

Java	Značenje
Uvod	
Program Eclipse IDE for Java Developers (verzija 3.7.1 u trenutku pisanja ovog dokumenta) u kojem ćemo kucati Java kod, možete skinuti sa stranice http://www.eclipse.org/downloads/	
// komentar	/* komentar */ komentar (može biti u jednom redu ili između znakaova /* i */)
import java.awt.*; import acm.graphics.*; import acm.program.*;	početak programa (da bi koristili acm biblioteku potrebno je učitati acm.jar paket na sljedeći način: File ---> Properties ---> Java Build Path ---> Libraries ---> Add External JARs... ---> acm.jar)
public class imeFajla extends GraphicsProgram { public void run() { ... naredbe ... } }	glavni (izvršni) dio programa
Promjenjive (variable)	
	/* u Javi promjenjiva se mora deklarirati (definisati) prije nego se upotrijebi. Data deklaracija promjenjive važi samo u klasi u kojoj je definisana */
tip imeVarijable;	deklaracija varijable
tip imeV1, imeV2;	deklaracija dvije varijable
tip1 imeV1; tip2 imeV2, imeV3;	deklaracija tri varijable koje mogu biti različitog tipa
tip imeVarijable = vrijednost;	preporučljiva deklaracija varijable;
byte float char boolean short double int long	tipovi varijabli
imeVarijable = izraz;	znak dodjeljivanja
+ - * / %	sabiranje, oduzimanje, množenje, dijeljenje, ostatak pri dijeljenju (modul)
/* u Javi treba voditi računa da bar jedan od dva djeljena broja bude realan broj. U protivnom, kao rezultat djeljenja dva cijela broja, ćemo dobiti cijeli broj (biće izvršeno cjelobrojno djeljenje) */	
&& !(izraz)	logički operatori: konjunkcija, disjunkcija i negacija
== != > < >= <=	relacioni operatori (znakovi jednakosti, različito, veće od, manje od, veće ili jednako i manje ili jednako)
k=k+1; k++; k+=1; k=k-1; k--; k-=1;	povećavanje i smanjivanje vrijednosti varijable k za jedan
i=i+j; i+=j; c=c*d; c*=d; a=a-b; a-=b; e=e/f; e/=f;	ekvivalenti i slične upotrebe znakova dodjele
++k; --k; i=j++; i=++j;	
{ ... }	početak i kraj neke cjeline ili bloka (može i tijela petlje)
Konstante i tri korisne funkcije	
private static final int imeKonstante= vrijednost;	definisanje cijelobrojne konstante (ova naredba se treba nalaziti iza public void run(){ ...} bloka)
int n= (int) imeVarijableKojaSadržiRacionalanBr;	pretvaranje racionalnog broja u cijeli
getWidth()	vraća širinu grafičkog prozora
getHeight()	vraća visinu grafičkog prozora
pause(konstanta);	pauzira izvršenje programa

<u>Promjenjive (preciznije klase) koje koristimo u grafičkom programu</u>	
GLLabel imeObjekta; imeObjekta = new GLLabel("tekst", x, y);	definisanje objekta koji će sadržavati tekst dodjeljivanje vrijednosti ranije definisanom GLLabel objektu
GLLabel imeObjekta = new GLLabel("tekst", x, y);	definisanje i dodjeljivanje vrijednosti GLLabel objektu
GRect imeObjekta; imeObjekta = new GRect(x, y, širina, visina);	definisanje objekta koji će sadržavati pravougaonik dodjeljivanje vrijednosti ranije definisanom GRect objektu
GRect imeObjekta = new GRect(x, y, širina, visina);	definisanje i dodjeljivanje vrijednosti GRect objektu
GOval imeObjekta; imeObjekta = new GOval(x, y, širina, visina);	definisanje objekta koji će sadržavati elipsu dodjeljivanje vrijednosti ranije definisanom GOval objektu
GOval imeObjekta = new GOval(x, y, širina, visina);	definisanje i dodjeljivanje vrijednosti GOval objektu
GLine imeObjekta; imeObjekta = new GLine(x ₀ , y ₀ , x ₁ , y ₁);	definisanje objekta koji će sadržavati liniju dodjeljivanje vrijednosti ranije definisanom GLine objektu (linija od tačke (x ₀ , y ₀) do tačke (x ₁ , y ₁))
GLine imeObjekta = new GLine(x ₀ , y ₀ , x ₁ , y ₁);	definisanje i dodjeljivanje vrijednosti GOval objektu
<u>Vrijednosti koje možemo dodijeliti GObject-u (GLLabel, GRect, GOval, GLine)</u>	
imeObjekta.setColor(imeBoje);	postavka boje objektu (boje se učitavaju iz paketa java.awt. Ime boje može biti: Color.BLACK, Color.DARK_GRAY, Color.GRAY, Color.LIGHR_GRAY, Color.WHITE, Color.RED, Color.YELLOW, Color.GREEN, Color.CYAN, Color.BLUE, Color.MAGENTA, Color.ORANGE, Color.PINK)
imeObjekta.setLocation(x, y);	postavlja objekat na željenu lokaciju
imeObjekta.move(dx, dy);	pomjera objekat za željeni broj piksela u x ili y pravcu
<u>Vrijednosti koje možemo dodijeliti GLLabel-u</u>	
imeGLLabelObjekta.setFont(imeFonta);	postavlka fonta GLLabel objektu (imeFonta je u obliku "familija-stil-veličina". Neke od familija su Serif, SansSerif, Monospaced, Dialog, DialogInput. Stilovi su PLAIN, BOLD, ITALIC ili BOLDITALIC)
<u>Vrijednosti koje možemo dodijeliti GRect-u i GOval-u</u>	
imeObjekta.setFilled(vrijednost);	postavka da li će objekat biti ispunjen (vrijednost može biti true ili false)
imeObjekta.setFillColor(imeBoje)	postavka boje unutar objekta
<u>Prikaz i uklanjanje objekata i Random klasa</u>	
add(imeObjekta);	ispis objekta na ekran
removeAll();	uklanja sve prikazane objekte sa ekrana
private RandomGenerator imeGeneratora = RandomGenerator.getInstance();	uključivanje generatora slučajnih brojeva (da bi smo mogli uključiti ovu naredbu potrebno je učitati biblioteku acm.util sa naredbom import acm.util.*;)
int imeVarijable = imeGeneratora.nextInt(donjiCijBr, gornjiCijBr);	cijlei broj imeVarijable će dobiti vrijednost iz intervala {donjiCijeliBroj,...,gornjiCijeliBroj}
Color boja=imeGeneratora.nextColor();	varijabla boja će sadržavati slučajno odabranu boju

<u>Strukture grananja</u>		
if (uvjet) { naredbe }	if (uvjet) naredba else {...naredbe...}	naredba grananja if
if(uslov1) { ... } else if (uslov2) { ... } else if (uslov3) { } else { ... }		naredbe grananja if
uslov ? izraz1 : izraz2;		?: operator ako je uslov tačan izvrši izraz1, ako nije izvrši izraz2
<u>Ciklične strukture (petlje)</u>		
break;		naredba za izlaženje iz petlji ponavljanja
while(uvjet) { ... }		naredba while (prvi oblik)
while(true) { tijeloPetlje if(vrijednost==potrebnaVrijednost) break; ostatakTijelaPetlje }		naredba while (drugi oblik)
for (inicijalizacija; uvjet; izmjena) { tijeloPetlje }		naredbe for /* sljedeći primjer ispisuje brojeve od 10 do 1 */ for (i=10; i>=1; i--) cout<<i<<endl;
do { ... } while (uvjet);		naredba do-while
switch(varijabla) { case konstanta1: naredbe1 break; case konstanta2: naredbe2 break; ... default: naredbe break; }		naredba case
<u>Nizovi</u>		
tip imeNiza[]; ... //kad unesemo broj elemenata imeNiza = new tip[brojElementa+1];		dinamički način definiranja niza
int imeNiza[] = new int[brojElementa+1];		statički način definiranja niza
naredba imeNiza[brojElementa]		pristup nizu
int array_name[] = new int[brojRed][brojKolona];		definiranje matrice
naredba imeMatrice[red][kolona]		pristup matrici

<i>Naredbe koje su prikazane u drugačijem obliku</i>	
<u>Korisne metode za GraphicProgram klasi</u>	
<u>Metode za dodavanje i uklanjanje grafičkih objekata</u>	
void add(GObject imeGobjekta)	dodaje grafički objekat na platno na unutarnju zabilježenu lokaciju
void add(GObject ImeGobj, double x, double y)	dodaje grafički objekat na platno na datu lokaciju
void remove(GObject imeGobj)	uklanja specifični grafički objekat sa platna
void removeAll()	uklanja sve grafičke objekte sa platna
<u>Metoda za pronalaženje grafičkog objekta na određenoj lokaciji</u>	
GObject getElementAt(double x, double y)	vraća gornji prvi objekat koji sadrži datu tačku, ili null vrijednost ako takav objekat ne postoji
<u>Metoda za podršku grafičkim animacijama (dostupno samo u GraphicsProgramu)</u>	
void pause(double milisekunde)	pauzira izvršenje programa za dati vremenski interval, koji je mjerljiv u milisekundama
void waitForClick()	pauzira izvršenje programa sve dok korisnik ne klikne mišem bilo gdje na platnu
<u>Korisne metode naslijeđene iz superklasa</u>	
int getWidth()	vraća širinu platna, u pikselima
int getHeight()	vraća visinu platna, u pikselima
void setBackground(Color imeBoje)	mijenja boju pozadine platna
<u>Korisne metode u RandomGenerator klasi</u>	
static RandomGenerator getInstance()	vraća uzorak RandomGenerator klase, koja se dijeli kroz ostatak programa
int nextInt(int n)	vraća slučajan cijeli broj izabran između n vrijednosti u rangu od 0 do n-1, uključivo
int nextInt(int donjaGranica, int gornjaGranica)	vraća slučajan cijeli broj u rangu od donjaGranica do gornjaGranica uključivo
double nextDouble()	vraća slučajan racionalan broj d u rangu $0 \leq d < 1$
double nextDouble(double donjaGr, double gorGr)	vraća slučajan racionalan broj d iz polu-otvorenog intervala $donjaGr \leq d < gorGr$
boolean nextBoolean()	vraća boolean vrijednost koji je tačan otprilike 50% u vremenu
boolean nextBoolean(double p)	vraća boolean vrijednost koja je tačna sa vjerovatnoćom p, koja mora biti između 0 i 1
Color nextColor()	vraća slučajnu Java boju
void setSeed(long seed)	postavlja "seed" koji inicira na poč. tačku pse. niza
<u>Odazivi na rad sa mišem</u>	
da bi se mogle koristiti naredbe koje se odnose na rad sa mišem potrebno je na samom početku programa učitati biblioteku java.awt.event.* sa naredbom: <i>import java.awt.event.*;</i>	
public void init() { addMouseListeners(); }	uključivanje metoda koje će se primjeniti na odazive miša. Ovu naredbu kucamo odmah iza: public class imeFajla extends GraphicsProgram {
addMouseListeners();	ovu naredbu možemo kucati i unutar public void run() { ... } dijela
public void mouseClicked(MouseEvent e) { ... naredbe kao npr. add(imeObjekta, e.getX(), e.getY()); ... }	dogadjaj koji će se izvršiti kad se napravi klik mišem (u ovom primjeru, dodat će se objekat imeObjekta na platno, na mjesto na kojem je napravljen klik)

<u>MouseListener interface</u>	
void mousePressed(MouseEvent e)	poziva se kadgod je dugme miša pritisnuto
void mouseReleased(MouseEvent e)	poziva se kadgod je dugme miša pušteno
void mouseClicked(MouseEvent e)	pozivase kadgod se izvrši klik (pritisne i pusti dugme miša sa kratkim vremenom između)
void mouseEntered(MouseEvent e)	poziva se kadgod je miš iznad platna
void mouseExited(MouseEvent e)	poziva se kadgod je miš van platna
<u>MouseMotionListener interface</u>	
void mouseMoved(MouseEvent e)	poziva se kadgod se miš pomjera sa puštenim dugmetom
void mouseDragged(MouseEvent e)	poziva se kadgod se miš pomjera sa pritisnutim dugmetom
<u>Korisne metode u GLine klasi</u>	
<u>Konstrukcija</u>	
new GLine(double x0, double y0, double x1, double y1)	kreira novi GLine objekat koji spaja tačke (x0,y0) i (x1,y1)
<u>Metode da dobijemo i postavimo krajnje tačke nezavisno</u>	
void setStartPoint(double x, double y)	resetuje koordinate početne tačke linije na (x,y) bez mijenjanja krajnje tačke
GPoint getStartPoint()	vraća koordinate početne tačke linije
void setEndPoint(double x, double y)	resetuje koordinate krajnje tačke linije na (x,y) bez mijenjanja početne tačke
GPoint getEndPoint()	vraća koordinate krajnje tačke linije
<u>Osnovne metode u GPoint, GDimension i GRectangle klasama</u>	
GPoint, GDimension i GRectangle klase se mogu tumačiti kao promjenjive koje čuvaju koordinate tačke, veličine neke dimenzije i položaj i veličinu pravougaonika, ove klase ne mogu prikazati te objekte	
<u>GPoint konstruktor i metode</u>	
new GPoint(double x, double y)	kreira GPoint objekat koji sadrži koordinate vrijednosti x i y
double getX()	vraća x koordinatu od GPointa
double getY()	vraća y koordinatu od GPointa
<u>GRectangle konstruktor i metode</u>	
new GRectangle(double x, double y, double width, double height)	kreira novi GRectangle objekat koji sadrži četiri specifične vrijednosti
double getX()	vraća x komponentu od GRectangle-a
double getY()	vraća y komponentu od GRectangle-a
double getWidth()	vraća width komponentu od GRectangle-a
double getHeight()	vraća height komponentu od GRectangle-a
<u>GDimension konstruktor i metode</u>	
new GDimension(double width, double height)	kreira novi GDimension objekat koji sadrži vrijednosti width i height
double getWidth()	vraća width komponentu od GDimension
double getHeight()	vraća height komponentu od GDimension
<u>Dodavanje fotografije na platno</u>	
GImage imePromSadrziSl=new GImage("imeSlike.png");	promjenjiva imePromSadrziSl će dobiti vrijednost fotografije pod imenom imeSlike sa ekstenzijom png, koja se nalazi u folderu ...eclipse\imeRadnogProstora\imeProjekta\bin
add(imePromSadrziSl, x, y);	dodavanje slike na platno

<u>Metode podržane u svim GObject podklasama</u>			
void setLocation(double x, double y)	postavlja položaj ovog objekta na datu tačku		
void move(double dx, double dy)	pomjera objekat koristeći pomake dx i dy		
void movePolar(double r, double theta)	pomjera objekat r jedinica u smjeru theta, gdje je theta ugao u stepenima		
double getX()	vraća x-koordinatu objekta		
double getY()	vraća y-koordinatu objekta		
double getWidth()	vraća širinu objekta		
double getHeight()	vraća širinu objekta		
boolean contains(double x, double y)	provjerava da li je tačka unutar objekta		
void setColor(Color c)	postavlja boju objekta		
Color getColor()	vraća boju objekta		
void setVisible(boolean visible)	postavlja vidljivost\nevidljivost objekta		
boolean isVisible()	vraća true ako je objekat vidljiv		
void sendToFront()	šalje ovaj objekat ispred na platnu, koji može prikrit neki drugi objekat šaljući ga iza		
void sendToBack()	šalje ovaj objekat iza na platnu		
void sendForward()	šalje objekat naprijed jednu poziciju u poretku		
void sendBackward()	šalje objekat nazad jednu poziciju u sleganom poretku		
<u>Rad sa dugmadima</u>			
/* za korištenje naredbi za rad sa dugmadima potrebno je učitati biblioteku javax.swing.*, tj na samom početku programa ukucati import javax.swing.*; */			
addActionListeners();	uključivanje metoda koje će se primjeniti na odazive klika na dugme		
JButton imeDugmeta=new JButton("natpisNaD");	definisanje dugmeta		
add(imeDugmeta, pozicija);	dodavanje dugmeta na platno gdje je pozicija jedna od riječi: NORTH, SOUTH, EAST i WEST		
public void actionPerformed(ActionEvent e) { if (e.getActionCommand().equals("natpisNaD")) ... }	podprogram koji će se izvršiti kada se klikne na dato dugme (tačnije izvršice se naredbe koje se nalaze iza if naredbe)		
<u>Odazivi na rad sa tastaturom</u>			
addKeyListener();	uključivanje metoda koje će se primjeniti na odazive tastature		
<u>Standardni slušači metode za odaziv na događaje sa tastaturom</u>			
void keyPressed(KeyEvent e)	poziva se kadgod je tipka pritisnuta		
void keyReleased(KeyEvent e)	poziva se kadgod je tipka puštena		
void keyTyped(KeyEvent e)	poziva se kad je tipka "otipkana" (pritisnuta i puštena)		
<u>Virtualne tipke definisane u KeyEvent klasi</u>			
VK_A sve do VK_Z	VK_BACK_QUOTE	VK_CONTROL	VK_PAGE_UP
VK_0 sve do VK_9	VK_QUOTE	VK_ALT	VK_PAGE_DOWN
VK_COMMA	VK_SPACE	VK_META	VK_HOME
VK_F1 sve do VK_F12	VK_PERIOD	VK_NUM_LOCK	VK_END
VK_NUMPAD0 sve do VK_NUMPAD9		VK_SCROLL_LOCK	VK_ESCAPE
VK_SEMICOLON	VK_BACK_SPACE	VK_CAPS_LOCK	VK_PRINTSCREEN
VK_EQUALS	VK_DELETE	VK_UP	VK_INSERT
VK_OPEN_BRACKET	VK_ENTER	VK_DOWN	VK_HELP
VK_BACK_SLASH	VK_TAB	VK_LEFT	VK_CLEAR
VK_CLOSE_BRACKET	VK_SHIFT	VK_RIGHT	VK_SLASH