



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)**

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

**КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**Методическое обеспечение обучения решению задач на последовательности в курсе
программирования**

**Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями
подготовки)**

Направленность программы бакалавриата

«Информатика. Экономика»

Проверка на объем заимствований:
67,07 % авторского текста

Работа рекомендована к защите
рекомендована/не рекомендована
« 10 » март 2018 г.
зав. кафедрой И, ИТ и МОИ

Рузаков А.А.

Выполнил:
Студент группы ЗФ 513-094-5-1
Цыпышев Евгений Николаевич

Научный руководитель:
к.п.н., доцент кафедры ИИТ и МОИ
Паршукова Наталья
Борисовна



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ЮУрГГПУ»)**

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

**КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**Методическое обеспечение обучения решению задач на последовательности в курсе
программирования**

**Выпускная квалификационная работа
по направлению 44.03.05 Педагогическое образование (с двумя профилями
подготовки)**

Направленность программы бакалавриата

«Информатика. Экономика»

Проверка на объем заимствований:
_____ % авторского текста

Работа _____ к защите
рекомендована/не рекомендована

« ___ » _____ 20__ г.
зав. кафедрой И, ИТ и МОИ

_____ Рузаков А.А.

Выполнил:

Студент группы ЗФ 513-094-5-1
Цыпышев Евгений Николаевич

Научный руководитель:

к.п.н., доцент кафедры ИИТ и МОИ

_____ Паршукова Наталья
Борисовна.

**Челябинск
2018**

ОГЛАВЛЕНИЕ

Введение.....	3
Глава 1. Теоретические аспекты решения задач на последовательности в курсе программирования.....	4
1.1 Понятие последовательности.....	5
1.2 Задачи на последовательность в школьном курсе информатики и ИКТ ..	10
Выводы по 1 главе.....	21
Глава 2. Особенности обучения решению задач на последовательности в школьном курсе информатики и ИКТ МБОУ «СОШ № 92 г. Челябинска» .	22
2.1. Педагогическое обеспечение обучения решению задач на последовательности в школьном курсе информатики и ИКТ	22
2.2. Совершенствование педагогического обеспечения обучения решению задач на последовательности в школьном курсе информатики и ИКТ	28
Выводы по 2 главе.....	31
Заключение	32
Список литературы	32
Приложение	36

Введение

В государственном стандарте по информатике отмечается, что в результате изучения информатики и ИКТ на базовом уровне обучающийся должен изучить алгоритмизацию и программирование. Изучая программирование, обучающиеся лучше понимают суть работы компьютера, его возможности и ограничения. Программирование помогает обучающимся развивать свои навыки мышления.

В курсе программирования необходимо уделять внимание задачам на последовательности, так как прослеживается межпредметная связь, например, с математикой. В качестве примера можно привести систематические ряды, то есть последовательности, которые сформированы по определенному образцу. Также достаточно хорошо в этих задачах прорабатывается циклическая организация алгоритма и работа с массивами.

Каждый преподаватель информатики сталкивается с огромным количеством вопросов при построении преподавания обучающимся информатики и ИКТ: как построить презентацию материала, какие методические разработки следует использовать, в какой форме проводить занятия, какие практические задания формулировать, какие материалы использовать обучающимся во время учебы и другие. Все эти вопросы возникают из-за отсутствия четко и полно изложенных учебных материалов и учебников для изучения данной темы.

Количество часов на изучение темы не дает возможности полноценно изучить данную тему в школьном курсе. В этом и заключается несоответствие выделенного количества часов на изучение данной темы объему рассматриваемого материала за заданное количество часов, что отражает несоответствие требованиям выпускника по форме единого государственного экзамена. И здесь возникает проблема, как заинтересовать обучающихся программированием, как научить понимать и решать задачи.

Таким образом, наблюдается противоречие, между необходимостью расширения внимания к задачам на последовательности и тем, что задачам не уделяется достаточно внимания в курсе программирования, а акцент делается на типовые одинаковые алгоритмы из курса в курс, которые не связаны с математикой (отсутствуют межпредметные связи).

Все выше сказанное и определило актуальность исследования, цель и задачи.

Цель исследования – проанализировать методическое обеспечение обучения решению задач на последовательности в школьном курсе информатики и ИКТ в МБОУ «СОШ № 92 г. Челябинска» и разработать методические и дидактические материалы для изучения темы «Последовательности» в рамках прохождения темы «Циклы» в 9 классах.

Исходя из объекта, предмета и цели исследования, необходимо решить следующие **задачи**:

- изучить теоретические аспекты решения задач на последовательности в школьном курсе информатики и ИКТ;
- провести анализ методического обеспечения обучения решению задач на последовательности в школьном курсе информатики и ИКТ;
- разработать методическое обеспечение по обучению решению задач на последовательности в школьном курсе информатики и ИКТ.

Практическая значимость работы состоит в том, что методические разработки и дидактические материалы можно использовать для углубления понимания циклических алгоритмов в программировании.

Гипотеза: если в программу основного общего образования в рамках изучения темы «Циклы» курса программирования, включить вопросы, связанные с изучением темы «Последовательности», то это будет способствовать повышению качества изучения темы «Циклы», а так же способствовать осуществлению межпредметных связей на уроке информатики.

Глава 1. Теоретические аспекты решения задач на последовательности в курсе программирования

1.1 Понятие последовательности

С задачами на последовательности мы довольно часто сталкиваемся в практической деятельности. Например, при подсчете процентов по вкладам и кредитам используются арифметические и геометрические прогрессии, при рассмотрении состояния экономики рассматриваются последовательности Фибоначчи (например, правило золотого сечения).

Рассмотрим понятие последовательности.

Если каждому значению n из натурального ряда чисел $1, 2, \dots, n, \dots$ ставится в соответствие по определённому закону некоторое вещественное число, то множество занумерованных вещественных чисел можно называть числовой последовательностью или просто последовательностью.

Последовательность – это набор из бесконечного количества элементов, которые занумерованы, т. е. имеют порядковые номера: $1, 2, 3, 4, 5$ и так далее. То есть это, фактически, любая функция от натурального аргумента. Под числовой последовательностью обычно подразумевают такую, состоящую из вещественных чисел. На последовательностях строится вся основа математического анализа, и им уделяется много внимания на занятиях по программированию [1, с. 64].

Особенность числовых последовательностей – бесконечное количество элементов в них. Именно это мешает как пониманию, так и исследованию их, но в то же время, и несёт основную пользу этих объектов. Ведь нельзя ни записать, ни хотя бы представить себе сразу все элементы последовательности.

Проще всего задать последовательность просто формулой, зависящей от номера элемента n . Следует обратить внимание и на обозначение: если у нас есть последовательность из элементов $a_1, a_2, \dots, a_n, \dots$, то мы будем записывать её как $\{a_n\}$.

Можно придумать сколь угодно сложные (и красивые) примеры последовательностей. Формула, выражающая n -й элемент последовательности, может зависеть от $(n-1)$ -го элемента или даже от всех предыдущих. Вот, например, первые 1000 элементов последовательности $a_n = \cos(na_{n-1}) + a_{n-2}$.

Изучая суть последовательности необходимо обратить внимание на ее предназначение. В чистом виде задачи на последовательность встречаются довольно редко, но на их основе строится весь материал курса, то есть они являются базой для дальнейшего изучения программирования, как в рамках школьной программы, так и в дальнейшей образовательной или профессиональной сфере.

Например, анализ числовых последовательностей может пригодиться, если есть какой-то многошаговый процесс, и есть потребность узнать, чем он закончится. Хороший, почти классический пример такого процесса – изменение численности популяции [11, с. 14].

Пусть сотня людей живет в каком-нибудь абстрактном поселении. И пусть это не касается болезней, голода, войн и миграции: люди в нем только рождаются и умирают. Пусть μ – рождаемость, а ν – смертность. Это такие числа, что при населении в N жителей в среднем за год рождается μN жителей и умирает νN жителей. (Это необязательно целые числа, но фраза в среднем позволяет нам оперировать вещественными.) Тогда если $a(n-1)$ – население посёлка в $(n-1)$ -м году, то его население в n -ном выражается формулой:

$$a_n = (1 + \mu - \nu) * a_{n-1}$$

Таким образом, задав начальное население посёлка $a(0)=100$, вы можете получить последовательность, описывающую изменение его населения для остальной части истории (при этом не стоит забывать, насколько далёк мир от настоящего, в котором на демографию влияют сотни факторов).

Теперь вернёмся от случайностей к определённым. Внимательно посмотрев на нашу формулу, выражающую следующий член

последовательности через предыдущий, можно прийти к выводу, что вся последовательность описывается простым соотношением:

$$a_n = a(0) * (1 + \mu - \nu)^n$$

Усложнив модель и введя в неё дополнительные факторы и зависимости, формулу можно значительно усложнить. Решив данную задачу с учетом разных факторов (увеличение или снижение рождаемости, рост или уменьшение смертности) можно составить прогноз развития города (рост или вымирание).

Встает вопрос, чем же «закончится» последовательность? Чтобы научиться отвечать на такой вопрос, нужно уточнить понятие «закончится». Ведь последовательности бесконечны, и ничем не «заканчиваются» в прямом смысле этого слова. Рассмотрим конкретный пример. Тот же многошаговый процесс. Когда можно считать, что процесс закончился? Например, когда состояние изучаемого объекта больше не изменяется, то есть когда ничего не происходит. Это значит, что, начиная с некоторого элемента, все последующие элементы последовательности равны между собой. Тогда константа, которой они все равны, описывает некое конечное состояние, финал процесса.

Это хорошая характеристика, удобная и понятная; но очень мало последовательностей, встречающихся в задачах, так заканчивают свой путь. Взять ту же популяцию: ни вымирание, ни размножение не остановилось на каком-либо рубеже и никогда не остановится. При вымирании население будет убывать и сколь угодно близко приближаться к нулю, но никогда его не достигнет (так как в задачах используются вещественные числа, «средние», а не целые). И, тем не менее, когда виден процесс вымирания, так уверенно приводящий популяцию к нулю, интуитивно приходит понимание, что дело «заканчивается» нулём. Поэтому можно говорить, что последовательность «заканчивается» на константе, если при росте элементы этой последовательности сколь угодно близко сходятся к этой константе [1, с. 66].

Формальное и чистое определение этому явлению дал французский математик Огюстен Луи Коши. Оно звучит так: говорят, что числовая последовательность $\{a_n\}$ сходится к числу C , если для любого сколь угодно малого отклонения существует достаточно большой номер $N = N(\varepsilon)$ такой, что для всех последующих номеров $n > N$ элементы a_n этой последовательности отличаются от C не больше, чем на ε , то есть справедливо неравенство $|a_n - C| < \varepsilon$. При этом говорят также, что число C является пределом числовой последовательности $\{a_n\}$; предел обозначается значком $\lim_{n \rightarrow \infty} a_n$. Если попутно $C=0$, то последовательность $\{a_n\}$ называют бесконечно малой.

Именно к бесконечно малым последовательностям относится последовательность, описывающая популяцию города с отрицательным естественным приростом населения (смертностью выше рождаемости). Чтобы доказать, что последовательность $a_n = a(0) * (1 + \mu - \nu)^n$ удовлетворяет определению бесконечно малой последовательности, нужно для любого $\varepsilon > 0$ подобрать такое N , чтобы для всех $n > N$ гарантированно выполнялось неравенство $|a_n| < \varepsilon$. Для удобства давайте перепишем это неравенство, учтя конкретный вид последовательности:

$$a(0) * (1 + \mu - \nu)^n < \varepsilon$$

Модуль убран, потому что это число в левой части неравенства всегда заведомо положительно (в решении неинтересны ситуации с отрицательным или нулевым начальным населением, равно как и с естественным приростом населения меньше единицы). Теперь поделим всё на a_0 (оно ненулевое) и прологарифмируем неравенство:

$$\ln(1 + \mu - \nu)^n < \ln(\varepsilon/a_0)$$

Теперь воспользуемся свойствами логарифма:

$$n * \ln(1 + \mu - \nu) < \ln(\varepsilon/a_0)$$

Таким образом, мы получаем условие на n вида:

$$n > \frac{\ln\left(\frac{\varepsilon}{a_0}\right)}{\ln(1 + \mu - \nu)}$$

Знак изменился, поскольку при убыли населения (то есть $1 + \mu - \nu < 1$) логарифм в знаменателе отрицательный. При $\varepsilon < 1$ логарифм в верхней части также отрицательный (и увеличивается по модулю с уменьшением ε), так что можно взять:

$$N = \left\lceil \frac{\ln\left(\frac{\varepsilon}{a_0}\right)}{\ln(1 + \mu - \nu)} \right\rceil$$

где $\left\lceil \frac{\ln\left(\frac{\varepsilon}{a_0}\right)}{\ln(1 + \mu - \nu)} \right\rceil$ – операция округления вверх. Если же $\varepsilon > 1$, то можно

взять $N=1$. Так было подобрано искомое N для любого $\varepsilon > 0$, а значит, «официально» доказано, что последовательность бесконечно мала. Также можно предположить, что при отрицательном населении город вымирает.

Продолжив демографические исследования можно рассмотреть бесконечно большие последовательности. Если смертность равна рождаемости, то население не изменяется. Это также означает, что последовательность $\{a_n\}$ сходится к числу a_0 .

Если рождаемость выше смертности, то в этом случае последовательность неуклонно возрастает, всё больше отдаляясь и от нуля, и от любого другого числа. У неё просто нет предела в обычном смысле этого слова.

Последовательность $\{a_n\}$ называется бесконечно большой, если для любого сколь угодно большого числа $M > 0$ существует достаточно большой номер такой, что для всех последующих номеров $n > N$ элементы a_n этой последовательности больше M по модулю, то есть выполняется неравенство $|a_n| > M$ [11, с. 16].

Таким образом, последовательность – это набор из бесконечного количества элементов, которые занумерованы, т. е. имеют порядковые номера:

1, 2, 3, 4, 5 и так далее. Различают бесконечно большие и бесконечно малые последовательности. Бесконечно малая последовательность – это последовательность, предел которой равен нулю. Бесконечно большая последовательность – это последовательность, предел которой равен бесконечности.

1.2 Задачи на последовательность в школьном курсе информатики и ИКТ

Все задачи на последовательность в курсе программирования можно условно разделить на 3 группы (рисунок 1).

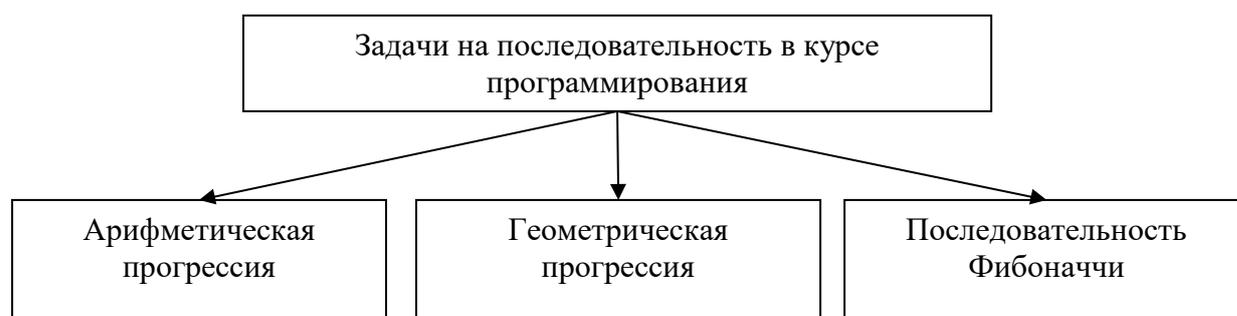


Рис. 1. Задачи на последовательность в курсе программирования

Рассмотрим представленные на рисунке 1 группы подробнее.

Арифметическая прогрессия – это такая последовательность a_1, \dots, a_n , что для всех $n > 1$ разность $a_n - a_{n-1}$ постоянна и равна одному и тому же числу d . Это число d называется знаменателем прогрессии.

Таким образом,

$$a_2 = a_1 + d;$$

$$a_3 = a_2 + d = a_1 + d + d = a_1 + 2d;$$

$$a_4 = a_3 + d = a_1 + 2d + d = a_1 + 3d;$$

....

$$a_n = a_{n-1} + d = a_1 + (n-2) \cdot d + d = a_1 + (n-1) \cdot d;$$

Иногда арифметической прогрессией называют бесконечную последовательность, в которой разница между соседними элементами постоянна. Тогда конечную последовательность a_1, \dots, a_n называют начальным участком арифметической прогрессии.

Рассмотрим пример задачи из курса программирования.

Пример 1. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
var a, b: integer;
begin
  sum:=0;
  a:=1;
  while sum < 78 do begin
    a:=a+4;
    sum:=sum+a;
  end;
  write(a);
end.
```

Решение.

Цикл `while` будет выполняться до тех пор, пока истинно условие $sum < 78$, т. е. переменная `sum` определяет количество выполнений тела цикла.

Значение `sum` является суммой первых n членов арифметической прогрессии. $b = \frac{2a_1 + (n-1)d}{2} n$, b — сумма первых n членов прогрессии, d — разность прогрессии, n — количество членов.

Цикл прервется, когда $sum = \frac{2s_1 + (n-1)d}{2} n < 78$.

Найдем n : $sum = (2s_1 + (n-1)d)n < 156, s_1 = 1, d = 4$ (т. к. `a:=a+4`). Для решения этого неравенства, нам нужно решить квадратичное уравнение $4n^2 - 2n - 156 = 0$, среди его корней нас интересуют только положительные, следовательно, $n = 6.5$

Используя метод интервалов, мы находим, что первое натуральное число n при котором нарушается условие, есть $n = 7$.

Подставив известные параметры в $a_n = a_1 + (n - 1)d$, получаем, что $a_7 = 25$ [3, с. 120].

Пример 2. Определите, что будет напечатано в результате работы программы (записано на разных языках программирования ниже):

Бэйсик	Паскаль
<pre> DIM N, S AS INTEGER N = 3 S = 0 WHILE N <= 7 S = S + N N = N + 1 WEND PRINT S </pre>	<pre> var n, s: integer; begin n := 3; s := 0; while n <= 7 do begin s := s + n; n := n + 1 end; write(s) end. </pre>
Си	Алгоритмический
<pre> #include void main() { int n, s; n = 3; s = 0; while (n <= 7) { s = s + n; n = n + 1; } printf("%d", s); } </pre>	<pre> алг нач цел n, s n := 3 s := 0 нц пока n <= 7 s := s + n n := n + 1 кц вывод s кон </pre>

Решение.

Цикл `while` будет выполняться до тех пор, пока будет истинным условие $n \leq 7$, т. е. переменная n определяет, сколько раз выполнится цикл.

Цикл выполнится $\frac{7-3}{1} + 1 = 5$ раз («+1» потому, что в дроби мы не учитываем то, что при 7 он тоже выполнится).

Заметим, что в s накапливается сумма арифметической прогрессии из 5 членов с разностью 1. Следовательно,

$$s = \frac{2 \cdot 3 + 1(5-1)}{2} \cdot 5 = 25.$$

Вторая группа задач на последовательности в курсе программирования – это геометрические прогрессии.

Геометрическая прогрессия – это числовая последовательность, первый член которой отличается от нуля, и каждый член, начиная со второго, равен предыдущему, умноженному на то же ненулевое число q . Это число q называют знаменателем геометрической прогрессии:

$$b_n = b_{n-1} \cdot q$$

Для геометрической прогрессии (b_n) имеем:

$$b_n = b_1 \cdot q^{n-1};$$

$$S_n = \frac{b_n q - b_1}{q - 1} = \frac{b_1 (q^n - 1)}{q - 1},$$

где $q \neq 1$ – знаменатель прогрессии, а S_n – сумма ее первых n членов.

При $q = 1$

$$S_n = b_1 \cdot n.$$

Чтобы определить геометрическую прогрессию, достаточно знать ее первый член и знаменатель прогрессии.

Сумма бесконечно убывающей геометрической прогрессии (b_n) ($|q| < 1$) определяется равенством:

$$S = \lim_{n \rightarrow \infty} S_n,$$

при этом

$$S = \frac{b_1}{1 - q}.$$

Последовательность (b_n) является геометрической прогрессией тогда и только тогда, когда для любого $n > 1$ верно равенство

$$b_n^2 = b_{n-1} \cdot b_{n+1}$$

В частности, если все

$$b_n > 0,$$

то

$$b_n = \sqrt{b_{n-1} \cdot b_{n+1}}.$$

Квадрат любого (кроме первого) члена геометрической прогрессии равен произведению равноудаленных от него членов:

$$b_n^2 = b_{n-k} \cdot b_{n+k}, k < n, k \in \mathbb{N}.$$

Если $|q| < 1$ и прогрессия имеет бесконечное число членов, то ее называют бесконечно убывающей геометрической прогрессией.

Рассмотрим примеры задач геометрической прогрессий в курсе программирования.

Пример 3.

1. Определите, что будет напечатано в результате работы следующего фрагмента программы:

```
var k, s: integer;
```

```
begin
```

```
  s:=0;
```

```
  k:=0;
```

```
  while s < 80 do begin
```

```
    s:=s+2*k;
```

```
    k:=k+4;
```

```
  end;
```

```
  write(s);
```

```
end.
```

Решение.

Цикл while выполняется до тех пор, пока истинно условие $s < 80$, т. е. переменная k определяет, сколько раз выполнится цикл.

Аккуратно выпишем все s и k:

s 0 0 8 24 48 80

k 0 4 8 12 16 20

При решении задачи необходимо помнить, что условие $s < 80$ проверяется только после $k:=k+4$, поэтому действие $s:=s+2*k$ последний раз выполнится для $k=16$.

Следовательно: ответ 80.

Пример 4. Определите, что будет напечатано в результате выполнения программы (записанной ниже на разных языках программирования).

Бейсик	Паскаль
<pre> DIM N, S AS INTEGER N = 0 S = 1 WHILE S <= 1000 S = S * 3 N = N + 3 WEND PRINT N </pre>	<pre> program B05; var n, s: integer; begin n := 0; s := 1; while s <= 1000 do begin s := s * 3; n := n + 3; end; write(n) end. </pre>
Си	Алгоритмический язык
<pre> #include void main() { int n, s; n = 0; s = 1; while (s <= 1000) { s = s * 3; n = n + 3; } printf("%d", n); } </pre>	<pre> алг нач цел n, s n := 0 s := 1 нц пока s <= 1000 s := s * 3 n := n + 3 кц вывод n кон </pre>

Решение:

Цикл `while` выполняется до тех пор, пока истинно условие $s \leq 1000$, т. е. переменная s определяет, сколько раз выполнится цикл.

Поскольку $3^6 < 1000 < 3^7 = 2187$, следовательно, цикл выполнится 7 раз. На 7-м шаге s станет равной 2187 и условие $s \leq 1000$ окажется не выполненным, цикл прервется. Следовательно, значение n будет равно $7 \cdot 3 = 21$.

Ответ: 21 [7, с. 44].

Следующая группа задач на последовательности – это последовательности Фибоначчи.

Последовательность Фибоначчи, которая стала известна благодаря фильму и книге "Код да Винчи", представляет собой последовательность чисел, полученную итальянским математиком Леонардо Пизанским, более известным как Фибоначчи, в тринадцатом веке. Последователи ученого заметили, что формула, которой подчинена данная последовательность цифр, находит свое отражение в окружающем нас мире и перекликается с другими математическими открытиями, тем самым открывая нам дверь в тайны мироздания.

Последовательность Фибоначчи – это математическая последовательность, каждый элемент которой равен сумме двух предыдущих. Обозначим некий член последовательности как x_n . Таким образом, получим формулу, справедливую для всего ряда: $x_{n+2} = x_n + x_{n+1}$. При этом порядок последовательности будет выглядеть так: 1, 1, 2, 3, 5, 8, 13, 21, 34. Следующим числом будет 55, так как сумма 21 и 34 равна 55. И так далее по такому же принципу.

Известный математик представил свою теорию в виде загадки. Звучит она примерно так. Вы можете поместить пару кроликов в ограниченном пространстве, чтобы узнать, сколько пар кроликов родится в течение одного года. Учитывая характер размножения этих животных, такой, что каждый месяц пара способна производить новую пару, а отовность к размножению у них появляется по достижении возраста двух месяцев, в результате чего он

получил свою знаменитую последовательность чисел: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 - где показано количество новых пар кроликов в каждом месяце.

Рассмотрим последовательность Фибоначчи и пропорциональное соотношение. Этот ряд имеет несколько математических нюансов, которые необходимо учитывать. Он, приближаясь все медленнее и медленнее (асимптотически), стремится к некоторой пропорциональной зависимости. Но это иррационально. Другими словами, это число с непредсказуемой и бесконечной последовательностью десятичных чисел в дробной части. Например, соотношение любого элемента ряда варьируется около цифры 1,618, то превосходя, то достигая его. Следующее по аналогии приближается к 0,618. Что есть обратно пропорциональным к числу 1,618. Если мы поделим элементы через один, то получим 2,618 и 0,382. Как вы уже поняли, они также являются обратно пропорциональными. Полученные числа называются коэффициентами Фибоначчи. А теперь объясним, для чего выполнялись эти вычисления.

Все окружающие объекты различаются по определенным критериям. Один из них - форма. Некоторые привлекают больше, некоторые меньше, а некоторые совсем не привлекают. Замечено, что симметричный и пропорциональный объект гораздо легче воспринимается человеком и вызывает чувство гармонии и красоты. Одно изображение всегда включает в себя части различных размеров, которые находятся в определенной связи друг с другом. Отсюда и ответ на вопрос о том, что называется Золотым сечением. Это понятие означает совершенные отношения целого и частей в природе, науке, искусстве и т. д.

Рассмотрим следующий пример. Берем отрезок любой длины и разделяем его на две части таким образом, что меньшая часть относится к большей части как к сумме (длине всего отрезка) к большей. Итак, примем отрезок с за величину один. Его часть, а будет равна 0,618, вторая часть b, выходит, равна 0,382. Таким образом, соблюдается условие Золотого сечения. Отношение отрезка с к а равняется 1,618. А отношение частей с и b – 2,618.

Получаем уже известные коэффициенты Фибоначчи. По тому же принципу строятся золотой треугольник, золотой прямоугольник и золотой кубоид. Также стоит отметить, что пропорциональное соотношение частей человеческого тела близко к золотому сечению.

Попробуем совместить теорию золотого сечения и известную последовательность итальянского математика. Начнем с двух квадратов первого размера. Затем добавим еще один квадрат второго размера сверху. Нарисуем следующую фигуру с длиной стороны, равной сумме двух предыдущих сторон. Аналогично нарисуйте квадрат пятого размера. И так можно продолжать бесконечно, пока не надоест. Главное, чтобы величина стороны каждого последующего квадрата была равна сумме сторон двух предыдущих. Получаем ряд полигонов, сторонами длины которых являются числа Фибоначчи. Эти фигуры называются прямоугольниками Фибоначчи. Рисуем плавную линию через углы наших полигонов и получаем... спираль Архимеда! Как известно, увеличение высоты шага этой фигуры всегда равномерное. Если включить фантазию, то полученный рисунок можно связать с раковиной моллюска. Отсюда можно сделать вывод, что последовательность Фибоначчи является основой пропорциональных, гармоничных отношений элементов в окружающем мире.

Если присмотреться, то спираль Архимеда (где-то явно, а где-то завуалирована) и, следовательно, принцип Фибоначчи прослеживается во многих привычных природных элементах, окружающих человека. Например, все те же ракушки моллюсков, соцветия обыкновенной брокколи, цветок подсолнечника, шишка хвойных растений и тому подобное. Если мы посмотрим немного дальше, мы увидим последовательность Фибоначчи в бесконечных галактиках. Даже человек, вдохновленный природой и принявший ее формы, создает объекты, в которых можно проследить вышеупомянутую последовательность. Пришло время вспомнить Золотое сечение. Наряду с закономерностью Фибоначчи прослеживаются принципы этой теории. Существует версия, что последовательность Фибоначчи является

своего рода тестом природы для адаптации к более совершенной и фундаментальной логарифмической последовательности золотого сечения, которая почти идентична, но не имеет начала и бесконечна. Закономерность природы такова, что у нее должна быть своя точка отсчета, с которой строится что-то новое. Соотношение первых элементов ряда Фибоначчи далеко от принципов золотого сечения. Однако, чем дальше мы его продолжаем, тем больше это расхождение сглаживается. Чтобы определить последовательность, нужно знать ее три элемента, которые идут один за другим. Для Золотой последовательности достаточно двух. Так как это и арифметическая, и геометрическая прогрессии. Рассмотрим пример из курса информатики.

Пример 5. Вывести на экран элемент ряда Фибоначчи по его порядковому номеру. Например, если на вход было подано 7, то программа должна вывести 13.

Пояснение: Ряд Фибоначчи - это последовательность натуральных чисел, где каждое последующее число является суммой двух предыдущих: 1 1 2 3 5 8 13 21 34 55 89 ...

Решение:

Для реализации программы, мы будем использовать современный язык программирования C#. Этот язык в большинстве случаев подходит для участия в экзаменах и олимпиадах по информатике. Программа будет консольным приложением.

Поскольку в задаче не указано никаких ограничений на максимальное значение введенного числа, то мы будем использовать наиболее емкий Тип, заданный для целых положительных чисел - `ulong`. Диапазон данного типа: 0: 18446744073709551615.

Реализация программы выглядит следующим образом (дополнительный код, связанный с консольным приложением опущен):

```
1 static void Main(string[] args)
2 {
```

```

3     ulong n = 0;
4     ulong res = 1;
5     ulong temp = 0;
6
7     //Считываем строку с клавиатуры
8     //и преобразовываем её в ulong
9     n = Convert.ToUInt64(Console.ReadLine());
10
11    for (ulong i = 1; i < n; i++)
12    {
13        res = res + temp;
14        temp = res - temp;
15    }
16
17    Console.WriteLine(res);
18    Console.ReadKey();
19
20 }

```

В переменной *res* всегда формируется новый член последовательности. В переменной *temp* предыдущий член [12, с. 29].

Таким образом, все задачи по последовательностям в школьном курсе информатики и ИКТ в работе были условно разделены на 3 группы: арифметическая и Геометрическая прогрессии, последовательность Фибоначчи. Арифметическая прогрессия – это такая последовательность a_1, \dots, a_n , что для всех $n > 1$ разность $a_n - a_{n-1}$ постоянна и равна одному и тому же числу d . Это число d называется знаменателем прогрессии. Геометрическая прогрессия - это числовая последовательность, первый член которой отличается от нуля, и каждый член, начиная со второго, равен предыдущему, умноженному на то же ненулевое число q . Это число q называется знаменателем геометрической прогрессии: $b_n = b_{n-1} \cdot q$. Последовательность Фибоначчи – это математическая последовательность, каждый элемент которой равен сумме двух предыдущих.

Выводы по 1 главе

Последовательность - это бесконечное множество пронумерованных элементов, т. е. имеющих порядковые номера: 1, 2, 3, 4, 5 и так далее. Существуют бесконечно большие и бесконечно малые последовательности. Бесконечно малая последовательность - это последовательность, предел которой равен нулю. Бесконечно большая последовательность - это последовательность, предел которой - бесконечность.

Все задачи по последовательностям в школьном курсе информатики и ИКТ в работе были условно разделены на 3 группы: арифметическая и Геометрическая прогрессии, последовательность Фибоначчи.

Арифметическая прогрессия – это такая последовательность a_1, \dots, a_n , что для всех $n > 1$ разность $a_n - a_{n-1}$ постоянна и равна одному и тому же числу d . Это число d называется знаменателем прогрессии.

Геометрическая прогрессия - это числовая последовательность, первый член которой отличается от нуля, и каждый член, начиная со второго, равен предыдущему, умноженному на то же ненулевое число q . Это число q называют знаменателем геометрической прогрессии: $b_n = b_{n-1} \cdot q$.

Последовательность Фибоначчи – это математическая последовательность, каждый элемент которой равен сумме двух предыдущих.

Глава 2. Особенности обучения решению задач на последовательности в школьном курсе информатики и ИКТ МБОУ «СОШ № 92 г. Челябинска»

2.1. Педагогическое обеспечение обучения решению задач на последовательности в школьном курсе информатики и ИКТ

Обучение решению задач на последовательности в школьном курсе информатики и ИКТ рассматриваются в теме «Циклы» и проводится в основном по учебным пособиям Н.Д. Угриновича [18], И.Г. Семакина [13], Н.В. Макарова [2].

Рассмотрим на примере учебных пособий 9 класса содержание обучения решению задач на последовательности.

Учебное пособие Н.Д. Угриновича содержит главу «Основы алгоритмизации и объектно-ориентированного программирования», которая включает в себя следующие параграфы (30 часов):

1. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования. Линейный алгоритм: алгоритмическая структура «ветвление», алгоритмическая структура «выбор», алгоритмическая структура «цикл».

2. Графические возможности объектно-ориентированного языка программирования Visual Basic.

Тема «1.2.4 Алгоритмическая структура «цикл» включает в себя рассмотрение двух типов структур: цикл со счётчиком; цикл по условию [18, с. 23].

Содержание учебника, относительно раздела программирования и алгоритмизации отвечает всем требованиям стандарта, представлена информация в интересном и доступном варианте. Рассматриваются основные языки: Visual Basic, Gambas и OpenOffice.org Basic.

В качестве отрицательных моментов следует выделить отсутствие примеров задач для решения и алгоритмов их решения.

Учебное пособие И.Г. Семакин включает раздел под названием «Управление и алгоритмы» (16 часов) и содержит следующие параграфы:

1. Управление и кибернетика.
2. Управление с обратной связью.
3. Определение свойства алгоритма.
4. Графический учебный исполнитель.
5. Вспомогательные алгоритмы и подпрограммы.
6. Циклические алгоритмы.
7. Ветвление и последовательная детализация.

Тема «Циклические алгоритмы» включает основные темы:

- команда цикла;
- цикл в процедуре;
- блок-схемы алгоритмов;
- цикл с условием.

В разделе представлена задача и пример решения.

Пример 6. Исходное положение: ГРИС – у левого края поля, направление – на восток. Требуется нарисовать горизонтальную линию через весь экран.

Задачу можно решить, написав 15 раз команду шаг (если поперек поля рисунок 15 шагов). Но есть и более короткий вариант программы. Вот он:

пока впереди не край повторять

нц

шаг

кц

Здесь использована команда, которая называется циклом. Формат команды цикла следующий:

пока <условие> повторять

нц

<тело цикла>

кц

Служебное слово `нц` обозначает начало цикла, `кц` – конец цикла. Это первая команда из СКИ, которая реализует обратную связь между графическим исполнителем и управляющим им компьютером. Она заключается в том, что проверяется, не вышел ли ГРИС на край поля, не грозит ли ему следующий шаг или прыжок в этом направлении аварией. Проверяемые условия звучат так: «впереди край?» или «впереди не край?». На что машина получает от исполнителя ответ «да» или «нет».

В приведенном примере проверяется условие «впереди не край?». Если «да», то делается шаг (т. е. выполняется <тело цикла>). Затем происходит возврат на проверку условия, и всё повторяется. Если проверка условия дает отрицательный результат (т. е. впереди край), то выполнение цикла завершается и исполняется следующая после цикла команда программы.

При программировании цикла важно думать о том, чтобы цикл был конечным. Цикл, записанный выше, конечный. Двигаясь в одном направлении, исполнитель обязательно достигнет края, и на этом выполнение цикла закончится.

Ситуация, при которой выполнение цикла никогда не заканчивается, называется зацикливанием. Пусть ГРИС находится в середине поля. Исполнение следующего цикла:

пока впереди не край повторять

`нц`

шаг

поворот

`кц`

никогда не закончится. ГРИС будет бесконечно рисовать квадратик, так как проверка условия «впереди не край?» всегда будет давать положительный ответ [13, с. 33].

В разделе «Программное управление работой компьютера» есть параграф под названием «Программирование циклов». Основные темы:

– этапы решения расчётной задачи на компьютере;

- программирование циклов на Паскале;
- тестирование и отладка программы.

В данном разделе прописан алгоритм решения задач, включающий в себя этапы:

1. Постановка задачи.
2. Математическая формализация.
3. Построение алгоритма.
4. Составление программы на языке программирования.
5. Отладка и тестирование программы.
6. Проведение расчетов и анализ полученных результатов.

Эту последовательность называют технологией решения задачи на компьютере.

Содержание по-своему интересно, и полностью отличается от содержания учебника Угринович Н. Г., здесь изложен совершенно иной материал, дополняющий уже имеющиеся знания на тему последовательности.

В качестве недостатков следует выделить безликость задач, что не позволяет школьникам вникнуть в решение и тем более усвоить его и применять в практической деятельности.

Учебное пособие Н.В. Макарова содержит раздел 2 Программное обеспечение информационных технологий (20 часов), который включает в себя следующие разделы для изучения:

Алгоритмы:

- линейные алгоритмы;
- разветвляющиеся алгоритмы;
- циклические алгоритмы.

Цикл с предусловием включает:

- использование цикла с предусловием для организации повторяющихся действий;
- тестирование циклических алгоритмов;

– использование цикла как универсального для решения разного вида задач;

– использование переменных разного типа для организации цикла;

Цикл с известным числом повторений:

– создание алгоритмов с известным числом повторений;

– тестирование циклических алгоритмов с известным числом повторений в пошаговом режиме;

– изменение параметров цикла.

Цикл постусловием:

– использование цикла с предусловием для решения задач;

– использование разных видов циклов в зависимости от поставленной задачи;

– использование типовых алгоритмов поиска минимума и максимума;

– использование данных разного типа при решении задач.

Рассмотрим пример решения задачи из данного учебника.

Пример 7. Требуется рассчитать время работы батарейки в часах с кукушкой, если известно, что заряда хватает примерно на 1000 звуковых сигналов «ку-ку». Однократный звуковой сигнал звучит, когда минутная стрелка показывает 30 минут. Начало каждого часа сопровождается повторением сигнала столько раз, сколько показывает часовая стрелка (от 1 до 12).

Расчетными данными для этой задачи являются:

t – обозначение текущего часа;

k – количество звуковых сигналов.

Алгоритм «Кукушка» представим на школьном алгоритмическом языке в таблице.

В этом алгоритме повторяются следующие действия:

– определение значения текущего часа;

– определение количества звуковых сигналов.

Эти действия составляют тело-цикла.

Цикл заканчивается, если количество поданных звуковых сигналов превысило 1000, что является признаком выработки ресурса батарейки [2, с. 180].

Таблица 1. Алгоритм «Кукушка» на школьном алгоритмическом языке

Алгоритм	Пояснения	
алг Кукушка	Заголовок алгоритма	
нач цел t, k	Описание типов переменных	
t:=0; k:=0	Начальные значения переменных t, k	
нц	Начало цикла	
t:= t + 1	Тело цикла	Значение текущего часа
k:= k + 1 + mod(t,12)		Общее количество сигналов на текущий час. mod(t,12) — остаток от деления t на 12
кц_при k>=1000	Проверка условия окончания цикла	
вывод "Количество часов =", t	Вывод количества часов на экран	
кон	Конец алгоритма	

Содержание учебника Макаровой Н.В. полностью соответствует требованиям, изложен совершенно иной материал, дополняющий уже имеющиеся знания на тему последовательности. Задачи имеют практические примеры, что позволяет лучше усвоить материал.

В качестве недостатков следует выделить отсутствие обучения школьников решения задач разными языками, что не позволяет вникнуть в решение и тем более усвоить его и применять в практической деятельности более широко.

Таким образом, содержание всех представленных и проанализированных здесь учебников, очень разнообразно, интересно, и каждое по-своему дополняют друг друга. В качестве проблем следует выделить:

1) в учебном пособии Н.Д. Угриновича отсутствуют примеры задач для решения и алгоритмов их решения;

2) в примерах задач представленных в учебных пособиях нет связи с реальной жизнью учеников (кроме учебника Н.В. Макаровой);

3) во всех учебных пособиях при рассмотрении задач не рассмотрена вариантность языков программирования и нет разбивки решения по сложности (уровнях освоения школьниками).

2.2. Совершенствование педагогического обеспечения обучения решению задач на последовательности в школьном курсе информатики и ИКТ

Для совершенствования педагогического обеспечения обучения решению задач на последовательности в школьном курсе информатики и ИКТ предлагается провести 5 практических уроков (40 минут) на тему «Циклы».

Цели уроков:

Образовательные:

1. Повторить понятие термина «Циклы»;
2. Научить применять циклы для решения задач на последовательности;
3. Способствовать осуществлению межпредметных связей на уроке информатики.

Развивающие:

1. Развивать навыки определения количества повторений цикла;
2. Способствовать развитию логического и алгоритмического мышления путем выполнения практической работы;
3. Развивать познавательный интерес и творческую активность.

Воспитательные:

1. Воспитывать культуру общения в группах;
2. Повышать интерес к практическому решению задач.

Тип уроков: изучение нового материала и закрепление ранее изученного.

Вид уроков: комбинированный.

Основные методы, технологии: элементы объяснительно-иллюстративного метода; частично-поискового; наглядный; практического; элементы групповой технологии, информационная технология.

На первом уроке изучается теоретическая часть, даются под запись основные формулы, разбираются примеры задач.

Второй, третий уроки направлены на практическое изучение арифметической и геометрической прогрессии, решение задач на данные темы.

Четвёртый урок направлен на практическое закрепление темы «Последовательности», решение задач на арифметическую, геометрическую прогрессии, последовательность Фибоначчи.

На пятом уроке предлагается провести самостоятельную контрольную работу.

Цели:

- образовательная: закрепление навыков в составлении программ с использованием различных циклов;
- развивающая: развитие умений обобщать, систематизировать и применять знания, полученные ранее;
- воспитательная: формирование навыков самостоятельной работы и воспитания самоконтроля учащихся.

Содержание урока:

1. Организационный момент.
2. Сообщение темы и цели урока.
3. Самостоятельная контрольная работа.

Учащимся предлагается выполнить тестовое задание.

Цель: проверить, владеют ли учащиеся всеми изученными видами последовательностей (арифметической прогрессии, геометрической прогрессии, последовательности Фибоначчи); умеют ли делать оптимальный выбор цикла для каждой конкретной задачи.

По итогам контрольной работы преподаватель выставит оценки.

Тест для учащихся размещен по ссылке:
<https://learningapps.org/display?v=pepb1v8o318>.

После выполнения теста учащиеся обмениваются задачами, составленными дома, и решают их на языке программирования.

Таким образом, для совершенствования педагогического обеспечения обучения решению задач на последовательности в школьном курсе информатики и ИКТ предлагается:

- 1) разработать планы уроков по решению задач на последовательности в школьном курсе информатики и ИКТ в рамках темы «Циклы»
- 2) представить алгоритм решения задач;
- 3) задачи представить в виде реальных жизненных примеров, чтобы ученикам было легче понять суть их решения;

В ходе работы был создан веб-ресурс, расположенный по адресу <http://d96179qr.beget.tech/>, на котором были размещены методическое и дидактическое обеспечение для учителя, связанное с изучением темы «Последовательности» в курсе программирования в 9 классе.

Выводы по 2 главе

Во второй главе квалификационной работы рассмотрены учебные пособия Н.Д. Угриновича, И.Г. Семакина, Н.В. Макаровой 9 класса на предмет наличия задач на последовательности. Было выявлено, что задачи на последовательности рассмотрены в теме «Циклы».

Содержание каждого представленного и проанализированного учебника очень разнообразно, интересно, и каждое по-своему дополняет друг друга. В качестве проблем выделены:

1) в учебном пособии Н.Д. Угриновича отсутствуют примеры задач для решения и алгоритмов их решения;

2) в примерах задач представленных в учебных пособиях нет связи с реальной жизнью.

3) во всех учебных пособиях при рассмотрении задач не рассмотрена вариантность языков программирования и нет разбивки решения по сложности (уровнях освоения школьниками).

Для совершенствования педагогического обеспечения обучения решению задач на последовательности в школьном курсе информатики и ИКТ в квалификационной работе предложено:

1) разработать планы уроков по решению задач на последовательности в школьном курсе информатики и ИКТ в рамках темы «Циклы»;

2) разработать тестовые задания для проверки знаний учеников.

Заключение

Подводя итоги проделанной работы, можно сделать вывод о том, что цель исследования была достигнута, а именно проанализировано методическое обеспечение обучения решению задач на последовательности в школьном курсе информатики и ИКТ в МБОУ «СОШ № 92 г. Челябинска» и разработаны методические и дидактические материалы для изучения темы «Последовательности» в рамках прохождения темы «Циклы» в 9 классах.

Разработанные уроки, рассчитанные на проведение занятий при изучении темы «Циклы» в 9 классе, с одной стороны способствуют развитию логического и алгоритмического мышления путем выполнения практической работы, повышают качество изучения темы «Циклы»; с другой стороны способствуют осуществлению межпредметных связей на уроке информатики.

В процессе исследования были решены следующие задачи:

1. изучены теоретические основы по проблеме исследования;
2. разработаны уроки по изучению темы «Последовательности» в рамках изучения темы «Циклы» в 9 классе;
3. разработана программно–методическая поддержка уроков для учителя в виде образовательного портала.

В подтверждении гипотезы можно сказать, что курс уроков по изучению темы «Последовательности» в рамках изучения темы «Последовательности» ведут к развитию логического и алгоритмического мышления путем выполнения практической работы, повышают качество изучения темы «Циклы», способствуют осуществлению межпредметных связей на уроке информатики.

Список литературы

1. Бодунов, М.А. Математический анализ [Текст] : учеб. пособие / М.А. Бодунова, С.И. Бородина, В.В. Показеев, Б.Л. Теуш, О.И. Ткаченко. – М. : Норма, 2016. – 127 с.
2. Кушниренко, А.Г. Информатика. 7-9 кл. [Текст]: учебник для общеобразоват. учеб. заведений / А. Г. Кушниренко, Г. В. Лебедев, Я. Н. Зайдельман. – М. : Дрофа, 2016. – 336 с.
3. Лапчик, М.П. Методика преподавания информатики [Текст] : учеб. пособие для студ. пед. вузов / М.П. Лапчик, И.Г. Семакин, Е.К. Хеннер ; под общей ред. М. П. Лапчика. – М. : Академия, 2016. – 624 с.
4. Майер Е. И. Типология текстовых задач в Едином государственном экзамене по математике профильного уровня [Текст] / Е.И. Майер // Молодой ученый. – 2017. – №51. – С. 289-291.
5. Макарова, Н.В. Информатика и ИКТ [Текст] : учебник для общеобразоват. учеб. заведений / Н.В. Макарова, Е.Г. Кочурова, Г.С. Николайчук, Ю.Н. Нилова, Ю.Ф. Титова ; под ред. Н.В. Макаровой. – СПб. : Питер, 2016. – 416 с.
6. Макарычев, Ю.Н. Алгебра. 9 класс [Текст] : учебник для общеобразовательных учреждений / Ю.Н. Макарычев, Н.Г. Миндюк, С.Б. Суворова. - М. : Просвещение, 2009. – 271 с.
7. Масленкова В.А., Васенина А.А., Пешкова Д.А., Бондарева Я.А. Методика работы над алгоритмической задачей в процессе обучения школьников информатике [Текст] / В.А. Масленкова // Педагогика: традиции и инновации: материалы VIII Междунар. науч. конф. (г. Челябинск, январь 2017 г.). – Челябинск, 2017. – С. 76-78.
8. Масленкова В.А., Горячева К.Г., Пешкова Д.А., Арзьева Н.А., Васенина А.А., Фадеева А.С., Коваленко И.А., Бондарева Я.А., Молдыбаева А.И., Ширшикова М.Е. Задача по программированию с продолжением на

- уроках информатики [Текст] / В.А. Масленкова // Молодой ученый. – 2016. – №30.1. – С. 32-36.
9. Масленкова В.А., Горячева К.Г., Пешкова Д.А., Арзьева Н.А., Васенина А.А., Фадеева А.С., Коваленко И.А., Бондарева Я.А., Молдыбаева А.И., Ширшикова М.Е. Задачи по программированию с продолжением на языке Паскаль [Текст] / В.А. Масленкова // Молодой ученый. – 2016. – №30.1. – С. 43-46.
10. Масленкова В.А., Горячева К.Г., Пешкова Д.А., Арзьева Н.А., Васенина А.А., Фадеева А.С., Коваленко И.А., Бондарева Я.А., Молдыбаева А.И., Ширшикова М.Е. Алгоритмическая задача на уроке информатики и методика работы над ней [Текст] / В.А. Масленкова // Молодой ученый. — 2016. — №30.1. — С. 36-38.
11. Масленкова В.А., Горячева К.Г., Пешкова Д.А., Арзьева Н.А., Васенина А.А., Фадеева А.С., Коваленко И.А., Бондарева Я.А., Молдыбаева А.И., Ширшикова М.Е. Методика работы над алгоритмической задачей как средство подготовки обучающихся по информатике [Текст] / В.А. Масленкова // Молодой ученый. — 2016. — №30.1. — С. 22-27.
12. Монахова, О.А. Рекуррентные последовательности. Алгебра формальных рядов [Текст] : Методические рекомендации для студентов специальностей «математика–физика», «математика–информатика» / О.А. Монахова, Н.А. Осьминина. – Пенза, 2016. – 32 с.
13. Мордкович, А.Г. Алгебра. 9 класс [Текст] : учебник для общеобразовательных учреждений / А.Г. Мордкович, П.В. Семенов. – М. : Мнемозина, 2010. – 224 с.
14. Просветов, Г. И. Текстовые задачи и методы их решения [Текст] : учеб.-практ. Пособие / Г.И. Просветов. – М. : Альфа-Пресс, 2010. – 48 с.
15. Семакин, И.Г. Информатика [Текст] : учебник для 9 класса / И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. – М.: Бином. Лаборатория знаний, 2015. – 200 с.

16. Семакин, И.Г. Основы алгоритмизации и программирования [Текст] : учебник / И.Г. Семакин, А.П. Шестаков. – М.: Академия, 2016. – 304 с.
17. Семакин, И.Г. Основы алгоритмизации и программирования. Практикум [Текст] : учеб. пособие / И.Г. Семакин, А.П. Шестаков. – М.: Норма, 2013. – 144 с.
18. Софронова, Н.В. Теория и методика обучения информатике [Текст] : учеб. пособие / Н.В. Софронова. – М. : Издательство «Высшая школа», 2013. – 186 с.
19. Сухих, Н.А. Поурочные разработки по информатике: 9 класс [Текст] / Н.А. Сухих. – М. : Вако, 2017. – 288 с.
20. Угринович, Н.Д. Информатика [Текст] : учебник для 9 класса / Н.Д. Угринович. – М. : Бинوم. Лаборатория знаний, 2016. – 152 с.
21. Угринович, Н.Д. Информатика и ИКТ. Базовый уровень [Текст] : учебник для 11 класса / Н.Д. Угринович. – М. : Бинوم. Лаборатория знаний, 2016. – 188 с.
22. Угринович, Н.Д. Информатика. 10 класс. Базовый уровень [Текст]: учебник / Н.Д. Угринович. – М. : Бинوم. Лаборатория знаний, 2017. – 288 с.

Урок 1

Тема: Циклы

Цели:

Образовательные:

4. Повторить понятие термина «Циклы»;
5. Научить применять циклы для решения задач на последовательности;
6. Способствовать осуществлению межпредметных связей на уроке информатики.

Развивающие:

4. Развивать навыки определения количества повторений цикла;
5. Способствовать развитию логического и алгоритмического мышления путем выполнения практической работы;
6. Развивать познавательный интерес и творческую активность.

Воспитательные:

3. Воспитывать культуру общения в группах;
4. Повышать интерес к практическому решению задач.

Тип урока: изучение нового материала и закрепление ранее изученного.

Вид урока: комбинированный.

Основные методы, технологии: элементы объяснительно-иллюстративного метода; частично-поискового; наглядный; практического; элементы групповой технологии, информационная технология.

Ход урока

Орг. момент (2 мин)

- приветствие, проверка отсутствующих
- сообщение темы и целей урока

Актуализация знаний (10 мин)

- Чем характерен цикл?
- Написать на доске общий вид убывающего и возрастающего цикла с параметром (2 учащихся у доски)
- Разобрать у доски домашнее задание
- Работа по карточкам на местах – (4 учащихся)

Объяснение нового материала (15 мин)

Решение задач.

Задача 1 (арифметическая прогрессия): Пользователь вводит с клавиатуры сумму кредита и ежемесячный платеж. Необходимо составить программу для подсчета срока выплаты кредита.

Решение задачи на языке программирования:

алг Кредит

нач

цел сум,пл,мес

вывод "Введите сумму кредита: "

ввод сум

вывод нс,"Введите ежемесячный платеж: "

ввод пл

мес:=0

нц пока сум>0

сум:=сум-пл

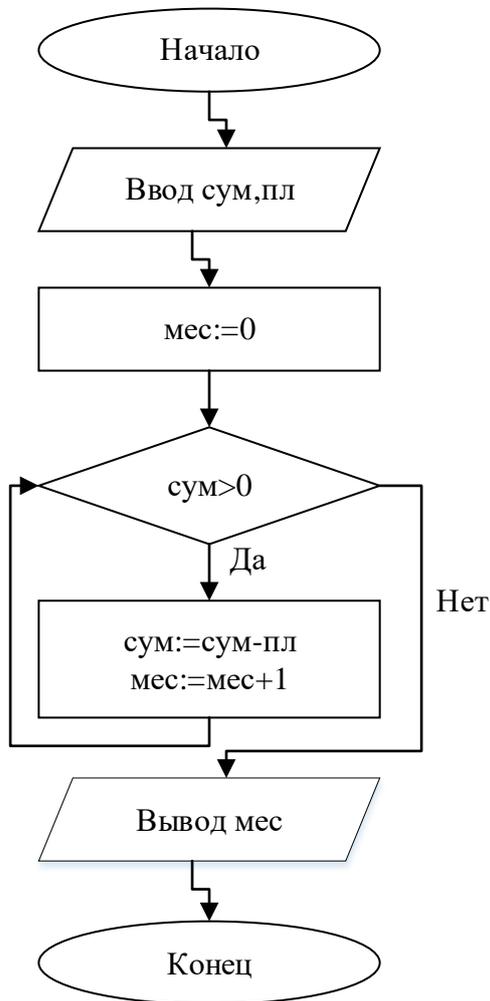
мес:=мес+1

кц

вывод нс,"Срок выплаты кредита - ",мес," мес."

кон

Блок-схема:



Задача 2 (геометрическая прогрессия): Некто продал лошадь за 156 рублей. Но покупатель, обретя лошадь, раздумал и возвратил продавцу, говоря: «Нет мне расчета покупать за эту цену лошадь, которая таких денег не стоит». Тогда продавец предложил другие условия: «Если по-твоему цена лошади высока, то купи ее подковные гвозди, лошадь же получишь тогда в придачу бесплатно. Гвоздей в каждой подкове 6. За первый гвоздь дай мне $1/4$ коп., за второй - $1/2$ коп., за третий - 1 коп., и т.д.» Покупатель, соблазненный низкой ценой, и желая даром получить лошадь, принял условия продавца, рассчитывая, что за гвозди придется уплатить не более 10 рублей. Кому будет выгодней такая сделка? Подсчитать стоимость гвоздей.

Решение задачи на языке программирования:

Алг Стоимость лошади

нач

вещ цен, сум

цен:=1/4

сум:=0

нц 24 раз

сум:=сум+цен

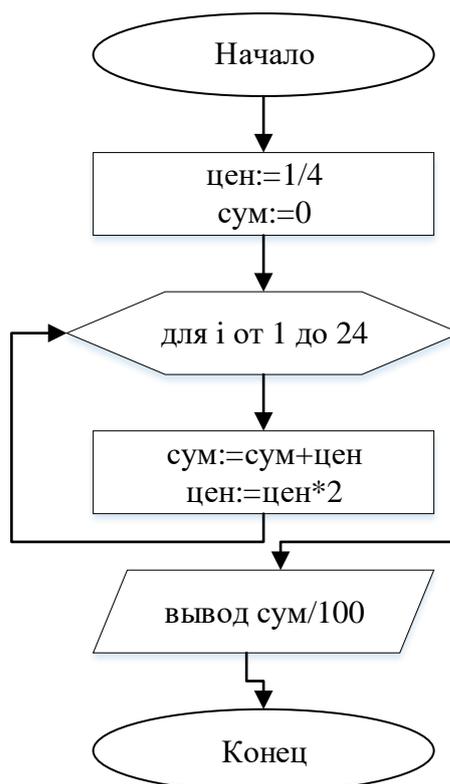
цен:=цен*2

кц

вывод "Стоимость гвоздей - ", сум/100, " рублей"

кон

Блок-схема:

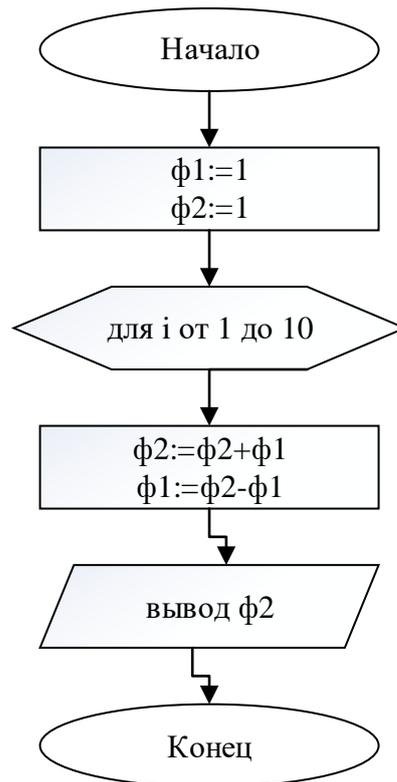


Задача 3 (последовательность Фибоначчи): В место, огороженное со всех сторон стеной, поместили пару кроликов, природа которых такова, что любая пара кроликов производит на свет другую пару каждый месяц, начиная со второго месяца своего существования. Сколько пар кроликов будет через год?

Решение задачи на языке программирования:

```
алг
нач
цел ф1,ф2
ф1:=1
ф2:=1
нц 10 раз
ф2:=ф2+ф1
ф1:=ф2-ф1
кц
вывод ф2, " кролика через год"
кон
```

Блок-схема:



Подведение итогов (3 мин)

- запись домашнего задания: учить конспект; составить задачу самостоятельно и решить ее;
- выставление оценок за работу на уроке.

Урок 2

Тема: Циклы

Цели:

Образовательные:

7. Научить применять циклы для решения задач на арифметические прогрессии;
8. Способствовать осуществлению межпредметных связей на уроке информатики.

Развивающие:

7. Развивать навыки определения количества повторений цикла;
8. Способствовать развитию логического и алгоритмического мышления путем выполнения практической работы;
9. Развивать познавательный интерес и творческую активность.

Воспитательные:

5. Воспитывать культуру общения в группах;
6. Повышать интерес к практическому решению задач.

Тип урока: Закрепление ранее изученного материала

Вид урока: комбинированный.

Основные методы, технологии: элементы объяснительно-иллюстративного метода; частично-поискового; наглядный; практического; элементы групповой технологии, информационная технология.

Ход урока

Орг. момент (2 мин)

- приветствие, проверка отсутствующих
- сообщение темы и целей урока

Объяснение нового материала по арифметической прогрессии (5 мин)

Решение задач.

Задача 1 При свободном падении тело прошло в первую секунду 6м, а в каждую следующую на 12м больше. Найдите глубину шахты, если свободно падающее тело достигло его дна через 15с после начала падения.

Решение задачи на языке программирования:

алг Падение

нач

цел глуб,расст

глуб:=0

расст:=6

нц 15 раз

глуб:=глуб+расст

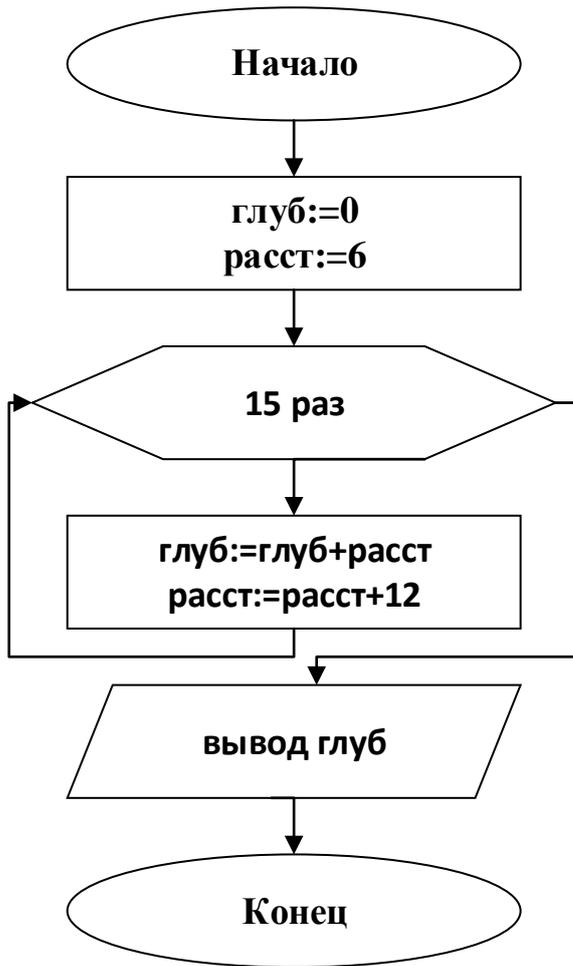
расст:=расст+12

кц

вывод глуб, " метров"

кон

Блок-схема:



Задача 2

Альпинисты в первый день восхождения поднялись на высоту 1200м, а затем каждый следующий день они проходили на 100м меньше, чем в предыдущий. За сколько дней они покорили высоту в 6000м?

Решение задачи на языке программирования:

```
алг Альпинисты
```

```
нач
```

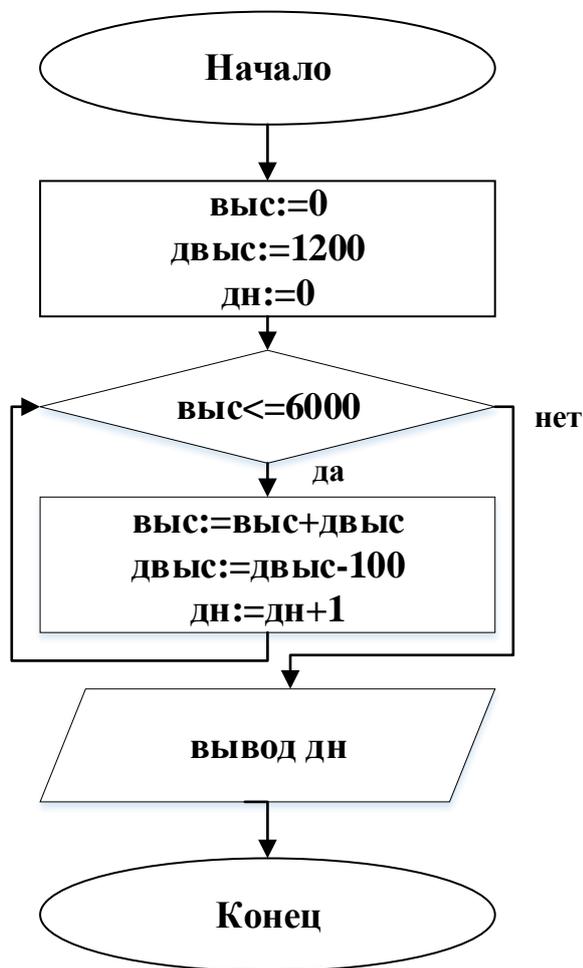
```
цел выс, двис, дн
```

```
выс:=0
```

```
двис:=1200
```

ДН:=0
 НЦ пока ВЫС<=6000
 ВЫС:=ВЫС+ДВЫС
 ДВЫС:=ДВЫС-100
 ДН:=ДН+1
 КЦ
 ВЫВОД "За ",ДН," дней"
 КОН

Блок-схема:



Задача 3

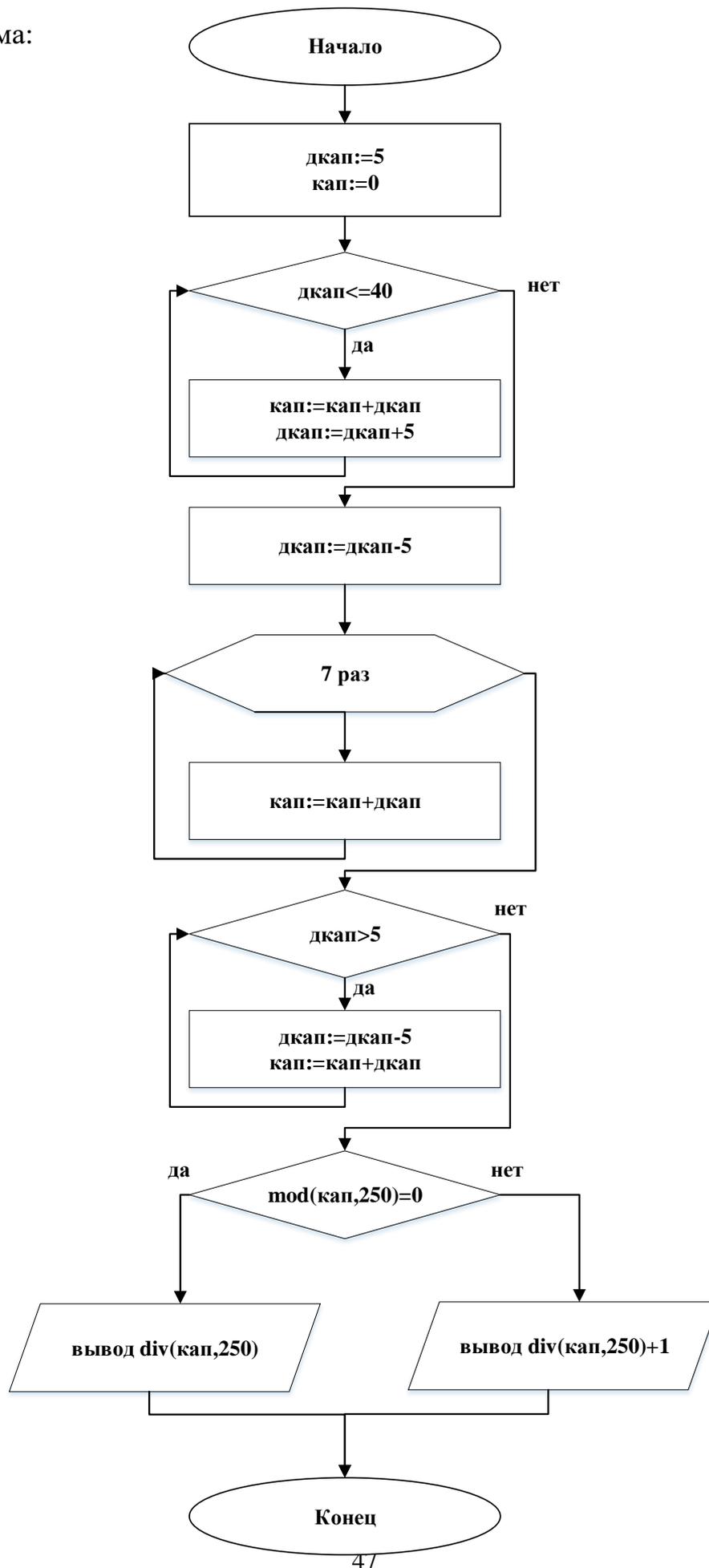
Больной принимает лекарство по следующей схеме: в первый день он принимает 5 капель, а в каждый следующий день — на 5 капель больше, чем

в предыдущий. Приняв 40 капель, он 7 дней пьет по 40 капель лекарства, а потом ежедневно уменьшает прием на 5 капель, доведя его до 5 капель. Сколько пузырьков лекарства нужно купить больному, если в каждом содержится 20 мл лекарства (что составляет 250 капель)?

Решение задачи на языке программирования:

```
алг
нач
цел дкап,кап
дкап:=5
кап:=0
нц пока дкап<=40
кап:=кап+дкап
дкап:=дкап+5
кц
дкап:=дкап-5
нц 7 раз
кап:=кап+дкап
кц
нц пока дкап>5
дкап:=дкап-5
кап:=кап+дкап
кц
если mod(кап,250)=0 то
вывод div(кап,250)," пуз."
иначе выводит div(кап,250)+1," пуз."
все
кон
```

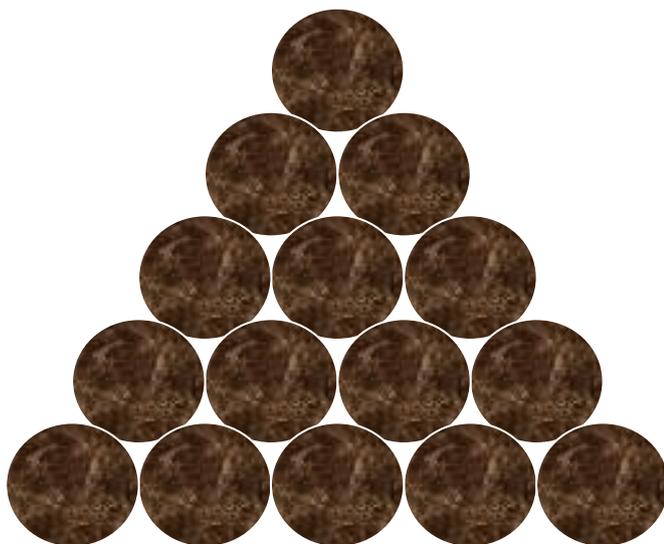
Блок-схема:



Подведение итогов (3 мин)

- запись домашнего задания: Решить задачу, составить блок-схему алгоритма программы:

При хранении бревен строевого леса их укладывают как показано на рисунке. Пользователь вводит с клавиатуры количество бревен в основании. Необходимо вывести на экран количество бревен в кладке.



- выставление оценок за работу на уроке.

Решение домашней задачи на языке программирования:

```
алг Кладка
```

```
нач
```

```
цел бряд,кб
```

```
кб:=0
```

```
вывод "Введите количество бревен в основании: "
```

```
ввод бряд
```

```
нц пока бряд>=1
```

```
кб:=кб+бряд
```

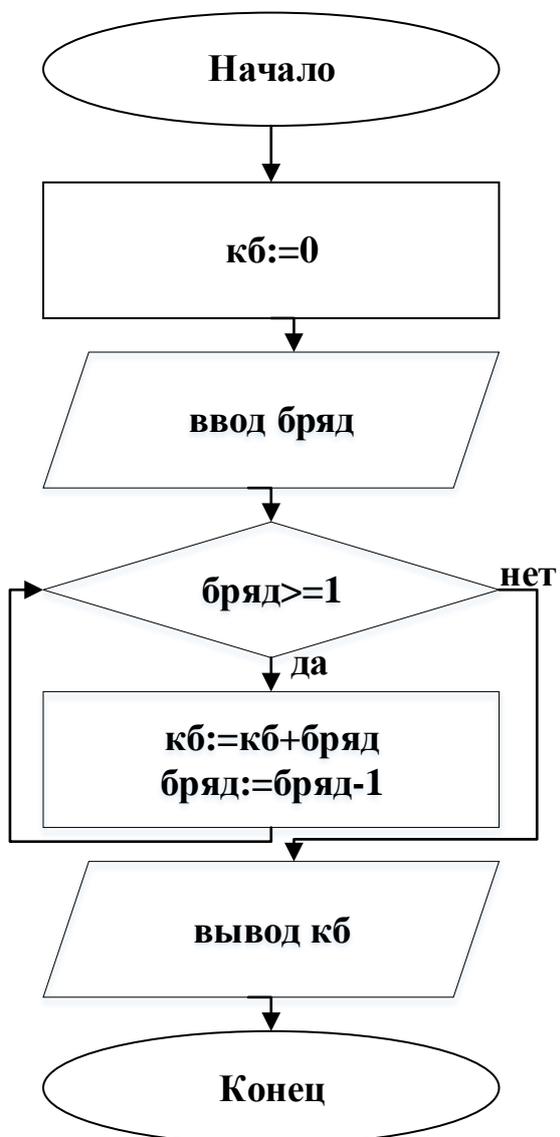
```
бряд:=бряд-1
```

```
кц
```

```
вывод кб," бревен в кладке"
```

КОН

Блок-схема:



Урок 3

Тема: Циклы

Цели:

Образовательные:

9. Научить применять циклы для решения задач на геометрические прогрессии;
10. Способствовать осуществлению межпредметных связей на уроке информатики.

Развивающие:

10. Развивать навыки определения количества повторений цикла;
11. Способствовать развитию логического и алгоритмического мышления путем выполнения практической работы;
12. Развивать познавательный интерес и творческую активность.

Воспитательные:

7. Воспитывать культуру общения в группах;
8. Повышать интерес к практическому решению задач.

Тип урока: Закрепление ранее изученного материала

Вид урока: комбинированный.

Основные методы, технологии: элементы объяснительно-иллюстративного метода; частично-поискового; наглядный; практического; элементы групповой технологии, информационная технология.

Ход урока

Орг. момент (2 мин)

- приветствие, проверка отсутствующих
- сообщение темы и целей урока

Объяснение нового материала по геометрической прогрессии (5 мин)

Решение задач.

Задача 1.

Бактерия, попав в живой организм, к концу 20-й минуты делится на две бактерии, каждая из них к концу следующих 20 минут делится опять на две и т.д. Найдите число бактерий, образующихся из одной бактерии к концу суток.

Решение задачи на языке программирования:

алг Бактерии

нач

вещ дел, тбак, бак

тбак:=1

бак:=0

дел:=24*60/20

нц int(дел) раз

бак:=бак+тбак

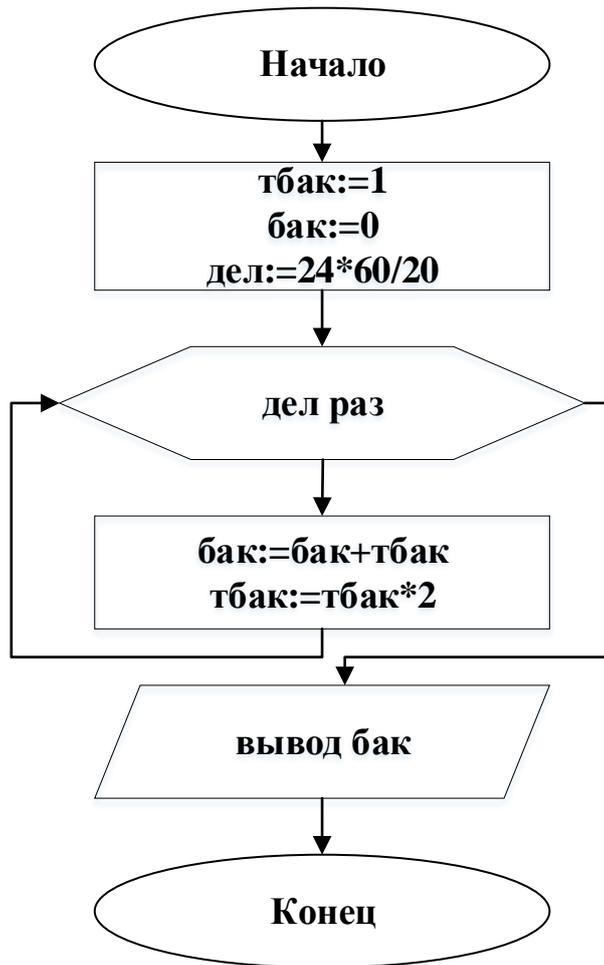
тбак:=тбак*2

кц

вывод "К концу суток будет ",бак," бактерий"

Кон

Блок-схема:



Задача 2

Автомобиль, двигаясь со скоростью 20 км/ч, начинает ускоряться. Причем за каждые 5 секунд разгона его скорость возрастает в 1.5 раза. Найти скорость автомобиля после 2 минут разгона.

Решение задачи на языке программирования:

```
алг Разгон  
нач  
вещ скор,кол  
кол:=2*60/5  
скор:=20
```

нц int(кол) раз

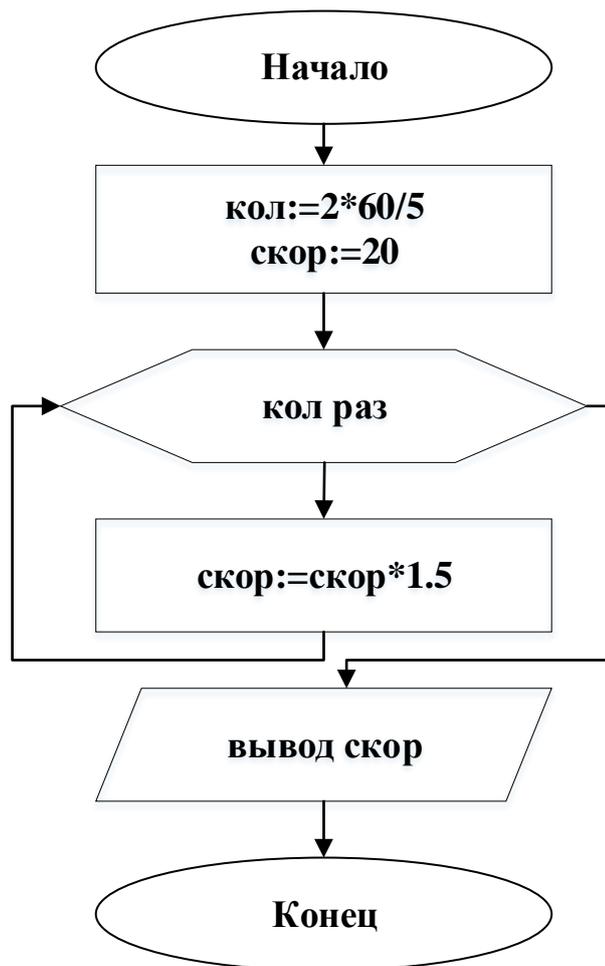
скор:=скор*1.5

кц

вывод "Скорость через 2 минуты - ",скор," км/ч"

кон

Блок-схема:



Задача 3

Если положить деньги на срочный вклад в сберегательный банк, то через год вклад увеличится на указанное количество процентов от исходной суммы. Ещё через год уже эта сумма увеличится на указанное количество процентов.

Пользователь вводит с клавиатуры размер вклада, годовой процент вклада и срок. Вывести на экран размер вклада после истечения срока.

Решение задачи на языке программирования:

алг

нач

вещ вклад,процент

цел срок

вывод "Введите размер вклада: "

ввод вклад

вывод "Введите срок: "

ввод срок

вывод "Введите процент: "

ввод процент

процент:=1+процент/100

нц срок раз

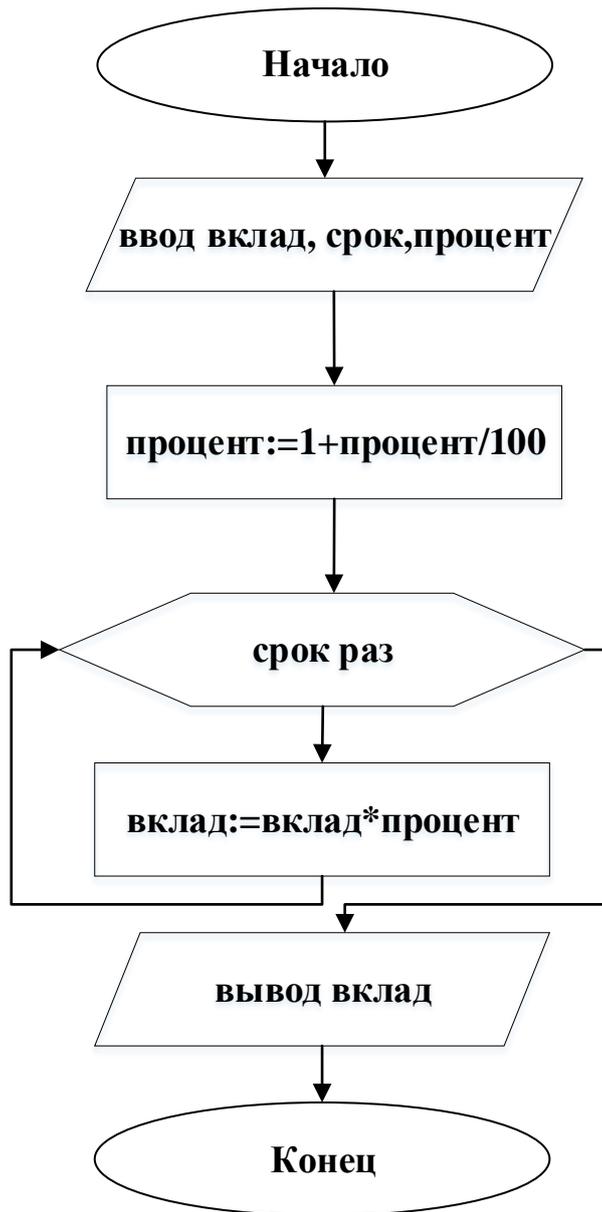
вклад:=вклад*процент

кц

вывод "Вклад после истечения срока: ",вклад

кон

Блок-схема:



Подведение итогов (3 мин)

- запись домашнего задания: Известно, что за единицу времени радиоактивное вещество теряет определенную часть своей массы (она переходит в другое вещество и энергию). Для каждого радиоактивного вещества определяется величина T – время периода полураспада. Через время T масса радиоактивного элемента уменьшается в 2 раза.

Пользователь вводит массу радиоактивного вещества, его период полураспада (в годах), показатель, во сколько раз должна уменьшиться масса.

Вывести на экран, за сколько лет масса радиоактивного вещества уменьшится в указанное количество раз.

Составить блок-схему алгоритма программы.

- выставление оценок за работу на уроке.

Решение домашней задачи на языке программирования:

алг Распад

нач

вещ масса, кмасса

цел rраз, период, кол

вывод "Введите исходную массу вещества: "

ввод масса

вывод "Введите период полураспада: "

ввод период

вывод "Укажите, во сколько раз масса вещества должна уменьшиться: "

ввод rраз

кол:=0

кмасса:=масса/rраз

нц пока масса>кмасса

масса:=масса/2

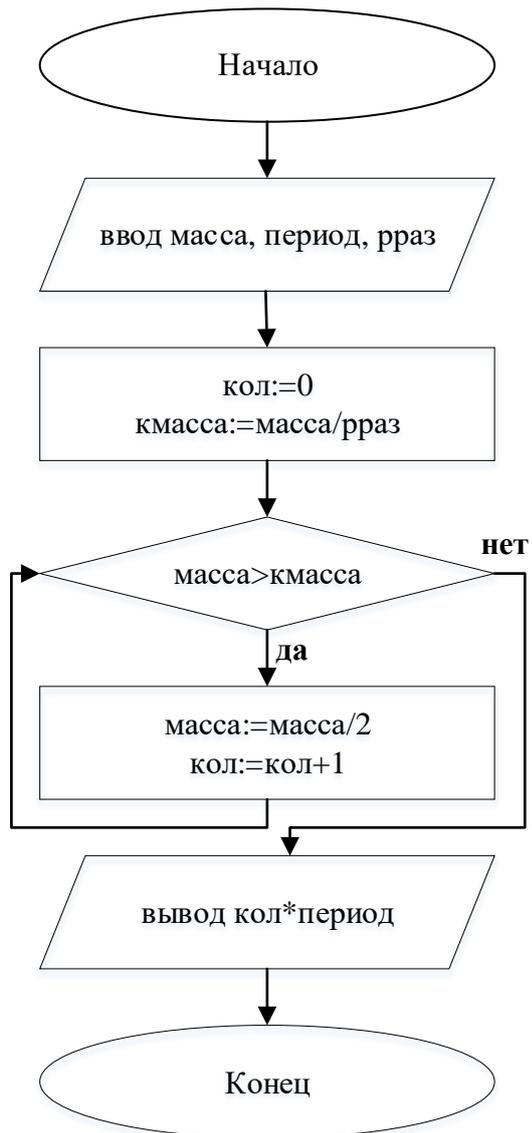
кол:=кол+1

кц

вывод "Масса вещества уменьшится в ",rраз," раз через ",кол*период," лет"

кон

Блок-схема:



Урок 4

Тема: Циклы

Цели:

Образовательные:

- 11.Закрепить полученные знания по последовательностям;
- 12.Способствовать осуществлению межпредметных связей на уроке информатики.

Развивающие:

- 13.Развивать навыки определения количества повторений цикла;
- 14.Способствовать развитию логического и алгоритмического мышления путем выполнения практической работы;
- 15.Развивать познавательный интерес и творческую активность.

Воспитательные:

9. Воспитывать культуру общения в группах;
- 10.Повышать интерес к практическому решению задач.

Тип урока: Закрепление ранее изученного материала

Вид урока: комбинированный.

Основные методы, технологии: элементы объяснительно-иллюстративного метода; частично-поискового; наглядный; практического; элементы групповой технологии, информационная технология.

Ход урока

Орг. момент (2 мин)

- приветствие, проверка отсутствующих
- сообщение темы и целей урока

Объяснение нового материала по числам Фибоначчи (5 мин)

Решение задач.

Задача 1.

Пользователь вводит с клавиатуры число (больше двух) – количество элементов последовательности Фибоначчи. Вывести их на экран. Определить, при каком значении происходит переполнение.

Решение задачи на языке программирования:

алг Фибоначчи

нач

цел ф1,ф2,кол

вывод "Введите количество элементов последовательности Фибоначчи: "

ввод кол

ф1:=1

ф2:=1

вывод ф1, " ",ф2," "

нц кол-2 раз

ф2:=ф2+ф1

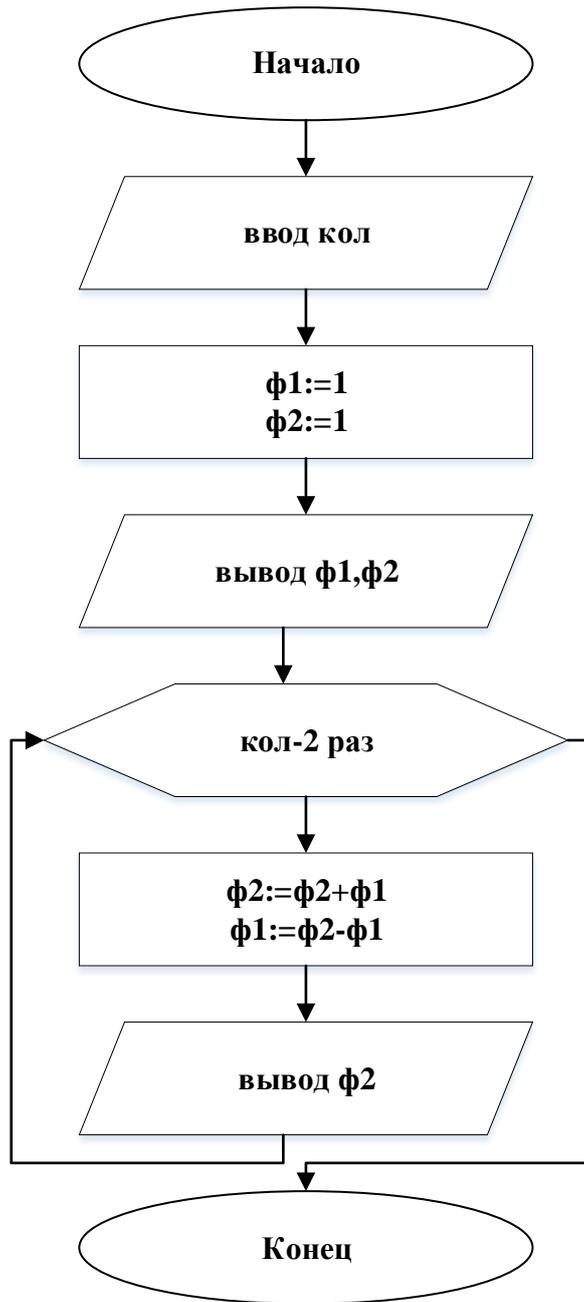
ф1:=ф2-ф1

вывод ф2," "

кц

кон

Блок-схема:



Задача 2

Дан ряд чисел 6 13 20, ... Определить, по какой закономерности образуется каждое последующее и написать программу для вычисления n-ного члена последовательности. Число n вводится с клавиатуры.

Решение задачи на языке программирования:

алг Последовательность

нач

цел a,n

ввод n

a:=6

нц n-1 раз

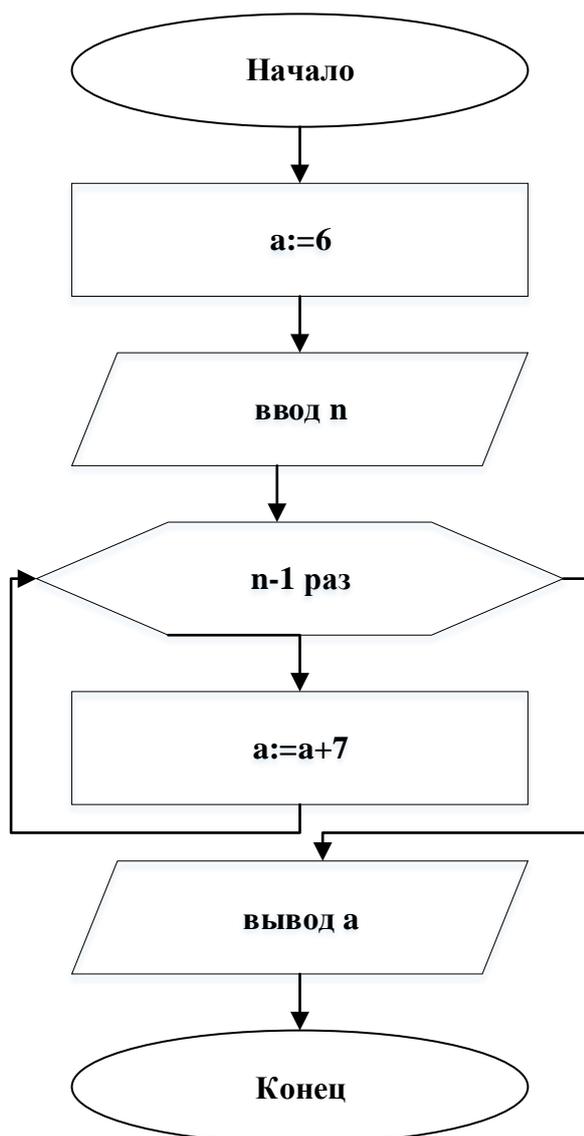
a:=a+7

кц

вывод n,"-ый член последовательности - ",a

кон

Блок-схема:



Задача 3

Дан ряд чисел 2 6 18, ... Определить, по какой закономерности образуется каждое последующее и написать программу для вычисления n-ного члена последовательности. Число n вводится с клавиатуры.

Решение задачи на языке программирования:

алг Последовательность

нач

цел a,n

ввод n

a:=2

нц n-1 раз

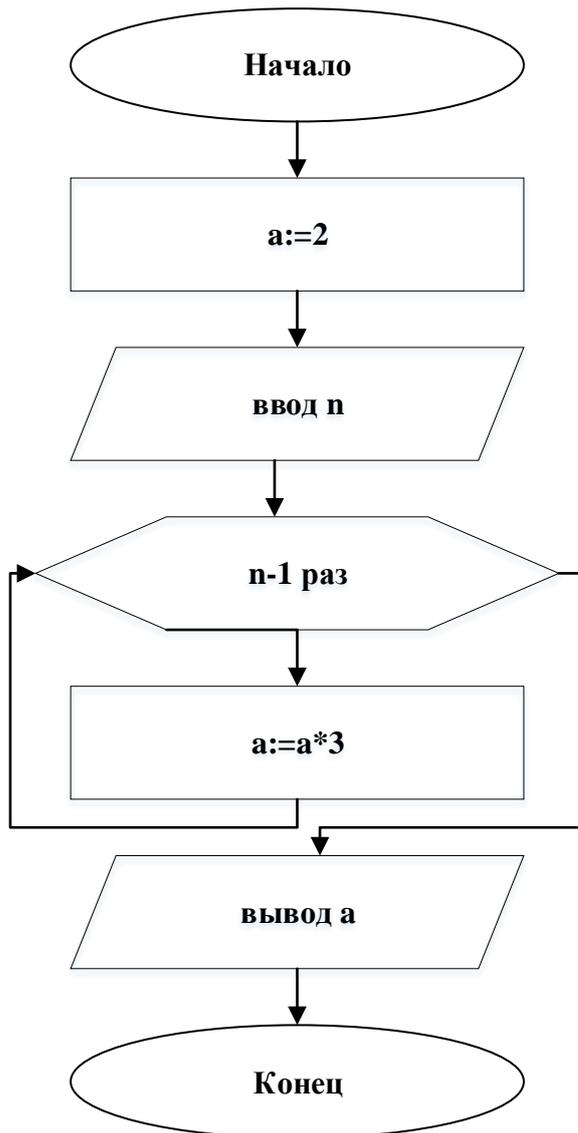
a:=a*3

кц

вывод n,"-ый член последовательности - ",a

кон

Блок-схема:



Подведение итогов (3 мин)

- запись домашнего задания:

- Доработать домашнее задание, данное на первом уроке
- Даны 3 первых члена арифметической или геометрической прогрессии (вводятся с клавиатуры). Составить программу для поиска n-ного члена прогрессии. Число n вводится с клавиатуры. В алгоритме программы определить тип прогрессии и использовать ветвление. Составить блок-схему алгоритма программы.

- выставление оценок за работу на уроке.

Решение домашней задачи на языке программирования:

алг Последовательность

нач

цел a_1, a_2, a_3, ana, n, d

вещ q, ang

вывод "1-ый член прогрессии: "

ввод a_1

вывод "2-ой член прогрессии: "

ввод a_2

вывод "3-ий член прогрессии: "

ввод a_3

вывод "Введите значение n : "

ввод n

если $a_2 + (a_2 - a_1) = a_3$ то

$d := a_2 - a_1$

$ana := a_1$

нц $n-1$ раз

$ana := ana + d$

кц

вывод n , "-ый член арифм. прогрессии - ", ana

иначе

$q := a_2 / a_1$

$ang := a_1$

нц $n-1$ раз

$ang := ang * q$

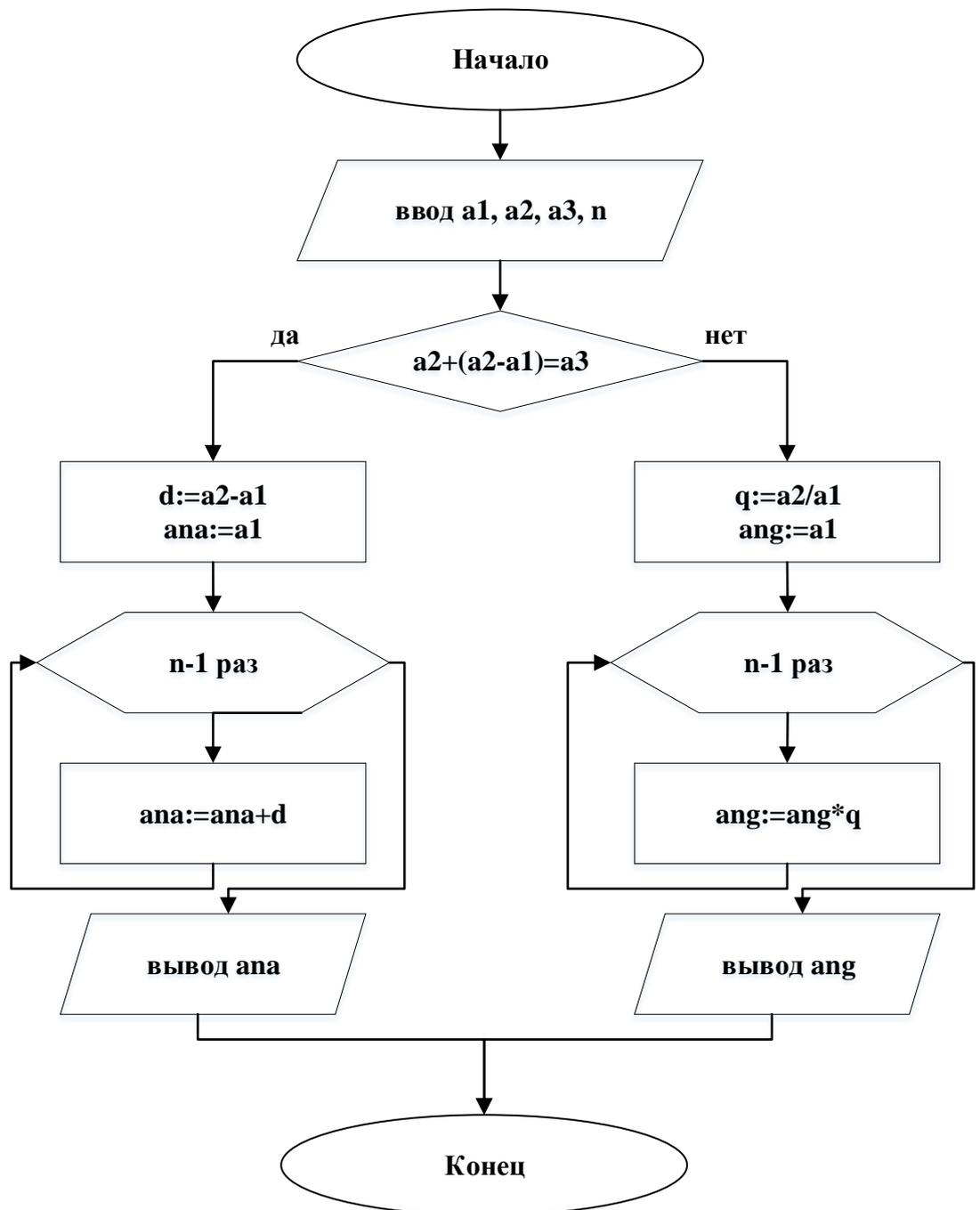
кц

вывод n , "-ый член геом. прогрессии - ", ang

все

КОН

Блок-схема:



Тест по теме «Последовательности»

1 / 10

Последовательность задана следующим образом:
 $a_1 = 2$, $a_n = 3a_{n-1} - 2$ при $n \geq 2$. Чему равно $a_5 - a_1$?

 54 52 56 2