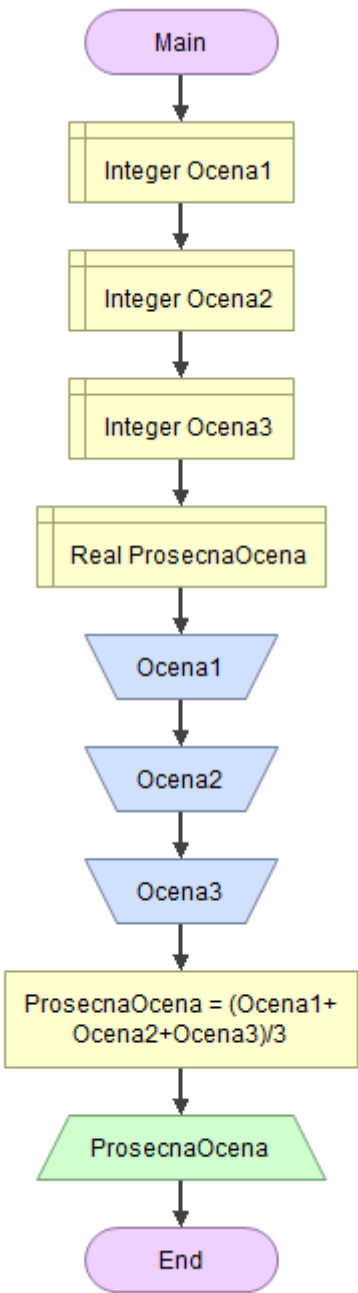
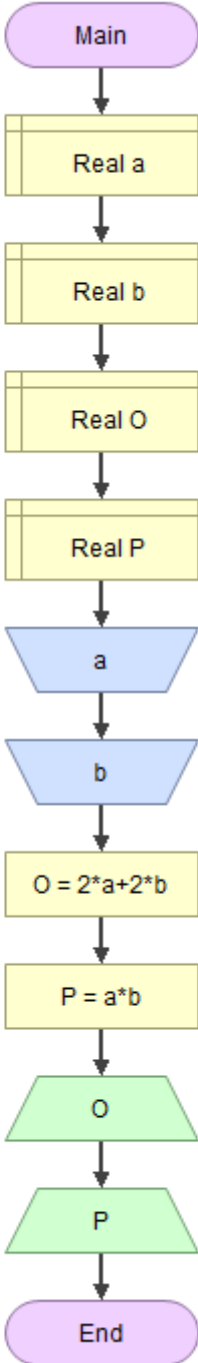


## Задаци:

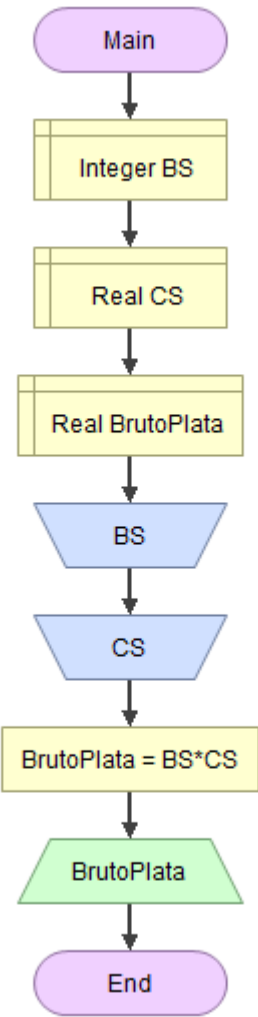
1. На основу три оцене израчунати просечну оцену из информатике.

| Задатак бр.1  | Задатак бр.1 - Pascal   |
|---|---|
|  <pre>graph TD; Start([Main]) --&gt; Decl1[Integer Ocena1]; Decl1 --&gt; Decl2[Integer Ocena2]; Decl2 --&gt; Decl3[Integer Ocena3]; Decl3 --&gt; Decl4[Real ProsečnaOcena]; Decl4 --&gt; Read1[/Ocena1/]; Read1 --&gt; Read2[/Ocena2/]; Read2 --&gt; Read3[/Ocena3/]; Read3 --&gt; Calc[ProsečnaOcena = (Ocena1 + Ocena2 + Ocena3) / 3]; Calc --&gt; Print[/ProsečnaOcena/]; Print --&gt; End([End]);</pre> <p>The flowchart illustrates the program's execution flow. It begins with a 'Main' terminal, followed by the declaration of three integer variables: 'Ocena1', 'Ocena2', and 'Ocena3'. Next, a real variable 'ProsečnaOcena' is declared. The program then prompts the user to input three grades, labeled 'Ocena1', 'Ocena2', and 'Ocena3'. These values are used in a calculation: <math>ProsečnaOcena = (Ocena1 + Ocena2 + Ocena3) / 3</math>. The result is then displayed, and the program ends at the 'End' terminal.</p> | <pre>program MyProgram; uses Math, SysUtils;  { Main } var   Ocena1 : integer;   Ocena2 : integer;   Ocena3 : integer;   ProsečnaOcena : real;  begin   ReadLn(Ocena1);   ReadLn(Ocena2);   ReadLn(Ocena3);   ProsečnaOcena := (Ocena1 + Ocena2 + Ocena3) / 3;   WriteLn(ProsečnaOcena); end.</pre> |

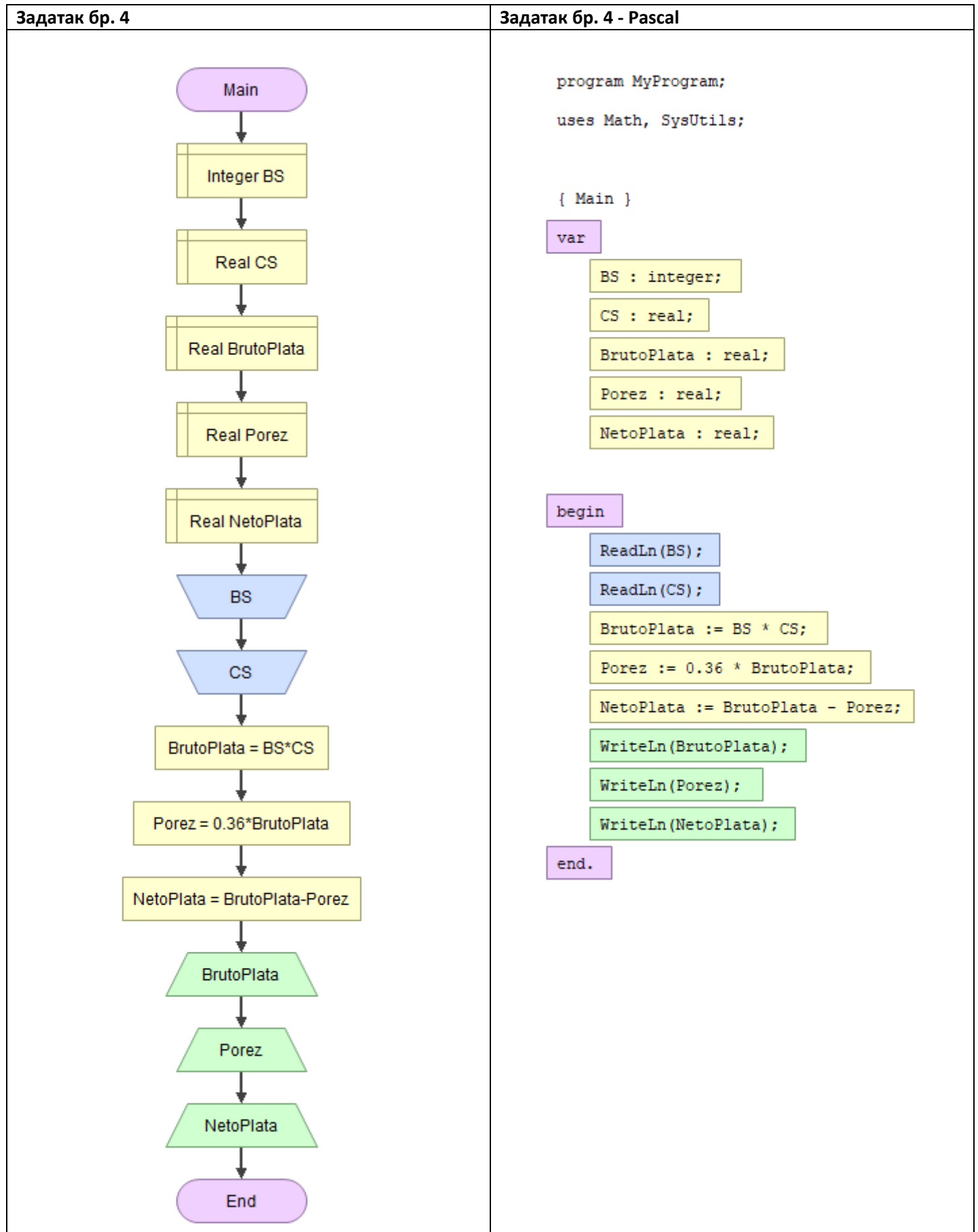
2. На основу страница правоугоника израчунати обим и површину правоугоника.

| Задатак бр.2  | Задатак бр.2 - Pascal   |
|---|---|
|  <pre>graph TD; Main([Main]) --&gt; Real_a[Real a]; Real_a --&gt; Real_b[Real b]; Real_b --&gt; Real_O[Real O]; Real_O --&gt; Real_P[Real P]; Real_P --&gt; Input_a[/a/]; Input_a --&gt; Input_b[/b/]; Input_b --&gt; Calc_O["O = 2*a+2*b"]; Calc_O --&gt; Calc_P["P = a*b"]; Calc_P --&gt; Output_O[/O/]; Output_O --&gt; Output_P[/P/]; Output_P --&gt; End([End]);</pre> <p>The flowchart illustrates the algorithm for calculating the perimeter and area of a rectangle. It starts with a 'Main' terminal, followed by declarations for real variables 'a', 'b', 'O', and 'P'. The process steps are: reading 'a', reading 'b', calculating 'O = 2*a + 2*b', calculating 'P = a*b', and finally outputting 'O' and 'P' before reaching the 'End' terminal.</p> | <pre>program MyProgram;  uses Math, SysUtils;  { Main } var   a : real;   b : real;   O : real;   P : real;  begin   ReadLn(a);   ReadLn(b);   O := 2 * a + 2 * b;   P := a * b;   WriteLn(O);   WriteLn(P); end.</pre> <p>The Pascal code implements the same logic as the flowchart. It declares real variables 'a', 'b', 'O', and 'P'. The 'begin' block contains the input for 'a' and 'b', the calculation of 'O' as <math>2 * a + 2 * b</math>, the calculation of 'P' as <math>a * b</math>, and the output of 'O' and 'P' using 'WriteLn'. The program ends with 'end.'</p> |

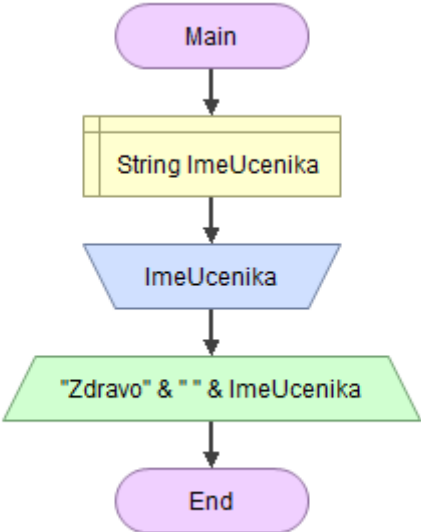
3. На основу броја радних сат и цене радног сата израчунати бруто плату радника.

| Задатак бр. 3   | Задатак бр. 3 - Pascal   |
|---|--|
|  <pre>graph TD; Main([Main]) --&gt; BS[Integer BS]; BS --&gt; CS[Real CS]; CS --&gt; BrutoPlata[Real BrutoPlata]; BrutoPlata --&gt; BS_in[BS]; BS_in --&gt; CS_in[CS]; CS_in --&gt; Calc[BrutoPlata = BS * CS]; Calc --&gt; BrutoPlata_out[BrutoPlata]; BrutoPlata_out --&gt; End([End]);</pre> <p>The flowchart illustrates the algorithm for calculating gross pay. It starts with a 'Main' terminal, followed by declaring 'Integer BS' and 'Real CS'. Then, 'Real BrutoPlata' is declared. The process steps are: reading 'BS', reading 'CS', calculating 'BrutoPlata = BS * CS', and finally outputting 'BrutoPlata' before reaching the 'End' terminal.</p> | <pre>program MyProgram; uses Math, SysUtils;  { Main } var   BS : integer;   CS : real;   BrutoPlata : real;  begin   ReadLn(BS);   ReadLn(CS);   BrutoPlata := BS * CS;   WriteLn(BrutoPlata); end.</pre> |

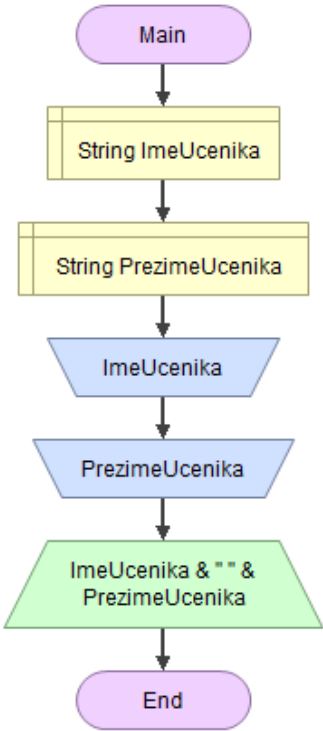
4. На основу броја радних сат и цене радног сата израчунати бруто плату, порез и нето плату радника. Порез је 36% од бруто плате.



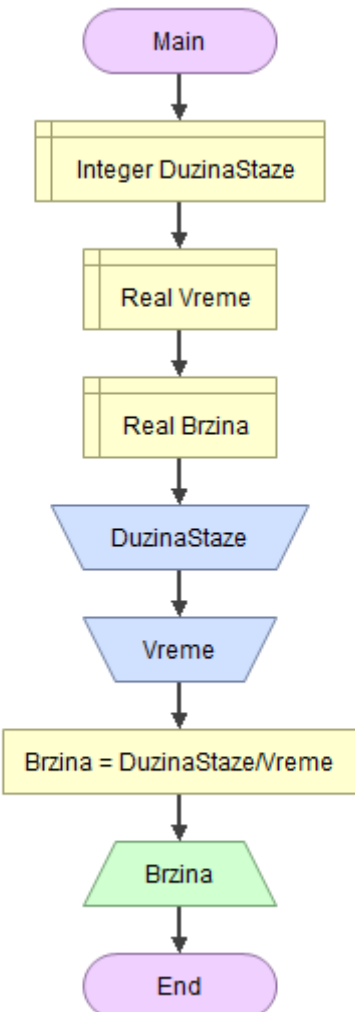
5. На основу унетог имена програм треба да штампа поздрав **Здраво + унето име**.

| Задатак бр. 5  | Задатак бр. 5 - Pascal  |
|--|---|
|  <pre>graph TD; Main([Main]) --&gt; String[String ImeUcenika]; String --&gt; ImeUcenika[/ImeUcenika/]; ImeUcenika --&gt; Output[/"Zdravo" &amp; " " &amp; ImeUcenika/]; Output --&gt; End([End]);</pre> <p>The flowchart illustrates the logic of the program. It starts with a 'Main' process, followed by a 'String ImeUcenika' process, then an 'ImeUcenika' process, then an output process displaying '"Zdravo" &amp; " " &amp; ImeUcenika', and finally an 'End' process.</p> | <pre>program MyProgram; uses Math, SysUtils;  { Main } var   ImeUcenika : string;  begin   ReadLn(ImeUcenika);   WriteLn('Zdravo' + ' ' + ImeUcenika); end.</pre> |

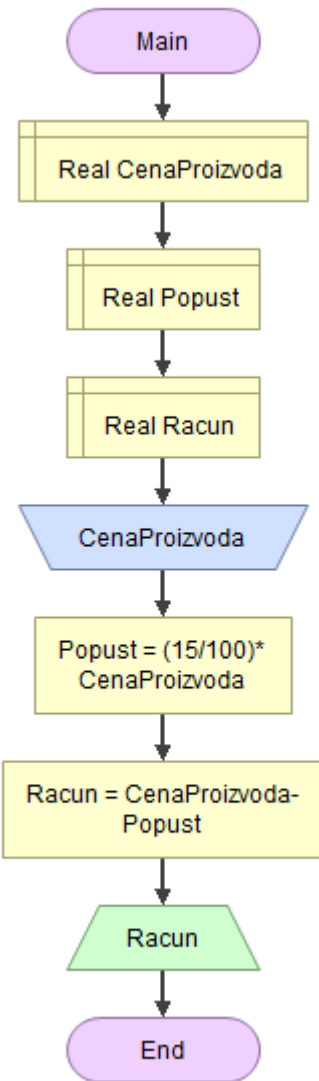
6. На основу унетог имена и презимена, одвојено, програм треба да штампа **унето име + унето презиме**.

| Задатак бр. 6   | Задатак бр. 6 - Pascal  |
|---|---|
|  <pre>graph TD; Main([Main]) --&gt; ImeUcenikaString[String ImeUcenika]; ImeUcenikaString --&gt; PrezimeUcenikaString[String PrezimeUcenika]; PrezimeUcenikaString --&gt; ImeUcenikaInput[/ImeUcenika/]; ImeUcenikaInput --&gt; PrezimeUcenikaInput[/PrezimeUcenika/]; PrezimeUcenikaInput --&gt; Output[/ImeUcenika &amp; "" &amp; PrezimeUcenika/]; Output --&gt; End([End]);</pre> | <pre>program MyProgram;  uses Math, SysUtils;  { Main }  var   ImeUcenika : string;   PrezimeUcenika : string;  begin   ReadLn(ImeUcenika);   ReadLn(PrezimeUcenika);   WriteLn(ImeUcenika + ' ' + PrezimeUcenika);  end.</pre> |

7. На основу основу дужине претрчане стазе (у метрима) и постигнутог времена (у секундама) треба израчунати просечну брзину ученика .

| Задатак бр. 7   | Задатак бр. 7 - Pascal   |
|---|--|
|  <pre>graph TD; Main([Main]) --&gt; DuzinaStaze[Integer DuzinaStaze]; DuzinaStaze --&gt; Vreme[Real Vreme]; Vreme --&gt; Brzina[Real Brzina]; Brzina --&gt; DuzinaStaze_in[/DuzinaStaze/]; DuzinaStaze_in --&gt; Vreme_in[/Vreme/]; Vreme_in --&gt; Calc[Brzina = DuzinaStaze/Vreme]; Calc --&gt; Brzina_out[/Brzina/]; Brzina_out --&gt; End([End]);</pre> | <pre>program MyProgram; uses Math, SysUtils;  { Main } var   DuzinaStaze : integer;   Vreme : real;   Brzina : real;  begin   ReadLn(DuzinaStaze);   ReadLn(Vreme);   Brzina := DuzinaStaze / Vreme;   WriteLn(Brzina); end.</pre> |

8. На основу цене производа, израчунати koliko треба платити тај производ, ако се приликом плаћања готовином добија попуст од 15%.

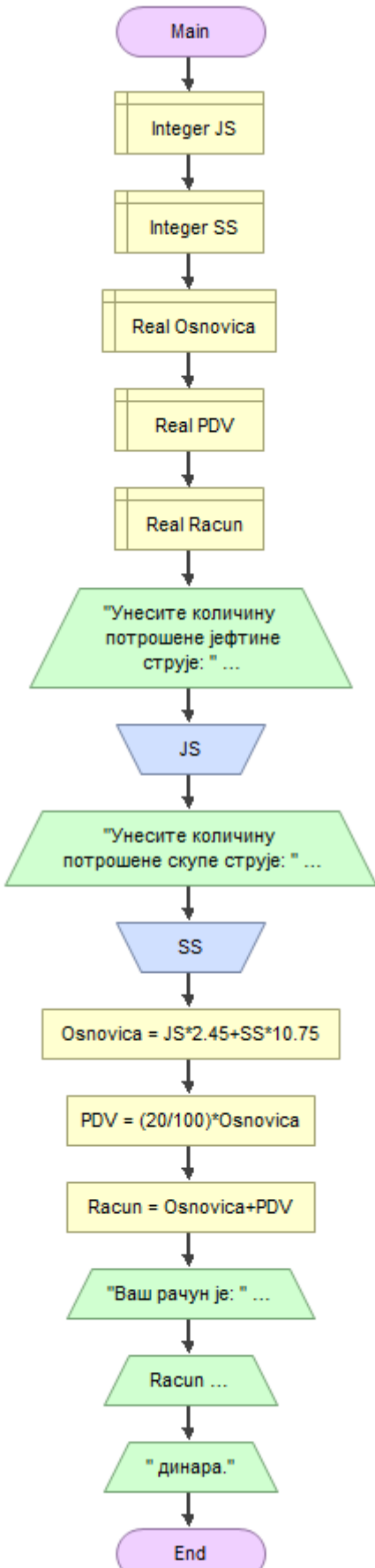
| Задатак бр. 8  | Задатак бр. 8 - Pascal  |
|--|---|
|  <pre>graph TD; Main([Main]) --&gt; CenaProizvoda[Real CenaProizvoda]; CenaProizvoda --&gt; Popust[Real Popust]; Popust --&gt; Racun[Real Racun]; Racun --&gt; CenaProizvodaInput[/CenaProizvoda/]; CenaProizvodaInput --&gt; PopustCalc[Popust = (15/100)* CenaProizvoda]; PopustCalc --&gt; RacunCalc[Racun = CenaProizvoda - Popust]; RacunCalc --&gt; RacunOutput[/Racun/]; RacunOutput --&gt; End([End]);</pre> <p>The flowchart illustrates the logic of the program. It starts with a 'Main' terminal, followed by three variable declarations: 'Real CenaProizvoda', 'Real Popust', and 'Real Racun'. An input operation 'CenaProizvoda' is performed. The discount 'Popust' is calculated as <math>(15/100) * \text{CenaProizvoda}</math>. The final amount 'Racun' is calculated as <math>\text{CenaProizvoda} - \text{Popust}</math>. The result 'Racun' is then outputted, and the program ends.</p> | <pre>program MyProgram; uses Math, SysUtils;  { Main } var   CenaProizvoda : real;   Popust : real;   Racun : real;  begin   ReadLn(CenaProizvoda);   Popust := 15 / 100 * CenaProizvoda;   Racun := CenaProizvoda - Popust;   WriteLn(Racun); end.</pre> |



9. На основу броја потрошених киловата јефтине и скупе струје израчунати основицу за палћање, ПДВ и укупан износ рачуна ако је киловат јефтине струје 2,45 динара и киловат скупе струје 10,75 динара. ПДВ рачунати по стопи од 20%.

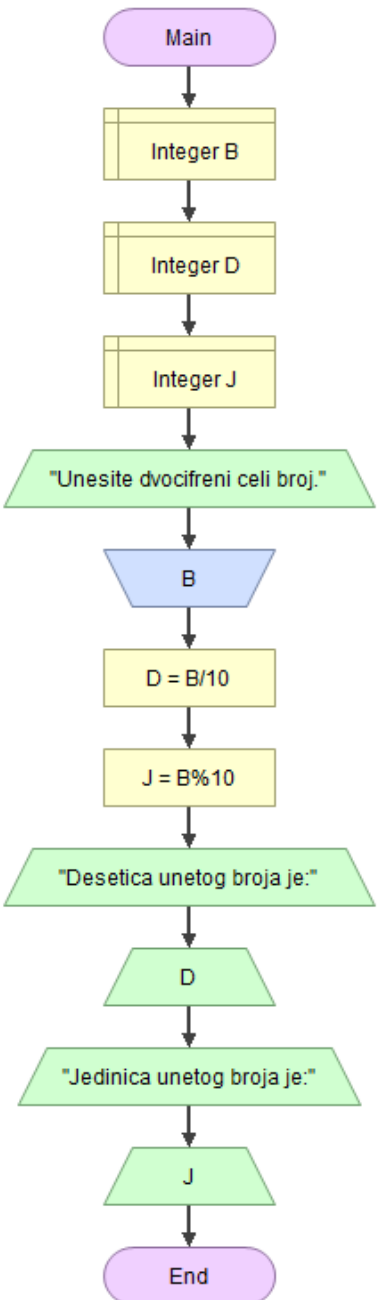
| Задатак бр. 9   | Задатак бр. 9 - Pascal   |
|---|--|
| <pre> graph TD     Main([Main]) --&gt; JS[Integer JS]     JS --&gt; SS[Integer SS]     SS --&gt; Osnovica[Real Osnovica]     Osnovica --&gt; PDV[Real PDV]     PDV --&gt; Racun[Real Racun]     Racun --&gt; JS_in[/JS/]     JS_in --&gt; SS_in[/SS/]     SS_in --&gt; Calc1[Osnovica = JS*2.45+SS*10.75]     Calc1 --&gt; Calc2[PDV = (20/100)*Osnovica]     Calc2 --&gt; Calc3[Racun = Osnovica+PDV]     Calc3 --&gt; Racun_out[/Racun/]     Racun_out --&gt; End([End])     </pre> | <pre> program MyProgram; uses Math, SysUtils;  { Main } var     JS : integer;     SS : integer;     Osnovica : real;     PDV : real;     Racun : real;  begin     ReadLn(JS);     ReadLn(SS);     Osnovica := JS * 2.45 + SS * 10.75;     PDV := 20 / 100 * Osnovica;     Racun := Osnovica + PDV;     WriteLn(Racun); end.     </pre> |

9а. На основу броја потрошених киловата јефтине и скупе струје израчунати основицу за палћање, ПДВ и укупан износ рачуна ако је киловат јефтине струје 2,45 динара и киловат скупе струје 10,75 динара. ПДВ рачунати по стопи од 20%. При уносу количине потрошене струје и при штампању рачуна треба штампати и одговарајуће текстуалне поруке.

| Задатак бр. 9а  | Задатак бр. 9а - Pascal   |
|---|---|
|  <pre> graph TD     Main([Main]) --&gt; JS[Integer JS]     JS --&gt; SS[Integer SS]     SS --&gt; Osnovica[Real Osnovica]     Osnovica --&gt; PDV[Real PDV]     PDV --&gt; Racun[Real Racun]     Racun --&gt; Input1[/Унесите количину потрошене јефтине струје: " .../]     Input1 --&gt; JS_val[/JS/]     JS_val --&gt; Input2[/Унесите количину потрошене скупе струје: " .../]     Input2 --&gt; SS_val[/SS/]     SS_val --&gt; Osnovica_calc[Osnovica = JS*2.45+SS*10.75]     Osnovica_calc --&gt; PDV_calc[PDV = (20/100)*Osnovica]     PDV_calc --&gt; Racun_calc[Racun = Osnovica+PDV]     Racun_calc --&gt; Output1[/Ваш рачун је: " .../]     Output1 --&gt; Racun_print[/Racun .../]     Racun_print --&gt; Dinara[/" динара."/]     Dinara --&gt; End([End])     </pre> | <pre> program MyProgram;  uses Math, SysUtils;  { Main } var     JS : integer;     SS : integer;     Osnovica : real;     PDV : real;     Racun : real;  begin     Write('Унесите количину потрошене јефтине струје: ');     ReadLn(JS);     Write('Унесите количину потрошене скупе струје: ');     ReadLn(SS);     Osnovica := JS * 2.45 + SS * 10.75;     PDV := 20 / 100 * Osnovica;     Racun := Osnovica + PDV;     Write('Ваш рачун је: ');     Write(Racun);     WriteLn(' динара. ');  end.     </pre> |

10. Nacrtať algoritam i napisati program (NA i NP) koji će uneti dvocifreni celi broj da razdvoji na deseticu i jedinicu.

B-dvocifreni broj; D-desetica; J-jedinica

| Задатак бр. 10   | Задатак бр. 10 - Pascal   |
|--|---|
|  <pre>graph TD     Start([Main]) --&gt; B[Integer B]     B --&gt; D[Integer D]     D --&gt; J[Integer J]     J --&gt; Input[/"Unesite dvocifreni celi broj."/]     Input --&gt; ReadB[/B/]     ReadB --&gt; CalcD[D = B/10]     CalcD --&gt; CalcJ[J = B%10]     CalcJ --&gt; OutputD[/"Desetica unetog broja je:"/]     OutputD --&gt; PrintD[/D/]     PrintD --&gt; OutputJ[/"Jedinica unetog broja je:"/]     OutputJ --&gt; PrintJ[/J/]     PrintJ --&gt; End([End])</pre> | <pre>program MyProgram; uses Math, SysUtils;  { Main } var   B : integer;   D : integer;   J : integer;  begin   WriteLn('Unesite dvocifreni celi broj.');</pre> <pre>  ReadLn(B);   D := B / 10;   J := B mod 10;   WriteLn('Desetica unetog broja je:');   WriteLn(D);   WriteLn('Jedinica unetog broja je:');   WriteLn(J); end.</pre> |

**11.** Nacrtati algoritam i napisati program (**NA i NP**) koji će za unetu cenu proizvoda (**CP-cena proizvoda, celi broj**) da izračuna potreban broj novčanica za plaćanje računa (**BN5000**-broj novčanica od 5000 dinara; **BN2000**-broj novčanica od 2000 dinara; **BN1000**-broj novčanica od 1000 dinara; **BN500**-broj novčanica od 500 dinara; **BN200**-broj novčanica od 200 dinara; **BN100**-broj novčanica od 100 dinara; **BN50**-broj novčanica od 50 dinara; **BN20**-broj novčanica od 20 dinara; **BN10**-broj novčanica od 10 dinara; **BN5**-broj novčanica od 5 dinara; **BN1**-broj novčanica od 1 dinara).