



## Методические рекомендации по теме «Язык программирования Python, алгоритм и виды алгоритмов»

### Цель:

- объяснение понятий алгоритм и программа, задачи программирования и функции программиста.

### Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с профессиональной деятельностью программиста;
- знакомство с алгоритмами как основной профессиональной деятельности программиста
- популяризация ИТ специальностей и ранняя профориентация школьников;
- развитие интеллектуальных способностей, логического и критического мышления.

### Планируемые результаты

Личностные: обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных целей и задач.

Предметные: обучающиеся получают представления о профессии и особенностях профессиональной деятельности программиста, алгоритмах как базе программирования и программировании как процессе создания программных продуктов; про возникновение и развитие языка программирования Python.

Метапредметные: обучающиеся получают возможность научиться определять и строить базовые логические конструкции, классифицировать их по типам; научиться формализовать логические конструкции с помощью схем и строить логическое рассуждение.

### Материалы к занятию

Приложение 1: Сценарный план видеоролика.

Приложение 2: Домашнее задание и практика.

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии.

## Ход проведения урока

### 1. Организационный момент.

#### Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, что такое программирование, чем занимается программист и какие «плюсы/минусы» в его работе, научимся понимать алгоритмы и выполнять основные математические операции в Python).

*Проблемная дискуссия* по вопросам:

- Что такое в вашем понимании ввод и вывод информации/данных?
- В каких областях жизни и профессиональной деятельности мы встречаемся с этими понятиями?
- Как понятия ввода и вывода данных связаны с типами компьютерного оборудования?
- Как вы думаете: каким словом (функцией) в программировании можно обозначить ввод данных/вывод данных?

*Итоги дискуссии* (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- программирование - процесс создания программы - кода, выполняющего определенную задачу и решающего проблемы людей;
- программисты - люди, занимающиеся программированием (как профессионалы, так и любители);
- даже если ты программируешь дома действия робота пылесоса, стиральной машины - ты уже можешь считаться в какой-то мере программистом; программирование - “азбука будущего”, основы программирования будут полезны абсолютно всем, а не только тем, кто выберет для себя ИТ специальность.

Преподаватель называет ученикам тему и цели урока.

### 2. Вводный блок.

#### Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия)

*\*см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1)*

### **3. Теоретический блок 1.**

#### **Определение, свойства и типы алгоритмов.**

Преподаватель предлагает ученикам просмотреть ролик (при необходимости останавливая трансляцию и комментируя дополнительно) и ответить на вопросы:

- Что такое алгоритм?
- Какие свойства алгоритма вы узнали?
- Какие три типа базовых алгоритмов используются в программировании?

*\*см. сцены 3 – 7*

### **4. Блок заданий 1.**

#### **Типы алгоритмов.**

Задания выполняются в сопровождении видеоролика с использованием таймера (3 мин) с дальнейшей проверкой в парах учеников.

Оценивается правильность составления, сложность и креативность по 10 бальной шкале, ориентированность на алгоритмы из реальной жизни. Рекомендуется провести выборочный разбор нескольких заданий, акцентируя типичные недочеты.

*\*см. сцены 8 – 9*

### **5. Теоретический блок 2.**

#### **Программирование, плюсы и минусы в работе программиста.**

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Согласны ли они с плюсами и минусами работы программиста?
- Если не согласны, то почему?
- Какие еще аргументы можно привести в пользу и против работы программистом?

*\*см. сцены 10 – 11*

## **6. Блок заданий 2.**

### **Логический код.**

Задание выполняется с помощью видеоролика с использованием таймера для решения (5 мин) и демонстрацией разбора задачи.

*\*см. сцены 12*

## **7. Теоретический блок 3.**

### **Терминал IDLE, основные операции.**

Продолжение демонстрации ролика с последующим запуском терминала IDLE для знакомства с интерфейсом терминала, программированием математических вычислений, изучением особенностей синтаксиса. Дальнейшие практические действия выполняются в терминале IDLE с последующим разбором решения.

*\*см. сцены 13*

## **8. Блок заданий 3.**

### **Математические действия.**

Продолжение демонстрации ролика с последующим переходом в терминал IDLE для выполнения более сложных математических вычислений в Python. Задание выполняется с таймером, после завершения работы таймера продолжается демонстрация ролика и производится разбор заданий.

*\*см. сцены 14*

## **9. Теоретический блок 4.**

### **Терминал IDLE, основные операции (продолжение).**

Продолжение демонстрации ролика с последующим запуском терминала IDLE для знакомства с интерфейсом терминала, программированием математических вычислений, изучением особенностей синтаксиса. Дальнейшие практические действия выполняются в терминале IDLE с последующим разбором решения.

*\*см. сцены 15*

## **10. Блок заданий 4.**

### **Математические действия.**

Продолжение демонстрации ролика с последующим переходом в терминал IDLE для выполнения более сложных математических вычислений в Python. Задание выполняется с таймером, после завершения работы таймера продолжается демонстрация ролика и производится разбор заданий.

*\*см. сцены 16 (Приложение 1: Сценарный план ролика)*

### **11. Рефлексия. Сообщение домашнего задания.**

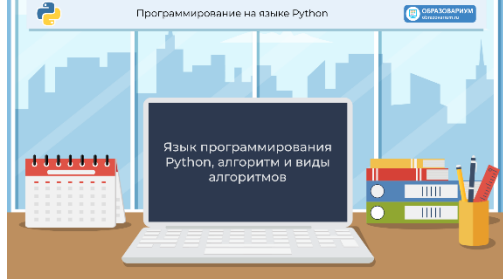

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Подводятся итоги и делаются выводы. Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

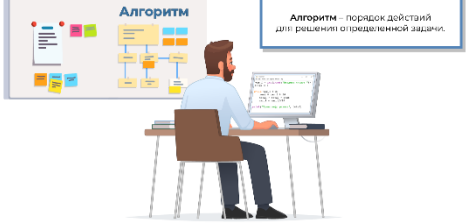


*\*см. сцены 17*

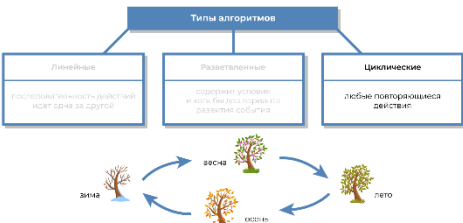
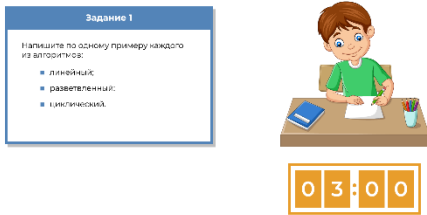
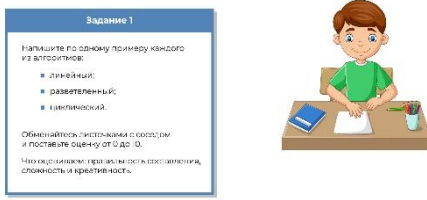
## Сценарный план видеоролика

В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

Таблица. Содержание видеоролика

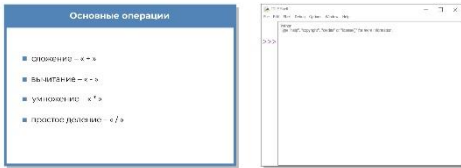
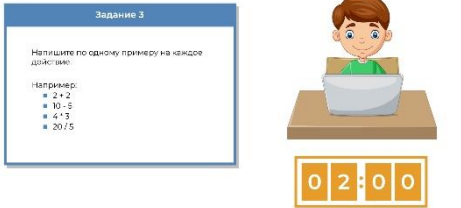
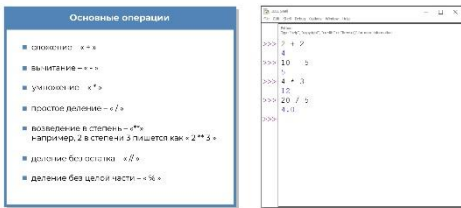
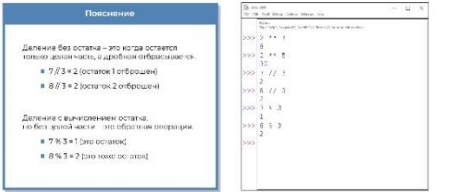
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока.  Язык программирования Python, алгоритм и виды алгоритмов		1 2
	При необходимости, останавливаем трансляцию и дополнительно комментируем.  С этого урока мы начинаем знакомство с языком программирования высокого уровня «Python».		

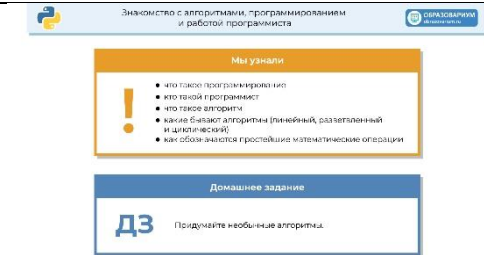
<p>Теоретический блок 1.</p> <p><b>Определение, свойства и типы алгоритмов</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Перейдем к изучению новой темы.</p> <p>Давайте рассмотрим, в чем заключается программирование и работа программиста, а также определим понятие «<b>алгоритм</b>» – мы в дальнейшем будем его очень часто использовать.</p> <p>Алгоритм – порядок действий для решения определенной задачи.</p>	<p>Язык программирования Python, алгоритмы и виды алгоритмов</p>  <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p><b>Свойства алгоритмов</b></p> <ol style="list-style-type: none"> <li>1. Конечность</li> <li>2. Определенность</li> <li>3. Ввод данных</li> <li>4. Результативность</li> <li>5. Универсальность</li> <li>6. Эффективность</li> </ol> <p>Существуют <b>три типа</b> алгоритмов: линейные, разветвленные и циклические.</p> <p><b>В линейном алгоритме</b> последовательность действий идет одна за другой.</p>	<p>Язык программирования Python, алгоритмы и виды алгоритмов</p>  <p>Сцена 4</p>	
	<p><b>В разветвленном алгоритме</b> обязательно содержится условие и хотя бы два варианта развития события.</p> <p>Признаком условия здесь выступают слова «<b>если – то – иначе</b>».</p>	<p>Язык программирования Python, алгоритмы и виды алгоритмов</p>  <p>Сцена 5</p>	<p>Сцена 6</p>

	<p><b>Циклический алгоритм</b> описывает повторяющиеся действия.</p> <p>Его примером может служить любая повторяющаяся операция: чередование времен года, бег секундной стрелки.</p>	<p>Язык программирования Python, алгоритмы и виды алгоритмов</p>  <p>Сцена 7</p>	
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Задание 1. Примеры алгоритмов</b></p> <p>Напишите по одному примеру каждого из алгоритмов:</p> <ul style="list-style-type: none"> <li>• линейный,</li> <li>• разветвленный,</li> <li>• циклический.</li> </ul> <p>Текст вопроса желательно сделать внутри функции <b>input</b>.</p> <p>Не забудьте, что в проекте вводится и сравнивается <b>текстовая</b> информация!</p> <p><b>Разбор задания 1.</b></p> <p>Предлагаем ученикам провести взаимопроверку и оценить работы друг друга. Приводим критерии, на которые нужно ориентироваться при оценивании.</p>	<p>Язык программирования Python, алгоритмы и виды алгоритмов</p>  <p>Сцена 8</p> <p>Язык программирования Python, алгоритмы и виды алгоритмов</p>  <p>Сцена 9</p>	<p>8 9</p>



<p>Теоретический блок 2.</p> <p><b>Программирование, плюсы и минусы в работе программиста</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Функция языков программирования.</p> <p><b>Программирование</b> – процесс создания компьютерных программ с использованием языков программирования</p> <p><b>Программист</b> – это переводчик между обычным человеком и компьютером, специалист, создающий для программы исходный код</p> <p><i>Обсудите плюсы и минусы работы программиста.</i></p> <p><b>Плюсы</b> в работе программиста</p> <ul style="list-style-type: none"> <li>- высокая средняя зарплата</li> <li>- востребованность</li> <li>- свободный график</li> <li>- возможность работы удаленно</li> </ul> <p><b>Минусы</b> в работе программиста</p> <ul style="list-style-type: none"> <li>- сидячий образ</li> <li>- необходимость постоянно совершенствоваться</li> </ul>	<p>Сцена 10</p>	<p>10 11</p>
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p>Напишите программу, следуя которой можно пройти от стрелки до звездочки.</p> <p>Использовать можно только следующие команды: «<b>вперед</b>» и количество клеток, поворот <b>налево</b>, поворот <b>направо</b>.</p> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<p>Сцена 12</p>	<p>12</p>

<p>Теоретический блок 3.</p> <p><b>Терминал IDLE, основные операции</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Основные математические операции:</p> <p>Для <b>сложения</b> и <b>вычитания</b> используются знаки «+» и «-»</p> <p>Для <b>умножения</b> – «*», для обычного <b>деления</b> – знак «/»</p>	 <p>Знакомство с алгоритмами, программированием и работой программиста</p> <p>Сцена 13</p>	13
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 3</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Напишите по одному примеру на каждое действие. Например:</p> <p>2+2 10-5 4*3 20/5</p>	 <p>Знакомство с алгоритмами, программированием и работой программиста</p> <p>Сцена 14</p>	14
<p>Теоретический блок 4.</p> <p><b>Терминал IDLE, основные операции (продолжение)</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Продолжим изучение основных математических операций:</p> <ul style="list-style-type: none"> <li>- возведение в степень – «**»</li> <li>- деление без остатка – «//»</li> <li>- деление без целой части – «%»</li> </ul>	 <p>Знакомство с алгоритмами, программированием и работой программиста</p> <p>Сцена 15</p>	15
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 4</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Напишите по одному примеру на каждое действие:</p> <ul style="list-style-type: none"> <li>• возведение в степень</li> <li>• деление без остатка</li> <li>• деление без целой части.</li> </ul>	 <p>Знакомство с алгоритмами, программированием и работой программиста</p> <p>Сцена 16</p>	16

	<i>После окончания времени провести разбор решения, анализируя результаты выполненных операций.</i>		
Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги:</b></p> <ul style="list-style-type: none"> <li>- узнали, что такое программирование, чем занимается программисты и какие «плюсы/минусы» в его работе;</li> <li>- что такое алгоритм, какие бывают алгоритмы;</li> <li>- познакомились с основными математическими операциями в Python.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 17</p>	17

## Приложение 2

### Домашнее задание.

Попробуйте придумать несколько примеров собственных алгоритмов. Чем необычнее – тем лучше.

### Практика.

Алгоритм «Бинарный поиск»

Имеется отсортированная по возрастанию последовательность случайных чисел (могут быть представлены любые числа).

11	25	33	47	54	69	72	81	96
----	----	----	----	----	----	----	----	----

Каким будет алгоритм поиска произвольного числа из последовательности, при условии, что первая ячейка выбрана заранее (до объявления числа), а следующие ячейки идут только по возрастанию или только по убыванию?

Сколько попыток максимально вам потребуется?

## Алгоритм «Пузырьковая сортировка»

Дана неотсортированная последовательность чисел:

9	7	16	6	1	4	18
---	---	----	---	---	---	----

Как ее можно отсортировать, если разрешено сравнивать между собой только любые соседние ячейки?

*Ответы:*

1. Если перебирать последовательность подряд, начиная с начала, то может понадобиться до 9 попыток. Но это не самый оптимальный путь. Лучше начинать с середины последовательности, где сейчас число 54. Тогда если загаданное число больше, то дальше идем по возрастанию, иначе – по убыванию. Тогда максимально понадобится 5 попыток.
2. Это один из самых известных алгоритмов сортировки. Здесь нужно последовательно **сравнивать значения соседних элементов и менять числа местами, если предыдущее оказывается больше последующего**. Таким образом, элементы с большими значениями оказываются в конце списка, а с меньшими – остаются в начале. Рассмотрим на примере:

9, 7, 16, 6, 1, 4, 18  
7, 9, 16, 6, 1, 4, 18  
7, 9, 6, 16, 1, 4, 18  
7, 6, 9, 16, 1, 4, 18  
6, 7, 9, 16, 1, 4, 18  
6, 7, 9, 1, 16, 4, 18  
6, 7, 1, 9, 16, 4, 18  
6, 1, 7, 9, 16, 4, 18  
1, 6, 7, 9, 16, 4, 18  
1, 6, 7, 9, 4, 16, 18

1, 6, 7, 4, 9, 16, 18  
1, 6, 4, 7, 9, 16, 18  
1, 4, 6, 7, 9, 16, 18

### Приложение 3

#### **Краткие организационно-методические рекомендации по организации работы на занятии.**

«Язык программирования Python, алгоритм и виды алгоритмов».

*Перед началом* ролика педагог может поинтересоваться: занимался ли кто из детей программированием и что они знают о Python.

*Включается ролик* и просматривается блок теоретического материала № 1, после которого учитель может дать дополнительную информацию по представленной теме. Рекомендуется записать на флип-чарте основную информацию ролика – типы алгоритмов

Перед просмотром ролика с блоком задания № 1 необходимо обеспечить учащихся листочками и ручками для работы.

По окончании блока разбора № 1 можно поинтересоваться, какие работы получили высокие баллы и зачитать эти алгоритмы вслух.

После просмотра блока теоретического материала № 2 можно снова сделать паузу для рассмотрения вопроса – интересна ли ребятам профессия программиста после того, что они услышали.

Перед просмотром ролика с блоком задания № 2 необходимо обеспечить учащихся листочками и ручками для работы. По окончании блока разбора № 2 можно поинтересоваться – получилось ли у кого написать абсолютно точный алгоритм и отметить отличившихся учеников.

Перед блоком заданий № 3 необходимо проследить, чтобы у всех был открыт компьютерный терминал для практической работы.

Также рекомендуется продублировать на флип-карте информацию об основных операциях, для чего ролик можно поставить на паузу.

После каждого задания необходимо чтобы ребята запускали свой код и отслеживать возможные ошибки в написании. В случае их появления рекомендуется объяснить почему они появились

***После просмотра ролика*** – если остается время – можно сделать самостоятельно аналогичные проекты в терминале или проверить качество усвоенного теоретического материала.