

Leksion 4

Loops

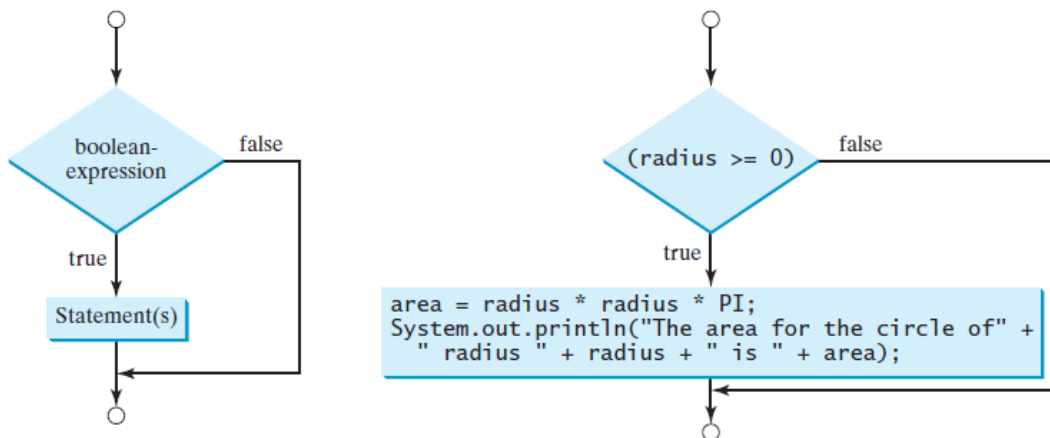
Përbajtja

Kushti if

Nje kusht if eshte nje instruksion qe i mundeson nje programi te percaktojë rrugët alternative të ekzekutimit.

Sintaksa e pergjithshme e tij eshte:

```
if (boolean-expression) {  
    statement(s);  
}
```



Kushti if-then është më themelore e të gjitha kushte të kontrollit. Ajo i tregon programin tuaj për të ekzekutuar një seksion të caktuar të kodit vetëm nëse një test i caktuar e vlerëson të vërtetë. Për shembull, klasa Biciklete mund t'i lejojë frenat të ulin shpejtësinë e biçikletës vetëm nëse biçikleta është në lëvizje. Një zbatim i mundshëm i metodës applyBrakes mund të jetë si më poshtë

```
void applyBrakes() {  
    // Kushti "if": bicikleta duhet te jete ne levizje  
    if (isMoving){  
        // kushte atehere (then): Ulni shpejtesine  
        currentSpeed--;  
    }  
}
```

Nëse ky test e vlerëson të gabuar (që do të thotë se biçikleta nuk është në lëvizje), kontrolli shkon në fund të deklaratës if-then.

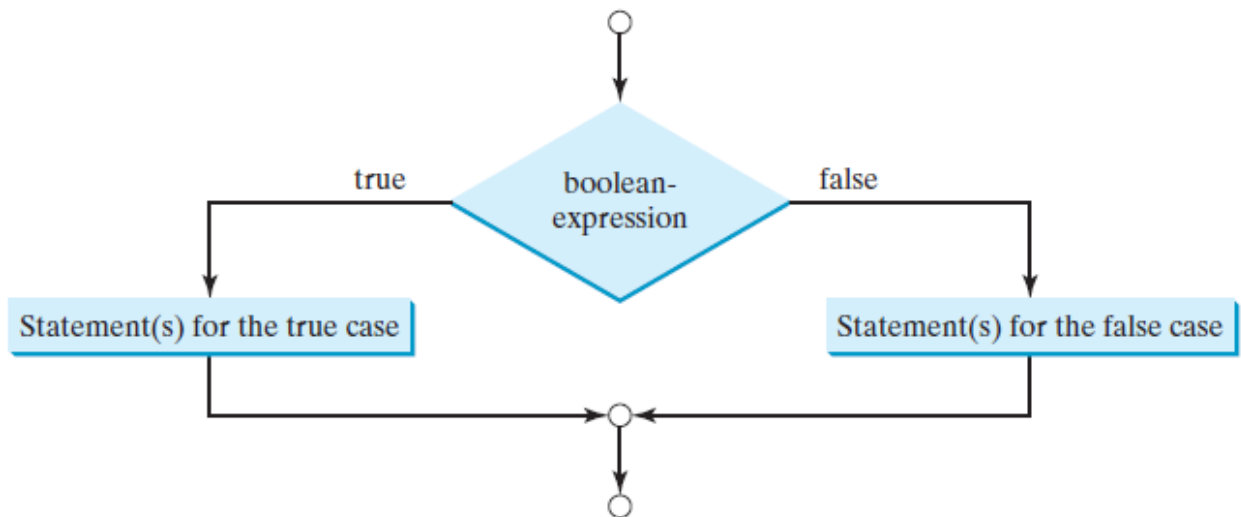
The if-then-else Statement

Deklarimi `if-then-else` na jep një rrugë të mesme të ekzekutimit, kur një kusht "`if`" e vlerëson të gabuar.

Sintaksa e `if-then-else` është si mëposhte:

```
if (boolean-expression) {  
    statement(s)-for-the-true-case;  
}  
else {  
    statement(s)-for-the-false-case;  
}
```

Grafikisht:



Nëse shprehja `boolean` vlerëson të vërtetë (`True`), deklarimet për rastin e vërtetë ekzekutohet; përndryshe, deklarimet për rastin e gabuar (`False`) ekzekutohen. Për shembull, merrni parasysh kodin e mëposhtëm:

```
if (rreze >= 0) {  
    siperfaqe = rreze * rreze * 3.1459;  
    System.out.println("Siperfaqja e rrethit me rreze " +  
        rreze + " është " + siperfaqe);  
}  
else {  
    System.out.println("Input negative");  
}
```

Ju mund të përdorni një deklaratë `if-then-else` në metodën `applyBrakes` për të marrë ndonjë veprim nëse frenat zbatohen kur biçikleta nuk është në lëvizje. Në këtë rast, veprimi është thjesht të shtypni një mesazh gabimi ku thuhet se biçikleta është ndalur.

```
void applyBrakes() {
```

```

    if (isMoving) {
        currentSpeed--;
    } else {
        System.err.println("Bicikleta eshte e ndaluar!");
    }
}

```

Shembulli i mëposhtëm, IfElseDemo, cakton një notë bazuar në vlerën e një rezultati testimi: një A për një rezultat prej 90% ose më lart, një B për një rezultat prej 80% ose më lart, e kështu me radhë.

```

class IfElseDemo {
    public static void main(String[] args) {

        int testscore = 76;
        char grade;

        if (testscore >= 90) {
            grade = 'A';
        } else if (testscore >= 80) {
            grade = 'B';
        } else if (testscore >= 70) {
            grade = 'C';
        } else if (testscore >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        System.out.println("Grade = " + grade);
    }
}

```

Output i kodit do te jete:

```
Grade = C
```

Shembull:

```

import java.util.Scanner;

public class SubtractionQuiz {
    public static void main(String[] args) {
        // 1. Gjenero 2 numra te rastesishem njeshifror
        int numri1 = (int)(Math.random() * 10);
        int numri2 = (int)(Math.random() * 10);
    }
}

```

```

// 2. Nese numri1 < numri2, ndrysho numri1 me numri2

if (numri1 < numri2) {

    int temp = numri1;

    numri1 = numri2;
    numri2 = temp;
}

// 3. Pyesni perdoruesin te pergjigjet "Cili eshte
rezultati i number1 - number2?"
System.out.print
    ("Sa bejne " + numri1 + " - " + numri2 + "? ");
Scanner input = new Scanner(System.in);
int answer = input.nextInt();

// 4. Grade the answer and display the result
if (numri1 - numri2 == answer)
    System.out.println("Ju jeni te sakte!");
else {
    System.out.println("Pergjigjia juaj eshte e gabuar.");
    System.out.println(numri1 + " - " + numri2 +
        " bejne " + (numri1 - numri2));
}
}
}

```

Instruksionet switch, break, continue

Java na jep një instruksion Switch për të trajtuar kushte të shumëfishta efikasiteti. Ky instruksion kontrollon për të parë nëse statusi përputhet me vlerat 0, 1, 2, 3. Komanda switch i kontrollon të gjitha opsionet e mundshme të krijuara më parë nga vete ne, dhe më pas ekzekuton intruksionet që ndodhen mbrenda kllapave gjarpëruese të opsionit që rezulton të jetë i vete të (pra për të cilin plotësohet kushti).

Ndryshe nga kushtet if-else dhe if-then-else, instruksioni switch mund të ketë disa alternativa të mundshme të ekzekutimit. Një switch punon me llojet e të dhënave me byte, short, char dhe int Gjithashtu punon me klasën String dhe disa klasa të veçanta që mbyllin lloje të caktuara primitive: Character, Byte, Short dhe Integer

Më poshtë është sintaksa e plotë e komandës switch:

```

switch (kushti) {
    case vlera1: intruksioni(s)1;
        break;
    case vlera2: intruksioni(s)2;
        break;
    ...
    Case vleraN: intruksioni(s)N;
        break;
    default: intruksioni(s)-per-default;
}

```

Komanda switch duhet të ndjek rregullat e mëposhtme:

- Shprehja Switch duhet të japë një vlerë të tipit **char**, **byte**, **short**, apo **int** dhe duhet të jetë gjithmonë e mbyllur në kllapa.
- Vlerat **vlera1**, dhe **vleraN** duhet të kenë të njejtin tip të dhëne (data type) njësoj si vlera e shprehjes **switch**. Vini re se **vlera1** dhe **vleraN**, janë konstante, duke nënkuptuar kështu që ato nuk mund të mbajnë vlera të tjera (nuk janë variabla).
- Kur vlera e njëres prej rasteve të parashikuara (deklaratat) përputhet me vlerën e shprehjes Switch, instruksionet e këtij rasti (ato që ndodhen brenda kllapave gjarpëruese përkatëse) fillojnë të ekzekutohen deri sa të ndeshemi në një deklaratë pushimi (break) apo të arrijmë në fund të deklaratave të shprehjes **Switch**.
- Fjala kyce **break**, është opsionale (jo e detyrueshme). Shprehja break, e ndalon në mënyre të menjëhershme ekzekutimin e mëtejshëm të kodit.
- Rasti i shënuar si **case-default**, i cili është opsional, përdoret për ti treguar programit se çfarë duhet të ekzekutojë nëse asnjë nga të gjithë rastet e mësipërme të parashikuara nuk përputhet.
- Rastet kontrollohen nga programi sipas radhës që ato janë renditur nga programuesi, por radha e rasteve nuk ka rëndësi (përfshirë edhe rastin default). Megjithatë është një praktikë e mirë programimi, që të ndiqet një arsyetim në krijimin e tyre dhe rasti default të lihet në fund.

Kujdes! Mos harroni të përdorni **break** sa herë që është e nevojshme. Kur njëra nga opsionet përputhet, instruksionet brenda saj fillojnë të ekzekutohen derisa të arrijë tek një komandë **break**, ose tek fundi i kodit.

Shembulli i mëposhtëm i kodit, SwitchDemo, deklaron një muaj të quajtur int, vlera e të cilit përfaqëson një muaj. Kodi tregon emrin e muajit, bazuar në vlerën e muajit, duke përdorur instruksionin switch.

Shembull SwitchDemo

```
public class SwitchDemo {
    public static void main(String[] args) {

        int month = 8;
        String monthString;
        switch (month) {
            case 1: monthString = "Janar";
                    break;
            case 2: monthString = "Shkurt";
                    break;
            case 3: monthString = "Mars";
                    break;
            case 4: monthString = "Prill";
                    break;
            case 5: monthString = "Maj";
                    break;
            case 6: monthString = "Qershor";
                    break;
            case 7: monthString = "Korrik";
                    break;
            case 8: monthString = "Gusht";
                    break;
            case 9: monthString = "Shtator";
                    break;
            case 10: monthString = "Tetor";
                    break;
            case 11: monthString = "Nentor";
                    break;
            case 12: monthString = "Dhjetor";
                    break;
            default: monthString = "Muaj jo i vlefshem";
                    break;
        }
        System.out.println(monthString);
    }
}
```



```
}
```

Output:

Gusht

Shembull Switch

```
public class ShembullSwitch {  
    public static void main(String[] args) {  
        for (int i=0; i <= 3 ; i++)  
        {  
            switch(i)  
            {  
                case 0:  
                    System.out.println("i eshte 0");  
                    break;  
  
                case 1:  
                    System.out.println("i eshte 1");  
                    break;  
  
                case 2:  
                    System.out.println("i eshte 2");  
                    break;  
  
                default:  
                    System.out.println("i eshte me  
i madh se 2");  
            }  
        }  
    }  
  
    // CONSOLE  
    /*  
    i eshte 0  
    i eshte 1  
    i eshte 2  
    i eshte me i madh se 2  
    */  
}
```

While Loop

Realizon krijimin e një cikli për të kryer pune ciklike, si më poshte.

```
public class LoopWhile {
    public static void main(String[] args) {
        int i = 5;
        while (i > 0){
            System.out.println("Ky eshte cikli while");
            i--;
        }
    }
}
```

Console

```
Ky eshte cikli while
Ky eshte cikli while
Ky eshte cikli while
Ky eshte cikli while
Ky eshte cikli while
```

Pra sintaksa e while është:

```
while (kushti është i vërtetë){
```

```
    //veprime
```

```
}
```

Pra për sa kohë kushti është i vërtetë, ekzekuto kushtin brenda kllapave të while.

Shembull – TestimiOnline

Shpjego se cfarë bën kodi i mëposhtëm.

```
import java.util.Scanner;
public class Testimi{
public static void main(String args[])
    {
// Deklaroj variablat që do më nevojiten
final int nr_pyetjesh = 5;
int numero_pyetje = 0, pergj_sakta = 0, pergj_josakte = 0;
long koha_e_nisjes = System.currentTimeMillis();
    String output = " ";

// Ndërtoj objektin Scanner
    Scanner input = newScanner(System.in);

while (numero_pyetje < nr_pyetjesh){
//1. Gjenero dy numra të rastësishëm një shifror
int nr1 = (int) (Math.random() * 10);
int nr2 = (int) (Math.random() * 10);

//2. Nqs nr1 është më i vogël se numri 2 => ndërro vendet
if (nr1 < nr2){
int temp = nr1;
        nr1 = nr2;
        nr2 = temp;
    }
}
```

```

//3. Afisho pyetjen
System.out.println("Sa është " + nr1 + " - " + nr2 + " ?");

//4. Marr përgjigjen dhe e vlerësoj
int pergjigja = input.nextInt();
if (nr1 - nr2 == pergjigja){
System.out.println("E sakte");
pergj_sakta++;
    }
else {
System.out.println("E pasakte");
System.out.println("Pergjigja e sakte eshte " + (nr1-nr2));
pergj_josakte++;
    }//mbaron if-else

//5. Kaloj në pyetjen pasardhëse
numero_pyetje++;
    }//mbaron while

long koha_e_mbarimit = System.currentTimeMillis();
System.out.println("ju keni dhene " + pergj_sakta + " pergjigje te
sakta dhe " + pergj_josakte + " pergjigje te pasakta");

long koha_e_testit = koha_e_mbarimit - koha_e_nisjes;
System.out.println("Afati i plotesimit te testit eshte: " +
koha_e_testit + " milisekonda");
    }//mbaron metoda main
} //mbaron klasa Testimi

```

Console

Sa eshte 7 - 7 ?

0

E sakte

Sa eshte 8 - 3 ?

7

E pasakte

Pergjigja e sakte eshte 5

Sa eshte 9 - 3 ?

8

E pasakte

Pergjigja e sakte eshte 6

Sa eshte 7 - 6 ?

1

E sakte

Sa eshte 4 - 0 ?

4

E sakte

ju keni dhene 3 pergjigje te sakta dhe 2 pergjigje te pasakta

Afati i plotesimit te testit eshte: 12933 milisekonda

Kodi realizon gjenerimin e dy numrave të rastësishëm nëpërmjet metodës *currentTimeMillis()*. Më pas ai afishon një mesazh ku i kërkon përdoruesit që të shkruaj rezultatin e pyetjes, që është rezultati i zbritjes së numrave. Për të mos dhënë asnjëherë një rezultat negativ, kodi ka kushtin që nëse numri i parë i gjeneruar është më i vogël se numri i dytë i gjeneruar, ata të ndërrojnë vlerat me njëri-tjetrin.

Është ndërtuar objekti dhe variabli për lexim pergjigja. Nëse pergjigja që fut përdoruesi është e saktë, atëherë kodi do të afishojë“e saktë” dhe do ta ruaj këtë pergjigje të saktë në numruesin e pergjigjeve të sakta **pergj_sakta**. Përndryshe në **pergj_josakte**. I gjithë kodi është futur në një cikël while, ku ne kemi parapërcaktuar që do të kemi vetëm 5 pyetje.

Pas mbarimit të ciklit **while**, kodi afishon numrin e pergjigjeve të sakta dhe jo të sakta si dhe diferencën e kohës së nisjes së testit me atë të mbarimit në milisekonda.

5.2 Përdorimi i DIALOG BOXES. GUI (Graphical User Interface).

- Shumë aplikime përdorin dritaret – dialog boxes – për të shfaqur output.
- Zakonisht dialog boxes janë dritare në të cilat programi shfaq mesazhe të rëndësishme për përdoruesit.
- Klasa **JOptionPane** mundëson shfaqjen e dritareve nga programi me mesazhe të rëndësishme për përdoruesin – që quhen message dialogs.

Libraria javax.swing ka shumë klasa që na ndihmojnë të krijojmë GUI.

- Komponentet e GUI lehtësojnë futjen e të dhënave nga përdoruesi dhe afishimin e të dhënave nga programi.
- Metoda showMessageDialog klasës JOptionPane shfaq një dialog box që mban një mesazh.
- Kërkon dy parametra
- I pari ndihmon aplikimin java ku duhet ta pozicionojë në ekran message box-in. Nqs argumenti i parë është null, dialog box shfaqet në mënyrë default në qendër të ekranit
- Parametri i dytë është stringu që do të shfaqet në ekran (**Kujdes!** stringun duhet ta vendosni në thonjza dyshe)

Metoda showMessageDialog e klasës JOptionPane është një metode statike.

- Thërritet me emrin e klasës e ndjekur me pikë (.) dhe emrin e metodës, psh. **ClassName.methodName(arguments)**
- **Vëmendje!** Vëmë re që nuk krijojmë një objekt të klasës **JOptionPane** për të përdorur metodën e tij statike **showMessageDialog**.

Një input- dialog lejon përdoruesin të fusë të dhëna në program.

- Metoda showInputDialog klasës JOptionPane shfaq një dritare për të vendosur input.
- Përmban një prompt dhe një fushë (të njohur si text field), në të cilën përdoruesi mund të fusë të dhëna.
- Metoda **showInputDialog** kthen një String që do të mbajë karakteret e shtypura nga përdoruesi.
- Nqs shtypim **Cancel** ose tastin **Esc**, metoda do të kthejë **null**.

Shembull – showInputDialog

Shikoni më poshtë se si mund ta përdorni metodën *showInputDialog*

```

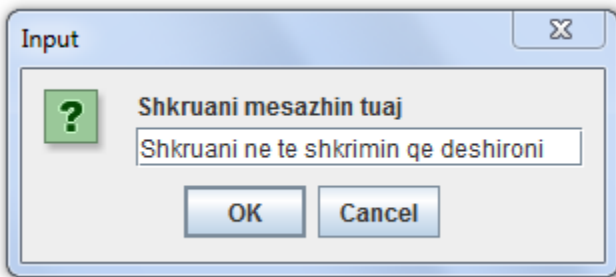
import javax.swing.JOptionPane;
public class ShfaqDritare {
    public static void main(String[] args) {

        // krijoj variablin mesazhi per te ruajtur mesazhin qe do
shkruhet
        String mesazhi = JOptionPane.showInputDialog("Shkruani
mesazhin tuaj");

        System.out.println(mesazhi);
    }
}

```

Nga ekzekutimi i këtij kodi, do t'ju shfaqet dritarja e mëposhtme:



Shkruani në të, shkrimin që deshironi dhe shtypni enter që ai të shfaqet në Console si më poshtë:

Console

Shkruani ne te shkrimin qe deshironi

Shembull – showInputDialog dhe showMessageDialog

Në këtë shembull do të kuptoni se si mund ti alternoni këto dy metoda për të arritur përdorimin sa më eficient të tyre:

```
import javax.swing.JOptionPane;

public class DialogBoxTest {
    public static void main( String[] args )
    {
        String emri = JOptionPane.showInputDialog("Si e keni emrin?");
        String mesazhi = String.format("Mireerdhe %s ne mesazhin e aplikimit Java!", emri);
        JOptionPane.showMessageDialog(null, mesazhi );
    }
}
```

