**Урок**

**Тема: “** Запис логічних виразів мовою програмування. Операції порівняння. Алгоритмічна конструкція розгалуження та її графічне подання. Оператори розгалуження.**”**

**Дата: 27.09.2016 р.**

**Мета уроку:**

***навчальна:*** навчити розв’язувати задачі з використанням команди

розгалуження та

записувати їх мовою програмування.

***виховна:*** виховувати у учнів цікавість до предмета, бережливість, почуття відповідальності

***розвиваюча:*** розвивати у учнів логічне мислення, уяву, увагу, пам'ять.

**Тип уроку:** урок засвоєння нових знань

**Обладнання:**

* комп'ютер, інтерактивна дошка.

**План уроку:**

1. Організаційний момент.
2. Актуалізація знань.
3. **Вивчення нового матеріалу з поетапним закріпленням**.
4. Домашнє завдання
5. Запитання учнів.
6. Підсумок уроку

**Хід уроку:  
I. Організаційний момент.**

Привітання, перевірка присутніх. Пояснення ходу уроку.

**II. Актуалізація знань.**

1. Що таке алгоритм?
2. Типи алгоритмів.
3. Що таке програма?
4. Які мови програмування ви знаєте?
5. Структура програми, написаної мовою програмування Паскаль.
6. Змінна, стала, типи змінних.
7. Арифметичні операції.
8. Порядок виконання операцій.
9. Команда присвоювання.
10. Стандартні математичні функції.
11. Процедура читання.
12. Процедура запису.

# Основні поняття математичної логіки: логічні константи, логічні змінні, логічні вирази.

# Логічні операції: кон’юнкція, диз’юнкція, заперечення.

# Таблиці істинності.

# Логічні формули.

**ІІІ. Вивчення нового матеріалу**

**Повторення**

**Лінійні алгоритми**, це алгоритми, в яких всі етапи рішення задачі виконуються строго послідовно.

**Розгалуженим** називають такий **алгоритм**, в якому вибирається один з декількох можливих варіантів обчислювального процесу.

Кожен подібний шлях називається **гілкою алгоритму**.  
Ознакою розгалуження алгоритму є наявність операцій перевірки умови.

Розрізняють два види умов - прості і складені.

**Простою умовою** називається вираз, що складається з двох арифметичних виразів або двох текстових величин, пов'язаних одним із знаків:

**< -** Менше, ніж ...

**> -** Більше, ніж ...

   <= - Менше, ніж ... або дорівнює

**>=** - Більше, ніж ... або дорівнює

**<>** - Не дорівнює

**=**  - Дорівнює.

**Вираз,** про який, при підстановці в нього деяких значень змінних, можна сказати правдивий (вірний) чи помилковий (невірний) називаються **булевим (логічним**) виразом.

**Змінна**, яка може приймати одне з двох значень: True (правда) або False (брехня), називається **булевой (логічного)** змінної.

**Складеною умовою** називається вираз, в якому використовуються логічні операції **and, or, not**.

**Логічна операція** **and** дає результат **true** тоді і тільки тоді, коли обидва операнди мають значення **true**.

**Логічна операція** **or** дає результат **true** тоді, коли хоча б один операнд має значення **true**.

**Логічна операція** **not** завжди дає результат, протилежний значенню її операнда.

Логічну операцію and ще називають логічним множенням, а логічну операцію or - логічним додаванням.

Кожна програма повинна здійснювати контроль за допустимостью величин, що беруть участь в обчисленнях. Тут ми стикаємося з розгалуженням нашого алгоритму в залежності від умови. Для реалізації таких умовних переходів у мові Паскаль використовують оператори **If і Else**, а також оператор безумовного переходу **Goto**.

У загальному виглядіу повна форма умовного оператора має вигляд:

if <логичний вираз>   
  then  
    <оператор 1>   
  else   
    <оператор 2>

Умовний оператор працює за наступним алгоритмом:

Спочатку обчислюється значення логічного виразу, розташованого за службовим словом IF. Якщо його результат істина, виконується <оператор 1>, розташований після слова THEN, а дії після ELSE пропускаються; якщо результат невірний, то, навпаки, дії після слова THEN пропускаються, а після ELSE виконується <оператор 2>. Якщо в якості оператора повинна виконатися серія операторів, то вони беруться в операторні дужки begin-end. Конструкція Begin ... End називається складеним оператором.

if <логічний вираз>  
  then  
    begin   
      оператор 1;  
      оператор 2;  
      ...  
    end  
  else   
    begin  
      оператор 1;  
      оператор 2;  
      ...  
    end;

Більшість операторів в програмах на мові Паскаль закінчуються крапкою з комою, але після деяких операторів крапка з комою не ставиться.

**Загальні правила вживання крапки з комою:**1. Кожний опис змінної і визначення константи закінчуються крапкою з комою.  
2. Кожен оператор у тілі програми завершується крапкою з комою, якщо відразу за ним не йдуть зарезервовані слова **End, Else, Until**.  
3. Після певних зарезервованих слів, таких, як **Then, Else, Var, Const, Begin**, ніколи не ставиться крапка з комою.

Неповна форма умовного оператора має наступний вигляд:

if <логічний вираз>   
  then  
    <оператор>;

Якщо вираз, розташований за службовим словом IF. в результаті дає істину, виконуються дії після слова THEN, в іншому випадку ці дії пропускаються.

Часто доводиться розглядати не два, а більшу кількість варіантів. Це можна реалізувати, використовуючи декілька умовних операторів. Коли оператор if з'являється всередині іншого оператора if, вони вважаються **вкладеними.**

**Приклади**

**Задача 1.** Розв’язати квадратне рівняння.

Program kv\_rivnyannya;

Uses CRT

var a,b,c:real;

x1,x2:real;

D:real;

begin

clrscr;

write('a=>');

readln(a);

write('b=>');

readln(b);

write('c=>');

readln(c);

D:=sqr(b)-4\*a\*c;

if D>=0 then

Begin

X1:=(-b+sqrt(D))/(2\*a);

X2:=(-b-sqrt(D))/(2\*a);

writeln('x1=>',x1:4:2);

writeln('x2=>',x2:4:2);

end

else

writeln(‘Nema koreniv y rivnyannya’

readln;

end.

**Задача** . Залежно від розміру суми, розмір податку з неї розраховується за такою схемою: якщо сума не перевищує деяку величину ***а*** , то податок не вираховується, якщо сума більша за ***а*** , але не перевищує ***b*** ,то податок становить 10%, якщо сума більша за ***b*** , але не перевищує ***с*** , то податок становить 25% і якщо сума більша за ***с*** , то податок становить 50%. Визначити, який податок буде нарахований із суми розміром ***S***.

Program podatok;  
Var  
  a, b, c s,p: real;   
Begin  
  writeln(‘Введить числа a, b, c’);   
  readln(a,b,c);

writeln(‘Введите суму’);   
  readln(s);  
  if (s<=a) then  
    p:=0

  if (s>a) and (s<=b) then

    p:=s\*0.1;

  if (s>b) and (s<=c) then

    p:=s\*0.25;

  if (s>c) then

    p:=s\*0.5;

  writeln(‘p=’,p:5:2);  
  readln;  
End.

**Задача** Дано три різні числа, найти среднє з них. Среднім називається число, яке більше найменшого з даних чисел, але меньше найбільшого.

Program seredne;

Var

a, b, c, ser: integer;

Begin

writeln(‘Введить числа a, b, c');

readln(a,b,c);

if ((a>b) and (b>c))or((a<b)and(b<c)) then

ser:=b;

if ((b>a) and (a>c))or((b<a)and(a<c)) then

ser:=a;

if ((a>c) and (c>b))or((c>a)and(c<b)) then

ser:=c;

writeln('seredne=',ser);

readln;

End.

Використовуючи команди розгалуження скласти програми для розв’язання наступних задач:

**Задача 1.** Складіть програму знаходження добутку двох найбільших з трьох введених з клавіатури чисел.

**Задача 2.** Дано два дійсних числа Х і У. Менше з них замінити їх напівсумою, а більше – їх подвоєним добутком. Результат вивести на екран монітора.

**VI. Підсумок уроку.**

Підведення підсумку уроку.