//Aritmetični operatorji: (rezultat je število)

|  |  |
| --- | --- |
| Operator | Namen |
| + | vsota števil, konkatenacija nizov (String-ov), pretvorba celoštevilskega števila v int |
| += | seštej in priredi vsoto, stakni in priredi niz (String) |
| - | razlika števil , aritmetična negacija števila |
| -= | odštej in priredi razliko |
| \* | zmnožek števil |
| \*= | zmnoži in priredi |
| / | količnik števil |
| /= | deli in priredi (divide and assign) |
| % | določi ostanek pri celoštevilskem deljenju (take remainder) |
| %= | določi ostanek pri celoštevilskem deljenju in priredi (take remainder and assign) |
| ++ | povečaj za 1 (increment by one) |
| -- | zmanjšaj za 1 (decrement by one) |

//Relacijski operatorji (rezultat je logična vrednost)

|  |  |
| --- | --- |
| Operator | Namen |
| > | večje kot (greater than) |
| >= | večje ali enako kot (greater than or equal to) |
| < | manjše kot (less than) |
| <= | manjše ali enako kot(less than or equal to) |
| == | test enakosti |
| != | test različnosti |

//Logični operatorji

|  |  |
| --- | --- |
| Operator | Namen |
| ! | logična negacija (boolean NOT) |
| != | logična neenakost (not equal to) |
| && | logični IN (boolean AND) |
| || | logični ALI (boolean OR) |
| == | logična enakost (boolean equals) |

//Bitni operatorji

|  |  |
| --- | --- |
| Operator | Namen |
| ~ | bitna negacija oz. eniški komplement (bitwise NOT) |
| | | bitni ALI (bitwise OR) |
| |= | bitni ALI in prirejanje (bitwise OR and assign) |
| ^ | bitni ekskluzivni ALI (bitwise XOR) |
| ^= | bitni ekskluzivni ALI in prirejanje (bitwise XOR and assign) |
| & | bitni IN (bitwise AND) |
| &= | bitni IN in prirejanje (bitwise AND and assign) |
| >> | pomik bitov v desno z upoštevanjem predznaka (shift bits right with sign extension) |
| >>= | pomik bitov v desno z upoštevanjem predznaka in prirejanje (shift bits right with sign extension and assign) |
| << | pomik bitov levo (shift bits left) |
| <<= | pomik bitov levo in prirejanje (shift bits left and assign) |
| >>> | pomik bitov desno brez upoštevanja predznaka (unsigned bit shift right) |
| >>>= | pomik bitov desno brez upoštevanja predznaka in prireditev (unsigned bit shift right and assign) |

//Nekategorizirani operatorji

|  |  |
| --- | --- |
| = | prirejanje (assignment) |
| ?: | pogojni operator (conditional) |
| . | operator dosega (selekcijski operator, selection) |
| :: | operator dosega (selekcijski operator, selection) |
| ( ) | vsiljena pretvorba tipa (cast), združevanje v aritmetičnih izrazih |
| [] | deklaracija tabele, kreiranje tabele, indeksiranje el. tabele |
| new | kreira nov objekt ali tabelo elementov |
| instanceof | preverjanje tipov (type checking) |

OSNOVE:

---------------------------------------------------------------

public static void main(String[] args){

}

---------------------------------------------------------------

import java.util.\*;

import java.util.Scanner;

Scanner vhod=new Scanner (System.in);

System.out.println();

System.out.print ();

String ime=vhod.next();

String ime=vhod.nextLine();

int a=vhod.nextInt();

Double a=vhod.nextDouble();

------------------------------------------------------------

//branje znaka preko niza

String znakovni\_niz=vhod.next();

char znak=znakovni\_niz.charAt(0);

------------------------------------------------------------

//if struktura

if (ime.equals("LJUBLJANA")){

}else{

}

//if struktura (razširjena struktura)

if (ime.equals("LJUBLJANA")){

}else if(pogoj1){

} else if(pogoj2)

}else{

}

----------------------------------------------------------------------------------

//pogojno logični operator

stevilo=(A>=B) ? "A je večje ali enako B":"B je večje od A";

//switch struktura

switch (spremenljivka) {

 case 1:

 break;

 case 2:

 break;

 default:

 }

--------------------------------------------------------------

//while struktura

while (spremenljivka!=0){

 }

--------------------------------------------------------------

//do while struktura

do {

}while(spremenljivka!=0);

--------------------------------------------------------------

//for struktura

for (int stevec=1; stevec<=stevilo; stevec++) {

 }

--------------------------------------------------------------

//dolžina niza

int dolzina\_niza=znakovni\_niz.length();

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//metoda, brez argumentov in ne vrača vrednosti

metoda();// klic metode

public static void metoda(){

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//metoda, brez argumentov, vrača vrednost

metoda();// klic metode

public static int vraca\_vrednost metoda(){

return lokalna\_spremenljivka;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//metoda z argumenti, ne vrača vrednosti

int a=5;

metoda(a);// klic metode

public static void metoda(int lokalna\_spremenljivka){

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//metoda z argumenti, vrača vrednost

int a=5;

metoda(a);// klic metode

public static int vraca\_vrednost metoda(int lokalna\_spremenljivka\_a){

 return lokalna\_spremenljivka\_b;

}

----------------------------------------------------------------------

Matematične metode

Math.sqrt(double spremenljivka); //kvadratni koren

Math.pow(double osnova, double eksponent); //metoda za potencirati število

**VIZUALNO PROGRAMIRANJE-»DOGODKOVNO« PROGRAMIRANJE**

 //INICIALIZACIJA VNOSNEGA POLJA

jTextField.setText("0");

//V. P. BRANJE VNOSNEGA POLJA jTextField

a=Double.parseDouble(jTextField.getText());

b=Integer.parseInt(jTextField.getText());// pretvorba znakovnega niza v celoštevilčno spremenljivko

ime\_priimek=jTextField.getText(); //BRANJE ZNAKOVNEGA NIZA

tab[i]=jTextField.getText();

//V. P. IZPIS V LABELO

jLabel.setText(""+c);

jLabel.setText(tab[0]+" "+tab[1]+" "+tab[2]);

//V. P. IZPIS V jTextArea

jTextArea.setText(""+c);

jTextArea.setText(tab[0]+" "+tab[1]+" "+tab[2]);

jTextArea.append("abc"+"\n"); //izpis se nadaljuje v novi vrstici, "\n"-skok v novo vrstico

//PRETVORBA SPREMENLJIVKE char V String ZA append METODO

jTextArea1.append(String.valueOf('a'));

//V. P. NASTAVITEV jTextArea ELEMENTA ZA PRIKAZ BESEDILA BREZ MOŽNOSTI UREJANJA

jTextArea1.setEditable(false);

jTextArea1.setCursor(null);

//V. P. SPREMINJANJE IKONE OKNA

import java.awt.Toolkit;//vnos knjižnice razredov

setIcon();//KLIC METODE setIcon

private void setIcon() {

 setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource("cube.jpg")));

 }

//V. P ODPIRANJE NOVEGA OKNA

 Okno2 a = new Okno2();

 a.setVisible(true);

//V. P METODA ZA ZAPIRANJE OKNA

import java.awt.Toolkit;

import java.awt.event.WindowEvent;

private void close() {

 WindowEvent oknoExit=new WindowEvent(this,WindowEvent.WINDOW\_CLOSING);

 Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(oknoExit);

 }

//PRIKAZ SLIKE V LABELI

jLabel.setVisible(true);

//SPREMINJANJE BARVE ELEMENTA label

import java.awt.Color;

jLabel1.setBackground(Color.red);

jLabel1.setOpaque(true);

//APLIKACIJA SE ZAPRE

System.exit(0);

//ODPRE SE BRSKALNIK IN DEFINIRANA INTERNETNA SPLETNA STRAN

Desktop povezava = Desktop.getDesktop();

povezava.browse(new URI("https://www.google.si"));

//ODPRE SE PROGRAM NA RAČUNALNIKU

Process p = Runtime.getRuntime().exec("C:/Program Files/Notepad++/notepad++.exe");

//OPOMBA: V NASLOVU ZAMENJAŠ ZNAK »\« ZA ZNAK »/«

//OKNU OB ZAGONU APLIKACIJE SE NE DA SPREMINJAT VELIKOSTI

setResizable(false);

//ODPIRANJE DOKUMENTA

Runtime.getRuntime().exec("rundll32 url.dll,FileProtocolHandler "+"C:\\Users\\peterz\\Desktop\\JAVA\_PZ.docx");

**TE ZADEVE SO V PROGRAMU VIŠJIH LETNIKOV (od 3. Letnika dalje)**

//tabele

String[] del\_dn\_v\_tednu={"pon","tor","sre","čet","pet"};

//Deklaracija tabele:
int[] tab;
String[] tab1;
double[] tab2;
char[] tab3;

//Rezervacija prostora za tabelo v pomnilniku:
tab=new int[10];
tab1=new String[10];
tab2=new double[10];
tab3=new char[10];

//Deklarcaija z rezervacijo:
int[] tab=new int[10];
String[] tab1=new String[10];
double[] tab2=new double[10];
char[] tab3=new char[10];

//Naključno generirano število na intervalu [a,b]
int s=(int)(Math.random()\*(b-a+1)+a);

// iz ascii v karakter oz. znak

c=(char)tab[i];

// 2D tabele

int a=0;

int[][] tabela = new int[2][2];

 for (int stolpec = 0; stolpec < tabela.length; ++ stolpec) {

 for (int vrstica = 0; vrstica < tabela.length; ++ vrstica) {

 }

 }

while (st\_tab!=tabela.length){

 System.out.println("element v diagonali: "+ tabela[st\_tab][st\_tab]);

 st\_tab++;

}

//iz znaka v niz

niz=String.valueOf(znak);

//ascii koda znaka

int ascii = (char) znak;

//metoda endsWith

String ime="PETER";

boolean odgovor;

odgovor=ime.endsWith("a");//true ali false

//iz znakovnega niza v tabelo znakov

char[] tabela = znakovni niz.toCharArray();

//metoda Character.isDigit

boolean odgovor = Character.isDigit(znak);// odgovor je true, če je znak število

Deklariranje razreda, spremenljivk kot Final

Java omogoča deklariranje nekega razreda, spremenljivk kot končnega final, kar pomeni, da iz njega ni mogoče izpeljati nobenega razreda več. Obstajata vsaj dva razloga za uporabo tega mehanizma: varnostni, kot recimo škodovanje sistemu, vdor med zasebne informacije itd in načrtovalni. Ko se odločimo, da je nek razred doživel svojo končno stopnjo razvoja, ga je pametno proglasiti za dokončnega. Razred deklariramo kot dokončen s pomočjo ključne besede final pred ključno besedo class. Spodnji primer deklarira razred kot dokončen:

 final class imeRazreda extends imeNadrazreda implements imeVmesnika {…}

Deklaracija javnih razredov, spremenljivk

Modifikator, ki ga lahko uporabimo v deklaraciji razreda, spremenljivk, je modifikator javen public. S tem omogočimo, da lahko razred uporabljajo tudi objekti, ki se nahajajo zunaj trenutnega paketa. Če tega ne napišemo, je razred dostopen samo tistim objektom, ki so v istem paketu. Po dogovoru se ključna beseda public napiše na začetku deklaracije razreda. Če razred deklariramo kot javen, tako da imajo tudi drugi objekti in razredi dostop do njega, se njegova deklaracija spremeni v:

 public final class imeRazreda extends imeNadrazreda implements imeVmesnika {…}

--------------------------------------------------------------------------------------------------------------------------------------------

//Delo z datotekami:

//razredi potrebni za delo z datotekami

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.PrintStream;

//glavna metoda s »podaljškom« v imenu

public static void main(String[] args) throws FileNotFoundException{

}

//kreiranje datoteke

PrintStream v\_datoteko = System.out;

File file = new File("izhodna\_datoteka.txt");

FileOutputStream fos = new FileOutputStream(file);//kreira file

PrintStream ps = new PrintStream(fos);

System.setOut(ps);

//od tu naprej se izpisuje v datoteko

System.setOut(v\_datoteko);//konec pisanja v datoteko izhodna\_datoteka.txt

-----------------------------------------------------------------



# DODATEK



Slika 1 PRINT FORMAT (PRIMER 1)



Slika 2 PRINT FORMAT (PRIMER 2)



Slika 3 PRINT FORMAT (PRIMER 3)



Slika 4 PRINT FORMAT (PRIMER 4)



Slika 5 PRINT FORMAT (PRIMER 5)



Slika 6 PRINT FORMAT (PRIMER 6)