Министерство образования Республики Беларусь

Учреждение образования

«Гомельский государственный университет

имени Франциска Скорины»

УТВЕРЖДАЮ

учитель информатики

Кондратьева Н. Г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата утверждения)

План - конспект

зачётного урока по информатике на тему

«ОПЕРАТОРЫ ЦИКЛА»

в 8 «А» классе

ГУО «Средняя школа № 30 г. Гомеля»

Студент-практикант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Г. Малкова

Отметка за проведение \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ учитель информатики

урока Н. Г. Кондратьева

Преподаватель

кафедры математического

анализа и ДУ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И. В. Парукевич

Гомель 2019

**Дата:** 06. 02. 2019

**Тема:** «Оператор цикла».

**Класс:** 8 «А»

**Цели урока:**

*Познавательные (обучающие):*

* Ввести понятие цикла.
* Изучить организацию цикла с предусловием, параметром.
* Познакомить со структурными элементами цикла.
* Выяснить условия работы цикла с предусловием, параметром.

*Развивающие:*

* развивать умение анализировать, систематизировать полученные знания;
* развивать творческое и логическое мышление;
* развивать инициативность, самостоятельность и внимание;
* укладываться в ограниченные сроки, уметь давать самооценку работы.

*Воспитательная:*

* развивать ответственность за результат выполненной работы.
* воспитание познавательной потребности, интереса к предмету.

**Тип урока:** усвоение новых знаний.

**Оборудование:** компьютерный класс, персональный компьютер, среда программирования Pascal.

**Структура урока:**

1. Организационный момент – 1 минута.

2. Повторение теоретического материала – 4 минуты.

3. Изучение нового материала – 20 минут.

3.1 Цикл с предусловием.

3.2 Цикл с параметром (со счетчиком).

4. Первичное осмысление и закрепление новой темы – 5 минут.

5. Практическая работа – 10 минут.

6. Подведение итогов урока. Рефлексия – 4 минуты.

7. Постановка задания на дом.– 1 мин.

**Литература:**

1. Златопольский Д.М. Я иду на урок информатики: Задачи по программированию. 7-11 классы: Книга для учителя. - М.: Издательство «Первое сентября», 2001.
2. Абрамов С.А. и др. Задачи по программированию М: Наука, 2003.
3. Гуденко Д.А. Сборник задач по программированию. С-Пб.: «Компас», 2005г.
4. Р. З. Ахметсафина, Е. М. Бронштейн и др. Методические указания по решению тестовых заданий. Информатика. Уфа, 2000.

*«Я хочу, чтобы компьютер был моим слугой,*

*а не господином, поэтому я должен уметь*

*быстро и эффективно объяснить*

*ему, что делать.»*

*Мацумото Юкихиро*

**Ход урока.**

1. **Организационный момент.**

Здравствуйте ребята! Все готовы к уроку? Присаживайтесь, пожалуйста. Все присутствуют на уроке? Хорошо, давайте начнем наш урок.

Учащиеся приветствуют учителя, отвечают на вопросы.

1. **Повторение теоретического материала.**

– Давайте вспомним, с какими структурами алгоритмов мы знакомились на прошлых занятиях?

– (*предполагаемые ответы*)

* линейная структура (действия следуют одно за другим)
* структуры ветвления и выбор (действия выполняются в одном из направлений в зависимости от истинности проверяемого условия)

– Вам часто приходится повторять одни и те же действия. Например:

* каждый день посещать занятия;
* 5 раз посмотреть понравившийся мультфильм;
* перечитывать текст задания, пока оно не станет понятным.

Выполнение даже очень простых операций может поставить человека в тупик, если их надо повторить тысячи раз. И здесь нам на помощь приходит алгебраическая структура Цикл.

**Цикл** – это многократное выполнение одинаковой последовательности действий.

Существуют циклы двух видов:

* цикл с **известным** числом шагов
* цикл с **неизвестным** числом шагов (цикл с условием)

Чтобы понять разницу, давайте разберем следующий пример. Перед нами стоит задача заточить **20** карандашей из старой коробочки. Мы можем составить алгоритм, состоящий из повторяющейся последовательности следующих действий:

* Достать из коробки карандаш
* Заточить карандаш
* Отложить карандаш в контейнер

Сколько раз нам необходимо повторить эти действия?

– (*предполагаемый ответ*) 20 раз

– Для отслеживания количества повторений служит управляющая переменная, которую называют счетчиком или параметром.

А если перед нами стоит задача заточить все карандаши из старой коробочки?

Мы можем составить алгоритм, состоящий из повторяющейся последовательности следующих действий. Сколько раз нам необходимо повторить эти действия?

– (*предполагаемый ответ*) Столько раз, сколько карандашей.

– А их количество нам неизвестно. Коробка вообще может быть пустой. Мы будем совершать действия, ЕСЛИ в коробке есть карандаши. Для того, чтобы действия совершались, необходимо выполнение УСЛОВИЯ. И эти циклы называются циклами с условием.

1. **Изучение нового материала.**

Где в жизни мы можем наблюдать цикличность (повторение)?

Дни недели, времена года, смена дня и ночи, смена месяцев в году и т.д.

Перенесёмся в Древнюю Грецию – страну великих ученых, поэтов и легендарных героев.   
“...Сизиф, сын бога повелителя всех ветров Эола, был основателем города Коринфа.

Тяжкое наказание несет Сизиф в загробной жизни за все коварства, за все обманы, которые совершил он на земле. Он осужден вкатывать на высокую, крутую гору громадный камень. Напрягая все силы, трудится Сизиф. Все ближе вершина; еще усилие, и окончен будет труд Сизифа; но вырывается из рук его камень и с шумом катится вниз. Снова принимается Сизиф за работу.

Так вечно катит камень Сизиф и никогда не может достигнуть цели – вершины горы...”

Нельзя ли помочь бедняге?

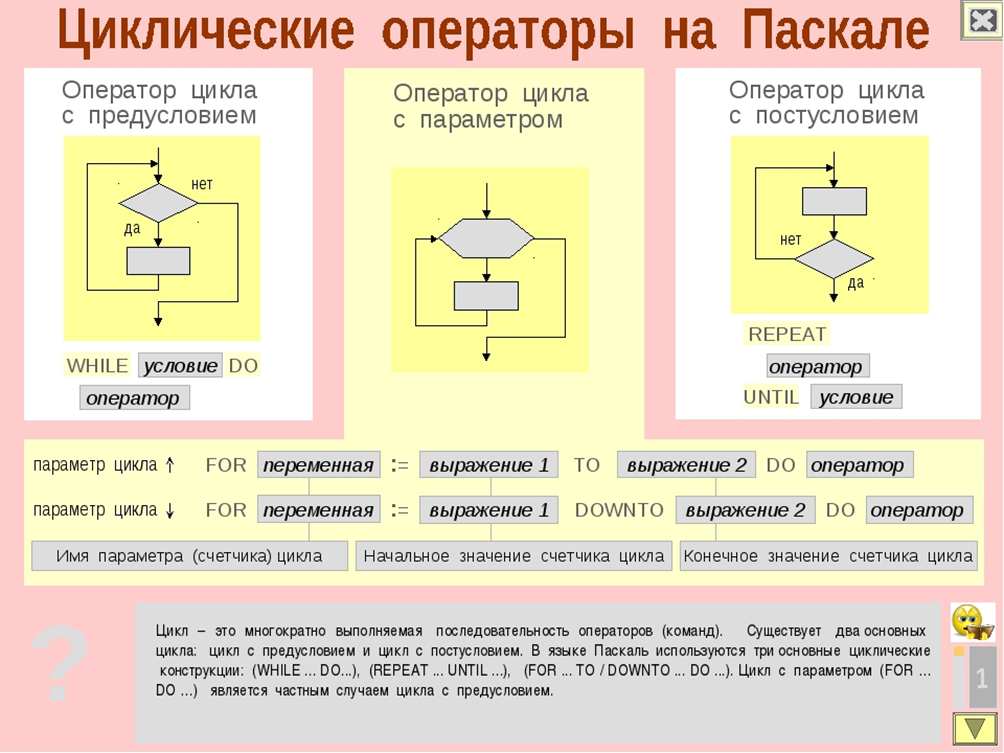
Вот что интересно – дайте прочитать древний миф программисту, и он скажет: “Ничего страшного. Речь идет о выполнении бесконечного цикла. ”

Цикл является одной из важнейших алгоритмических структур и представляет собой последовательность операторов, которая выполняется неоднократно. В программах, связанных с обработкой данных или вычислениями, часто приходится выполнять циклически повторяющиеся действия. Циклы позволяют записать такие действия в компактной форме.

Циклы принадлежат к числу управляющих операторов.

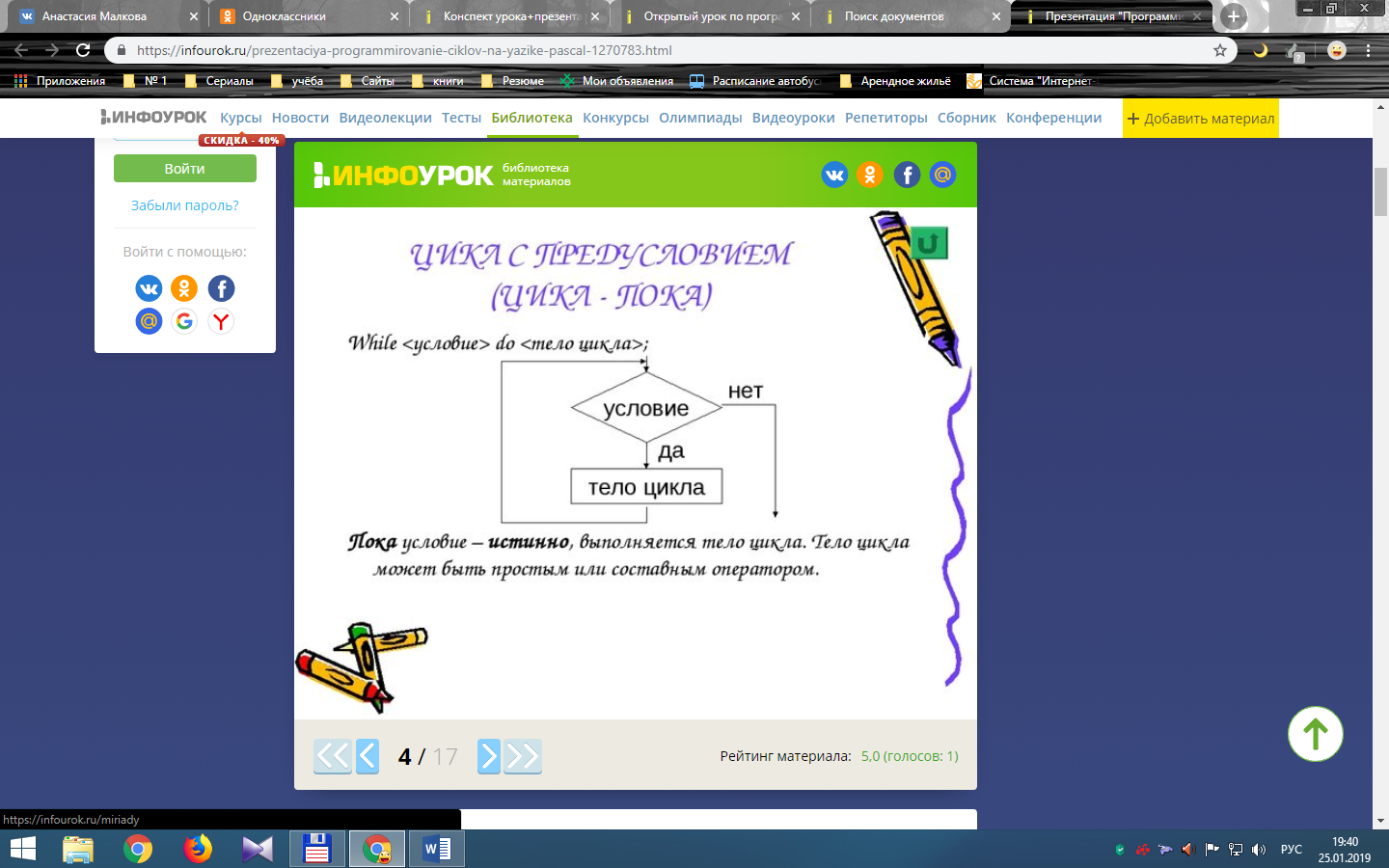
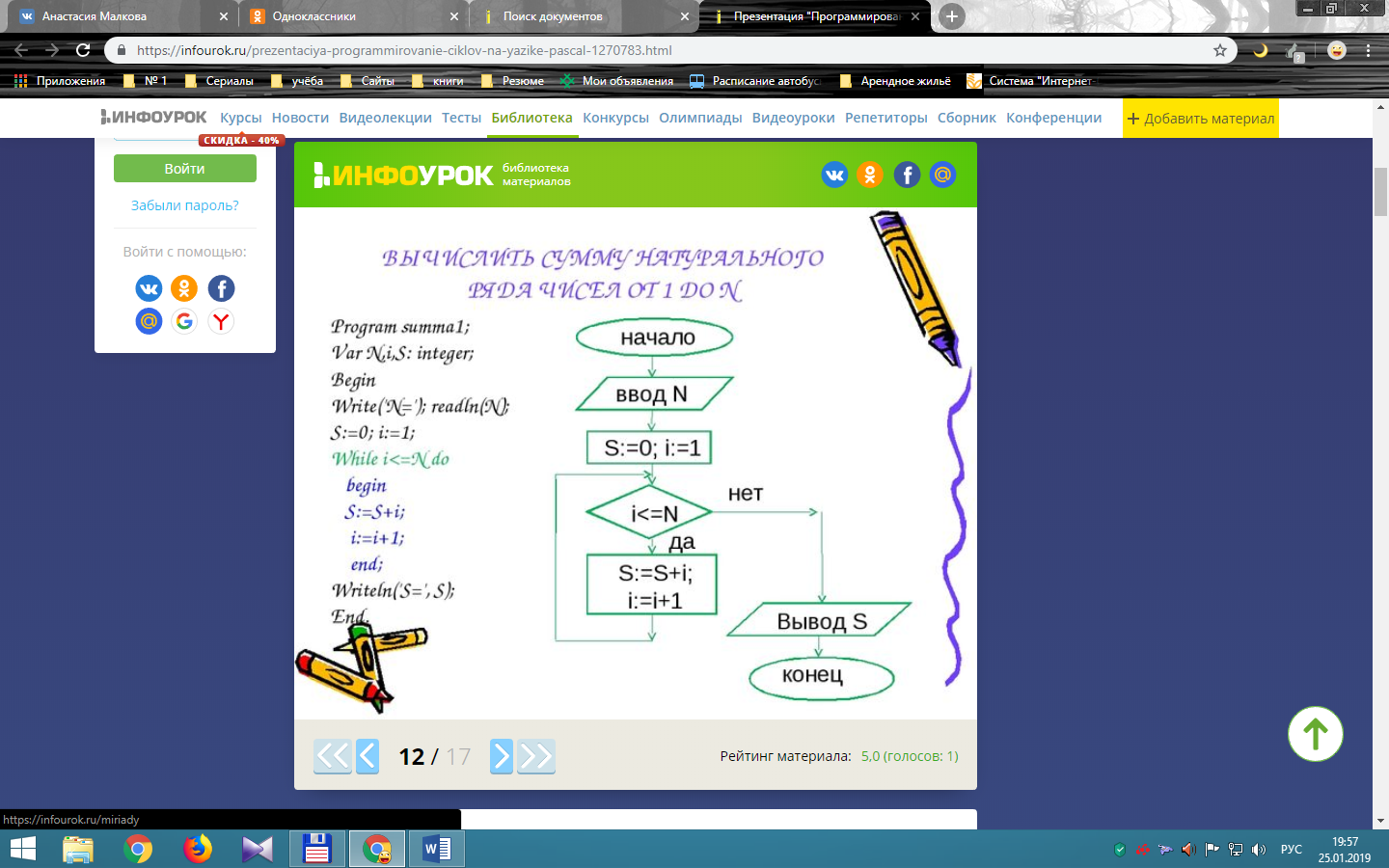
В языке программирования ОР имеется три разновидности цикла (два итерационных и один арифметический):

* Цикл с предусловием (цикл “пока” – whi1е);
* Цикл с постусловием (цикл “до тех пор, пока” repeat...unti1);
* Цикл с параметром (со счетчиком) (цикл “для” – for ... to / downto);



Каждая из разновидностей цикла имеет свои особенности, для каждой из них есть свой круг задач, наиболее естественно решаемых именно с ее помощью.

Сегодня на уроке мы поговорим о циклах с предусловием и параметром (со счетчиком).

* 1. *****Цикл с предусловием.***

«До тех пор пока верно условие, выполнять следующие действия: …».

– Цикл с предусловием имеет следующий вид:

***While****<условие>****do*** {эта часть называется заголовком цикла}

***begin***

*оператор 1;*

*оператор 2;* {эта часть называется телом цикла}

*………….;*

*оператор n;*

***end****;*

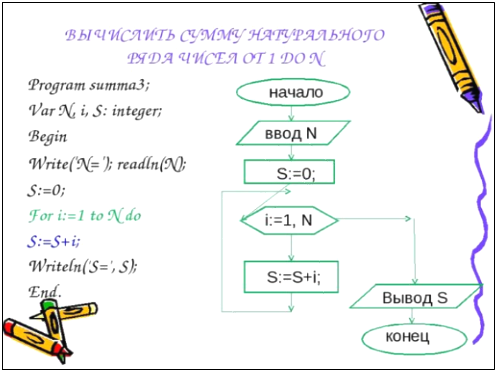
Телом цикла может быть и группа операторов, заключенная в операторные скобки **begin... end**(то есть составной оператор).

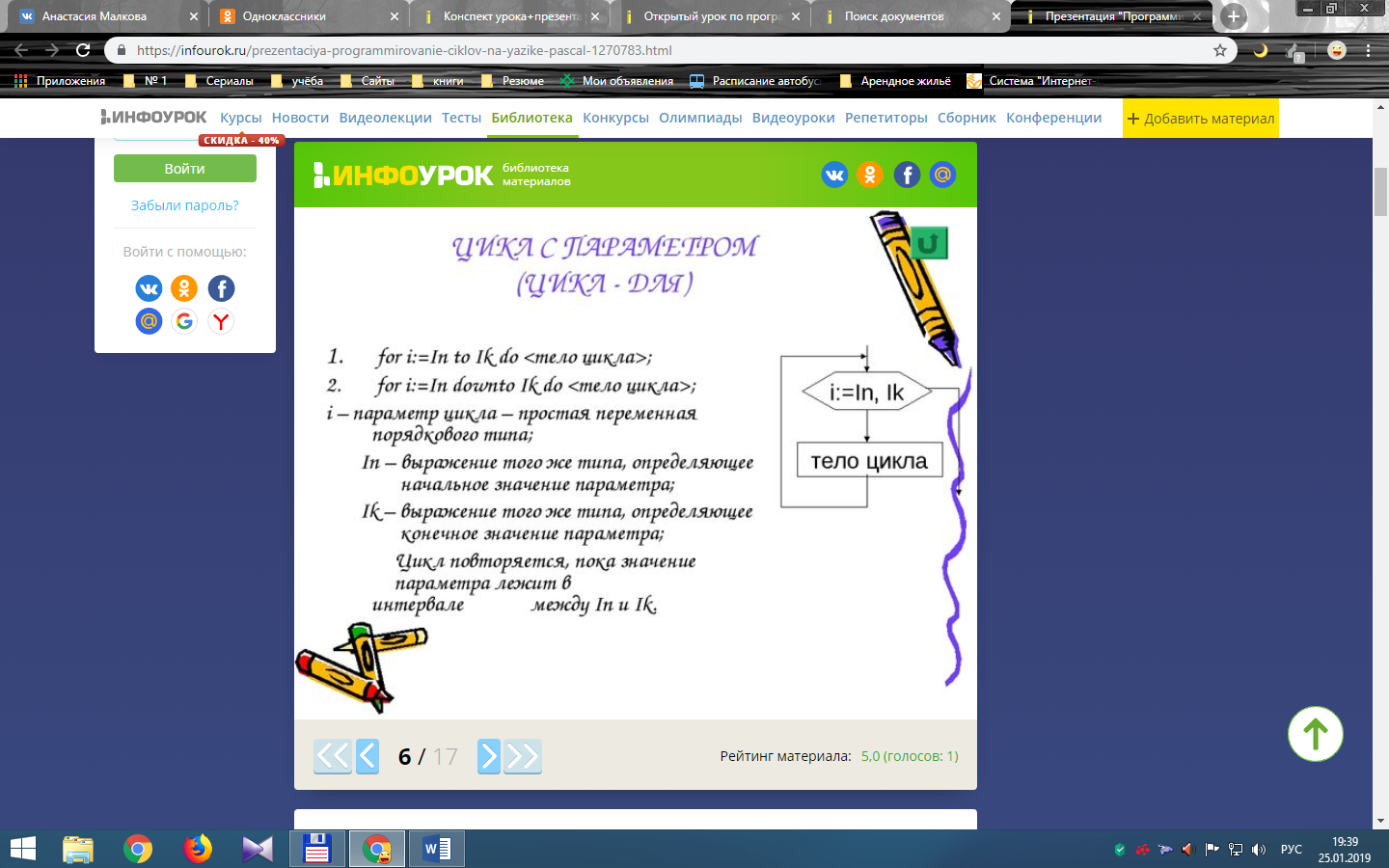
Цикл с предусловием выполняется до тех пор, пока истинно условие в заголовке цикла, причем оно проверяется вначале, потом исполняется оператор.

Переменным, входящим в условие, должны быть присвоены определенные значения до входа в цикл.

В теле цикла должны быть операторы, которые в какой-то момент изменят значение условия, сделав его ложным. Если этого не случится, цикл будет бесконечным.

При возникновении в программе бесконечного цикла говорят, что программа “зациклилась”. Зациклившуюся программу приходится останавливать одновременным нажатием клавиш CtrL + Break, иначе она будет выполняться вечно (точнее, до первого отключения компьютера).

* 1. *****Цикл с параметром (со счетчиком).***



– Цикл со счетчиком имеет следующий вид:

***for i :=*** *начальное\_значение* ***to****конечное\_значение* ***do ...  
оператор;***

Алгоритм, соответствующий инструкции for, представлен на рисунке

– Здесь переменная i, называемая управляющей переменной цикла for (или его счетчиком), является произвольным идентификатором, который объявляется как переменная порядкового (чаще всего целого) типа.

- При выполнении оператора for сначала вычисляется значение выражения начальное\_значение, затем вычисляется значение выражения конечное\_значение, далее управляющая переменная цикла последовательно пробегает все значения от начального до конечного.

- В том случае, когда начальное значение оказывается больше конечного значения, тело цикла не будет выполняться вовсе. Начальное и конечное значения остаются неизменными в ходе выполнения всего цикла fог.

- Параметр цикла i, если он целого типа, пробегает все значения с приращением 1, и его текущее значение не должно изменяться операторами внутри цикла. Переменную-счетчик можно использовать внутри цикла (но ни в коем случае не изменять). Такое изменение не запрещено правилами языка, но его последствия будут непредсказуемы.

- После завершения цикла параметр i считается неопределенным.

Параметр цикла меняется от начального значения до конечного с шагом – 1.

1. **Первичное осмысление и закрепление новой темы.**
   1. ***Портфель программиста:***

Сколько раз программа выведет слово *«привет»*?

*var n: integer;*

*begin n:=0;*

*while n<9 do*

*begin writeln(‘привет');*

*n:=n+2*

*end;*

*end.*

Ответ: 5 раз. А что будет, если убрать begin и end?

* 1. ***Шиворот-навыворот.***

Определите значение переменной S после выполнения программы:

*Var a,S: integer;*

*Begin*

*S:=0;*

*For a:=5 downto 1 do*

*S:=s+2\*a;*

*Writeln('S=', S);*

*End.*

Ответ: S=30

- Что ж, мы молодцы, теперь давайте решим с вами еще одну задачу.

Необходимо просуммировать числа от 1 до 50 (включительно) и вывести на экран полученный результат.

Сначала составим математическую модель:

*S:=1+2+3+…+50;*

Ну а теперь можно написать программу:

*PROGRAM Sum;*

*VAR i,S: integer;*

*BEGIN*

*S:=0;*

*For i:=1 to 50 do*

*S:=S+i;*

*Writeln (‘Сумма чисел от 1 до 50 равна ’,S);*

*END.*

- Молодцы, хорошо справились.

- Ну а теперь можно порешать задачи. Усаживайтесь за компьютеры.

1. **Практическая работа.**

**Задача №1**

Написать программу, которая 10 раз выводит на экран Ваши имя и фамилию.

**Задача №2**

Написать программу, которая выводит таблицу квадратов первых 10 целых положительных чисел.

**Задача №3.\***

Найти среднее арифметическое N первых натуральных чисел.

***Задача 1. Решение:***

*Program а1;*

*Var i:integer;*

*Begin*

*For i:=1 to 10 do*

*Writeln (‘Иванов Иван’);*

*Readln;*

*End.*

***Задача 2. Решение:***

*Program a2;*

*Var a:integer;*

*Begin*

*For a:=1 to 10 do*

*Writeln (a,’ ‘,sqr(a));*

*Readln;*

*End.*

***Задача 3. Решение:***

*Var i:word;*

*N:integer;*

*Sa: real;*

*Begin*

*Readln (n);*

*Sa:=0;*

*For i:=1 to n do*

*Sa:=sa+I;*

*Sa:=sa/n;*

*Writeln(sa:0:4);*

*Readln;*

*End.*

1. **Подведение итогов урока. Рефлексия.**

– Итак, мы познакомились с циклами, важнейшей составной частью большинства программ. Но как нам быть с героем древнегреческого мифа? Мы выяснили, что циклическое действие, которое он выполняет, будет конечным, если существует условие, при выполнении (или невыполнении) которого цикл должен завершиться. Ограничение может быть и по числу повторений цикла. От цикла, который выполняет Сизиф, немного пользы, ведь каждый раз он повторяет одно и то же действие. Цикл в программе при каждом новом повторении выполняет действие, которое хотя бы немного отличается о предыдущего. Так, Сизиф мог бы, скажем, каждый раз вкатывать на гору камень меньшего размера, чем предыдущий. Условием окончания нелегкого сизифова труда было бы в этом случае выкатыванием на гору камня, например, весом в 1 грамм. Следовательно помочь Сизифу можно, изменив алгоритм его работы, введя в него “правильный” цикл.

– Завершим мы тему “Циклы” решением задач на следующем уроке. В каждой задаче вначале нужно придумать алгоритм, а затем написать программу.

– Закончите фразу «Сегодня на уроке я …»

(Выслушать 2-3 ответа.)

– Подведем итоги сегодняшнего занятия. Все вы сегодня поработали хорошо, справились с заданиями, а оценки мы выставим вместе:

Оценивание: самооценка + оценка преподавателя = итоговая.

1. **Постановка задания на дом.**

– Подумайте над решением следующей задачи: Найти произведение натуральных чисел от 1 до *n*.

Молодцы! Спасибо вам за урок! До свидания!