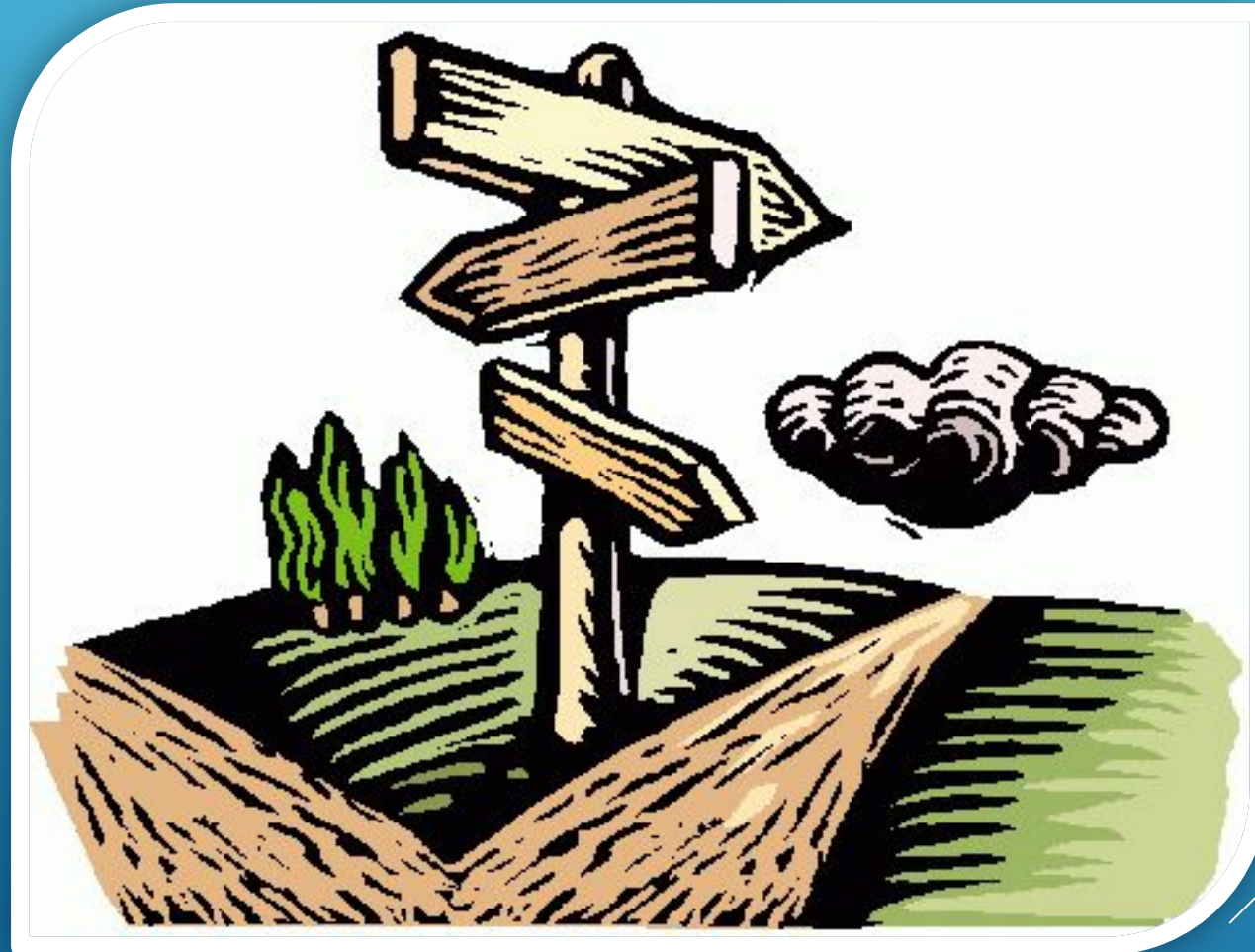


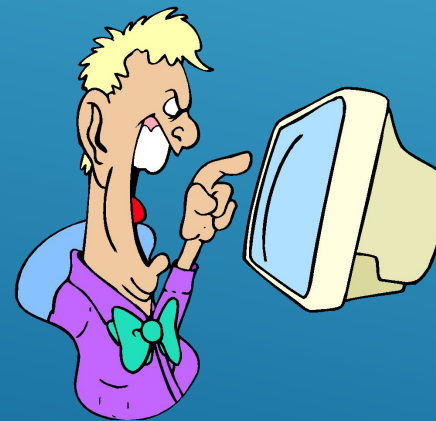
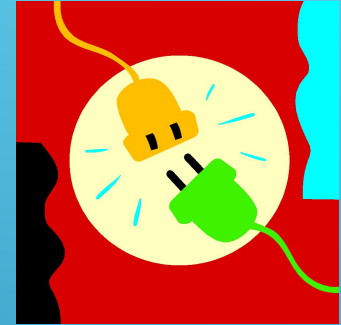
Сьогодні

16.11.2016

# Оператори розгалуження



# Повторимо правила поведінки та безпеки в комп'ютерному класі



# Пригадаємо

1. Який алгоритм ми називаємо лінійним?
2. Який алгоритм ми називаємо розгалуженим?
3. Які існують типи розгалуження?



# Оператори розгалуження

Повне розгалуження з одинарними командами

Тоді  
**if** <умова> **then** <ком.1> **else** <ком.2> ;  
Якщо Інакш  
е

---

A:=2

```
if A>5 then 'A більше 5'else 'A менше  
5'
```

# Оператори розгалуження

Не повне розгалуження з одинарними командами

**if** <умова> **then** <ком.1> ;

---

A:=2

if A>5 then 'A більше 5';

# Оператори розгалуження

Повне розгалуження з декількома командами

```
if <умова> then
  begin
    <ком.1>;
    <ком.N>;
  end
else
  begin
    <ком.1>;
    <ком.N>;
  end;
```

```
if (a+b)>=10 then
  begin
    c:=(a+b)*0,15;
    Label5.Caption:=FloatToStrF(c,ffFixed,
    5,1);
  end
else
  begin
    c:=(a+b)*0,02;
    Label5.Caption:=FloatToStrF(c,ffFixed,
    5,1);
  end;
```

A:=5; B:=4;

# Оператори розгалуження

Не повне розгалуження з декількома командами

```
if <умова> then
  begin
    <ком.1>;
    <ком.N>;
  end;
```

```
if (a+b)>=10 then
  begin
    A:=5; B:=4;
    c:=(a+b)*0,15;
    Label5.Caption:=FloatToStrF(c,ffFixed,
    5,1);
  end;
```

# Оператори розгалуження

## Перевірка числа на парність

The screenshot displays the Delphi 7 IDE interface. The central window shows a form titled "Оператори розгалуження" (Branching Operators) with the following UI elements:

- Title: "Перевірка числа на парність" (Evenness Check)
- Input field: "Введіть число: 0" (Enter number: 0)
- Button: "Перевірити" (Check)
- Output field: "Число **результат**" (Number **result**)

The Object TreeView on the left shows a form with components: Button1, Edit1, Label1, Label2, Label3, and Label4. The Object Inspector at the bottom left shows the properties of the selected component (TForm1).

The code editor on the right shows the following code snippet:

```
TForm1 = class(TForm)
```

A blue callout box with white text is overlaid on the code editor, containing the following instruction:

**Виводячи результат в label рекомендовано відформатовати як буде відобразатись результат і текст по завершенню стерти (залишити об'єкт label пустим)**



# Оператори розгалуження

## Перевірка числа на парність

The screenshot displays the Delphi 7 IDE. The central window, titled "Оператори розгалуження", shows a form titled "Перевірка числа на парність". The form contains a label "Введіть число:" followed by a text box containing the number "0". Below the text box is a button labeled "Перевірити". Underneath the button is a label "Число:". The left sidebar shows the Object TreeView with a tree structure for "Form1" containing "Button1", "Edit1", and four "Label" components. The Object Inspector at the bottom left shows the properties of the selected "Form1" object, including "Caption" set to "розгалуження" and "Color" set to "clSkyBlue". The code editor on the right shows the "implementation" section of a unit, with a placeholder comment "{SR \*.dfm}" and "end.".

**Складемо алгоритм виконання на дошці за допомогою блок-схеми за умови, що користувач буде вводити лише цілі числа**

```
implementation  
  
{SR *.dfm}  
  
end.
```

# Оператори розгалуження

## Перевірка числа на парність

```
var
  Form1: TForm1;
  a: integer;
implementation
  {$R *.dfm}

procedure TForm1.Button1Click(Sender
begin
  a:=StrToInt(Edit1.Text);
  if (a mod 2)>0 then
    Label4.Caption:='не парне'
  else
    Label4.Caption:='парне';
end;

end.
```

**Integer – лише для цілих чисел**

**StrToInt(Edit1.Text); –  
рядкові дані в ціле  
ЧИЛО**

**IntToStr(a); - Число А в  
рядковий тип**

# Оператори розгалуження

## Перевірка числа на парність

```
var
  Form1: TForm1;
  a:integer;
  b:string;
implementation
  {$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  a:=StrToInt(Edit1.Text);
  b:=IntToStr(a);
  if (a mod 2)>0 then
    Label4.Caption:=b+' не парне'
  else
    Label4.Caption:=b+' парне';
  end;
end.
```

**Вдосконалимо з  
ВИВОДОМ ЧИСЛА**

```
Form1.Button1Click(Sender: TObject);
Edit1.Text);
);
0 then
tion:=b+' не парне'
tion:=b+' парне';
```

Modified Insert Code/Diagram/

# Оператори розгалуження

## Нарахування заробітної плати

Скласти програму нарахування заробітної плати з наступними умовами:

- Має вводитись ім'я отримувача
- Має вводитись стаж та ставка (ціла ставка рівна 4750 грн)
- Якщо стажу немає то від величини ставки буде відніматись 20%
- Якщо стаж від 1 до 5 років від величини ставки буде відніматись 15%
- В усіх інших випадках буде відніматись від величини ставки 5%
- Результат виводу даних має бути таким – **«Робітник «Ім'я» має отримати «Сума» грн»**

# Оператори розгалуження

## Нарахування заробітної плати

The screenshot displays the Delphi 7 IDE with a project titled "Delphi 7 - Project1". The main window shows a form titled "Нарахування заробітної плати" (Payroll Calculation). The form contains three input fields: "Введіть ім'я робітника" (Enter employee name), "Введіть стаж" (Enter experience) with a value of 0, and "Введіть ставку" (Enter rate) with a value of 0. A "Підрахувати" (Calculate) button is positioned to the right of the input fields. Below the inputs, a red text label reads "Тут буде відображатись результат" (The result will be displayed here).

The Object TreeView on the left shows the form structure with components: Form1, Button1, Edit1, Edit2, Edit3, Label1, Label2, Label3, and Label4. The Object Inspector at the bottom left shows the properties of the selected Form1, including its caption "Нарахування" and client width of 469.

The code editor on the right shows the Unit1.pas file with the following code:

```
unit Unit1;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;  
  
type  
  TForm1 = class(TForm)  
    Label1: TLabel;  
    Edit1: TEdit;  
    Label2: TLabel;  
    Edit2: TEdit;  
    Label3: TLabel;  
    Edit3: TEdit;  
    Button1: TButton;  
    Label4: TLabel;  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form1: TForm1;  
  
implementation
```

# Оператори розгалуження

## Нарахування заробітної плати

```
var
  Form1: TForm1;
  namerob, fintsen: string;
  stazh: integer;
  stavka, tsena: real;
const
  odst=4750;
implementation
{$R *.dfm}
```

# Оператори розгалуження

## Нарахування заробітної плати

```
procedure TForm1.Button1Click(Sender: TObject);
begin
namerob:=Edit1.Text;
stazh:=StrToInt(Edit2.Text);
stavka:=StrToFloat(Edit3.Text);
if stazh<1 then
begin
    tsena:=(stavka*odst)-(stavka*odst)*0.2;
    fintsen:=FloatToStrF(tsena,ffFixed,10,2);
    label4.Caption:='Робітник '+namerob+' має отримати '+fintsen+ ' грн';
end
end
```

# Оператори розгалуження

## Нарахування заробітної плати

```
procedure TForm1.Button1Click(Sender: TObject);
begin
namerob:=Edit1.Text;
stazh:=StrToInt(Edit2.Text);
stavka:=StrToFloat(Edit3.Text);
if stazh<1 then
begin
tsena:=(stavka*odst)-(stavka*odst)*0.2;
fintsena:=FloatToStrF(tsena,ffFixed,10,2);
label4.Caption:='Робітник '+namerob+' має отримати '+fintsena+' грн';
end
else if stazh<=5 then
begin
tsena:=(stavka*odst)-(stavka*odst)*0.15;
fintsena:=FloatToStrF(tsena,ffFixed,10,2);
label4.Caption:='Робітник '+namerob+' має отримати '+fintsena+' грн';
end
end
```



# Оператори розгалуження

## Нарахування заробітної плати

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  namerob:=Edit1.Text;
  stazh:=StrToInt(Edit2.Text);
  stavka:=StrToFloat(Edit3.Text);
  if stazh<1 then
  begin
    tsena:=(stavka*odst)-(stavka*odst)*0.2;
    fintsen:=FloatToStrF(tsena,ffFixed,10,2);
    label4.Caption:='Робітник '+namerob+' має отримати '+fintsen+ ' грн';
  end
  else if stazh<=5 then
  begin
    tsena:=(stavka*odst)-(stavka*odst)*0.15;
    fintsen:=FloatToStrF(tsena,ffFixed,10,2);
    label4.Caption:='Робітник '+namerob+' має отримати '+fintsen+ ' грн';
  end
  else
  begin
    tsena:=(stavka*odst)-(stavka*odst)*0.05;
    fintsen:=FloatToStrF(tsena,ffFixed,10,2);
    label4.Caption:='Робітник '+namerob+' має отримати '+fintsen+ ' грн';
  end;
end;
```

# Оператори розгалуження

## Нарахування заробітної плати

```
var
  Form1: TForm1;
  namerob, fintsenas: string;
  stazh: integer;
  stavka, tsena: real;
const
  odst=4750;
```

```
namerob:=Edit1.Text;
stazh:=StrToInt(Edit2.Text);
stavka:=StrToFloat(Edit3.Text);
if stazh<1 then
begin
  tsena:=(stavka*odst)-(stavka*odst)*0.2;
  fintsenas:=FloatToStrF(tsena,ffFixed,10,2);
  label4.Caption:='Робітник '+namerob+' має отримати '+fintsenas+ ' грн';
end
else if stazh<=5 then
begin
  tsena:=(stavka*odst)-(stavka*odst)*0.15;
  fintsenas:=FloatToStrF(tsena,ffFixed,10,2);
  label4.Caption:='Робітник '+namerob+' має отримати '+fintsenas+ ' грн';
end
else
begin
  tsena:=(stavka*odst)-(stavka*odst)*0.05;
  fintsenas:=FloatToStrF(tsena,ffFixed,10,2);
  label4.Caption:='Робітник '+namerob+' має отримати '+fintsenas+ ' грн';
end;
```

**Перепишіть  
програмний  
КОД В ЗОШИТ**

# Оператори розгалуження

Демонстрування результату

Нарахування заробітної плати

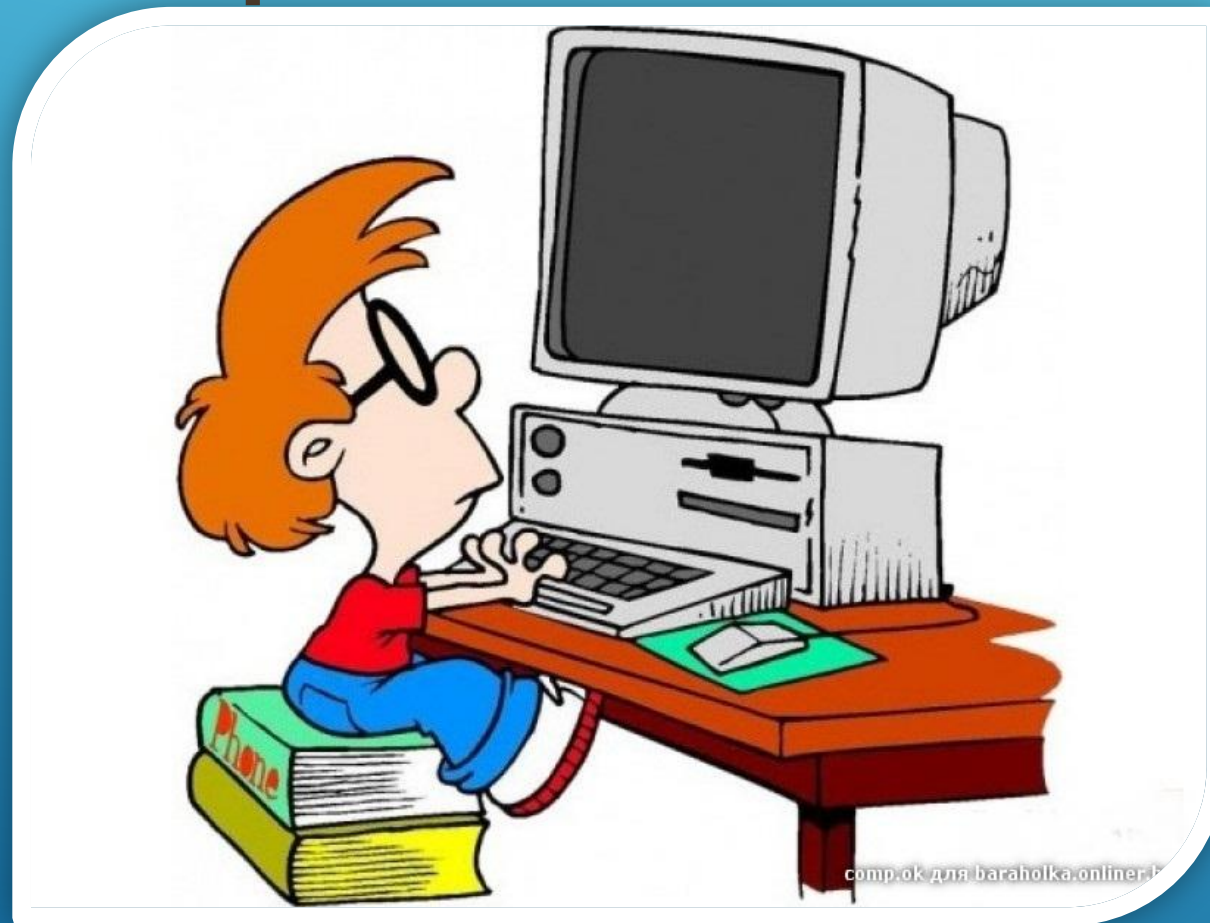
Введіть ім'я робітника

Введіть стаж

Введіть ставку

Підрахувати

# Працюємо за комп'ютером



# Працюємо за комп'ютером

## Нарахування заробітної плати

Скласти програму нарахування заробітної плати з наступними умовами:

- Має вводитись ім'я та прізвище отримувача в окремих полях
- Має вводитись стаж та ставка (ціла ставка рівна 5230 грн)
- Якщо стажу немає то від величини ставки буде відніматись 21,5%
- Якщо стаж від 1 до 7 років від величини ставки буде відніматись 17%
- В усіх інших випадках буде відніматись від величини ставки 7%
- Результат виводу даних має бути таким – « **«Прізвище» «Ім'я» має отримати «Сума» гривень»**

# Повторюємо

1. Які існують типи розгалуження?
2. Яка структура умови, яка містить підумови?
3. Як перевести ціле число в рядковий тип?



# ДОМАШНЄ ЗАВДАННЯ

Опрацювати матеріал  
вивчений на уроці



До нових  
зустрічей!

