

<b>Школа:</b>		
<b>Дата:</b>	<b>ФИО учителя:</b>	
<b>Класс:</b>	<b>Участвовали:</b>	<b>Не участвовали:</b>
<b>Тема урока:</b> Основные алгоритмические конструкции		
<b>Цели обучения, которые достигаются на данном уроке</b>		
<b>Цели урока</b>		
<b>Критерии оценивания</b>	Учащийся: <ul style="list-style-type: none"> <li>- поясняет принцип работы операторов;</li> <li>- исправляет ошибки в программе с оператором;</li> <li>- самостоятельно разрабатывает алгоритм и программу для решения задачи</li> </ul>	
<b>Воспитание ценностей</b>	Данный урок направлен на развитие ценностей академической честности, сплоченности и умения работать в команде, ответственности и лидерства. Привитие ценностей осуществляется посредством установления правил работы в группе, оказания поддержки менее способным учащимся.	
<b>Предварительные знания</b>	Синтаксис языка программирования Java. Отличия его от других языков.	
<b>Межпредметные связи</b>	Математика	
<b>Запланированные этапы урока</b>	<b>Запланированная деятельность на уроке</b>	<b>Ресурсы</b>
<b>Начало урока</b> _7_ мин	<b>1. Организационный момент.</b> Мотивация к учебной деятельности. Целеполагание. Совместно с учащимися определяются цели урока <b>2. Повторение материала</b> с целью актуализации знаний. Какие языки программирования вы знаете? На каких из них вы можете написать программу? Какие средства нужны, что бы создать исполняемый файл из вашего кода?	Презентация
<b>Середина урока</b> _25_ мин	<b>Основные алгоритмические конструкции</b> Различают четыре основные алгоритмические конструкции: линейный, ветвящийся, циклический и рекурсивный алгоритм. Эти конструкции по отдельности и в комбинации применяются для построения сложных алгоритмов. Рассмотрим каждую из этих конструкций более подробно. <div style="text-align: center;"> <pre> graph TD     A[Алгоритмические конструкции] --&gt; B[линейный алгоритм]     A --&gt; C[ветвящийся алгоритм]     A --&gt; D[циклический алгоритм]     A --&gt; E[рекурсивный алгоритм]           </pre> </div> <p>Линейный алгоритм представляет собой последовательность действий, которые выполняются строго друг за другом и только один раз. Такая алгоритмическая конструкция довольно проста, очень надежна и стабильна.</p> <p>Ветвящийся алгоритм является разновидностью линейной алгоритмической конструкции, которая обеспечивает выбор между двумя альтернативными вариантами в зависимости от значений</p>	<a href="https://stepik.org/lesson/13020/step/1?unit=4361">https://stepik.org/lesson/13020/step/1?unit=4361</a>

входных данных. Эта конструкция позволяет алгоритму реагировать на изменяющиеся данные и адаптироваться к ним.

Циклический алгоритм представляет собой последовательность действий, которые могут выполняться несколько раз в зависимости от значений входных данных. Эта конструкция применяется очень часто при обработке набора однотипных данных.

Рекурсивный алгоритм включает в себя последовательность действий, в процессе выполнения которых он обращается сам к себе. На практике встречается не так много задач, требующих применения этой алгоритмической конструкции, однако, если это необходимо, то рекурсивная алгоритмическая конструкция делает алгоритм более компактным и гибким.

Рассмотрим применение наиболее простых линейных и ветвящихся алгоритмических конструкций на конкретных примерах.

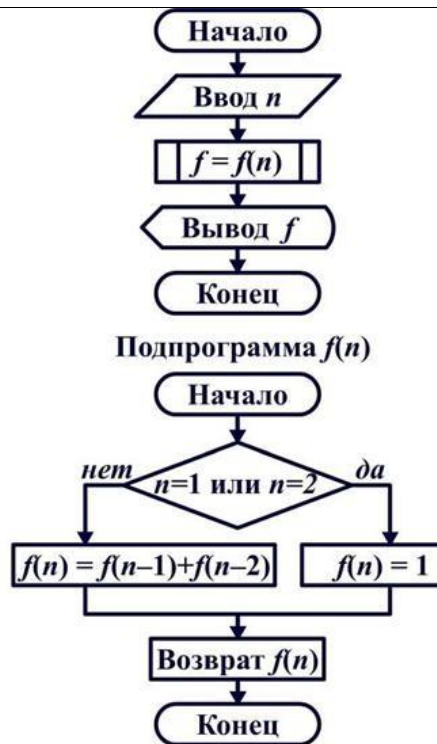
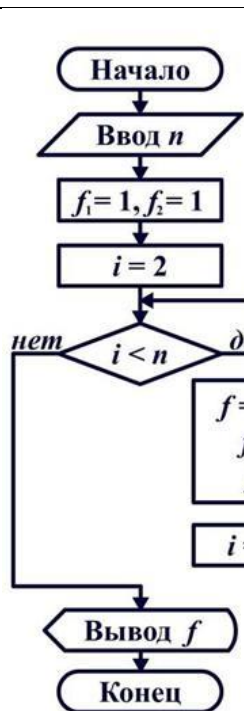
При использовании линейной алгоритмической конструкции (рисунок

6.4, а) сначала происходит ввод с клавиатуры первого члена арифметической прогрессии  $a_1$ , ее разности  $d$  и количества членов арифметической прогрессии  $n$ . На основании введенных данных по формуле вычисляется сумма арифметической прогрессии  $S$ , значение которой выводится на экран монитора.

При использовании ветвящейся алгоритмической конструкции (рисунок

6.4, б) осуществляется ввод значения температуры  $T$ . Если значение температуры  $T$  больше 20, то на экран монитора выводится текст «Жарко», иначе - выполняется еще одно условие.

Если значение температуры  $T$  больше 10, то на экран монитора выводится текст «Тепло», иначе - выводится текст «Холодно».



Рассмотрим принципы использования циклического и рекурсивного алгоритмов на примере вычисления чисел Фибоначчи, представляющих собой

ряд чисел, в котором каждое последующее число равно сумме двух предыдущих: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 и т. д.

При использовании циклической алгоритмической конструкции (рисунок

6.5, а) сначала происходит ввод номера определяемого элемента  $n$ , затем двум переменным  $f_1$  и  $f_2$  присваиваются единичные значения. Далее выполняется циклическое выполнение действий  $(n - 2)$  раз, при которых переменной  $f$  присваивается сумма значений переменных  $f_1$  и  $f_2$ . После чего происходит изменение значений переменных: значению переменной  $f_1$  присваивается значение  $f_2$ , значению переменной  $f_2$ , присваивается значение  $f$ . После работы цикла переменная  $f$  содержит значение определяемого числа Фибоначчи, которое выводится на экран монитора.

При использовании рекурсивной алгоритмической конструкции (рисунок

6.5, б) количество действий алгоритма значительно сократится. В основной программе также происходит ввод номера определяемого элемента  $n$ , но затем вызывается подпрограмма, возвращающая значение определяемого числа Фибоначчи, которое выводится на экран монитора. В подпрограмму передается лишь значение  $n$ , при

	<p>этом происходит проверка значения <math>n</math>, с помощью ветвящейся алгоритмической конструкции. Если значение <math>n</math> равно единице или двойке, то подпрограмма возвращает значение, равное единице, иначе происходит вызов этой же подпрограммы, но уже с аргументами <math>(n - 1)</math> и <math>(n - 2)</math>, результат выполнения этих подпрограмм складывается и возвращается в вызывающую ветку алгоритма. Такая вложенная структура позволяет получить значение определяемого числа Фибоначчи, равного сумме двух предыдущих элементов.</p>	
<p><b>Конец урока _8_ мин</b></p>	<p><b>4. Рефлексия</b> Учитель возвращается к целям урока, обсуждая уровень их достижения. Для дальнейшего планирования уроков учащимся задаются вопросы: - что узнал, чему научился; - что осталось непонятным; - над чем необходимо работать. Вопросы обсуждаются устно.</p>	<p>Стикеры</p>
<p><b>Дифференциация – каким образом Вы планируете оказать больше поддержки? Какие задачи Вы планируете поставить перед более способными учащимися?</b></p>	<p><b>Оценивание – как Вы планируете проверить уровень усвоения материала учащимися?</b></p>	<p><b>Охрана здоровья и соблюдение техники безопасности</b></p>