать блок-схемы; – сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python; – объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами); – научить применять функции при написании программ на языке программирования Python;

	<ul style="list-style-type: none"> – научить отлаживать и тестировать программы, делать выводы о работе этих программ. <p>2. Воспитательные:</p> <ul style="list-style-type: none"> - воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности; - сформировать культуру занятий, направленную на воспитание личностных и социальных качеств; <p>3. Развивающие:</p> <ul style="list-style-type: none"> - развить познавательные процессы (внимание, восприятие, логическое мышление, память), - развить креативность, - развить способности к самореализации.
Ожидаемые результаты	<p>В рамках курса «Программирование на языке Python» обучающиеся овладевают следующими знаниями, умениями и способами деятельности:</p> <ul style="list-style-type: none"> – умеют составлять алгоритмы для решения задач; – умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python; – владеют основными навыками программирования на языке Python; – умеют отлаживать и тестировать программы, написанные на языке Python.
Особые условия	<ul style="list-style-type: none"> – Операционная система Windows 7 Профессиональная или выше; – Интерпретатор Python версии 3.7 и выше; – IDE JupyterPyCharm; – FoxitReaderили другойпросмотрщик PDF файлов; – WinRAR
Возможность реализации в сетевой форме	нет
Возможность реализации в электронном формате с применением дистанционных технологий	при необходимости возможна реализация в электронном формате с применением дистанционных технологий.
Материально-техническая база	<p>Материально-техническое обеспечение:</p> <ul style="list-style-type: none"> – столы для компьютера; – компьютерные стулья; – шкафы для дидактических материалов, пособий;

- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JupyterLab/PyCharm;
- FoxitReader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

РАЗДЕЛ 1

«КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ОБРАЗОВАНИЯ: ОБЪЕМ, СОДЕРЖАНИЕ, ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ»

1.1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» разработана и реализуется в соответствии:

1. Федеральный закон Российской Федерации «Об образовании в Российской Федерации» от 29 декабря 2012 года №273-ФЗ;

2. Приказ Министерства образования и науки РФ «Об утверждении Порядка организации осуществления образовательной деятельности по дополнительным общеобразовательным программам» от 09.11.2018г. №196;

3. Приказ Министерства образования и науки РФ «Об утверждении порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ» от 9 января 2014 года № 2;

4. Концепция развития дополнительного образования детей, утвержденная распоряжением Правительства Российской Федерации от 4 сентября 2014 года № 1726-р;

5. Постановление Главного государственного санитарного врача Российской Федерации «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» от 4 июля 2014 года № 41;

6. Федеральный закон Российской Федерации «Об основных гарантиях прав ребенка в РФ» от 24.07.1998 г. № 124-ФЗ (с изменениями от 20.07.2000 г.; 22.08; 21.12.2004г.; 26,30.06.2007г.);

7. Государственная программа Российской Федерации «Развитие образования» на 2021-2025 годы;

8. Устав МБУ ДО СЮТ.

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации до 2020 года и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают

раскрытие личностных резервов обучающихся и создание соответствующей образовательной среды.

Общепедагогическая направленность занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных обучающихся. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание обучающимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения обучающихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если обучающиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Направленность (профиль) программы

Данная программа имеет выраженную техническую направленность, что позволяет осуществлять воспитание ребят через творческую активность. Программа дополняет основное образование, предоставляет школьнику свободу выбора вида деятельности, делая его досуг содержательным, дает возможность осознать себя, свои предпочтения в любой из них.

Содержание обучения, представленное в программе «Программирование на языке Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

Актуальность программы состоит в том, что активизация познавательного процесса позволяет обучающимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Программа направлена на социально-экономическое развитие муниципального образования и региона в целом.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов обучающихся, их дальнейшей профессиональной ориентации.

Новизна программы

Новизна Программы 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 программы выражена в подборе интерактивных и практикоориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

Отличительные особенности программы:

Ключевой особенностью курса является его направленность на формирование у обучающихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Практическая значимость:

В рамках предлагаемого курса «Программирование на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов обучающихся (внимание, память, логика), освоения приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования предметных и метапредметных результатов обучающихся особенно важно, т.к. именно они активизирует процесс индивидуально-личностного становления обучающихся.

Адресат программы

Дополнительная общеразвивающая программа «Программирование на языке Python» рассчитана на детей 12–18 лет, проявляющих интерес к информационно-коммуникационным технологиям. Обучающиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений законных представителей обучающихся (самих обучающихся с 14 лет). Группы формируются из школьников разного возраста на добровольной внеконкурсной основе.

Программа составлена с учётом индивидуальных и возрастных особенностей обучающихся. Психолого-педагогические особенности обучающихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

- Срок реализации программы: 1 год.
- Режим обучения: 2 раза в неделю по 1 часа, или 1 раз в неделю по 2 часа.

- Форма обучения: очная.
- Формирование групп осуществляется по желанию детей.
- Возраст обучающихся – 7-16 лет.
- Состав группы постоянный.
- Количество обучающихся в объединении: 8 – 12 человек.

Уровень программы – Ознакомительный.

Объем программы - 72 часа.

Режим работы в группах:

- 2 раза в неделю по 1 часу или 1 раз в неделю по 2 часа.
- с перерывами по 10-15 минут для игр на воздухе или разминке, а также для отдыха глаз. 15 минут после занятий необходимы для приведения рабочего места в порядок.
- Форма занятий групповая с индивидуальным подходом.

Методические принципы программы

- принцип развития: развитие индивидуальных способностей, общей культуры, навыков творческой продуктивной деятельности обучающихся;
- принцип демократизма: право каждого субъекта системы технического творчества на выбор своей траектории развития;
- принцип дифференциации и индивидуализации образования: выявление и развитие способностей обучающихся в области технического творчества, обеспечение их развития в соответствии с потенциалом, индивидуальными возможностями и интересами;
- принцип культуросообразности: ориентация на потребности общества и личности обучающихся, единство человека и социокультурной среды, адаптация детей к современным условиям жизни общества;
- принцип единства учебного и воспитательного процесса;
- принцип систематичности и последовательности: вначале используются репродуктивный и интерактивный методы усвоения знаний с постепенным введением проблемного метода обучения, метода проекта.

Формы обучения и занятий

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности обучающихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала педагогом, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе – обучение обучающихся навыкам самостоятельной творческой деятельности.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п.

Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов,

проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

Особенности организации образовательного процесса

В программе предусмотрено участие детей с особыми образовательными потребностями: детей-инвалидов и детей с ограниченными возможностями здоровья; талантливых (одаренных, мотивированных) детей; детей, находящихся в трудной жизненной ситуации.

В программе предусмотрена возможность занятий по индивидуальной образовательной траектории (по индивидуальному учебному плану).

1.2 ЦЕЛЬ И ЗАДАЧИ ПРОГРАММЫ

Цель: 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

Образовательные:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Воспитательные:

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;

Развивающие:

- развить познавательные процессы (внимание, восприятие, логическое мышление, память),
- развить креативность,
- развить способности к самореализации.

1.3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Реализация программы предполагает достижение обучающимися следующих личностных, метапредметных и предметных результатов:

Предметные:

- умение 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 на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Личностные:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

В рамках курса «Программирование на языке Python» обучающиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.

1.4 СОДЕРЖАНИЕ ПРОГРАММЫ

Учебно-тематический план

№ п/п	Наименование тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
	Инструктаж по технике безопасности в компьютерном классе	2	2		Опрос
1.	Раздел 1. Знакомство с	4	1,5	2,5	Тестирование,

	языком Python				решение практических задач
1.1	Общие сведения о языке	1	1		
1.2.	Режимы работы Практическая работа: Режимы работы с Python	1	0,5	0,5	
1.3	Тест №1. Знакомство с языком Python	2		2	Тестирование, решение практических задач
2.	Раздел 2. Переменные и выражения	8	4	4	
2.1	Переменные Практическая работа: Переменные	2	1	1	
2.2	Выражения Практическая работа: Выражения	2	1	1	
2.3	Ввод и вывод	2	1	1	
2.4	Задачи на элементарные действия с числами Практическая работа Задачи на элементарные действия с числами	2	1	1	
3	Раздел 3. Условные предложения	10	2	8	Тестирование, Решение практических задач
3.1	Логические выражения и операторы. Практическая работа: Логические выражения	2	1	1	
3.3	Множественное ветвление Практическая работа: Множественное ветвление	2	1	1	
3.5	Самостоятельная работа №1 по теме «Условные операторы».	2		2	
3.6	Зачетная работа №1. "Составление программ с ветвлением".	2		2	
3.7	Тест №3. "Условные операторы".	2		2	
4	Раздел 4. Циклы	10	3	7	
4.1.	Занятие12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2	1	1	
4.2.	Занятие13. Оператор цикла for Практическая работа Решение задачи с циклом for.	2	1	1	
4.3.	Занятие16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	2	1	1	
4.4	Самостоятельная работа №2 "Составление программ с циклом"	2		2	

4.5	Тест №4. Циклы	2		2	
5	Раздел 5. Функции	10	3	7	
5.1.	Создание функций Практическая работа Создание функций	2	1	1	Тестирование, решение практических задач
5.2.	Локальные переменные Практическая работа Локальные переменные	2	1	1	
5.3.	Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2	1	1	
5.4	Самостоятельная работа №3 по теме "Функции"	2		2	
5.5	Тест № 5. Функции	2		2	
6	Раздел 6. Строки - последовательности символов	6	2	4	
6.1.	Строки Практическая работа: Строки	2	1	1	Решение практических задач
6.2.	Срезы строк	2	1	1	
6.3	Примеры решения задач со строками Практическая работа: Решение задач со строками.	2		2	
7	Раздел 7. Сложные типы данных	10	6	4	
7.1.	Списки Практическая работа: Списки	2	1	1	Тестирование, Решение практических задач
7.2	Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	2	1	1	
7.3	Матрицы	1	0,5	0,5	
7.4	Тест №7. Списки	1	0,5	0,5	
7.5	Кортежи	1	1		
7.6	Введение в словари	1	1		
7.7	Множества в языке Python	2	1	1	
8	Раздел 8. Стиль программирования и отладка программ	12	6	6	
8.1	Стиль программирования	2	1	1	Решение практических задач, презентация проекта, рефлексия.
8.2	Отладка программ	2	1	1	
8.3	Зачет по курсу «Программирование на языке Python»	6	2	4	
8.3	Что дальше?	2	2		
	ВСЕГО	72	29,5	42,5	

1.5 СОДЕРЖАНИЕ УЧЕБНО-ТЕМАТИЧЕСКОГО ПЛАНА

Содержание программы

Вводное занятие (2 часа).

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python (4 часов)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python

Практическая работа 1.2. Режимы работы с Python

Обучающиеся должны знать/понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Обучающиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения (8 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой. Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами. Тест №2. Выражения и операции.

Обучающиеся должны знать/понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения (10 часов)

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения. Практическая работа 3.2. "Условный оператор". Практическая работа 3.3. Множественное ветвление. Практическая работа 3.4. "Условные операторы"

Самостоятельная работа №1. Решение задач по теме "Условные операторы". Зачетная работа №1. "Составление программ с ветвлением".

Тест №3. "Условные операторы".

Обучающиеся должны знать/ понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы (10 часов)

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

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа №2 "Составление программ с циклом"

Тест №4. Циклы

Творческая работа №1. "Циклы"

Обучающиеся должны знать/ понимать:

- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Обучающиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции (6 часов)

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.
Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа №3 по теме "Функции" Тест №5. Функции

Обучающиеся должны знать/ понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Обучающиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки- последовательности символов (10 часов)

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Обучающиеся должны знать/ понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Обучающиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Раздел 7. Сложные типы данных (10 часов)

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы

для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест №7. Списки

Обучающиеся должны знать/понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

Раздел 8. Стиль программирования и отладка программ (12 часов)

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Обучающиеся должны знать/понимать:

- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Обучающиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

РАЗДЕЛ 2
«КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ,
ВКЛЮЧАЮЩИХ ФОРМЫ АТТЕСТАЦИИ»

2.1 КАЛЕНДАРНЫЙ - УЧЕБНЫЙ ГРАФИК

№ п/п	Дата проведения занятия		Наименование тем	Количество часов		
	План	Факт		Всего	Теория	Практика
1			Инструктаж по технике безопасности в компьютерном классе	2	2	
			Раздел 1. Знакомство с языком Python	4	1,5	2,5
2			Общие сведения о языке	1	1	
3			Режимы работы. Практическая работа: Режимы работы с Python	1	0,5	0,5
4			Тест №1. Знакомство с языком Python	2		2
			Раздел 2. Переменные и выражения	8	4	4
5			Переменные Практическая работа: Переменные	2	1	1
6			Выражения Практическая работа: Выражения	2	1	1
7			Ввод и вывод	2	1	1
8			Задачи на элементарные действия с числами. Практическая работа Задачи на элементарные действия с числами	2	1	1
			Раздел 3. Условные предложения	10	2	8
9			Логические выражения и операторы. Практическая работа: Логические выражения	2	1	1
10			Множественное ветвление Практическая работа: Множественное ветвление	2	1	1
11			Самостоятельная работа №1 по теме «Условные операторы».	2		2
12			Зачетная работа №1. "Составление программ с ветвлением".	2		2
13			Тест №3."Условные операторы".	2		2
			Раздел 4. Циклы	10	3	7
14			Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2	1	1
15			Оператор цикла for Практическая работа Решение	2	1	1

		задачи с циклом for.			
16		Занятие16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	2	1	1
17		Самостоятельная работа №2 "Составление программ с циклом"	2		2
18		Тест №4. Циклы	2		2
		Раздел 5. Функции	10	3	7
19		Создание функций Практическая работа Создание функций	2	1	1
20		Локальные переменные Практическая работа Локальные переменные	2	1	1
21		Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2	1	1
22		Самостоятельная работа №3 по теме "Функции"	2		2
23		Тест № 5. Функции	2		2
		Раздел 6. Строки - последовательности символов	6	2	4
24		Строки Практическая работа: Строки	2	1	1
25		Срезы строк	2	1	1
26		Примеры решения задач со строками Практическая работа: Решение задач со строками.	2		2
		Раздел 7. Сложные типы данных	10	6	4
27		Списки Практическая работа: Списки	2	1	1
28		Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	2	1	1
29		Матрицы	1	0,5	0,5
30		Тест №7. Списки	1	0,5	0,5
31		Кортежи	1	1	
32		Введение в словари	1	1	
33		Множества в языке Python	2	1	1
		Раздел 8. Стиль программирования и отладка программ	12	6	6
34		Стиль программирования	2	1	1
35		Отладка программ	2	1	1
36		Зачет по курсу «Программирование на языке Python»	6	2	4
37		Что дальше?	2	2	
		ВСЕГО	72	29,5	42,5

2.2 ФОРМЫ АТТЕСТАЦИИ/КОНТРОЛЯ

Входящий контроль осуществляется при комплектовании группы в начале учебного года. Цель – определить исходный уровень знаний обучающихся, определить формы и методы работы с обучающимися.

Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу обучающихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточный контроль осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

Итоговый контроль осуществляется в конце учебного года.

Форма контроля: защита творческого проекта.

Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

Способы оценивания уровня достижений обучающихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты обучающихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать последующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.);
- проверка достигаемых обучающимися результатов производится в следующих формах:
 - текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
 - текущая диагностика и оценка педагогом деятельности учащихся;

- итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;
- Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

2.3 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Входящий контроль: *Тестирование (Приложение 1)*

Промежуточный контроль: *Тестирование, решение задач (Приложение 2)*

Итоговый контроль: *обучающимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.*

2.4 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2025 года.

Методы обучения:

1. **Словесный:** объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.
2. **Наглядный:** применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. **Практический:** индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. **Интерактивный:** создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Технологии:

1. **Технология проблемного диалога.** Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.
2. **Технология коллективного взаимообучения** («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучающихся самостоятельность и коммуникативные умения.
3. **Игровая технология.** Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.
4. **Элементы здоровьесберегающих технологий** являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал обучающихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

2.5 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Материально-техническое обеспечение:

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и обучающихся;
- канцтовары;

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JatBrainsPyCharm;
- FoxitReader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

2.6 ВОСПИТАТЕЛЬНЫЙ КОМПОНЕНТ

В центре воспитательного процесса МБУ ДО Станция юных техников находится личностное развитие обучающихся, формирование духовно-нравственных ценностей, способности к осуществлению ответственного выбора собственной индивидуальной образовательной траектории, способности к успешной социализации в обществе.

Профессионализм педагога способствует обеспечению позитивной динамики развития личности ребенка. Сотрудничество, партнерские отношения педагога и обучающегося, сочетание усилий педагога по развитию личности ребенка и усилий самого ребенка по своему саморазвитию - являются важным фактором успеха в достижении поставленной цели.

Цель воспитательной работы в рамках реализации программы «Программирование на языке Python» формирование общей культуры как составляющей личности ребёнка через воспитание в творческом коллективе, приобщение к техническому творчеству, развитие творческого созидательного мировоззрения.

Задачи:

- ✓ воспитание патриотизма, гражданского сознания;
- ✓ развитие лидерских качеств через самоуправление и организацию коллективно-творческой деятельности;
- ✓ построение системы взаимодействия с семьёй и местным социумом, реализация творческого потенциала обучающихся.

Приоритеты воспитательной деятельности:

- ✓ создание условий для воспитания здоровой, счастливой, свободной, ориентированной на труд личности;
- ✓ формирование внутренней позиции личности по отношению к окружающей социальной действительности;

Особенности организуемого воспитательного процесса

План воспитательной работы составлен в соответствии со Стратегией развития воспитания в РФ на период до 2025 года.

Настоящий План имеет следующую структуру:

Основные направления воспитательной работы:

- гражданское воспитание;
- развитие социальных институтов воспитания (семья, консультирование родителей и т.д.);
- духовно-нравственное воспитание; приобщение детей к культурному наследию; популяризация научных знаний. Приоритеты воспитательной деятельности:

1. создание условий для воспитания здоровой, духовной, самостоятельной личности, обогащенной научными знаниями, готовой к сознательной творческой деятельности, нравственному поведению;

2. формирование внутренней позиции личности по отношению к окружающей социальной действительности.

Особенности организуемого воспитательного процесса.

Содержание воспитательной работы в учреждении планируется по направлениям в соответствии с ФГОС, Стратегией развития воспитания в РФ на период до 2025г.:

1. Духовно-нравственное воспитание.
2. Социальное воспитание.
3. Спортивно-оздоровительное воспитание
4. Общеинтеллектуальное воспитание.
5. Общекультурное воспитание.

Виды, формы и содержание деятельности: комбинированное занятие; беседа; игра; викторина, творческая встреча, соревнование, экскурсия и др.

Воспитательное значение игры, ее влияние на развитие ребенка трудно переоценить. Игра органически присуща детскому возрасту и при умелом руководстве со стороны взрослых способна творить чудеса. Ленивого она может сделать трудолюбивым, незнайку - знающим, неумелого - умельцем. Игра помогает сплотить детский коллектив, включить в активную деятельность детей замкнутых и застенчивых. В играх воспитывается сознательная дисциплина, дети приучаются к соблюдению правил, справедливости, умению контролировать свои поступки, правильно и объективно оценивать поступки других.

Основными формами работы с детьми в объединениях являются фронтальная, групповая, индивидуальная.

Планируемые результаты.

- будет сформирована общественная активность личности, сформирована гражданская позиция;
- будет выстроена система взаимодействия с семьёй и местным социумом; сформирована потребность в реализации творческого потенциала обучающихся.

Календарный план воспитательной работы

Мероприятия	Сроки проведения	Цель
Духовно-нравственное воспитание		
День Конституции РФ	декабрь	• формирование у обучающихся понимания значения Конституции и государственных символов Российской Федерации

		<ul style="list-style-type: none"> воспитание чувства уважения, гордости, патриотизма расширение кругозора и повышение общей культуры учащихся
Социальное воспитание		
Беседа «Мои цели в жизни»	Февраль - март	Формирование у детей ценного отношения к своему здоровью и здоровому образу жизни, профилактика алкоголизма и наркомании. Социально-нравственное оздоровление молодежной среды.
Спортивно-оздоровительное		
Групповые выходы на оздоровительные и спортивно-массовые мероприятия.	Сентябрь - октябрь	Создание атмосферы доброжелательности, дружеского состязания и коллективного общения
Общеинтеллектуальное		
мероприятие по информатике "И в шутку, и всерьез"	март	воспитывать самостоятельность, целеустремленность, умение работать в команде, ответственность в достижении цели.
Общекультурное		
Совместные выходы учебной группы в музей.	Апрель-май	Развивать интерес детей к истокам русской национальной культуры

План-график воспитательной работы в объединении «Программирование на языке Python»

дата		тема
сентябрь		Игра-приветствие
		Беседа: «Техника безопасности при работе в Компьютерном классе»
октябрь		Беседа: «Как вести себя на занятиях в объединении»
		Беседа: «Бережное отношение к имуществу СЮТ»
ноябрь		Беседа: «Правила перехода улиц и дорог»
		Беседа: «Поддержание чистоты и порядка в объединении»
декабрь		Беседа: «Главные правила вежливости в общении»
		Инсценировка детей на тему: «Самый вежливый у нас...».
		Беседа: «Подарок–символ уважения».
январь		«Золотые руки»-почему так говорят.

		Беседа: «Чаще улыбайся, если хочешь нравиться людям».
		Беседа: «Что такое справедливость».
февраль		Беседа-викторина «История Олимпийских игр»
		Беседа: «История авиации, флота ,бронетанковой техники».
		Беседа: «Закаляйся, если, хочешь быть здоров».
март		Беседа: «Масленица–древний славянский праздник»
		Беседа: «Урок вежливости»
апрель		Игровая минутка «1апреля - день смеха»
		Беседа-викторина «Шагает эра программирования вперед»
май		Беседа: «Патриотизм и любовь к Родине».
		Беседа-размышление: «Мой лучший друг».
		Беседа: «Не пробовать, не начинать» (о вреде алкоголя и пагубных привычек).

3. СПИСОК ЛИТЕРАТУРЫ

Нормативно-правовые акты и документы

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ
2. Приказ Министерства образования и науки РФ от 29.08.2013 №1008 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Письмо Министерства образования Российской Федерации от 18 июня 2003 №28-02-484/16 «Требования к содержанию и оформлению образовательных программ дополнительного образования детей»,
4. Постановление Главного государственного санитарного врача Российской Федерации от 4.07.2014 г. «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».
5. Стратегия инновационного развития Российской Федерации на период до 2020 года. Распоряжение правительства Российской Федерации от 08.12.2011 № 2227-р.
6. Концепция развития дополнительного образования детей. Распоряжение Правительства Российской Федерации от 24 апреля 2015 года № 729-р.
7. Стратегия развития воспитания в Российской Федерации на период до 2025 года». Распоряжение Правительства Российской Федерации от 29 мая 2015 года № 996-р.

Литература для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

Литература для родителей и обучающихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

Диагностические материалы

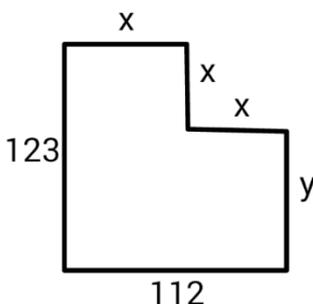
Входящий контроль

Низкий уровень: 0-70% выполненных заданий;
 Средний уровень: 70-85% правильно выполненных заданий;
 Высокий уровень: 85-100% правильно выполненных заданий.

Для обучающихся – 14-17 лет

Фамилия, Имя	
--------------	--

1. Из прямоугольника вырезали квадрат со стороной x , получилась фигура как на рисунке. Чему равна сумма цифр y ?



2. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

3. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, если по зеленым — 31 кусок.

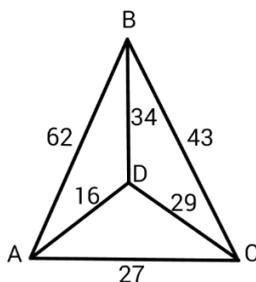
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

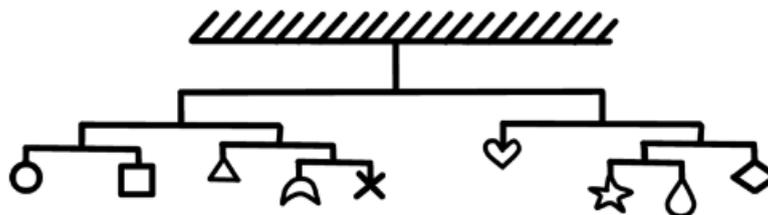
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



6. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

7. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

8. Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

Значит, некоторые клёны быстро желтеют.

Да

Нет

9. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

Для обучающихся 11-13 лет

Фамилия, Имя.	
---------------	--

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

--

3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

--

4. Как с помощью только одной палочки образовать на столе треугольник?

--

5. В каком месяце болтливая девочка говорит меньше всего?

--

6. Что становится на треть больше, если его поставить вверх ногами?

--

7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

--

8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.

Вопрос: Сколько кусков пиццы получилось у Юры.

--

9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.

Вопрос: Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

--

Итоговый контроль
Диагностическая карта

№/п	Имя	Входящая Диагностика	Промежуточная диагностика	Итоговая диагностика
1				
2				
3+				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

Низкий уровень

Обучающиеся должны знать/ понимать:

- технику безопасного поведения во время занятий;
- правила поведения в общественных местах,
- понятие программы Python; общую структуру программы;
- основные типы данных;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов условием;
- формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;
- способ описания кортежа;
- способ описания словаря;
- основные операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Обучающиеся должны уметь:

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе;

- выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога;
- определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

Средний уровень

Обучающиеся должны знать/ понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- основные шаги работы над проектом, его презентации.

Обучающиеся должны уметь:

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;

- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень

Обучающиеся должны знать/ понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приемы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- Общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;

- основные рекомендации при написании программ;
 - правила и этапы работы над проектом;
 - приемы успешной презентации проекта.
- Обучающиеся должны уметь:*
- выполнить установку программы;
 - выполнить простейшую программу в интерактивной среде;
 - написать комментарии в программе;
 - решать задачи на элементарные действия с числами;
 - использовать условный оператор;
 - создавать сложные условия с помощью логических операторов;
 - определять вид цикла, наиболее удобный для решения поставленной задачи;
 - использовать цикл с условием;
 - определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
 - создавать и использовать функции;
 - использовать механизм параметров для передачи значений;
 - описывать строки;
 - соединять строки;
 - находить длину строки;
 - вырезать часть строки;
 - находить подстроку в строке;
 - находить количество слов в строке;
 - описывать списки;
 - вводить элементы списка;
 - выводить элементы списка;
 - выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
 - использовать вложенные списки;
 - приводить примеры использования вложенных списков (матриц);
 - описывать множества;
 - определять принадлежность элемента множеству;
 - вводить элементы множества;
 - выводить элементы множества.
 - определять вид ошибок и находить ошибки в программе.
 - составлять алгоритмы для решения задач;
 - реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
 - отлаживать и тестировать программы, написанные на языке Python;
 - понимать учебную задачу, анализировать достижение результата;
 - делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
 - понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
 - планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
 - создать проектировочную команду и организовать ее деятельность;
 - разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
 - самостоятельно подготовить презентацию и защитить проект.

Вопросы для промежуточного контроля по усвоению материала
(Промежуточный контроль проводится в последнюю неделю декабря)

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:

11111·1111111

— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)

Запишите число **1.2345e3** в виде десятичной дроби.

Составьте и запишите выражение для вычисления:

2014.0¹⁴(Возвестив 14 степень)

Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

Расставьте скобки в выражении

a and b or not a and not b

в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).

Найдите результат выражения для заданных значений *a* и *b*

Учитывайте регистр символов при ответе.

a = True

b = False

a and b or not a and not b

Отметьте выражения, значения которых равны True:

"239" < "30" and 239 < 30

"239" < "30" and 239 > 30

"239" > "30" and 239 < 30

"239" > "30" and 239 > 30

Укажите результат выражения:

"123" + "42"

Какое значение будет у переменной *i* после выполнения фрагмента программы?

i = 0

while i <= 10:

i = i + 1

if i > 7:

i = i + 2

Сколько итераций цикла будет выполнено в этом фрагменте программы?

i = 0

while i <= 10:

i = i + 1

if i > 7:

i = i + 2

Сколько всего знаков * будет выведено после исполнения фрагмента программы:

i = 0

while i < 5:

print('*')

```
if i % 2 == 0:  
    print('**')  
if i > 2:  
    print('***')  
i = i + 1
```

Определите, какое значение будет иметь переменная *i* после выполнения следующего фрагмента программы:

```
i = 0  
s = 0  
while i < 10:  
    i = i + 1  
    s = s + i  
if s > 15:  
    break  
i = i + 1
```

Определите, какое значение будет иметь переменная *i* после выполнения следующего фрагмента программы:

```
i = 0  
s = 0  
while i < 10:  
    i = i + 1  
    s = s + i  
if s > 15:  
    continue  
i = i + 1
```

Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран. Поддерживаемые операции: +, -, /, *, mod, pow, div, где mod — это взятие остатка от деления, pow — возведение в степень, div — целочисленное деление. Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!". Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты. Для числа π в стране Малевии используют значение 3.14. Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где a, b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (Пример: 123321)

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и сразу после этого выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, после этого считывание продолжать не нужно.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если $n = 7$, то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером $n \times n$, заполненную числами от 1 до n^2 по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь $n=5$)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 10485556620218183357344113440560018432977890985

Владелец Саянова Наталья Андреевна

Действителен с 21.05.2024 по 21.05.2025