

Навчально-виховний комплекс №12

Рівненської міської ради

**Цикл уроків
з інформатики в 11 класі
фізико-математичного профілю
за темою «Вказівки повторення у мові програмування»**

Підготувала

учитель інформатики

навчально-виховного комплексу №12

Рівненської міської ради

Ковальова Людмила Миколаївна

Рівне

2015

Пояснювальна записка

Тема «Вказівки повторення у мові програмування» є складовою частиною великої теми «Основи алгоритмізації та програмування».

Наведені конспекти п'яти уроків призначені для загальноосвітніх навчальних закладів класів фізико-математичного профілю.

Метою циклу уроків є формування теоретичної бази знань учнів стосовно призначення та правил описування вказівок повторення, а також практичних умінь та навичок: складання й реалізація найпростіших циклічних алгоритмів, введення та налагоджування програм з операторами циклу на комп'ютері.

У роботі пропонується цикл уроків із застосуванням інтерактивної, проектної технологій навчання, що підвищує ефективність уроку та зацікавлює учнів до вивчення теми «Вказівки повторення».

Тематично-поурочне планування теми «Вказівки повторення»

№	Тема уроку	Тип уроку	Зміст уроку	Обладнання
1	Вказівка повторення та її опис мовою програмування	Урок вивчення нового матеріалу	Поняття про вказівку повторення та її використання при розв'язуванні задач. Типи циклів та їх оформлення мовою програмування Паскаль. Призначення та суть операторів циклу, особливості використання операторів циклу	Картки для інтерактивної вправи
2	Створення програм з циклами	Урок формування вмінь і навичок	Задачі на знаходження суми та добутку скінченої кількості чисел, знаходження кількості елементів з певними властивостями	Картки для гри «Лото», картки для інтерактивної вправи

3	Вкладені цикли	Комбінований урок	Поняття про вкладені цикли, створення програм з вкладеними циклами	
4	Створення і реалізація програм з циклами	Практична робота	Задачі на створення програм із командами повторення в середовищі програмування	Інструктивні картки для практичної роботи
5	Створення програми-тесту	Урок систематизації і узагальнення знань	Програма тестового характеру	

Урок 1

Тема. Вказівка повторення та її опис мовою програмування

Мета:

- дати поняття про вказівку повторення та її використання при розв'язуванні задач, про типи циклів та їх оформлення мовою програмування Паскаль, ознайомити з призначенням та суттю операторів циклу, особливостями використання операторів циклу; навчити встановлювати ситуації, у яких доцільніше використовувати певний оператор циклу;
- розвивати кругозір учнів, практичні навички роботи з ПК, пам'ять, уміння працювати у групі;
- виховувати чуйність, взаємодопомогу, дружбу в класі.

Тип уроку: вивчення нового матеріалу.

Обладнання: картки для інтерактивної вправи

Епіграф уроку: «Без повторень немає глибини» Г. Ландау

План уроку

- I. Мотивація навчальної діяльності учнів (2 хв.)
- II. Повідомлення теми та мети уроку (1 хв.)
- III. Надання необхідної інформації (3 хв.)
- IV. Інтерактивна вправа (20 хв.)
- V. Практична робота у середовищі програмування TP 7.1 (15 хв.)
- VI. Підбиття підсумків уроку, оцінювання результатів (2 хв.)
- VII. Повідомлення домашнього завдання (2 хв.)

Перебіг уроку

I. Мотивація навчальної діяльності учнів

Циклічність — загальна властивість розвитку та існування будь-якої біологічної системи. Циклічність в природі, в розмноженні, в життєдіяльності клітини підтримує необхідний баланс для їх існування. Якщо подивитись глобально на винаходи людства, то помітно, що практично всі вони моделюють природні об'єкти. Не виключення і цикл як одна з базових логічних конструкцій.

Формую проблемну ситуацію.

Складіть програму для виведення на екран всіх цілих чисел від 1 до 5; від 1 до 10; від 1 до 1000.

(Очікується використання великої кількості вказівок виведення.)

А чи можна знайти більш раціональний спосіб виконання цього завдання? Яку закономірність слід реалізувати?

Відповідь на це запитання допоможе знайти сьогоднішній урок.

II. Повідомлення теми та мети уроку

Отже, тема нашого уроку «Вказівка повторення та її опис мовою програмування». Після цього уроку ви зможете

- пояснювати призначення та суть операторів циклу;
- наводити особливості використання операторів циклу;
- встановлювати ситуації, у яких доцільніше використовувати певний оператор циклу;
- набудете навичок роботи у групі.

III. Надання необхідної інформації

Цикл — це послідовність операторів, що може виконуватися більше одного разу.

Цикли використовуються при розв'язуванні таких задач, де необхідно повторити деяку послідовність команд більше одного разу.

Відомо два типи команди повторення, що суттєво різняться:

- цикл із відомою заздалегідь кількістю повторів;
- цикл із невідомою кількістю повторів.

Щоб це пояснити, розглянемо приклад. Вчителька в першому класі дає дітям завдання: «Діти, напишіть, будь ласка, десять літер «А» та рядочок літер «Б». Чим відрізняються ці два завдання? В першому випадку зразу ж відомо, скільки разів необхідно повторювати виконання команди «напишіть літеру А», а в другому — кількість літер «Б» буде залежати від різних факторів: почерку дитини, розміру літер, відстані між літерами тощо. Ясно, що в першому випадку ми можемо чітко обумовити кількість повторів, а в другому — необхідно знайти таку умову, перевіряючи яку, дитина зможе закінчити свою

роботу.

Даю дітям можливість знайти цю умову та придумати свої подібні варіанти циклів.

У мові програмування Паскаль існує три типи циклів: *for*, *repeat* та *while*. Якщо кількість повторень відома заздалегідь, використовується оператор *for*, якщо кількість повторень невідома, застосовуються оператори *repeat* або *while*.

IV. Інтерактивна вправа

1) Розповідаю учням про мету вправи за методом «Ажурна плитка», правила, послідовність дій і кількість часу на виконання завдання.

Вправа «Ажурна плитка» дає можливість на уроці вам працювати разом, щоб вивчити значну кількість інформації за короткий проміжок часу, а також заохочує вас допомагати один одному вчитися навчаючись.

Під час роботи за методом «Ажурна плитка» маєте бути готовими працювати в різних групах.

Спочатку ви вивчаєте матеріал у «домашній» групі. Потім в іншій групі ви будете виступати в ролі «експертів» з питання, над яким працювали в «домашній» групі, та отримувати інформацію від представників інших груп.

В останній частині цієї вправи ви знову повернетесь у свою «домашню» групу, для того щоб поділитися тією новою інформацією, що її вам надали учасники інших груп.

2) Учні отримують картку одного з кольорів (синього, червоного, зеленого) з номером на ній (від 1 до 5). Сформувалося три «домашні» групи по 5 осіб у кожній.

3) Пропоную учням об'єднатися у групи відповідно до кольорів карток, які ви отримали («домашні» групи). У «домашніх» групах учні знайомляться зі змістом карток, обмінюються думками, користуючись текстом на картках, роблять у зошитах конспекти.

Зміст синіх карток

Оператор повторення *for*.

Цей оператор повторення інакше називається циклом з параметром (або лічильником). Він складається із заголовку та тіла циклу і може бути поданий у двох форматах:

1. *for* <параметр циклу> := <S1> *to* <S2> *do*

begin <серія>; *end*;

2. *for* <параметр циклу> := <S1> *downto* <S2> *do*

begin <серія>; *end*;

де S1 і S2 — вирази, що визначають початкове і кінцеве значення параметру циклу;

for.. do — заголовок циклу; <серія> — тіло циклу. Тіло циклу може бути простим або складеним.

Цей оператор забезпечує виконання тіла циклу доти, доки не будуть перебрані всі значення параметра циклу від початкового до кінцевого.

Параметр циклу, його початкове і кінцеве значення повинні належати до одного типу даних. При цьому можливе використання будь-якого скалярного типу: цілого (*integer, byte, word, longint*); булівського (*boolean*); символьного (*char*). Дійсний тип даних тут не використовується. Значення параметру циклу послідовно збільшується при *for... to* або зменшується при *for... downto* на одиницю при кожному повторі.

В операторі *for* після *do* може знаходитися складений оператор, у тілі якого *заборонені оператори, що змінюють значення параметру циклу*. Після нормального завершення циклу значення параметру циклу дорівнює кінцевому значенню.

Цикл може не виконатися зовсім, але ніколи не може «зациклитися», на відміну від наступних двох операторів повторення.

Приклад програми з оператором *For*

Умова задачі. Вивести на екран сьогоднішній день тижня 300 разів.

Program Example_1;

Uses crt;

Var i: integer;

```
Begin  
Clrscr;  
For i:=1 to 300 do  
Write(' Понеділок ');  
End.
```

Зміст червоних карток

Оператор повторення *repeat*

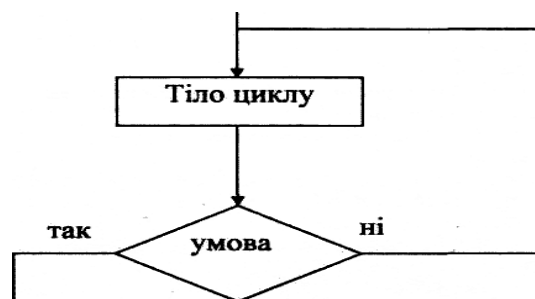
Наступний оператор циклу складається із заголовку (*repeat*), тіла та умови закінчення (*until*). Формат опису:

```
repeat  
<серія> until < умова >;
```

Умова — вираз булівського типу. Про умови ми з вами вже говорили в темі «Команда розгалуження», і всі особливості їх застосування та оформлення в цій команді нічим не відрізняються від попереднього випадку.

В цьому циклі спочатку виконується серія (тіло циклу), а потім перевіряється умова виходу з циклу. Саме тому ця команда повторення інакше називається *циклом із післяумовою*. Якщо умова виходу з циклу хибна (*false*), цикл активізується ще раз, якщо результат істинний (*true*), відбувається вихід із циклу.

Мовою блок-схем цей цикл описується таким чином:



Такий оператор повторення має наступні характерні риси:

- використовується у випадках, коли користувачу не відома заздалегідь кількість повторень;
- тіло циклу виконується хоча б один раз;
- тіло циклу виконується, поки умова хибна *False*;
- у тілі може знаходитися будь-яка кількість операторів без операторних

дужок (*begin... end*);

- принаймні один оператор у тілі циклу повинен змінювати значення умови, інакше цикл буде виконуватися нескінченно.

Для передчасного виходу з циклу можна присвоїти параметру циклу значення, що виходить за діапазон дозволених значень.

Приклад програми з оператором *Repeat*

Умова задачі. Перевірка коректності введення. Дано три числа, що задають величини кутів трикутника. Визначити, чи можна побудувати трикутник, що має задані кути. Якщо ні, користувач повинен ввести інші дані.

Розв'язання:

```
Program Example_2;
```

```
Uses crt;
```

```
Var a,b,c: real;
```

```
Begin
```

```
Clrscr;
```

```
Repeat
```

```
Write('Введіть довжини сторін трикутника: ');
```

```
Readln(a,b,c);
```

```
Until
```

```
(a>0)and(b>0)and(c>0)and(a+b+c)=180;
```

```
End.
```

Зміст зелених карток

Оператор повторення *while*

Оператор *while* аналогічний оператору *repeat*, але перевірка умови виконання тіла циклу виконується на самому початку оператора, тому цикл інакше називається *циклом із передумовою*.

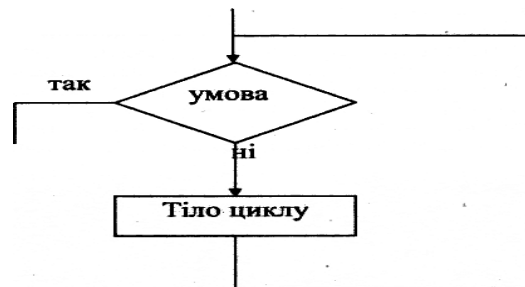
Формат опису:

```
while < умова > do begin <серія>; end;
```

Умова — булівський вираз, а *серія* — простий або складений оператор. Перед кожним виконанням тіла циклу обчислюється значення виразу умови.

Якщо результат є істинним (*true*), тіло циклу виконується, а інакше відбувається вихід із циклу і перехід до першого після *while* оператору. Якщо перед першим виконанням циклу значення виразу було хибним *false*, то тіло циклу взагалі не виконується і відбувається перехід на наступний оператор.

Мовою блок-схем цей цикл описується таким чином:



Як і в попередньому випадку, програміст сам повинен подбати про переприсвоєння значення змінної, що визначає умову виходу з циклу, інакше він буде нескінченним.

Приклад програми з оператором *While*

Умова задачі. Дано натуральне число *N*. Визначити кількість цифр у числі.

Розв'язок:

Program Example_3;

Uses crt;

Var N: longxnt; Counter: integer;

Begin

Clrscr;

Write('Введіть число: ');

Readln(N);

Counter:= 0;

While N > 0 do

Begin

Counter:=Counter+1; {Підрахунок кількості цифр}

N:=N div 10; {Відкидання останньої цифри}

End;

Writeln(' Кількість цифр у заданому числі дорівнює ', N);

Readkey;

End.

4) Пропоную учням об'єднатися в «експертні» групи, в яких учні з різних «домашніх» груп повідомляють інформацію про «свої» оператори циклу.

5) Пропоную учням знову об'єднатися в «домашні» групи і обмінятися щойно отриманою інформацією.

6) Клас збирається у спільне коло для підбиття підсумків вправи. За методом бесіди обговорюю з учнями новий матеріал.

V. Практична робота у середовищі програмування TP 7.1.

Задача. Знайти суму всіх натуральних чисел від 1 до 100.

Розв'язання:

Program Example_1;

Uses crt;

Var Sum, i: integer;

Begin

Clrscr; Sum:= 0;

For i:= 1 to 100 do Sum:= Sum + i;

Writeln('Sum= ', Sum); Readkey;

End.

Резервна задача: знайти добуток всіх натуральних чисел з проміжку [a;b], які є точними квадратами цілих чисел.

VI. Підбиття підсумків уроку, оцінювання результатів

Як, на вашу думку, можна використати циклічні конструкції при створенні програми-тесту? (підводжу до підсумку про застосування циклу для повторення тесту спочатку, або для встановлення пароля)

VII. Повідомлення домашнього завдання

- опрацювати конспект
- скласти програму, яка запитуватиме, скільки Вам років, доки ви не введете „100”.
- Задача. Знайти суму всіх непарних натуральних чисел від 3 до 100.

Урок 2

Тема. Створення програм з циклами

Мета:

- вдосконалити методику складання програм з використанням циклів, навчити застосовувати їх для розв'язування типових задач;
- розвивати вміння працювати з середовищем програмування, цілеспрямованість;
- виховувати здатність долати перешкоди на життєвому шляху.

Тип уроку: формування вмінь і навичок.

Обладнання: картки для гри «Лото», картки для інтерактивної вправи

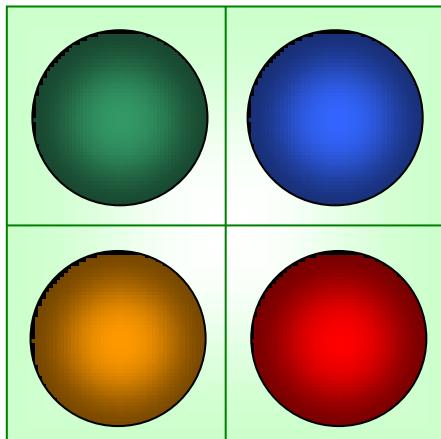
План уроку

- I. Актуалізація опорних знань (2 хв.)
- II. Мотивація навчальної діяльності учнів (1 хв.)
- III. Оголошення теми та мети уроку (1 хв.)
- IV. Надання необхідної інформації (1 хв.)
- V. Інтерактивна вправа (20 хв.)
- VI. Практична робота у середовищі програмування TP 7.1 (15 хв.)
- VII. Підбиття підсумків, оцінювання результатів (3 хв.)
- VIII. Повідомлення домашнього завдання (2 хв.)

Перебіг уроку

I. Актуалізація опорних знань

Гра „Лото”. Учні на ігровому полі вибирають фішку певного кольору, відкривають її і отримують номер запитання, на яке повинні дати відповідь.



- Вказівка розгалуження
- Цикл з параметром
- Цикл з передумовою
- Цикл з післяумовою

II. Мотивація навчальної діяльності учнів

Учням пропонується скласти алгоритм знаходження середньої температури за місяць та підрахунку кількості теплих ($t^{\circ}\text{C} > 0$) і холодних ($t^{\circ}\text{C} < 0$) днів. Учні доходять висновку, що основним завданням у цьому алгоритмі є знаходження суми всіх середньодобових температур та організація підрахунку кількості відповідних днів. Повідомляю, що при розв'язуванні різноманітних обчислювальних задач часто виникає потреба підраховувати кількість якихось подій або якихось даних, що задовольняють певним умовам, накопичувати суму або добуток. Для розв'язання таких задач також використовуються оператори циклу.

III. Оголошення теми та мети уроку

Після цього уроку ви зможете:

- пояснювати алгоритм знаходження суми та добутку скінченної кількості чисел, знаходження кількості елементів з певними властивостями;
- наводити особливості знаходження суми та добутку скінченної кількості чисел, підрахунку кількості елементів з певними властивостями;
- складати алгоритми на знаходження суми, добутку скінченної кількості чисел, знаходження кількості елементів з певними властивостями та реалізовувати ці алгоритми на одній з мов програмування;
- набути навичок роботи у групі.

IV. Надання необхідної інформації

Для обчислення суми, добутку або кількості під ці величини виділяється спеціальна змінна відповідного типу. Далі, поєднуючи оператор циклу з оператором присвоєння, можна організувати накопичення суми, добутку або кількості. В кожному з цих випадків накопичення відбувається за такою схемою: спеціальна змінна, де буде відбуватися накопичення, - це один кошик, у який весь час докладають дані.

V. Інтерактивна вправа

Учням пропонується виконати інтерактивну вправу за методом «Ажурної

плитки». Для цього учні мають об'єднатися в групи. А саме: учні отримають картку одного з кольорів (синього, червоного або жовтого) з номером на ній (від 1 до 5). У результаті формуються три "домашніх" групи по 5 учнів у кожній. Групи отримують картки із завданнями. Кожній «домашній» групі пропонується ознайомитися зі змістом карток, розглянути наведені блок-схему та програму, відповісти на питання, обговорити висновки, наведені в картах, записати у зошит програму та висновки.

Зміст синіх карток

Обчислення суми

Скласти програму для обчислення суми цілих чисел від 1 до 100.

Для накопичення суми введемо спеціальну змінну, наприклад S. Під час обчислення суми початкове значення змінної S, де накопичується сума, має дорівнювати нулеві. Для цього використовують команду присвоєння $S:=0$. Оскільки треба перебрати натуральні числа від 1 до 100 з кроком 1, то доцільно скористатися оператором циклу з параметром.

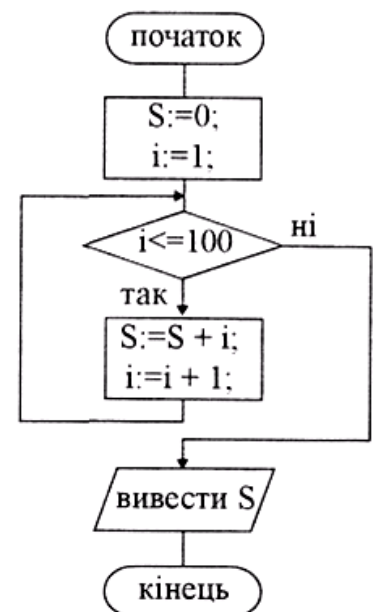
Ознайомтеся з блок-схемою алгоритму накопичення суми та розгляньте програму, яка реалізує цей алгоритм (файл prog1.pas):

```
var s,i:integer;  
begin  
  s:=0;  
  for i:=1 to 100 do s:=s+i;  
  writeln('s=',s);  
end.
```

Змінна i набуватиме значення: 1, 2, 3,...,100,
а змінна S - такі: 0, 1, 3, 6, 10, 15, ... ,5050.

Як треба змінити програму, щоб вона обчислювала суму чисел від 200 до 300?

Висновки: Для накопичення суми треба:



- ввести змінну для обчислення суми;
- надати цій змінній початкового значення 0;
- організувати цикл, у тілі якого записати оператор
сума:=сума+доданок;

Зміст червоних карток

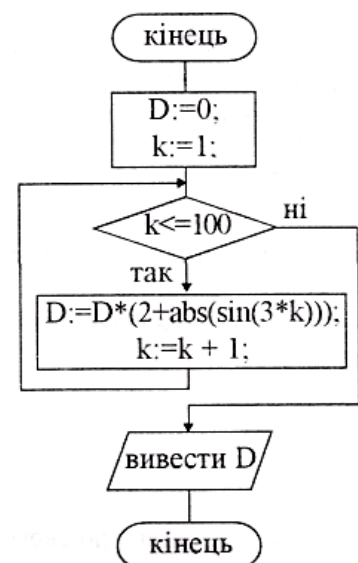
Обчислення добутку

Обчислити добуток перших п'яти елементів послідовності, значення яких знаходяться за формулою: $a_k = 2 + |\sin 3k|$, $k = 1, 2, \dots, 5$. Для накопичення добутку введемо спеціальну змінну, наприклад, D. Під час обчислення добутку початкове значення змінної, де накопичується добуток, має дорівнювати одиниці. Отже, $D:=1$. Оскільки треба перебрати натуральні числа (індекси елементів) від 1 до 5 з кроком 1, то доцільно скористатися оператором циклу з параметром.

Ознайомтеся з блок-схемою алгоритму накопичення добутку та розгляньте програму, яка реалізує цей алгоритм (файл prog2.pas):

```
var d, k:integer;
begin d:=1;
for k:=1 to 5 do d:=d*(2+abs(sin(3*k)));
writeln ( 'd=' ,d: 7 : 1 );
end.
```

Змінна k набуватиме значення: 1, 2, 3,...,5. Як треба змінити програму, щоб вона обчислювала добуток перших десяти елементів?



Висновки:

Для накопичення добутку треба:

- ввести змінну для обчислення добутку;
- надати цій змінній початкового значення 1;
- організувати цикл, у тілі якого оператор добуток:=добуток * множник;

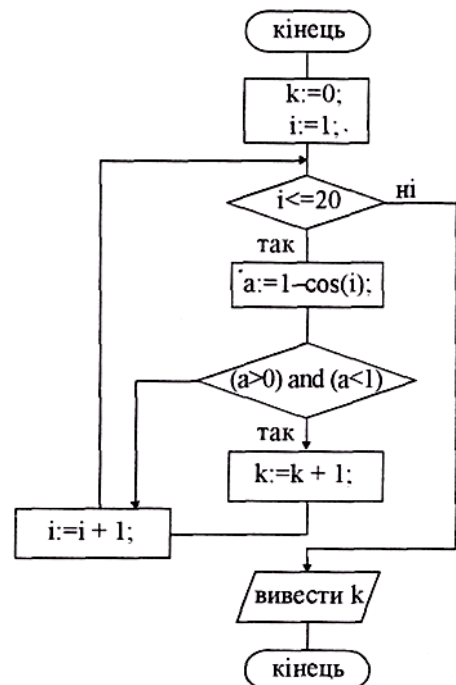
Обчислення кількості елементів з певними властивостями

Скільки елементів послідовності $a_i = 1 - \cos i$, де $i=1, 2, \dots, 20$ задовольняють умову $0 < a_i < 1$?

Для обчислення кількості елементів, що задовольняють вказану умову, введемо спеціальну змінну, наприклад, k . Під час обчислення кількості початкове значення змінної, де буде накопичуватися кількість, має дорівнювати нулеві ($k:=0$;). Оскільки треба перебрати натуральні числа (індекси елементів) від 1 до 20 з кроком 1, то доцільно скористатися оператором циклу з параметром, у тілі якого має бути команда $k:=k+i$;

Ознайомтесь з блок-схемою алгоритму обчислення кількості та розгляньте програму, яка реалізує цей алгоритм (файл prog3.pas):

```
var a:real; i, k:integer;
begin
k:=0;
for i:=1 to 20 do
begin
a:=1-cos(i);
if (a>0)and(a<1)
then k:=k+1;
end;
writeln('k=',k) ;
end.
```



Як треба змінити програму, щоб за її допомогою можна було б знайти кількість елементів, більших за 1?

Висновки:

Для накопичення кількості елементів з певними властивостями треба:

- ввести змінну для обчислення кількості;
- надати цій змінній початкового значення 0;
- організувати цикл, у тілі якого записати оператор $\text{кількість} := \text{кількість} + 1$;

2) Учням пропонується об'єднатися в групи відповідно до своїх номерів («експертні» групи). У кожну експертну групу потраплять представники з кожної «домашньої» групи. Учні з різних «домашніх» груп, користуючись зробленим конспектом, повідомляють правила знаходження суми, добутку та кількості елементів з деякою властивістю.

3) Пропонується знову об'єднатися в «домашні» групи, учасники яких обмінюються між собою відомостями, отриманими в «експертних» групах, зіставляються правила обчислення суми, добутку та кількості.

4) «Домашні» групи отримують завдання на картках.

Зміст карток

Скласти алгоритми розв'язання задач. Записати ці алгоритми мовою програмування. Порівняти отримані програми. Перевірити, чи відповідають ці програми висновкам, наведеним у картках з попередньої вправи.

Задача 1. Підрахувати кількість теплих днів за один місяць за умови, що середньодобова температура вводиться з клавіатури.

Задача 2. Знайти середню температуру за місяць за умови, що середньодобова температура вводиться з клавіатури.

Задача 3. Знайти середнє геометричне п'яти чисел, введених з клавіатури.

5) Клас об'єднується в загальне коло для підбиття підсумків вправи. Представники кожної «домашньої» групи повідомляють свій варіант однієї з програм до задач 1-3.

VI. Практична робота у середовищі програмування TP 7.1

Учням пропонується виконати програми задач 1 – 3 за комп'ютером.

VII. Підбиття підсумків, оцінювання результатів

Запитання для бесіди:

1. Які оператори мови програмування Паскаль використовуються для знаходження суми, добутку та кількості елементів з певною властивістю?
2. Яким повинно бути початкове значення змінної, у якій передбачається накопичувати суму?
3. Яким має бути початкове значення змінної, у якій передбачається накопичувати добуток?

4. Яким повинно бути початкове значення змінної, у якій передбачається накопичувати кількість?
5. Яка загальна схема знаходження суми (добутку, кількості)?
6. Які помилки можуть виникнути при знаходженні суми, добутку та кількості?
7. Як ви гадаєте, чи виконали ми ті завдання, що поставили на початку уроку?
8. Які труднощі виникали у вас протягом уроку?
9. Що ви порадите однокласникам для подолання труднощів, що виникали?

VIII. Повідомлення домашнього завдання

Повторити теоретичний матеріал.

Розв'язати задачу:

Умова: Капосний папуга навчився висмикувати у дідусі Василя волосся, яке ще залишилося у того на голові. Почавши з однієї волосини, він щодня збільшував порцію вдвічі. Через скільки днів дідусеві не знадобиться гребінець, якщо спочатку в нього на голові було аж N волосин?

Рекомендації: аналізувати наявність волосся на голові необхідно після того, як папуга вже висмикнув чергову порцію волосся. А «знущення» над дідусем скінчиться тоді, коли гребінець йому стане непотрібним, тобто кількість волосся на голові дорівнюватиме нулю. Зверніть увагу, що в цій задачі змінна S використовується для підрахунку чергової порції волосся, що підлягає висмикуванню капосним папугою.

Урок 3

Тема. Вкладені цикли

Мета:

- навчити створювати програми з вкладеними циклами;
- розвивати логічне мислення, уміння працювати з середовищем програмування, логіку мислення. Розвивати різні види пам'яті, уміння виразити свою думку;
- виховувати загальнолюдські цінності особистості.

Тип уроку: комбінований

План уроку

- I. Актуалізація опорних знань. Перевірка домашнього завдання (4 хв.)
- II. Мотивація вивчення нового матеріалу (3 хв.)
- III. Пояснення нового матеріалу (15 хв.)
- IV. Закріплення вивченого, робота у середовищі програмування (20 хв.)
- V. Підведення підсумків. Повідомлення д/з (3 хв.)

Перебіг уроку

I. Актуалізація опорних знань. Перевірка д/з.

- що таке лічильник?
- що таке лічильник в програмуванні? Як він описується і працює?

II. Мотивація вивчення нового матеріалу.

Який принцип роботи спідометра? Демонструю результат роботи програми «Спідометр»

Program Spidometr;

uses crt;

var i,j,k:integer;

Begin

Writeln('Імітація роботи спідометра');

for i:=0 to 9 do

for j:=0 to 9 do

for k:=0 to 9 do

begin

```
        clrscr;  
        gotoxy(40,20);  
        write(i,' ',j,' ',k,' ');  
        delay(2000);  
    end;  
  
    readln;  
  
End.
```

III. Пояснення нового матеріалу

Повідомляю тему, мету уроку.

Тілом циклу може бути знову циклічна структура. Такі цикли називають вкладеними. Цикл, що містить у собі інший цикл, називається зовнішнім, а цикл, що міститься в тілі іншого циклу, - внутрішнім. Як зовнішній, так і внутрішній цикли можуть бути з параметром, передумовою чи післяумовою.

Вкладені цикли характеризуються рівнями вкладення. Зовнішній цикл має рівень вкладень 0, внутрішній - 1. Якщо ж тілом цього внутрішнього циклу знову є цикл, то його рівень вкладення буде 2 і т.д. У цьому випадку цикл з рівнем вкладення 1 є внутрішнім щодо циклу 2. Обмежень на глибину вкладення циклів (тобто кількість рівнів) у мові Паскаль немає, вони виникають на етапі реалізації внаслідок обмежень пам'яті конкретного комп'ютера.

Параметри циклів у випадку вкладених циклів змінюються так: спочатку змінюється параметр внутрішнього циклу, набуваючи всіх своїх значень. Потім зовнішній цикл змінить значення на один крок і знову параметр внутрішнього циклу набуде всіх значень. Так триває доти, доки параметр зовнішнього циклу не набуде всіх своїх значень.

Задача 1. Вивести на екран таблицю множення чисел від 1 до 9.

Розв'язання:

```
Program rrrr;  
var i, j : byte;  
begin  
    for i:=1 to 9 do  
        begin
```

```
writeln;
for j:=1 to 9 do write(i,'*',j,'=',i*j:8)
    end;
end.
```

Задача 2. Обчислити:

$$1+1/2+1/3+1/4+\dots+1/n+2/1+2/2+2/3+2/4+\dots+2/n+3/1+3/2+3/3+\dots+3/n.$$

Розв'язання:

```
Program rrrr;
var i, j, n : byte;
begin
    readln (n);
    for i:=1 to 3 do
        begin
            writeln;
            for j:=1 to n do S:=S+i/j
                end;
            write('S=',S:8)
        end.
```

IV. Закріплення вивченого, робота у середовищі програмування

ТР 7.1

Пропоную учням розв'язати самостійно та опрацювати програму у середовищі програмування:

Вивести на екран таблицю множення чисел: від 10 до 19; від 20 до 29; ...

V. Підведення підсумків. Повідомлення д/з

Обчислити:

$$10+10/2+10/3+10/4+20/1+20/2+20/3+20/4+30/1+30/2+30/3+30/4.$$

(Аналізую розв'язання задачі)

Урок 4

Тема. Створення і реалізація програм з циклами

Мета:

- закріпити навички роботи учнів із середовищем програмування, його інтерфейсом. Перевірити та оцінити уміння учнів створювати програми із командами повторення в середовищі програмування та підлагоджувати їх;
- розвивати практичні навички роботи з ЕОМ та клавіатурою, логічне мислення;
- виховувати самовідданість та вольові риси характеру.

Тип уроку: практична робота

Обладнання: інструктивні картки для практичної роботи

План роботи

- I. Організаційний момент (10 хв.)
- II. Повідомлення теми і мети практичної роботи (5 хв.)
- III. Практична робота (25 хв.)
- IV. Повідомлення домашнього завдання (2 хв.)
- V. Підведення підсумків роботи. Оцінювання результатів (3 хв.)

Перебіг уроку

I. Організаційний момент

Перевіряю готовність учнів класу до практичної роботи, повторюємо правила техніки безпеки при роботі за комп'ютером. Пояснюю принципи проведення практичної роботи та оцінювання результатів.

II. Повідомлення теми і мети практичної роботи

III. Практична робота

Учні працюють за інструктивними картками. Вчитель скеровує індивідуальну самостійну роботу учнів, надає необхідні консультації.

IV. Повідомлення домашнього завдання

Скласти завдання для тесту за обраною вами темою

V. Підведення підсумків практичної роботи. Оцінювання результатів

Практична робота «Створення програм з циклами»

Мета: Закріпити вміння створювати і реалізовувати програми з циклами

Хід роботи

1. Завантажте ТР.7.1. Створіть новий документ і збережіть його
2. Виберіть рівень складності завдання. Номер задачі, яку потрібно розв'язати, співпадає з номером варіанту. Створіть програму для розв'язання задачі
3. Продемонструйте результати роботи викладачу. Підготуйте та здайте звіти.

I РІВЕНЬ

Знайти значення:

- 1) $1*3*5*...*101$;
- 2) $(1+0.1)(2+0.2)...(9+0.9)$;
- 3) $(1+2)+(1+2+3)+...+(1+2+...+50)$;
- 4) $(100+10 \cos 0.1)(100+10 \cos 0.2)...(100+10 \cos 10)$.

II РІВЕНЬ

- 1) Протабулювати значення функції $y=\sin 3x^2$ проміжку $[-5;6]$ з кроком 0,8.
- 2) Протабулювати значення функції $y=\cos^2 x$ проміжку $[-2;8]$ з кроком 0,6.
- 3) Протабулювати значення функції $y=\cos |x|$ проміжку $[-4;4]$ з кроком 0,3.
- 4) Протабулювати значення функції $y=\ln z^2$ проміжку $[-5;3]$ з кроком 0,4.

III РІВЕНЬ

За даними співвідношеннями визначити n-й член числової послідовності ($n>2$):

- 1) $x_n = x_{n-1} + x_{n-2}$; $x_0 = x_1 = 1$;
- 2) $x_n = 2x_{n-1} + 3x_{n-2}$; $x_0 = 0, x_1 = 9$;
- 3) $x_n = x_{n-1} + x_{n-2} + x_{n-3}$; $x_0 = x_1 = 1, x_2 = 6$;
- 4) $x_n = x_{n-1} + 4x_{n-3}$; $x_0 = x_1 = x_2 = 2$;

IV РІВЕНЬ

Обчислити:

- 1) $\sum_{i=1}^{100} \sum_{j=1}^{50} (i+j)$;
- 2) $\sum_{i=1}^{30} \sum_{j=1}^i (1+i^2+j^2)$;
- 3) $\sum_{i=1}^{20} \sum_{j=1}^{30} (i/j)$;
- 4) $\sum_{i=10}^{100} \sum_{j=5}^{50} \cos(i-j)$.

Примітка: знак $\sum$ означає суму всіх елементів з вказаного проміжку.

Наприклад $\sum_1^{40} x$ -це сума виду $1+2+3+...+39+40$; $\sum_1^{40} (x^2-1)$ -сума виду $(0+3+8+...+(40^2-1))$.

Урок 5

Тема. Створення програми-тесту

Мета:

- навчитись застосовувати здобуті знання, вміння та навички при написанні програми для тестування знань. Закріпити вміння і навички створювати і реалізовувати алгоритми з використанням команд розгалуження та повторення;
- розвивати творчі здібності учнів, практичні навички роботи з ЕОМ;
- виховувати почуття відповідальності за свої вчинки.

Тип уроку: урок систематизації і узагальнення знань

План уроку

- I. Організаційний момент (2 хв.)
- II. Повідомлення теми і мети проектної діяльності (3 хв.)
- III. Робота над проектом „ТЕСТ” (15 хв.)
- IV. Налаштування програм (10 хв.)
- V. Презентація та захист проектів. (10 хв.)
- VI. Підбиття підсумків. Оцінювання результатів(5 хв.)

Перебіг уроку

I. Організаційний момент

Об'єднання учнів у групи.

II. Повідомлення теми і мети проектної діяльності

Вивчення теми «Вказівки повторення» ми завершимо творчим проектом «Створення програми-тесту». Однією з цілей даного уроку є показати, як може бути застосований вивчений матеріал при створенні навчальних засобів. Вдома ви склали тести із обраної вами теми. На уроці ми спробуємо написати програму-тест, яка буде визначати правильний або не правильний варіант відповіді вказав користувач, а також підраховувати кількість набраних балів та виставляти оцінку за роботу. Цей тест буде навчального, а не контролюючого призначення, тому користувач буде мати

змогу відповідати на питання до отримання привальної відповіді і лише потім переходити до іншого запитання.

III. Робота над проектом „ТЕСТ”

Учні протягом уроку реалізують тестуючу програму контролюючого характеру. Вчитель скеровує індивідуальну самостійну роботу учнів, надає необхідні консультації

IV. Налагодження програм

V. Презентація та захист проектів.

1. Виступ-презентація проектів
2. Обговорення

VI. Підбиття підсумків. Оцінювання результатів

Проводиться бесіда: прошу учнів висловити думку, чи залежить якість програм від професійного рівня фахівця? Які якості повинен мати фахівець? Чи досягнуто мету, яку поставили на початку уроку?