Краснодарский край, Кореновский район, город Кореновск,

муниципальное общеобразовательное автономное некоммерческое учреждение

средняя общеобразовательная школа №19 имени Героя России С.А. Наточего

муниципального образования Кореновский район

Методическая разработка урока Информатики и ИКТ в

соответствии с требованиями ФГОС по теме

«Подготовка к ЕГЭ № 16. Рекурсивные алгоритмы. Процедуры и функции»

Автор разработки: учитель информатики и ИКТ Бирюкова Е.А.

Кореновск

2021

**Цели занятия:**

**Предметные:** - формирование представления о рекурсивном объекте и рекурсивном алгоритме; освоение приемов применения рекурсивных подпрограмм на языке PascalABC.

**Метапредметные: -** умение самостоятельно планировать пути достижения целей; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

**Личностные:** - алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе; представление о программировании как сфере возможной профессиональной деятельности.

**Решаемые учебные задачи:**

1. напомнить сущность понятия вспомогательного алгоритма, подпрограммы;
2. познакомить с правилами оформления подпрограммы в виде процедуры;
3. познакомить с правилами оформления подпрограммы в виде функции;
4. познакомить с примерами работы рекурсивных процедур.

**Вид занятия:**изучение нового материала.

**Тип урока:**интегрированный.

**Метод проведения:**словесный, наглядный, практическая работа.

**Формы работы:** фронтальная, выполнение практической работы, работа в группах.  
**ТСО:**  мультимедийный проектор, компьютер учителя.

**Структура занятия:**

1. Организационный момент
2. Мотивация учебной деятельности. Актуализация опорных знаний
3. Изучение нового материала
4. Закрепление нового материала. Решение задач.
5. Рефлексия
6. Сообщение домашнего задания

**Ход занятия**

|  |  |  |
| --- | --- | --- |
| Этапы урока | Действия учителя | Действия учеников |
| **1.Организационный момент** | Учитель приветствует учеников, проверяет присутствие в классе | Ученики готовятся к уроку |
| **2. Мотивация учебной деятельности (создание ситуации успеха)** | Что такое подпрограмма?  Какие виды подпрограмм существуют в языке Паскаль?  Что такое формальные и фактические параметры?  В каком случае используется в программе процедура?  В каком случае удобно использовать в программе функцию?  Давайте ответим на вопрос: «Процедура или функция может ли содержать вызов других процедур или функций?»  Сформулируйте цели нашего урока. | Подпрограмма – это специальным образом оформленный алгоритм.  В языке Паскаль существует два вида подпрограмм: процедура(PROCEDURE) и функция (FUNCTION).  Формальные – условные обозначения в описании процедуры, описываются в ее заголовке. Фактические – перечисляются при вызове процедуры и с ними требуется выполнить процедуру.  Процедура используется в случаях, когда в подпрограмме необходимо получить несколько результатов.  Когда в подпрограмме необходимо получит единственный результат.  Ученики высказывают свои версии ответов.  Ученики формулируют цели урока. |
| 3.Изучение нового материала сопровождается презентацией. | **Устно** Основной ошибкой циклической программы является зацикливание, то есть бесконечное повторение тела цикла. Но есть вид циклических алгоритмов, которые применяют этот прием для решения класса задач. Эта структура называется **рекурсией**.  Слово «рекурсия» происходит от латинского слова «recursio» - возвращение. Рекурсивные определения как мощный аналитический аппарат используется во многих областях науки, особенно в математике: арифметическая прогрессия (an+1=an+d); геометрическая прогрессия (bn+1=bnq, ); числа Фибоначчи, каждый элемент которого является суммой двух предыдущих;факториал (n!=1\*2\*3\*4\*5…).  **(2 слайд)**  **Рекурсивные алгоритмы** — это алгоритмы, которые решают поставленные задачи с помощью приведения их к разрешению одной или более аналогичных задач, но в более коротком их представлении.  Рекурсивностью в Паскале могут обладать как функции, так и процедуры.  **(3 слайд)**  Процедуры и функции помещаются в раздел описаний программы. Для обмена информацией между процедурами и функциями и другими блоками программы существует механизм **входных** и **выходных параметров**. Входными параметрами называют величины, передающиеся из вызывающего блока в подпрограмму (исходные данные для подпрограммы), а выходными - передающиеся из подрограммы в вызывающий блок (результаты работы подпрограммы).  **(4 слайд)**  **Одна и та же подпрограмма может вызываться неоднократно, выполняя одни и те же действия с разными наборами входных данных.** Параметры, использующиеся при записи текста подпрограммы в разделе описаний, называют **формальными**, а те, что используются при ее вызове - **фактическими**.  (**5 и 6 слайд)** Формат описания процедуры и функции. | Ученики слушают учителя, записывают основные понятия. |

**4. Закрепление нового материала. Решение задач.**

**(7 слайд)** Рассмотрим задачу, решаемую с помощью функции.

**Алгоритм вычисления функции F(n) задан следующими соотношениями:**

**F(n) = 2 при n = 1**

**F(n) = F(n–1) + 5n2, если n > 1**

**Чему равно значение функции F(39)?**

Запишем решение в тетрадь из слайда. Затем копируем в среду программирования PASCALABC . И смотрим результат через проектор.

program rrr5;

function F( n: integer ): integer;

begin

IF n=1 then F:= 2;

IF n>1 then F:=F(n-1)+5\*n\*n;

end;

begin

writeln( F(39) ) ;

end.

**(8 слайд)** Алгоритм вычисления функции F(n) задан следующими соотношениями:

F(n) = n – 3 при n > 16

F(n) = 2·F(n+1) + 2n + 3, если n  16

Чему равно значение функции F(2)?

Самостоятельно составляем программу и обсуждаем. (5минут)

program rrr9;

function F( n: integer ): integer;

begin

if n>16 then f:=n-3;

if n<=16 then f:=2\*f(n+1)+2\*n+3;

end

begin

writeln(F(2)) ;

end.

Рассмотрим метод решения с помощью массива:

PROGRAM FF6;

FUNCTION F( N: INTEGER): iNTEGER;

VAR A: ARRAY[0..100] of integer; i: integer;

begin

A[1]:=1; A[2]:=1;

For i:=3 to n do

A[i]:=A[i-1]\*i-2\*A[i-2];

Result := a[n];

END;

begin

writeln (F(6);

END.

**(9 слайд)** Самостоятельно составляем программу и обсуждаем. (5минут)

**Алгоритм вычисления функции F(n) задан следующими соотношениями:**

**F(n) = 1+2n при n < 5**

**F(n) = 2·(n + 1)·F(n–2), если n делится на 3,**

**F(n) = 2·n + 1 + F(n–1) + 2·F(n–2), если n не делится на 3. Чему равно значение функции F(15)?**

**program** r15;

**function** F(n:integer): integer;

**begin**

**If** n<5 **then** F:=1+2\*n;

**If** (n **mod** 3=0)**and** (n>=5) **then** f:=2\*(n+1)\*f(n-2);

**If** (n **mod** 3<>0)**and** (n>=5) **then** f:=2\*n+1 +f(n-1)+2\*f(n-2);

**end**;

**begin**

writeln (F(15));

**end**.

**(10 слайд)** Рассмотрим задачу с двумя функциями

**Алгоритм вычисления функций F(n) и G(n) задан следующими соотношениями:**

**F(1) = G(1) = 1**

**F(n) = 2·F(n–1) + G(n–1) – 2n, если n > 1**

**G(n) = F(n–1) +2·G(n–1) + n, если n > 1**

**Чему равно значение F(14) + G(14)?**

function G(n:integer): integer; forward;

function F(n:integer): integer; forward;

function F(n:integer): integer;

begin

if n=1 then F:=1;

if n>1 then F:=2\*f(n-1)+G(n-1)-2\*n;

end;

function G(n:integer): integer;

begin

if n=1 then G:=1;

if n>1 then G:=F(n-1)+2\*G(n-1)+n;

end;

begin

writeln (F(14)+G(14));

end.

**(11 слайд)** Переходим к рассмотрению задач с процедурами.

**Определите, сколько символов \* выведет эта процедура при вызове F(140):**

procedure F(n: integer);

begin

write('\*');

if n >= 1 then begin

Write('\*');

F(n-1); F(n div 2);

end;

**(12 слайд)** Определите наименьшее значение n, при котором сумма чисел, которые будут выведены при вызове F(n), будет больше 1000000. Запишите в ответе сначала найденное значение n, а затем через пробел – соответствующую сумму выведенных чисел.

**procedure** F(n:integer);

**Begin**

writeln(n+1);

**if** n>1 **then begin**

F(n-1); F(n-2);

**end**;

Решение:

**var** s, n: integer;

**procedure** F(n:integer);

**begin**

s := s+n+1;

**if** n>1 **then begin**

s := s+n+5;

F(n-1);

f(n-2);

**end**;

**end**;

**begin**

n := 0;

**repeat**

n := n + 1;

s := 0;

F(n);

**until** s > 1000000;

writeln( n, ' ', s);

**end**.

**(13 слайд)** **Самостоятельно решаем.** Определите наименьшее значение *n*, при котором сумма чисел, которые будут выведены при вызове *F*(*n*), будет больше 3200000. Запишите в ответе сначала найденное значение *n*, а затем через пробел – соответствующую сумму выведенных чисел*.*

procedure F ( n: integer );

Begin

writeln(n\*n);

if n > 1 then begin

writeln(2\*n+1); F(n-2); F(n div 3);

end;

**Решение**

var s, n: integer;

procedure F ( n: integer );

begin

s:=s+n\*n;

if n > 1 then begin

s:=s+2\*n+1;

F(n-2); F(n div 3);

end;end;

begin

clrscr;

n := 0;

repeat

n := n + 1; s := 0;F(n);

until s > 3200000;

writeln(n, ' ', s);

end.

**Слайд(15-16)** **Решение задачи повышенной сложности**

Алгоритм вычисления функции *F*(*n*) задан следующими соотношениями:

*F*(*n*) = *n* + 3, при *n*  18

*F*(*n*) = (*n* // 3) · *F*(*n* // 3) + *n* – 12, при *n* > 18, кратных 3

*F*(*n*) = *F*(*n–*1) + *n* · *n* + 5, при *n* > 18, не кратных 3

Здесь «//» обозначает деление нацело. Определите количество натуральных значений *n* из отрезка [1; 1000], для которых все цифры значения *F*(*n*) чётные.

Var i, k, t, x, p:integer;

function F( n: integer ): integer;

begin

if n<=18 then F:=n+3;

if (n>18) and (n mod 3=0) then F:=(n div 3)\*F(n div 3)+n-12;

if (n>18) and (n mod 3<>0) then F:=F(n-1)+n\*n+5;

End;

Begin

k:=0; t:=0;

for i:=1 to 1000 do

begin

p:= f(i);

while p>0 do

begin

x:=p mod 10; p:=p div 10;

if (x mod 2<> 0) then k:=k+1;

end;

if k=0 then t:=t+1; k:=0;

end;

writeln(t); end.

5. Рефлексия

6.Сообщение домашнего задания.

**Литература**

1. ЕГЭ Информатика и ИКТ. Типовые экзаменационные варианты Л.Н. Евич , 2020.
2. Материалы сайта К.Ю. Полякова kpolyakov.spb.ru.