

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/311677784>

# Principe të programimit në Java: Arkitekturat dhe Interfejsat (kapitujt I–VI)

Book · January 2004

CITATIONS

0

READS

1,292

2 authors, including:



[Ridvan Bunjaku](#)

South East European University

5 PUBLICATIONS 4 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



OOP and programming techniques in practice [View project](#)

# Principe të Programimit në Java: Arkitekturat dhe Interfejsat

Ushtrime  
(Pjesa e parë: kapitujt I - VI)

nga David Schmidt me Ridvan Bunjakun

```
import java.awt.*;
import javax.swing.*;
import java.text.*;
/** ShkruesIBankes i shkruan transaksionet e bankes */
public class ShkruesIBankes3 extends JPanel
{ private int GJERESIA = 300; // gjeresia dhe thellesia e dritares se paraqitur
  private int LARTESIA = 200;
  private LlogariEBankes banka; // adresa e llogarise se bankes qe paraqitet
  private String transaksioni_i_fundit = ""; // transaksioni qe paraqitet

  /** Konstruktori ShkruesIBankes e inicializon shkruesin
   * @param titulli - teksti i shiritit te titullit
   * @param b - llogaria e bankes se paraqitur nga shkruesi (adresa) */
  public ShkruesIBankes3(String titulli, LlogariEBankes b)
  { banka = b;
    JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle(titulli);
    korniza_ime.setSize(GJERESIA, LARTESIA);
    korniza_ime.setVisible(true);
  }
}
```

Gjilan  
2004

# Përmbajtja

<b>HYRJE</b> .....	<b>1</b>
<b>I KOMPJUTERËT DHE PROGRAMIMI</b> .....	<b>2</b>
1.1 Ç'ËSHTË KOMPJUTERI?.....	2
1.2 PROGRAMIMI I KOMPJUTERËVE .....	3
(1.3 PROGRAMET JANË OBJEKTE).....	4
1.4 SISTEMET OPERATIVE DHE WINDOWS-I .....	5
(1.5 ARKITEKTURA E SOFTUERIT) .....	5
<b>II JAVA APLIKACIONET</b> .....	<b>6</b>
(2.1 APLIKACIONI DHE ARKITEKTURA E TIJ).....	6
2.2 SI NDËRTOHET DHE EKZEKUTOHET NJË APLIKACION.....	6
2.3 SI FUNKSIONON APLIKACIONI .....	6
2.3.1 (GJURMA) SHTEGU I EKZEKUTIMIT PËR APLIKACIONIN .....	8
2.4 SI NJË OBJEKT E KONSTRUKTON NJË TJETËR.....	9
(2.5 PËRMIRËSIMI I MESAZHEVE TË GABIMEVE TË KOMPILATORIT).....	10
(2.6 PËRMBLEDHJE) .....	10
2.7 USHTRIME TË PROGRAMIMIT .....	11
<b>III ARITMETIKA DHE NDRYSHORET</b> .....	<b>12</b>
3.1 ARITMETIKA E NUMRAVE TË PLOTË .....	12
3.2 MADHËSITË E EMËRTUARA: NDRYSHORET.....	13
3.2.1 NDRYSHORET MUND TË NDRYSHOJNË: NDARJA E VLERËS .....	16
3.3 ARITMETIKA ME THYESA: TIPI DOUBLE.....	21
3.4 TIPI BOOLEAN .....	22
3.5 (PRIORITETI) PËRPARËSIA E OPERATORËVE.....	23
3.6 STRINGJET, KARAKTERËT DHE VEPRIMET E TYRE .....	24
3.7 KONTROLLIMI I TIPEVE TË TË DHËNAVE .....	25
3.8 HYRJA (INPUT) PËRMES ARGUMENTEVE PROGRAMORE .....	26
(3.8.1 Konvertimi ndërmjet Stringjeve e Numrave dhe Formatizimi).....	26
3.8.2 Konvertimi i Temperaturës me Hyrje (Input).....	26
(3.9 DIAGNOSTIFIKIMI I GABIMEVE NË SHPREHJE) .....	29
(3.10 FJALËT KYÇE DHE IDENTIFIKATORËT).....	29
<b>IV HYRJA (INPUT), DALJA (OUTPUT) DHE GJENDJA</b> .....	<b>30</b>
4.1. HYRJA (INPUT) INTERAKTIVE .....	30
4.2 DALJA (OUTPUT) GRAFIKE .....	32
(4.2.1 PANELAT DHE KORNIZAT E TYRE) .....	32
4.2.2 PËRSHTATJA E PANELAVE ME TRASHËGIMI.....	32
4.3. FORMATI DHE METODAT PËR VIZATIM .....	34
4.3.1 Konstruktore-metodat dhe Objekti this .....	34
4.4 OBJEKTET ME GJENDJE: NDRYSHORET E FUSHAVE.....	36
4.5. TESTIMI I PROGRAMIT QË PËRDOR HYRJE.....	39

<b>V STRUKTURA E KOMPONENTEVE: NDËRTIMI I METODAVE DHE KLASAVE .....</b>	<b>41</b>
(5.1 METODAT) .....	41
5.2 METODAT PUBLIKE.....	41
5.2.1 <i>Disa Metoda Publike Themelore</i> .....	41
5.2.2 <i>Metodat Konstruktorë</i> .....	44
5.3 PARAMETRAT E METODAVE.....	46
5.3.1 FORMAT E PARAMETRAVE .....	47
5.4 CASE STUDY: KORNIZË DALËSE ME DESTINIM TË PËRGJITHSHËM.....	51
5.5 REZULTATET E METODAVE: FUNKSIONET .....	56
5.6 METODAT PRIVATE .....	59
<b>VI STRUKTURA E KONTROLLIT: URDHËRAT KUSHTËZUES DHE ARKITEKTURA E SOFTUERIT .....</b>	<b>65</b>
(6.1 RRJEDHA E KONTROLLIT DHE STRUKTURA E KONTROLLIT) .....	65
6.2 STRUKTURA KUSHTËZUESE E KONTROLLIT.....	65
6.2.1 URDHËRAT KUSHTËZUES TË VENDOSUR PËRBRENDË .....	67
(6.2.2 PROBLEMET SINTAKSORE ME KUSHTËZUES).....	69
6.3 VEPRIMET RELACIONALE (OPERATORËT LOGJIKË) .....	70
6.4 ZBATIMI I URDHËRAVE KUSHTËZUES.....	72
(6.5 NDRYSHIMI I RRJEDHËS SË KONTROLLIT).....	73
(6.6 URDHËRI SWITCH).....	73
(6.7 KOMPONENTET MODEL DHE KONTROLLUES).....	73
6.8 MENAXHER KONTOSH TË BANKËS (CASE STUDY) .....	74
6.8.10 OBJEKTE TË SHUMËFISHTA NGA E NJËJTA KLASË .....	77
(6.9 TESTIMI I METODAVE DHE KLASAVE) .....	77
<b>SHTOJCË: KËSHILLA PRAKTIKE PËR PROGRAMIM.....</b>	<b>78</b>
EMËRTIMI NË GJUHËN SHQIPE I EMRAVE NË KOD.....	78
NIVELET NË KOD DHE HIERARKIA .....	84
NXJERRJA E ASCII KODEVE .....	87
FJALËT KYÇE TË GJUHËS PROGRAMUESE JAVA.....	88

## Hyrje

Në këtë tekst janë përkthyer dhe përshtatur ushtrimet nga kapitujt I-VI të librit të David Schmidt: "Programming Principles in Java: Architectures and Interfaces", që mund ta gjeni në [www.cis.ksu.edu/santos/schmidt/ppj](http://www.cis.ksu.edu/santos/schmidt/ppj). Poashtu, në fund është edhe një shtojcë me artikuj këshillues për programim të mirë. E falënderoj autorin për lejimin e publikimit të këtij teksti.

Libri i Schmidt-it është i mrekullueshëm për fillestarët në programim dhe konkretisht në Java. Me anë të shembujve praktikë e shumë të afërt me cilindo lexues, autori e bën të këndshëm mësimin e Java-s dhe të teknikave të ndryshme të programimit.

Motivi për shkruarjen e këtij teksti në gjuhën shqipe lindi gjatë punës me studentë të cilët nuk e zotëronin mirë gjuhën angleze. Për shkak të pengesës së gjuhës ata nuk mund ta lexonin librin dhe t'i kuptonin kodet e ndryshme. Përshtatja ime e kodit në gjuhën shqipe (në masën sa është e mundur) doli të jetë e dobishme për ta.

Natyrisht, parapëlqehet që ky tekst të përdoret krahas librit të Schmidt-it. Poashtu, informata të dobishme mund të mirren edhe nga ueb-faqja e prof. Faton Berisha: [fberisha.netfirms.com](http://fberisha.netfirms.com).

### Për kë është ky tekst?

Ky tekst është shkruar për studentët që i ndjekin ushtrimet nga "Programimi dhe algoritmet" në Universitetin e Prishtinës, FSHMN, Departamenti i Matematikës, Drejtimi i Shkencës Kompjuterike. Mirëpo mund ta përdorin si tekst fillestar të gjithë ata që duan ta mësojnë gjuhën Java bashkë me disa teknika bashkëkohore të programimit dhe filozofinë e programimit të orientuar kah objektet.

### Gjuha e përdorur në tekst

Ky tekst nuk e ka për qëllim shmangien e mësimin të gjuhës angleze, por mësimin thelbësor të gjërave. Gjuha angleze, në të vërtetë, është kusht i domosdoshëm për Shkencën Kompjuterike.

I mirëpres të gjitha vërejtjet, këshillat dhe komentet rreth tekstit.

Ridvan Bunjaku, ing. mat.  
[ridvan.bunjaku@fshmn.uni-pr.edu](mailto:ridvan.bunjaku@fshmn.uni-pr.edu)

# **I Kompjuterët dhe Programimi**

## **1.1 Ç'është Kompjuteri?**

1. Nëse kompjuter është çdo qenie që mund t'i zbatojë urdhërat, atëherë jep shembuj të kompjuterëve nga jeta e përditshme.

Përgjigje:

Ushtari është kompjuter që i zbaton urdhërat që ia jep eprori i vet. Gjellëbërësja është kompjuter që bën ushqim të ngrohtë sipas urdhërit. Shitësi është kompjuter që i sjell mallërat që i kërkon ti. Kabina e telefonit është kompjuter që i zbaton urdhërat që ti ia jep, pasi ta futësh kartelën e rregullt. Vetura është kompjuter që lëviz sipas urdhërave tua. Natyrisht, ka edhe shumë shembuj të tjerë.

2. Shëno të gjitha pajisjet hyrëse-dalëse që janë të lidhura me kompjuterin që do ta përdorësh për ushtrimet e tua të programimit.

Përgjigje:

Pajisje hyrëse: tastiera, miu (mausi), mikrofoni.

Pajisje dalëse: monitori, altoparlantët, shtypësi.

Pajisje hyrëse-dalëse: kartela e rrjetit.

3. Llogaritësit-kalkulatorët janë kompjuterë. Cilat janë pajisjet hyrëse-dalëse për një llogaritës?

Përgjigje:

Pajisje hyrëse për llogaritësin është tastiera (bashkësia e tasteve) në të cilën ti i shtyp numrat dhe veprimet aritmetike. Pajisje dalëse është ekrani ose shiriti i letrës në të cilin paraqiten rezultatet.

## 1.2 Programimi i kompjuterëve

1. Gjeje një libër me receta dhe studioje një nga recetat e tij. Shëno foljet deklarative, emrat, dhe sasitë e sakta. Poashtu, rretho çdo udhëzim (instruksion) në recetë që ty të duket jopreciz.

Përgjigje:

Receta për përgatitjen e një çaji:

*2 filter-qese përvlohen me 200ml ujë. Përzihet 3-5 min. dhe lihet të qëndrojë 30 minuta në gotë të mbuluar.*

*Kullohet dhe ashtu i nxehtë të pihet.*

*Çaji i përgatitur i freskët pihet 2-3 herë në ditë.*

*Sheqer i shtohet sipas shijes.*

Foljet deklarative: përvlohen, përzihet, lihet të qëndrojë, kullohet, pihet, shtohet.

Emrat: filter-qese, ujë, minutë, gotë, çaj, sheqer, shije.

Udhëzime joprecize: përzihet 3-5 min.; "Ashtu i nxehtë" (a është çaji domosdo i nxehtë?); 2-3 herë në ditë, sipas shijes.

2. Aritmetika shpesh quhet "gjuha e parë programuese". Ta zëmë se kjo shprehje është një program

$$(3 + 2) - 1 + (6 + 5)$$

dhe ta zëmë se ti je kompjuter. Shëno hapat që i ndërmerr me lapsin tënd për t'i kryer (ekzekutuar) udhëzimet (instruksionet) në këtë program. (Domethënë, llogarite rezultatin e shprehjes "një llogaritje në hap").

Përgjigje:

Kryerja (ekzekutimi) në drejtimin prej majtas-djathtas, do t'i kalonte këta hapa:

1) mbledh 3 me 2, që jep 5 |  $5 - 1 + (6 + 5)$

2) zbrite 1, që jep 4 |  $4 + (6 + 5)$

3) mbledh 6 me 5, që jep 11 |  $4 + 11$

4) mbledh 4 me 11, që jep 15 | 15.

(Provoni në drejtimin prej djathtas-majtas)

3. Algjebra është gjuhë programuese. Shëno hapat që i bën për ta zgjidhur vlerën e  $x$ -it në këtë "program":

$$3y - x = 3 + 2x$$

(Udhëzim: hapi i parë është të shtosh  $x$  të dy anëve të barazimit, duke e dhënë barazimin e ri,  $3y - x + x = 3 + 2x + x$ .)

Përgjigje:

$3y - x = 3 + 2x$	shtojmë $x$ të dy anëve të barazimit
$3y - x + x = 3 + 2x + x$	i mbledhim $-x$ me $x$ në anën e majtë, që jep 0 (anulohen)
$3y = 3 + 2x + x$	i mbledhim $2x$ me $x$ në anën e djathtë
$3y = 3 + 3x$	shtojmë $-3$ të dy anëve të barazimit
$3y - 3 = 3 + 3x - 3$	ua ndërrojmë vendet $3x$ dhe $-3$ (ana e dj.) (vetia komutat.)
$3y - 3 = 3 - 3 + 3x$	i zbresim $3$ me $3$ në anën e djathtë
$3y - 3 = 3x$	pjesëtojmë secilin monom me $3$
$3y : 3 - 3 : 3 = 3x : 3$	pjesëtojmë $3y$ me $3$ në anën e majtë
$y - 3 : 3 = 3x : 3$	pjesëtojmë $3$ me $3$ në anën e majtë
$y - 1 = 3x : 3$	pjesëtojmë $3x$ me $3$ në anën e djathtë
$y - 1 = x$	ua ndërrojmë anët shprehjeve
$x = y - 1.$	fund.

**4. Qe një pjesë e vogël e një Java-programi:**

```
int x = 3 + 2;
int y = x + 1;
afishoNëKonsolë(y);
```

Cilat pjesë të këtij programi duken të jenë folje? emra? mbiemra? shprehje algjebrike?

Përgjigje:

Folje: = (ndaj vlerën), afishoNëKonsolë

Emra: x, y, 3, 2, 1

Mbiemra: int

Shprehje algjebrike: 3+2

**5.** Propozo një gjuhë programuese për të vizatuar rrathë dhe katrorë me ngjyra në një fletë. Cilat folje do t'i përfshish (p.sh. "vizato", "përshko")? emra ("rreth")? mbiemra ("i madh", "i errët")?

Përgjigje:

Folje: "vizato rrethin", "vizato katrorin", "përshko rrethin", "përshko katrorin", "fshij rrethin", "fshij katrorin", "ngjyros rrethin", "ngjyros katrorin", "vizato harkun", "fshij harkun"

Emra: rreth, katror, rreth i mbushur, katror i mbushur, ngjyrë

Mbiemra: i madh, i mesëm, i vogël, i errët, i ndritur etj.

**(1.3 Programet Janë Objekte)**



## **1.4 Sistemet Operative dhe Windows-i**

1. Përdore kompjuterin për ta filluar një program, p.sh. lojë apo përpunues fjalësh (word-processor). Shëno të gjitha dritaret që krijohen nga programi, shëno mënyrat se si ti ia jep të dhënat hyrëse programit, dhe shëno mënyrat se si e shfaq programi daljen.

Përgjigje:

Hapja e Word 2000:

Dritaret: dritarja kryesore e programit, dritarja e dokumentit aktiv.

Mënyrat se si ia jap të dhënat hyrëse: me tastierë, me maus.

Programi e shfaq daljen: përmes ekranit, përmes shtypësit, përmes altoparlantëve (tingujt e ndryshëm).

## **(1.5 Arkitektura e Softuerit)**

## **II Java Aplikacionet**

### **(2.1 Aplikacioni dhe Arkitektura e Tij)**

### **2.2 Si Ndërtohet dhe Ekzekutohet një Aplikacion**

1. Instalo ose një Rrethinë Zhvilluese të Integruar (Integrated Development Environment, IDE), ose Java Development Kit-in (JDK) dhe shkruaje, kompajloje dhe ekzekutoje programin vijues:

```
/** Tung afishon dy rreshta në konsolë */
public class Tung
{
    public static void main(String[] args)
    {
        System.out.println("Tungjatjeta botë!");
        System.out.println(2004);
    }
}
```

### **2.3 Si Funksionon Aplikacioni**

1. Shkruaje një Java-program që e afishon emrin dhe mbiemrin tënd në një rresht.

Përgjigje:

```
/** Emril e shenon emrin tim ne nje rresht */

public class Emril
{
    static public void main(String[] args)
    {
        System.out.println("Filan Fisteku");
    }
}
```

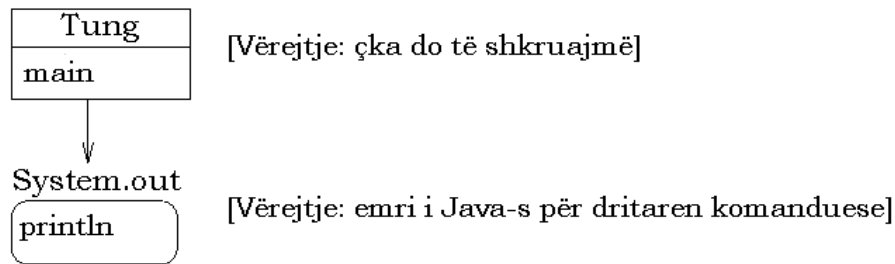
2. Shkruaje një Java-program që e afishon emrin dhe mbiemrin tënd, ashtu që emri yt është në një rresht të veçantë, dhe mbiemri yt është në rreshtin tjetër.

Përgjigje:

```
/** Emri2 e afishon emrin dhe mbiemrin tim ashtu qe
 * emri afishohet ne nje rresht te vecante,
 * dhe mbiemri afishohet ne rreshtin tjeter. */

public class Emri2
{ public static void main(String[] args)
  { System.out.println("Filan");
    System.out.println("Fisteku");
  }
}
```

3. Për të dy ushtrimet e mëparshme, shpjego pse programi yt e ka arkitekturën e paraqitur si në figurën e mëposhtme. (Kjo e qartëson faktin se programe të ndryshme mund të ndërtohen në të njëjtin stil të arkitekturës).

Përgjigje:

Të dy programet e kanë një klasë me një metodë `main`; metoda i dërgon mesazhe objektit `System.out`.

### 2.3.1 (Gjurma) Shtegu i Ekzekutimit për Aplikacionin

1. Shkruaje një shteg të ekzekutimit (execution trace) për një program që e shkrove nga ushtrimet e mëhershme.

Përgjigje:

Do ta shkruajmë shtegun e ekzekutimit të programit `Emril.java`.

```
/** Emril e shenon emrin tim ne nje rresht */
public class Emril
{ static public void main(String[] args)
  { System.out.println("Filan Fisteku");
  }
}
```

Atëherë kur aplikacioni `Emril` nisët nga përdoruesi, fajli `Emril.class` kopjohet në RAM dhe bëhet objekt `Emril`.

Që një përshkrim të objektit `Emril` (dhe `System.out`, që tashmë ekziston) në RAM.

**Emril**

```
static public void main(String[] args)
{ System.out.println("Filan Fisteku");
}
```

**System.out**

```
...
println(...)
{ udhëzimet për të shkruar tekst }
```

Objektet rrijnë ("pushojnë") në RAM dhe presin mesazhe. Kështu, për ta filluar punën, kompjuteri (më saktësisht JVM - Java Virtual Machine) ia dërgon objektit `Emril` një mesazh për ta filluar metodën e vet `main`.

Kur `Emril` e pranon mesazhin, ai në të vërtetë e nis metodën `main`, urdhërat e së cilës ekzekuten, një nga një. Do ta përdorim një shënues, `>`, për të theksuar se cili urdhër kryhet (ekzekutohet) më parë.

**Emril**

```
static public void main(String[] args)
{ > System.out.println("Filan Fisteku");
}
```

**System.out**

```
...
println(...)
{ udhëzimet për të shkruar tekst }
```

Urdhëri, `System.out.println("Filan Fisteku")`, ia dërgon mesazhin, `println("Filan Fisteku")` objektit `System.out`. Argumenti "Filan Fisteku" është ajo çka do të afishojë metoda. (Thonjëzat nuk afishohen, por duhet t'i përfshini, megjithatë). Ky është një shembull se si një objekt ia dërgon një mesazh objektit tjetër, duke e zgjuar nga pushimi.

Mesazhi i dërgohet objektit `System.out`, i cili e vë në punë metodën e vet `println`. Në ndërkohë, objekti `Emril` pret:

**Emril**

```
static public void main(String[] args)
{ > PRITE PËRFUDIMIN E System.out
}
```

**System.out**

```
...
println(...)
{ > udhëzimet për të shkruar tekst }
```

Përfundimisht, `System.out` e përmbush kërkesën dhe tekstin, `Filan Fisteku`, paraqitet në dritaren komanduese. Sapo të ndodhë kjo, `System.out` e lajmëron objektin `Emril` se mund të vazhdojë me urdhërin tjetër, dhe `System.out` kthehet në gjendjen e vet të "pushimit".

Sapo të mbarojë `println`, në trupin e metodës `main` më nuk ka asgjë për t'u ekzekutuar.

**Emril**

```
static public void main(String[] args)
{ ...
> }
```

**System.out**

```
...
println(...)
{ udhëzimet për të shkruar tekst }
```

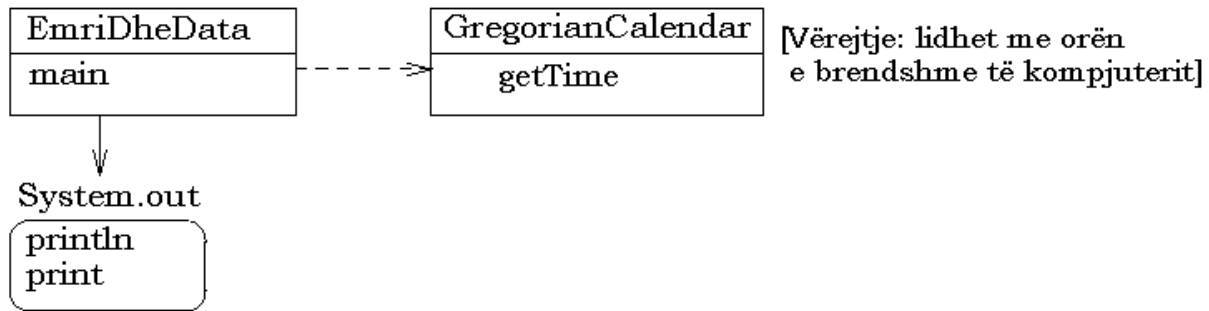
dhe objekti `Emril` lajmëron se ka përfunduar.

## 2.4 Si një Objekt e Konstrukton një Tjetër

1. Shkruaje programin që e afishon emrin tënd me datën dhe kohën. Kompajloje dhe ekzekutoje. Ekzekutoje përsëri. Krahase të dhënat dalëse nga dy ekzekutimet.

Përgjigje:

```
import java.util.*;
/** EmriDheData e afishon emrin tim dhe datën e kohën e sakte. */
public class EmriDheData
{ public static void main(String[] args)
  { System.out.print("Filan Fisteku --- ");
    // Urdheri vijues e krijon një objekt:
    GregorianCalendar c = new GregorianCalendar();
    System.out.println(c.getTime()); // kerko kohen nga c dhe afisho pergjigjen
    System.out.println();
    System.out.println("Fund.");
  }
}
```



*Diagrami klasor i aplikacionit që e afishon emrin dhe kohën*

Figura e ka një arkitekturë interesante: `EmriDheData` është kontrolluesi, sepse kontrollon se çka do të ndodhë; `System.out` është pamja dalëse, sepse ia prezenton përdoruesit "pamjen" e programit; dhe `GregorianCalendar` është modeli, sepse e "modelon" orën e kompjuterit. Ky është shembulli i parë i thjeshtë i arkitekturës Model-View-Controller (në të cilën do të mbështetemi gjatë kursit).

## (2.5 Përmirësimi i Mesazheve të Gabimeve të Kompilatorit)

## (2.6 Përmbledhje)

## 2.7 Ushtrime të Programimit

1. Shkruaje një Java-program që e afishon emrin tënd dhe adresën postare në tre ose më shumë rreshta. Pastaj, përdore një objekt `GregorianCalendar` për afishim të datës dhe kohës para emrit tënd.

2. Shkruaje një Java-program që duket sikur s'bën asgjë.

3. Krijë dy objekte `GregorianCalendar` duke i futur këta dy urdhëra,

```
System.out.println( new GregorianCalendar().getTime() );
System.out.println( new GregorianCalendar().getTime() );
```

në metodën `main`. Çka ndodh kur e ekzekuton programin?

4. Ndryshoje programin nga ushtrimi i kaluar si vijon: Fute ndërmjet dy urdhërave këtë Java-instruksion:

```
try { Thread.sleep(5000); }
catch (InterruptedException e) { }
```

Nuk do ta shqyrtojmë këtë urdhër kompleks - thënë thjesht, ai shkakton që metoda `main` të vonohet për 5000 milisekonda (dmth., 5 sekonda). Kompajloje dhe ekzekutoje programin. Çka ndodh?

5. Shkruaje Java-programin më të vogël (me numrin më të vogël të karakterëve) që mundesh. A ekziston programi më i madh i mundshëm (me numrin më të madh të karakterëve)?

6. Shkruaje një Java-program që e vizaton këtë vizatim në ekran:

```
  /\
 /  \
----
|  -  |
|  |  |
```

## **III Aritmetika dhe Ndryshoret**

### **3.1 Aritmetika e Numrave të Plotë**

Llogarite secilën nga këto shprehje ashtu si do ta bënte kompjuteri; trego të gjithë hapat e llogaritjes. Pasi ta kesh kryer llogaritjen, shkruaje secilën nga to përbrenda një Java-aplikacioni, si ky:

```
public class Test
{ public static void main(String[] args)
  { System.out.println(1 + ((2 - 4) * 3) + 5); }
}
```

1.  $6 * ((-2 + 3) * (2 - 1))$
2.  $6 * (-2 + 3) * (2 - 1)$
3.  $6 * -2 + 3 * (2 - 1)$
4.  $6 * -2 + 3 * 2 - 1$

Përgjigje:

1.  $6 * ((-2 + 3) * (2 - 1))$   
=>  $6 * (1 * (2 - 1))$   
=>  $6 * (1 * 1)$   
=>  $6 * 1$   
=> 6
2.  $6 * (-2 + 3) * (2 - 1)$   
=>  $6 * 1 * (2 - 1)$   
=>  $6 * 1 * 1$   
=>  $6 * 1$   
=> 6
3.  $6 * -2 + 3 * (2 - 1)$   
=>  $6 * -2 + 3 * 1$   
=>  $-12 + 3 * 1$   
=>  $-12 + 3$   
=> -9
4.  $6 * -2 + 3 * 2 - 1$   
=>  $-12 + 3 * 2 - 1$   
=>  $-12 + 6 - 1$   
=>  $-6 - 1$   
=> -7



## 3.2. Madhësitë e Emërtuara: Ndryshoret

1. Ndryshoje aplikacionin vijues ashtu që ta llogarisë vlerën e 3 njëzetqindarkave (20-centëshave) e 12 pesëqindarkave (5-centëshave); kompajloje dhe ekzekutoje.

```
/** Total - e llogarit sasinë e parave të imta që i kam */
public class Total
{ public static void main(String[] args)
  { int gjysma = 4;
    int njezeta = 2;
    int dhjeta = 0;
    int pesa = 1;
    int qindarka = 6;
    System.out.println("Për këto sasi metelikësh (monedhash):");
    System.out.print("Gjysma = ");
    System.out.println(gjysma);
    System.out.print("Njëzetqindarka = ");
    System.out.println(njezeta);
    System.out.print("Dhjetëqindarka = ");
    System.out.println(dhjeta);
    System.out.print("Pesëqindarka = ");
    System.out.println(pesa);
    System.out.print("Qindarka = ");
    System.out.println(qindarka);
    System.out.print("Shuma totale është ");
    System.out.println( (gjysma * 50) + (njezeta * 20)
      + (dhjeta * 10) + (pesa * 5) + (qindarka * 1) );
  }
}
```

### Përgjigje:

```
/** Total2 - e llogarit sasinë e parave të imta që i kam */
public class Total2
{ public static void main(String[] args)
  { int gjysma = 0;
    int njezeta = 3;
    int dhjeta = 0;
    int pesa = 12;
    int qindarka = 0;
    System.out.println("Për këto sasi metelikësh (monedhash):");
    System.out.print("Gjysma = ");
    System.out.println(gjysma);
    System.out.print("Njëzetqindarka = ");
    System.out.println(njezeta);
    System.out.print("Dhjetëqindarka = ");
    System.out.println(dhjeta);
    System.out.print("Pesëqindarka = ");
    System.out.println(pesa);
    System.out.print("Qindarka = ");
    System.out.println(qindarka);
    System.out.print("Shuma totale është ");
    System.out.println( (gjysma * 50) + (njezeta * 20)
      + (dhjeta * 10) + (pesa * 5) + (qindarka * 1) );
  }
}
```

**2. Ndryshoje (modifikoje) aplikacionin në figurën 3.1. ashtu që ta paraqesë vlerën e secilit grup të monedhës. Provoje aplikacionin për 5 njëzetqindarka, 1 pesëqindarkë dhe 1 qindarkë. Aplikacioni duhet të kthejë:**

Për këto sasi metelikesh (monedhash):  
 Njëzetqindarka = 5, që bëjnë 100 centë  
 Dhjetëqindarka = 0, që bëjnë 0 centë  
 Pesëqindarka = 1, që bëjnë 5 centë  
 Qindarka = 1, që bëjnë 1 centë  
 Shuma totale është 106 centë

### Përgjigje:

```
/** Total3 - e llogarit sasine e parave te imta qe i kam */
public class Total3
{ public static void main(String[] args)
  { int gjysma = 0;
    int njezeta = 5;
    int dhjeta = 0;
    int pesa = 1;
    int qindarka = 1;
    System.out.println("Per këto sasi metelikesh (monedhash):");
    System.out.print("Gjysma = ");
    System.out.println(gjysma + ", qe bejne " + gjysma*50 + " cente");
    System.out.print("Njëzetqindarka = ");
    System.out.println(njezeta + ", qe bejne " + njezeta*20 + " cente");
    System.out.print("Dhjetëqindarka = ");
    System.out.println(dhjeta + ", qe bejne " + dhjeta*10 + " cente");
    System.out.print("Pesëqindarka = ");
    System.out.println(pesa + ", qe bejne " + pesa*5 + " cente");
    System.out.print("Qindarka = ");
    System.out.println(qindarka + ", qe bejne " + qindarka*1 + " cente");
    System.out.print("Shuma totale eshte ");
    System.out.println( (gjysma * 50) + (njezeta * 20)
      + (dhjeta * 10) + (pesa * 5) + (qindarka * 1) );
  }
}
```

**3. Ndryshoje (modifikoje) programin Total4 ashtu që ta shfaqë daljen e vet në format dhjetor (decimal), p.sh.**

Shuma totale është 2.51 euro.

Shpjego se çka ndodh kur aplikacioni i ndryshuar përdoret me vlerat e dhëna në ushtrimin e kaluar. (Vështirësinë do ta rregullojmë në pjesën "Konvertimi ndërmjet Stringjeve e Numrave dhe Formatizimi").

```
/** Total4 e llogarit vleren e parave te imta qe i kam */
public class Total4
{ public static void main(String[] args)
  { int njezeta = 5;
    int dhjeta = 2;
    int pesa = 0;
    int qindarka = 6;
    System.out.println("Per keto vlera te metelikeve:");
    System.out.println("Njezetqindarka = " + njezeta);
    System.out.println("Dhjeteqindarka = " + dhjeta);
    System.out.println("Peseqindarka = " + pesa);
    System.out.println("Qindarka = " + qindarka);
    int krejt = (njezeta * 20) + (dhjeta * 10)
               + (pesa * 5) + (qindarka * 1);
    System.out.println("Shuma totale eshte " + (krejt / 100) + " euro e "
                      + (krejt % 100) + " cente");
  }
}
```

#### Përgjigje:

```
/** Total5 e llogarit vleren e parave te imta qe i kam */
public class Total5
{ public static void main(String[] args)
  { int njezeta = 5;
    int dhjeta = 2;
    int pesa = 0;
    int qindarka = 6;
    System.out.println("Per keto vlera te metelikeve:");
    System.out.println("Njezetqindarka = " + njezeta);
    System.out.println("Dhjeteqindarka = " + dhjeta);
    System.out.println("Peseqindarka = " + pesa);
    System.out.println("Qindarka = " + qindarka);
    int krejt = (njezeta * 20) + (dhjeta * 10)
               + (pesa * 5) + (qindarka * 1);
    System.out.println("Shuma totale eshte " + (krejt / 100) + "."
                      + (krejt % 100) + " euro");
  }
}
```

Kur i përdorim vlerat e ushtrimit të kaluar (5 njëzetqindarka, 5 pesëqindarka e 1 qindarkë) aplikacioni afishon:

Shuma totale eshte 1.6 euro

që në të vërtetë është 1 euro e 60 centë, në vend të 1 euro e 6 centë.

4. Sa për ushtrim, shkruaje një aplikacion në Java që afishon tabelë të shumëzimit të numrave ndërmjet 1 dhe 3, dmth., 1\*1, 1\*2, e kështu me radhë, deri në 3\*3.

### 3.2.1 Ndryshoret Mund të Ndryshojnë: Ndarja e Vlerës

1. Rishkruaje programin `BeriTIIma` ashtu që të bëjë para të imta nga 5-Euro, 1-Euro, 50-centë, 20-centë, 10-centë, 5-centë dhe 1-cent.

```
/** BeriTIIma i llogarit paratë e imta për sasi të ndryshoret
 * eurot dhe centet. */
public class BeriTIIma
{ public static void main(String[] args)
  { int eurot = 3;
    int centet = 46;
    int parate = (eurot * 100) + centet;
    System.out.println("20-centëshe = " + (parate / 20));
    parate = parate % 20;
    System.out.println("10-centëshe = " + (parate / 10));
    parate = parate % 10;
    System.out.println("5-centëshe = " + (parate / 5));
    parate = parate % 5;
    System.out.println("1-centëshe = " + parate);
  }
}
```

#### Përgjigje:

```
/** BeriTIIma5 i llogarit paratë e imta për sasi të ndryshoret
 * eurot dhe centet. */
public class BeriTIIma5
{ public static void main(String[] args)
  { int eurot = 33;
    int centet = 16;
    int parate = (eurot * 100) + centet;
    System.out.println("Nga 5-euro = " + (parate / 500));
    parate = parate % 500;
    System.out.println("Nga 1-euro = " + (parate / 100));
    parate = parate % 100;
    System.out.println("50-centëshe = " + (parate / 50));
    parate = parate % 50;
    System.out.println("20-centëshe = " + (parate / 20));
    parate = parate % 20;
    System.out.println("10-centëshe = " + (parate / 10));
    parate = parate % 10;
    System.out.println("5-centëshe = " + (parate / 5));
    parate = parate % 5;
    System.out.println("1-centëshe = " + parate);
  }
}
```

2. Cilat janë rezultatet e këtyre shprehjeve?

6/4; 6%4; 7/4; 7%4; 8/4; 8%4; 6/-4; -6%4; 6%-4.

Përgjigje.

$6/4 = 1;$   
 $6\%4 = 2;$   
 $7/4 = 1;$   
 $7\%4 = 3;$   
 $8/4 = 2;$   
 $8\%4 = 0;$   
 $6/-4 = -1;$   
 $-6\%4 = -2;$   
 $6\%-4 = 2;$

```
public class TestoVeprimet
{ public static void main(String[] args)
  { System.out.println(6/4);
    System.out.println(6%4);
    System.out.println(7/4);
    System.out.println(7%4);
    System.out.println(8/4);
    System.out.println(8%4);
    System.out.println(6/-4);
    System.out.println(-6%4);
    System.out.println(6%-4);
  }
}
```

**3.** Përdore algjebren për të shpjeguar se pse ndarja e vlerës,  $\text{paratë} = \text{paratë} \% 20$  në programin `MakeChange` e zbret në mënyrë korrekte numrin e 20-centëshave nga sasia fillestare e ndryshores `paratë`.

Përgjigje. Nga algjebra e dimë se i pjesëtueshmi mund të shkruhet si 'prodhimi i pjesëtuesit me herësin e plotë, plus mbetja'. Dmth.

$\text{paratë} == 20 * (\text{paratë} / 20) + (\text{paratë} \% 20)$

Prandaj, kur ndryshores `paratë` ia ndajmë vlerën  $\text{paratë} = \text{paratë} \% 20$ , ne në fakt ia heqim numrin e 20-centëshave nga sasia e mëparshme.

**4.** Shkruaje këtë varg të urdhërave në Java:

- Një ndryshore, `paratë_e_mija`, inicializohet me 12.
- `paratë_e_mija` zvogëlohet për 5.
- `paratë_e_mija` dyfishohet.
- `paratë_e_mija` resetohet, rivendoset në 1.
- Vlera e ndryshores `paratë_e_mija` i dërgohet në një `println`-mesazh objektit `System.out`.

Përgjigje.

```
int paratë_e_mija = 12;
paratë_e_mija = paratë_e_mija - 5;
paratë_e_mija = 2 * paratë_e_mija;
paratë_e_mija = 1;
System.out.println(paratë_e_mija);
```

**5. Shkruaje një shteg të ekzekutimit të këtij aplikacioni:**

```
public class Ushtrimi3
{ public static void main(String[] args)
  { int x = 12;
    int y = x + 1;
    x = x + y;
    y = x;
    System.out.println(x + " është i barabartë me " + y);
  }
}
```

Përgjigje.

```
Ushtrimi3
-----
|  main
|  { >int x = 12;
|    int y = x + 1;
|    x = x + y;
|    y = x;
|    System.out.println(x + " është i barabartë me " + y);
|  }
```

```
Ushtrimi3
-----
|  main      ----
|  { int x ==| 12 |
|            ----
|    >int y = x + 1;
|    x = x + y;
|    y = x;
|    System.out.println(x + " është i barabartë me " + y);
|  }
```

```
Ushtrimi3
-----
|  main      ----
|  { int x ==| 12 |
|            ----
|    >int y = 12 + 1;
|    x = x + y;
|    y = x;
|    System.out.println(x + " është i barabartë me " + y);
|  }
```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 12 |
|              ----
|    int y ==| 13 |
|              ----
|    >x = x + y;
|    y = x;
|    System.out.println(x + " është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 12 |
|              ----
|    int y ==| 13 |
|              ----
|    >x = 12 + 13;
|    y = x;
|    System.out.println(x + " është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 13 |
|              ----
|    ...
|    >y = x;
|    System.out.println(x + " është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 13 |
|              ----
|    ...
|    >y = 25;
|    System.out.println(x + " është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 25 |
|              ----
|    ...
|    >System.out.println(x + " është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 25 |
|              ----
|
|    ...
|    >System.out.println(25 + " është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 25 |
|              ----
|
|    ...
|    >System.out.println("25 është i barabartë me " + y);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 25 |
|              ----
|
|    ...
|    >System.out.println("25 është i barabartë me " + 25);
|  }

```

```

Ushtrimi3
-----
|  main      ----
|  { int x ==| 25 |
|              ----
|    int y ==| 25 |
|              ----
|
|    ...
|    >System.out.println("25 është i barabartë me 25");
|  }

```

dhe fraza, 25 është i barabartë me 25, paraqitet në konsolë (dritaren komanduese).



### 3.3 Aritmetika me Thyesa: Tipi double

1. Rishkruaje programin, CelsiusNeFahrenheit, në programin, FahrenheitNeCelsius, që e konverton një vlerë double të gradëve Fahrenheit në një vlerë double të gradëve Celsius dhe i afishon të dy vlerat.

```
/** CelsiusNeFahrenheit e konverton
 * nje vlere Celsius ne Fahrenheit. */
public class CelsiusNeFahrenheit
{ public static void main(String[] args)
  { int c = 22; // grade Celsius
    double f = ((9.0/5.0) * c) + 32;
    System.out.println("Per " + c + " grade Celsius,");
    System.out.println(f + " grade Fahrenheit");
  }
}
```

Përgjigje:

```
/** FahrenheitNeCelsius e konverton
 * nje vlere Fahrenheit ne Celsius */
public class FahrenheitNeCelsius
{ public static void main(String[] args)
  { double f = 27.5; // grade Fahrenheit
    double c = (f - 32) * 5 / 9;
    System.out.println("Per " + f + " grade Fahrenheit,");
    System.out.println(c + " grade Celsius");
  }
}
```

2. Dihet se një kilometër është 0.62137 e milës. Rishkruaje programin, CelsiusNeFahrenheit, në programin, KilometraNeMila, që e merr si hyrje një numër të plotë të kilometrave dhe e afishon si dalje një vlerë double të milave korresponduese.

3. Llogariti rezultatet e këtyre shprehjeve; ki kujdes rreth konvertimeve (casts) të numrave të plotë në double gjatë llogaritjeve:

- a)  $(5.3 + 7) / 2.0$
- b)  $(5.3 + 7) / 2$
- c)  $5.3 + (7 / 2)$
- ç)  $(1.0 + 2) + ((3 \% 4) / 5.0)$

Përgjigje:

- a)  $(5.3 + 7) / 2.0 == 12.3 / 2.0 == 6.15$
- b)  $(5.3 + 7) / 2 == 12.3 / 2 == 6.15$
- c)  $5.3 + (7 / 2) == 5.3 + 3 == 8.3$
- ç)  $(1.0 + 2) + ((3 \% 4) / 5.0) == 3.0 + (3 / 5.0) == 3.0 + 0.6 == 3.6$

### 3.4 Tipi boolean

1. Ndërtoje një Java-aplikacion të plotë që e përdor teknikën në këtë paragraf (në libër) për ta përcaktuar se a është një temperaturë e dhënë në Celsius më e ngrohtë se një temperaturë e dhënë në Fahrenheit. Testoje aplikacionin me 40 gradë Celsius dhe 40 gradë Fahrenheit.

Përgjigje:

```
public class KrahasoTemp
{
    public static void main(String[] args)
    {
        double C = 40;
        double F = 40;
        double C_e_konvertuar = ((9.0/5.0) * C) + 32;
        System.out.print(C + " Celsius më e ngrohtë se " + F + " Fahrenheit? ");
        System.out.println(C_e_konvertuar > F);
        // ose, duke e përdorur operatorin ( ? : )
        // System.out.println((C_e_konvertuar > F ? "e vërtetë" : "e pavërtetë"));
    }
}
```

2. Ndërtoje një Java-aplikacion të cilit i tregohet një sasi e 10-centëshave dhe një sasi e 5-centëshave dhe ai e afishon se a është sasia e 10-centëshave më pak e vlefshme sesa sasia e 5-centëshave. Testoje aplikacionin me 4 dhjetë-centësha dhe 6 pesë-centësha. (Për punë të pavarur)

3. Llogarite rezultatin e këtyre shprehjeve:

- a)  $(3 * 2) \geq (-9 - 1)$
- b)  $3 * 2 \neq 5.5 + 0.4 + 0.1$

Përgjigje:

- a)  $(3 * 2) \geq (-9 - 1)$   
=>  $6 \geq (-9 - 1)$   
=>  $6 \geq -10$   
=> true
- b)  $3 * 2 \neq 5.5 + 0.4 + 0.1$   
=>  $6 \neq 5.5 + 0.4 + 0.1$   
=>  $6 \neq 5.9 + 0.1$   
=>  $6 \neq 6.0$   
=>  $6.0 \neq 6.0$   
=> false

### 3.5 (Prioriteti) Përparësia e Operatorëve

1. Llogariti përgjigjet për secilën nga këto shprehje:

a)  $6 * -2 + 3 / 2 - 1$

b)  $5.3 + 7 / 2 + 0.1$

c)  $3 * 2 \% 4 / 5.0 * 2 * 3$

Përgjigje:

a)  $6 * -2 + 3 / 2 - 1$   
 $\Rightarrow -12 + 3 / 2 - 1$   
 $\Rightarrow -12 + 1.5 - 1$   
 $\Rightarrow -11.5 - 1$   
 $\Rightarrow -12.5$

b)  $5.3 + 7 / 2 + 0.1$   
 $\Rightarrow 5.3 + 3.5 + 0.1$   
 $\Rightarrow 8.8 + 0.1$   
 $\Rightarrow 8.9$

c)  $3 * 2 \% 4 / 5.0 * 2 * 3$   
 $\Rightarrow 6 \% 4 / 5.0 * 2 * 3$   
 $\Rightarrow 2 / 5.0 * 2 * 3$   
 $\Rightarrow 0.4 * 2 * 3$   
 $\Rightarrow 0.8 * 3$   
 $\Rightarrow 2.4$

(Java e jep 2.4000000000000004 si rezultat në llogarinë e fundit për shkak të gabimeve të rrumbullakësimit.)

2. Një mënyrë tjetër për ta kuptuar prioritetin (përparësinë) e operatorëve është se prioritetet janë rregulla për t'i futur kllapat në shprehje. Për shembull, prioritetet na tregojnë se shprehja  $1 + 2.0 * 3$  do të duhej të kllapëzohet si

$1 + (2.0 * 3)$ . Ngjashëm,  $5 - 3 / 2 * 4 + 6 * -2 / -3 + 1.5$  kllapëzohet si  $((5 - ((3 / 2) * 4)) + ((6 * -2) / -3)) + 1.5$ .

Për secilën nga shprehjet në ushtrimin e kaluar, shkruaje shprehjen me kllapat e veta.

Përgjigje:

- a)  $((6 * -2) + (3 / 2)) - 1$
- b)  $(5.3 + (7 / 2)) + 0.1$
- c)  $(( (3 * 2) \% 4) / 5.0) * 2 * 3$

### 3.6 Stringjet, Karakterët dhe Veprimet e tyre

1. Llogaritni rezultatet e këtyre shprehjeve:

- a)  $2 + ("a" + " ") + "bc"$
- b)  $1 + "" + 2 + 3$
- c)  $1 + 2 + "" + 3$
- ç)  $1 + 2 + 3 + ""$
- d)  $1 + "" + (2 + 3)$

Përgjigje:

- a)  $2 + ("a" + " ") + "bc" == 2 + "a " + "bc" == "2a " + "bc" == "2a bc"$
- b)  $1 + "" + 2 + 3 == "1" + 2 + 3 == "12" + 3 == "123"$
- c)  $1 + 2 + "" + 3 == 3 + "" + 3 == "3" + 3 == "33"$
- ç)  $1 + 2 + 3 + "" == 3 + 3 + "" == 6 + "" == "6"$
- d)  $1 + "" + (2 + 3) == 1 + "" + 5 == "1" + 5 == "15"$

2. Qe dy stringje:

```
String t = "abc ";  
String u = "ab";
```

Çka afishohet me këta urdhëra?

- a) `System.out.println(t.equals(u));`
- b) `System.out.println(u.charAt(1) == t.charAt(1));`
- c) `System.out.println(t.length() - u.length());`
- ç) `System.out.println(u + 'c');`
- d) `System.out.println(t.trim());`
- e) `System.out.println(t.toUpperCase());`

Përgjigje:

- a) `false`
- b) `true`
- c) `2`
- ç) `abc`
- d) `abc`
- e) `ABC`

### 3.7 Kontrollimi i Tipeve të Të Dhënave

Po e zëmë se ti je Java-kompajleri.

**1. Pa i llogaritur përgjigjet në këto shprehje, parashikoje tipin e secilit rezultat të shprehjes (ose a ka gabim tipi):**

- a) `5.3 + (7 / 2)`
- b) `3 * (1 == 2)`
- c) `1 < 2 < 3`
- ç) `"a " + 1 + 2`
- d) `("a " + 1) * 2`

Përgjigje:

- a) `double + (int / int) => double + int => double`
- b) `int * boolean => gabim tipi`
- c) `boolean < int => gabim tipi`
- ç) `String + int + int => String + int => String`
- d) `(String + int) * int => String * int => gabim tipi`

**2. Cili nga këta urdhëra përmban gabim të tipit?**

```
int x = 3.5;
double d = 2;
String s = d;
d = (d > 0.5)
System.out.println(s * 3);
```

Përgjigje:

<code>int x = 3.5;</code>	- përmban gabim - humbet preciziteti
<code>double d = 2;</code>	- nuk përmban gabim ( <code>int =&gt; double</code> )
<code>String s = d;</code>	- gabim i tipit
<code>d = (d &gt; 0.5)</code>	- gabim i tipit
<code>System.out.println(s * 3);</code>	- gabim i tipit ( <code>String * int</code> )

## 3. 8 Hyrja (Input) Përmes Argumenteve Programore

### (3.8.1 Konvertimi ndërmjet Stringjeve e Numrave dhe Formatizimi)

### 3.8.2 Konvertimi i Temperaturës me Hyrje (Input)

1. Kompajloje dhe ekzekutoje kodin `FahrenheitNeCelsius` tri herë me argumentet programore 20, 22 dhe -10, përkatësisht.

```
import java.text.*;
/** FahrenheitNeCelsius2 konverton vlerën e temperatures */
public class FahrenheitNeCelsius2
{ public static void main(String[] args)
  { int f = new Integer(args[0]).intValue();
    double c = (5.0/9.0) * (f - 32);
    System.out.println("Për gradë Fahrenheit: " + f + ",");
    DecimalFormat formatuesi = new DecimalFormat("0.0");
    System.out.println("gradë Celsius: " + formatuesi.format(c));
  }
}
```

Përgjigje: Në vijën komanduese:

```
java FahrenheitNeCelsius 20
java FahrenheitNeCelsius 22
java FahrenheitNeCelsius -10
```

Në IDE - jepen argumentet në vendin përkatës. (P.sh. në jGRASP: 'Run - Run Arguments' e hap vendin e gatshëm ku mund t'i shkruajmë argumentet programore)

2. Ripunoje programin që i bën t'ima paratë, `BeriTIma`, ashtu që eurot dhe centët jipen si argumente programore, p.sh.,

```
java BeriTIma 3 46
```

Përgjigje:

```

/** BeriTima i llogarit paratë e imta për sasitë në ndryshoret
 * eurot dhe centet. */
public class BeriTimaArg
{ public static void main(String[] args)
  { int eurot = new Integer(args[0]).intValue();
    int centet = new Integer(args[1]).intValue();
    int parate = (eurot * 100) + centet;
    System.out.println("20-centëshe = " + (parate / 20));
    parate = parate % 20;
    System.out.println("10-centëshe = " + (parate / 10));
    parate = parate % 10;
    System.out.println("5-centëshe = " + (parate / 5));
    parate = parate % 5;
    System.out.println("1-centëshe = " + parate);
  }
}

```

**3. Ripunoje njërin ose të dy programet: CelsiusNeFahrenheit dhe KilometraNeMila, që i ke shkruar si zgjidhje të ushtrimeve të kaluara ("Aritmetika me Thyesa: Tipi double"), ashtu që programet të përdorin argumente programore.**

Përgjigje:**Kodi original:**

```

/** CelsiusNeFahrenheit e konverton
 * nje vlere Celsius ne Fahrenheit. */
public class CelsiusNeFahrenheit
{ public static void main(String[] args)
  { int c = 22; // grade Celsius
    double f = ((9.0/5.0) * c) + 32;
    System.out.println("Per " + c + " grade Celsius,");
    System.out.println(f + " grade Fahrenheit");
  }
}

```

**Kodi i ripunuar:**

```

/** CelsiusNeFahrenheit e konverton
 * nje vlere Celsius ne Fahrenheit. */
public class CelsiusNeFahrenheitArg
{ public static void main(String[] args)
  { int c = new Integer(args[0]).intValue(); // grade Celsius
    double f = ((9.0/5.0) * c) + 32;
    System.out.println("Per " + c + " grade Celsius,");
    System.out.println(f + " grade Fahrenheit");
  }
}

```

**4. Të themi se një program, Test, është nisur me argumentet programore**

12 345 6 7.89. Shkruaji urdhërat inicializues që i lexojnë argumentet dhe e vendosin stringun "12" në një ndryshore, s; e vendosin numrin e plotë 6 në një ndryshore, i; e vendosin numrin double 7.89 në ndryshoren d.

**Përgjigje:**

```
String s = args[0];
int i = new Integer(args[2]).intValue();
double d = new Double(args[3]).doubleValue();
```

**5. Shkruaje një varg prej tre urdhërash që: e lexon një argument programor si numër të plotë: i, e lexon një argument programor si numër të plotë, j; e afishon true ose false, varësisht se a është vlera e i-së më e madhe se sa e j-së.****Përgjigje:**

```
int i = new Integer(args[0]).intValue();
int j = new Integer(args[1]).intValue();
System.out.println(i > j);
```

**6. Ushtrimi 3 në pjesën: "Madhësitë e Emërtuara: Ndryshoret", pati vështirësi për t'i afishuar si duhet përgjigjet kur sasia e afishuar e euro-centëve përmbante sasi më të vogël se 10 të centëve. Përdore klasën DecimalFormat për ta zgjidhur këtë problem. Provoje zgjidhjen tënde me hyrjen prej 5 njëzetqindarkash, një pesëqindarke dhe një qindarke.****Përgjigje:****Kodi i lartpërmendur:**

```
/** Total4 e llogarit vleren e parave te imta qe i kam */
public class Total4
{ public static void main(String[] args)
  { int njezeta = 5;
    int dhjeta = 2;
    int pesa = 0;
    int qindarka = 6;
    System.out.println("Per keto vlera te metelikeve:");
    System.out.println("Njezetqindarka = " + njezeta);
    System.out.println("Dhjeteqindarka = " + dhjeta);
    System.out.println("Peseqindarka = " + pesa);
    System.out.println("Qindarka = " + qindarka);
    int krejt = (njezeta * 20) + (dhjeta * 10)
               + (pesa * 5) + (qindarka * 1);
    System.out.println("Shuma totale eshte " + (krejt / 100) + "."
                      + (krejt % 100) + " euro");
  }
}
```



Kodi i rregulluar:

```
import java.text.*;
/** Total6 e llogarit vleren e parave te imta qe i kam
 * afishon me formatues. */
public class Total6
{ public static void main(String[] args)
  { int njezeta = 5;
    int dhjeta = 0;
    int pesa = 1;
    int qindarka = 1;
    System.out.println("Per keto vlera te metelikeve:");
    System.out.println("Njezetqindarka = " + njezeta);
    System.out.println("Dhjeteqindarka = " + dhjeta);
    System.out.println("Peseqindarka = " + pesa);
    System.out.println("Qindarka = " + qindarka);
    int krejt = (njezeta * 20) + (dhjeta * 10)
               + (pesa * 5) + (qindarka * 1);
    DecimalFormat formatues = new DecimalFormat("0.00");
    System.out.println("Shuma totale eshte "
      + formatues.format(krejt/100.0) + " euro");
  }
}
```

### **(3.9 Diagnostifikimi i Gabimeve në Shprehje)**

### **(3.10 Fjalët Kyçe dhe Identifikatorët)**

## **IV Hyrja (Input), Dalja (Output) dhe Gjendja**

### **4.1. Hyrja (Input) Interaktive**

1. Shkruaje një varg urdhërash që e përdorin `JOptionPane` për të lexuar në mënyrë interaktive emrin e një personi (string) dhe moshën e vet (numër i plotë).

Përgjigje:

```
import javax.swing.*; // ne fillim te kodit

String emri = JOptionPane.showInputDialog("Emri:");
String m = JOptionPane.showInputDialog("Moshë:");
int mosha = new Integer(m).intValue();
```

2. Shkruaje një aplikacion: `class Test`, që e lexon në mënyrë interaktive një numër të plotë dhe e afishon në dritaren komanduese rrenjën katrore të tij. Testoje këtë program: çka ndodh kur ti, si përdorues, e keqpërdor duke e shkruar emrin tënd në vend të numrit? Po kur si hyrje e shtyp vetëm OK? Kur refuzon të shkruash çkado dhe shkon në drekë?

Përgjigje:

```
import javax.swing.*;

/** Test - e lexon ne menyre interaktive nje numer te plote
 * dhe e afishon ne dritaren komanduese rrenjen katrore te tij */
public class Test
{
    public static void main(String[] args)
    {
        String n = JOptionPane.showInputDialog("Numri i plote:");
        int numri = new Integer(n).intValue();
        System.out.println(Math.sqrt(numri));
    }
}
```

**3. Rregulloje programin Total, ashtu që të përdorë hyrje interaktive.**

```

/** Total - e llogarit sasinë e parave të imta që i kam */
public class Total
{ public static void main(String[] args)
  { int gjysma = 0;
    int njezeta = 3;
    int dhjeta = 0;
    int pesa = 12;
    int qindarka = 0;
    System.out.println("Për këto sasi metelikësh (monedhash):");
    System.out.print("Gjysma = ");
    System.out.println(gjysma);
    System.out.print("Njëzetqindarka = ");
    System.out.println(njezeta);
    System.out.print("Dhjetëqindarka = ");
    System.out.println(dhjeta);
    System.out.print("Pesëqindarka = ");
    System.out.println(pesa);
    System.out.print("Qindarka = ");
    System.out.println(qindarka);
    System.out.print("Shuma totale është ");
    System.out.println( (gjysma * 50) + (njezeta * 20)
      + (dhjeta * 10) + (pesa * 5) + (qindarka * 1) );
  }
}

```

**Përgjigje:**

```

import javax.swing.*;
/** TotalInteraktiv - e llogarit sasinë e parave të imta që i kam
 * duke përdorur hyrje interaktive. */
public class TotalInteraktiv
{ public static void main(String[] args)
  { String gj = JOptionPane.showInputDialog("Numri i gjysmave:");
    int gjysma = new Integer(gj).intValue();
    String nj = JOptionPane.showInputDialog("Numri i njezetave:");
    int njezeta = new Integer(nj).intValue();
    String dh = JOptionPane.showInputDialog("Numri i dhjetave:");
    int dhjeta = new Integer(dh).intValue();
    String p = JOptionPane.showInputDialog("Numri i pesave:");
    int pesa = new Integer(p).intValue();
    String q = JOptionPane.showInputDialog("Numri i qindarkave:");
    int qindarka = new Integer(q).intValue();
    System.out.println("Për këto sasi metelikësh (monedhash):");
    System.out.print("Gjysma = ");
    System.out.println(gjysma);
    System.out.print("Njëzetqindarka = ");
    System.out.println(njezeta);
    System.out.print("Dhjetëqindarka = ");
    System.out.println(dhjeta);
    System.out.print("Pesëqindarka = ");
    System.out.println(pesa);
    System.out.print("Qindarka = ");
    System.out.println(qindarka);
    System.out.print("Shuma totale është ");
    System.out.println( (gjysma * 50) + (njezeta * 20)
      + (dhjeta * 10) + (pesa * 5) + (qindarka * 1) );
  }
}

```

## 4.2 Dalja (Output) Grafike

### (4.2.1 Panelat dhe Kornizat e tyre)

### 4.2.2 Përshtatja e Panelave me Trashëgimi

Qe kodi për klasët `PaneliIm` dhe `KornizeTest3`:

(Numrat majtas janë vetëm për t'i numëruar rreshtat, dmth. nuk janë pjesë e kodit.)

```
// PaneliIm.java
1 import java.awt.*;
2 import javax.swing.*;
3 /** PaneliIm e krijon nje panel me ngjyra */
4 public class PaneliIm extends JPanel
5 { /** paintComponent e mbush panelin me gjerat qe do te paraqiten
6   * @param g - "penda" grafike qe i vizaton gjerat */
7   public void paintComponent(Graphics g)
8   { int gjeresia_e_kornizes = 300;
9     int lartesia_e_kornizes = 200;
10    g.setColor(Color.white);
11    g.fillRect(0, 0, gjeresia_e_kornizes, lartesia_e_kornizes);
12    g.setColor(Color.red);
13    int skaji_majtas = 105;
14    int lart = 70;
15    // vizatoje nje drejtkendesh
16    int gjeresia = 90;
17    int lartesia = 60;
18    g.drawRect(skaji_majtas, lart, gjeresia, lartesia);
19    // vizatoje nje rreth te mbushur
20    int diametri = 40;
21    g.fillOval(skaji_majtas + gjeresia - diametri, lart, diametri, diametri);
22  }
23 }

// KornizeTest3.java
1 import javax.swing.*;
2 import java.awt.*;
3 /** KornizeTest3 e paraqet nje dritare grafike me ngjyra */
4 public class KornizeTest3
5 { public static void main(String[] args)
6   { JFrame korniza_ime = new JFrame();
7     // fute nje panel te ri ne kornize (frame):
8     korniza_ime.getContentPane().add(new PaneliIm());
9     // vere titullin ne kulmin e kornizes:
10    korniza_ime.setTitle("VizatuesIKornizesSime");
11    // nje menyre e lehte per ta ngjyrosur prapavijen e tere dritares
12    int gjeresia_e_kornizes = 300;
13    int lartesia_e_kornizes = 200;
14    korniza_ime.setSize(gjeresia_e_kornizes, lartesia_e_kornizes);
15    korniza_ime.setVisible(true);
16    System.out.println("Korniza u paraqit!");
17  }
18 }
```

Ndërroje klasën `PaneliIm` (dhe `KornizeTest3`) si vijon:

1. Rrethi i paraqitur vizatohet me ngjyrë të zezë. (Udhëzim: përdore `Color.black`.)
2. Madhësia e panelit është 400 me 400 piksela.
3. Prapavija e dritares është e verdhë. (Udhëzim përdore `Color.yellow`.)
4. Elipsa vizatohet me madhësi 80 me 40.
5. Drejtkëndëshi vizatohet në këndin e poshtëm të djathtë të dritares 300-me-200. (Udhëzim: Në këtë rast, këndi i sipërm i majtë i drejtkëndëshit do të ishte te piksel-pozita 210, 140.)

Përgjigje:

1. Pas rreshtit 19 e shtojmë rreshtin:

```
g.setColor(Color.black)
```

2. Rreshtat 8, 9:

```
8 { int gjeresia_e_kornizes = 400;  
9   int lartesia_e_kornizes = 400;
```

3. Rreshti 10 te `PaneliIm`:

```
g.setColor(Color.yellow);
```

- 4.

```
g.fillOval(skaji_majtas + gjeresia - diametri * 2, lart,  
           diametri * 2, diametri);
```

5. Rreshti 18 te `PaneliIm`:

```
g.drawRect(210, 140, gjeresia, lartesia);
```

Drejtkëndëshi nuk duket plotësisht në kornizë për shkak të hapësirës që e merr shiriti i titullit lart. Shiriti i merr disa piksela (në drejtimin vertikal), varësisht nga parametrat e rrethinës (Windows, Linux etj.).

## 4.3. Formati dhe Metodat për Vizatim

### 4.3.1 Konstruktor-metodat dhe Objekti this

Qe kodi i klasës VizatuesIOres:

```
import java.awt.*;
import javax.swing.*;
import java.util.*;
/** VizatuesIOres e vizaton nje ore ne nje panel. */
public class VizatuesIOres extends JPanel
{ public VizatuesIOres()
  { int gjeresia = 200; // gjeresia e ores
    // konstruktoje kornizen e ketij paneli:
    JFrame korniza_e_ores = new JFrame();
    // dhe fute kete panel (panelin _this_) ne kornizen e vet:
    korniza_e_ores.getContentPane().add(this);
    // tregoje kornizen:
    korniza_e_ores.setTitle("Ora");
    korniza_e_ores.setSize(gjeresia, gjeresia);
    korniza_e_ores.setVisible(true);
  }

  /** paintComponent e vizaton oren me kohen korrekte.
   * @param g - penda grafike qe e ben vizatimin */
  public void paintComponent(Graphics g)
  { int gjeresia = 200;
    g.setColor(Color.white);
    g.fillRect(0, 0, gjeresia, gjeresia); // ngjyrose prapavijen
    GregorianCalendar koha = new GregorianCalendar();
    int minutat = koha.get(Calendar.MINUTE);
    int oret = koha.get(Calendar.HOUR);
    int kendi_i_minutave = 90 - (minutat * 6);
    int kendi_i_oreve = 90 - oret * 30;
    // vizatoje oren si rreth te zi:
    int skaji_majtas = 50;
    int lart = 50;
    int diametri = 100;
    g.setColor(Color.black);
    g.drawOval(skaji_majtas, lart, diametri, diametri);
    // vizatoje akrepin e minutave te kuq, 10 piksela me te vogel,
    // me gjeresi prej 5 shkallesh
    g.setColor(Color.red);
    g.fillArc(skaji_majtas + 5, lart + 5, diametri - 10, diametri - 10,
              kendi_i_minutave, 5);
    // vizatoje akrepin e oreve te kalter, 50 piksela me te vogel,
    // me gjeresi prej -8 shkallesh
    g.setColor(Color.blue);
    g.fillArc(skaji_majtas + 25, lart + 25, diametri - 50, diametri - 50,
              kendi_i_oreve, -8);
  }

  /** Metoda main (e futur per testim) e monton oren ne kornizen e vet. */
  public static void main(String[] args)
  { new VizatuesIOres(); }
}
```

1. Modifikojë klasën `VizatuesIOres`, ashtu që akrepi i vet i orëve të jetë me gjatësi të njëjtë me të akrepit të minutave, por dyfish më i gjerë.

2. Përmirësoje pozitën e akrepit të orëve në klasën `VizatuesIOres` ashtu që të lëvizë gradualisht, e jo vetëm në orën e plotë. Përdore këtë formulë:

$$\text{kendi\_i\_orëve} = 90 - ((\text{orët} + (\text{minutat}/60.0)) * 30)$$

3. Bëje orën "digjitale" si edhe "analoge" duke e afishuar kohën e saktë, në shifra, në fundin e dritares.

#### Përgjigje:

Së pari, e ruajmë kodin si `VizatuesIOres2`. Pastaj i ndërrojmë të gjitha paraqitjet e emrit `VizatuesIOres` me `VizatuesIOres2` (p.sh. me një urdhër të zëvendësimit 'Replace'). Pastaj i bëjmë modifikimet tona. Kështu do t'i kemi të ruajtura të dy versionet e programit.

1.

```
// vizatoje akrepin e oreve te kalter,  
// me gjeresi prej -16 shkallesh  
g.setColor(Color.blue);  
g.fillArc(skaji_majtas + 5, lart + 5, diametri - 10, diametri - 10,  
         kendi_i_oreve, -16);
```

2. Ndërrohen rreshtat në metodën `paintComponent`:

```
GregorianCalendar koha = new GregorianCalendar();  
int minutat = koha.get(Calendar.MINUTE);  
int oret = koha.get(Calendar.HOUR);  
int kendi_i_minutave = 90 - (minutat * 6);  
int kendi_i_oreve = 90 - (int)((oret + (minutat/60.0)) * 30);
```

3. Në fund të metodës `paintComponent` shtohen rreshtat:

```
g.setColor(Color.black);  
g.drawString(koha.get(Calendar.HOUR) + ":" + koha.get(Calendar.MINUTE),  
            90, 170);
```

## 4.4 Objektet me Gjendje: Ndryshoret e Fushave

### 1. Qe kodi për ShembullIFushes:

```
import java.awt.*;
import javax.swing.*;
/** ShembullIFushes afishon ate se sa shpesh vizatohet ne ekran nje dritare */
public class ShembullIFushes extends JPanel
{ private int numri; // kjo ndryshore-fushe e ben numerimin se sa shpesh
                      // eshte vizatuar nja dritare; per kete arsye,
                      // vlere e saj duhet te jete gjithmone jonegative.

  /** ShembullIFushes e konstrukton dritaren. */
  public ShembullIFushes()
  { numri = 0; // dritarja nuk eshte vizatuar kurre
    // konstrukto kornizen e panelit dhe afishoje:
    JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    int lartesia = 200;
    korniza_ime.setSize((3*lartesia)/2, lartesia);
    korniza_ime.setVisible(true);
  }

  /** paintComponent e paraqet numrin e vizatimeve
   * @param g - penda grafike */
  public void paintComponent(Graphics g)
  { numri = numri + 1; // po vizatojme edhe nje here
    g.setColor(Color.black);
    int margjina = 25;
    int lartesia_e_rreshtit = 20;
    int rreshti_i_pare = 40;
    int rreshti_baze = rreshti_i_pare + (lartesia_e_rreshtit * numri);
    g.drawString("U vizatua " + numri + " here", margjina, rreshti_baze);
  }

  /** Metoda main i monton panelin dhe kornizen dhe i tregon ato. */
  public static void main(String[] args)
  { new ShembullIFushes(); }
}
```

### Kompajloje dhe ekzekutoje këtë aplikacion:

```
public class TregoDyShembujTeFushes
{ public static void main(String[] args)
  { ShembullIFushes a = new ShembullIFushes();
    ShembullIFushes b = new ShembullIFushes();
  }
}
```

Sa dritare i konstrukton? (Vërejtje: Ki kujdes kur të përgjigjesh - provo t'i lëvizësh dritaret që i sheh në ekran.) Kur e ikonifikon (minimizon) dhe e deikonifikon (rivendos) njërën nga dritaret ShembullIFushes, a ndikon kjo në paraqitjen e tjetrës? Në çfarë përfundimi arrin për gjendjet e brendshme të objekteve ShembullIFushes respektivisht?



Përgjigje:

- a) Aplikacioni i konstrukton dy dritare në të njëjtën pozitë me fillim në këndin e sipërm të majtë të ekranit.
- b) Ndikon në rastin kur dritarja që "fle" duhet të rivizatohet.
- c) Gjendjet e brendshme janë të pavarura nga njëra-tjetra. Secila i ka fushat e veta.

2. Përfundoje së shkruari klasën `ClockWriter2` (në libër) dhe bëje një eksperiment: Filloje aplikacionin, ikonifikoje dritaren që e gjeneron ai në ekran, dhe pas pesë minutash, rihape dritaren. Çfarë kohe sheh? Krahajoje këtë sjellje me atë të klasës `ClockWriter` në Figurën 17.

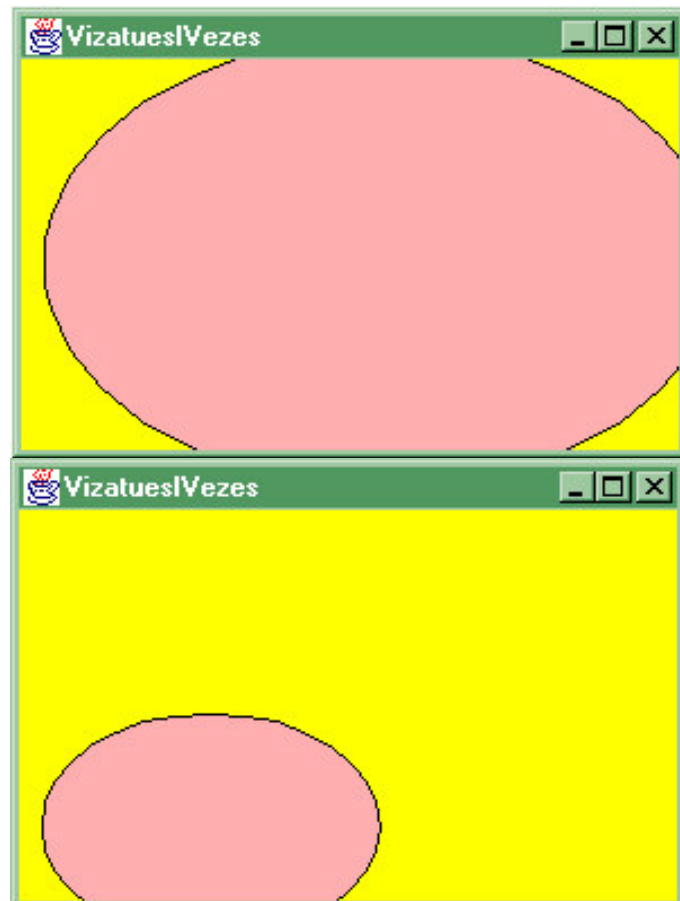
3. Qe një shembull komik që e ilustron përdorimin e fushave.

Shkruaje një klasë `VizatuesIVezes`, që e gjeneron një dritare grafike që e paraqet një vezë:

Sa herë që të ikonifikohet dhe të deikonifikohet dritarja `EggWriter`, veza rivizatohet sa gjysma e madhësisë së vet të mëparshme:

Hapja dhe mbyllja e shpeshtë e dritares bën që veza të tkurret në asgjë.

(Udhëzim: Përdore një fushë për ta mbajtur mend madhësinë e vezës që do të vizatohet; sa herë që metoda `paint` e rivizaton vezën, ajo poashtu e përgjysmon vlerën e fushës.)



Përgjigje:

```

import java.awt.*;
import javax.swing.*;
/** VizatuesIVezes e krijon nje dritare grafike qe e vizaton nje veze.
 * Sa here qe te rivizatohet dritarja, veza zvogelohet pergjysme */
public class VizatuesIVezes extends JPanel
{ private int GJERESIA = 300; // gjeresia e kornizes
  private int LARTESIA = 200; // lartesia e kornizes
  private Color NGJYRA_E_PRAPAVIJES = Color.yellow;
                                     // ngjyra per ta lyer kornizen

  private int gjeresia_e_vezes = 600; // duhet te jete jonegative
  private int lartesia_e_vezes = 400; // duhet te jete jonegative

  /** Konstruktori EggWriter e krijon dritaren dhe e ben te dukshme */
  public VizatuesIVezes()
  { JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("VizatuesIVezes");
    korniza_ime.setSize(GJERESIA, LARTESIA);
    korniza_ime.setVisible(true);
  }

  /** paintComponent e vizaton vezen.
   * @param g - "penda grafike" qe e vizaton vezen */
  public void paintComponent(Graphics g)
  { g.setColor(NGJYRA_E_PRAPAVIJES);
    g.fillRect(0, 0, GJERESIA, LARTESIA); // "fshije" tere prapavijen

    int kufiri_majtas = 10; // kufiri i majte: pozita horizontale e vezes
    int vija_baze = 190;    // vija-baze: ku te leshohet veza
    g.setColor(Color.pink); // ngjyra e vezes
    g.fillOval(kufiri_majtas, vija_baze - lartesia_e_vezes,
               gjeresia_e_vezes, lartesia_e_vezes);
    g.setColor(Color.black); // ngjyra e vijes kufizuese te vezes
    g.drawOval(kufiri_majtas, vija_baze - lartesia_e_vezes,
               gjeresia_e_vezes, lartesia_e_vezes);

    // rregulloji fushat ashtu qe veza te zvogelohet
    // kur te vizatohet ne heren tjeter
    gjeresia_e_vezes = gjeresia_e_vezes / 2;
    lartesia_e_vezes = lartesia_e_vezes / 2;
  }
}

```

Për ta provuar, e shkruajmë, p.sh., kodin vijues:

```

public class TregoVezen
{ public static void main(String[] args)
  { new VizatuesIVezes(); }
}

```

## 4.5. Testimi i Programit që Përdor Hyrje

1. Rishkruaje aplikacionin që i përimiton paratë: `BeriTIma`, ashtu që t'i kërkojë dy numra të plotë si hyrje (vlerën e eurove dhe vlerën e centëve) dhe e bën përimitimin. Pastaj përkufizo disa raste të testimit të tipit të "kutisë së zezë" (për të dhënat hyrëse); përkufizo disa raste të testimit të tipit të "kutisë së bardhë". Në bazë të eksperimenteve tua shkruaje një listë të formave të hyrjes interaktive që do të bënin që programi të punojë mirë, dhe shkruaje një listë të formave të hyrjes që do të mund të shkaktonin përgjigje të papritura apo jokorrekte.

Përgjigje: Aplikacioni i përmendur është ky:

```
/** BeriTIma i llogarit paratë e imta për sasitë në ndryshoret
 * eurot dhe centet. */
public class BeriTIma
{ public static void main(String[] args)
  { int eurot = 3;
    int centet = 46;
    int parate = (eurot * 100) + centet;
    System.out.println("20-centëshe = " + (parate / 20));
    parate = parate % 20;
    System.out.println("10-centëshe = " + (parate / 10));
    parate = parate % 10;
    System.out.println("5-centëshe = " + (parate / 5));
    parate = parate % 5;
    System.out.println("1-centëshe = " + parate);
  }
}
```

Programi i rishkruar:

```
import javax.swing.*;
/** BeriTIma2 i llogarit paratë e imta për sasitë në ndryshoret
 * eurot dhe centet. */
public class BeriTIma2
{ public static void main(String[] args)
  { String e = JOptionPane.showInputDialog("Numri i plote i eurove:");
    int eurot = new Integer(e).intValue();
    String c = JOptionPane.showInputDialog("Numri i centeve te mbetura:");
    int centet = new Integer(c).intValue();
    int parate = (eurot * 100) + centet;
    System.out.println("20-centëshe = " + (parate / 20));
    parate = parate % 20;
    System.out.println("10-centëshe = " + (parate / 10));
    parate = parate % 10;
    System.out.println("5-centëshe = " + (parate / 5));
    parate = parate % 5;
    System.out.println("1-centëshe = " + parate);
  }
}
```

Hyrjet e tipit të "kutisë së zezë" janë ato që do të mund t'i shkruante një përdorues tipik, jo-fort-i kujdesshëm. Për programin e përimitimit, do të mund t'i provonim këto vlera të eurove e të centëve të tipit 'kuti e zezë':

- 3 49
- 0 99
- 3 112
- 349 0
- -2 -12
- -2 12
- një dy

Hyrjet e tipit "kuti e bardhë" janë krijuar për të na ndihmuar t'i kuptojmë dhe t'i analizojmë urdhërat specifikë brenda programit. Programi i përimitimit i bën llogaritjet e tij me veprimin modulo, prandaj do të mund t'i krijojmë këto hyrje, që e përdorin urdhërin, `paratë = paratë % 20; (money = money % 20;)` dhe të tregojmë se çka ndodh kur t'imat mund të llogariten vetëm me 20-centësha dhe me 20-centësha e disa monedha:

- 0 60
- 0 61
- 1 00
- 0 65

Urdhërat tjerë modulo në program mund të provohen ngjashëm.

## V Struktura e Komponenteve: Ndërtimi i Metodave dhe Klasave

### (5.1 Metodot)

### 5.2 Metodot Publike

#### 5.2.1 Disa Metoda Publike Themelore

1. Qe klasët `AsciiArtVizatues` dhe `VizatoArt`:

```
// AsciiArtVizatues.java
/** AsciiArtVizatues permban metoda per te vizatuar Ascii art */
public class AsciiArtVizatues
{ /** Konstruktori AsciiArtVizatues e ben 'inicializimin'. */
    public AsciiArtVizatues()
    { System.out.println(); }

    /** afishoBlete e afishon nje blete */
    public void afishoBlete()
    { System.out.println("  ,-.");
      System.out.println("  \\_/_"); // karakteri '\\' duhet te shkruhet si '\\'
      System.out.println(">{|||}-");
      System.out.println("  /  \\");
      System.out.println("  ^-^");
      System.out.println();
    }

    /** afishoFlutur e afishon nje flutur */
    public void afishoFlutur()
    { System.out.println("  _ \\  _"); // karakteri '"' duhet te shkruhet si '\"'
      System.out.println("  (_\\|/_");
      System.out.println("  (/|\\");
      System.out.println();
    }

    /** afishoKalikuse e afishon nje kalikuse */
    public void afishoKalikuse()
    { System.out.println(" `m'"); // \' për '
      System.out.println(" (|)");
      System.out.println();
    }
}
```

```
// VizatoArt.java
/** VizatoArt afishon Ascii art dhe nje fjali */
public class VizatoArt
{ public static void main(String[] args)
  { AsciiArtVizatues vizatuesi = new AsciiArtVizatues();
    vizatuesi.afishoBlete();
    System.out.println("Ky eshte test.");
    vizatuesi.afishoFlutur();
    vizatuesi.afishoBlete();
  }
}
```

Të themi se e ndryshojmë klasën `VizatoArt` ashtu që të duket kështu:

```
public class VizatoArt2
{ public static void main(String[] args)
  { AsciiArtVizatues v = new AsciiArtVizatues();
    v.afishoFlutur();
    v.afishoFlutur();
  }
}
```

Çka afishohet në ekran?

Përgjigje: Në ekran afishohet një rresht i zbrazët (konstruktori) dhe dy flutura.

**2.** Shpjego se çka paraqitet në dritaren komanduese kur ekzekutohet aplikacioni vijues. Sa objekte `AsciiWriter` konstruktohen?

```
public class TestArt
{ public static void main(String[] args)
  { AsciiArtVizatues vizatuesi = new AsciiArtVizatues();
    vizatuesi.afishoBlete();
    new AsciiArtVizatues().afishoFlutur();
    vizatuesi.afishoKalikuse();
  }
}
```

Përgjigje: Në dritaren komanduese afishohet një rresht i zbrazët (konstruktori), një bletë, një rresht i zbrazët (konstruktori i objektit të ri të përkohshëm

`new AsciiArtVizatues().afishoFlutur()`) dhe një flutur, pastaj një kalikuse.

Konstruktohen dy objekte `AsciiArtVizatues`: `vizatuesi` dhe një objekt i përkohshëm, që krijohet në rreshtin e pestë. Sa herë që rrjedha e kontrollit kalon nëpër fjalën kyçe, `new`, krijohet një objekt i ri.

**3. Qe një klasë me një metodë publike:**

```
import javax.swing.*;
public class KlaseNdihmese
{ public KlaseNdihmese()
  { } // asgje per inicializim

  /** llogariteRrenjenKatrore e lexon nje numer te plote si hyrje
   * dhe e afishon rrenjen katrore te tij. */
  public void llogariteRrenjenKatrore()
  { String s = JOptionPane.showInputDialog("Shkruane nje numer:");
    double d = new Double(s).doubleValue();
    double rrenja = Math.sqrt(d);
    JOptionPane.showMessageDialog(null,
      "Rrenja katrore e " + d + " eshte " + rrenja);
  }
}
```

Shkruaje një aplikacion ashtu që metoda `main` e tij e thërret metodën për t'i ndihmuar përdoruesit që t'i llogarisë dy rrënjë katrore.

**Përgjigje:**

```
public class LlogaritiDyRrenje
{ public static void main(String[] args)
  { KlaseNdihmese llogarites = new KlaseNdihmese();
    llogarites.llogariteRrenjenKatrore();
    llogarites.llogariteRrenjenKatrore();
  }
}
```

**4. Shkruaje metodën që mungon për këtë aplikacion:**

```
import javax.swing.*;
/** GjatesiaEEmrit e llogarit gjatesine e dy emrave.
 * Hyrja: dy emra, secili i shkruar ne nje dialog hyres
 * Dalja: dialogjet qe i afishojne emrat dhe gjatesite e tyre. */
public class GjatesiaEEmrit
{ public static void main(String[] args)
  { KlaseNdihmese n = new KlaseNdihmese();
    n.lexoEmrinDheAfishoGjatesineETij();
    n.lexoEmrinDheAfishoGjatesineETij();
    JOptionPane.showMessageDialog(null, "Fund.");
  }
}

public class KlaseNdihmese
{ ...
  /** lexoEmrinDheAfishoGjatesineETij e lexon nje emer dhe e afishon
   * emrin me gjatesine e tij
   * Udhezim: per stringun, x, x.length() e kthen gjatesine e x */
  ...
}
```

Përgjigje:

Në klasën ndihmëse, e shtojmë metodën:

```
public class KlaseNdihmese
{
    ...
    /** lexoEmrinDheAfishoGjatesineETij e lexon nje emer dhe e afishon
     * emrin me gjatesin e tij
     * Udhezim: per stringun, x, x.length() e kthen gjatesine e x */
    private static void lexoEmrinDheAfishoGjatesineETij()
    {
        String emri = JOptionPane.showInputDialog("Emri");
        JOptionPane.showMessageDialog(null, "Emri: " + emri
                                         + "; Gjatesia: " + emri.length() );
    }
}
```

Metoda nuk punon për shkak se është deklaruar si private. Domethënë ajo mund të thirret vetëm nga ndonjë vend përbrenda klasës së vet. Për ta thirrur këtë metodë prej klasës `GjatesiaEEmrit`, duhet ta deklarojmë si publike (dmth. ta ndërrojmë fjalën kyçe `private` me fjalën kyçe `public`).

## 5.2.2 Metodat Konstruktorë

1. Qe një klasë me konstruktor dhe metodë publike:

```
public class KlaseAfishuese
{
    public KlaseAfishuese()
    {
        System.out.println("A");
    }

    public void afishoB()
    {
        System.out.println("B");
    }
}
```

Çka afishon ky aplikacion?

```
public class TestoKlasenAfishuese
{
    public static void main(String[] args)
    {
        KlaseAfishuese p = new KlaseAfishuese();
        p.afishoB();
        p.afishoB();
        new KlaseAfishuese().afishoB();
        KlaseAfishuese q = p;
    }
}
```



Përgjigje: Aplikacioni i dhënë afishon:

A  
B  
B  
A  
B

Gjatë ekzekutimit konstruktohen dy objekte, prandaj `a` afishohet dy herë. Urdhëri i fundit i ndarjes së vlerës nuk konstrukton objekt. Ai thjesht e kopjon adresën që e mban `p` - në `q`.

**2. Qe një klasë me konstruktor dhe metodë publike:**

```
public class Numerues
{ private int numri;

    public Numerues()
    { numri = 0; }

    public void numero()
    { numri = numri + 1;
      System.out.println(numri);
    }
}
```

**Çka afishon ky aplikacion?**

```
public class TestoNumeruesin
{ public static void main(String[] args)
  { Numerues a = new Numerues();
    a.numero();
    a.numero();
    Numerues b = new Numerues();
    b.numero();
    a.numero();
    Numerues c = a;
    c.numero();
    a.numero();
  }
}
```

Përgjigje:

1  
2  
1  
3  
4  
5

Shembulli `TestoNumeruesin` i krijon dy objekte, me fusha të ndryshme `numri`. Ndryshoret `c` dhe `a` i referohen të njëjtit objekt.

## 5.3 Parametrat e Metodave

### 1. Çka do të afishojë ky aplikacion:

```
public class KlaseAfishuese2
{ public KlaseAfishuese2() { }

    public void afishoEmrin(String emri)
    { String s = " ";
      System.out.println(emri + s + emri);
    }
}

public class TestoKlasenAfishuese2
{ public static void main(String[] args)
  { String s = "Filan";
    new KlaseAfishuese2().afishoEmrin(s + s);
  }
}
```

Përgjigje: Aplikacioni afishon:

FilanFilan FilanFilan

sepse parametri aktual llogaritet si FilanFilan. Vëreje se si nuk ka kurrfarë lidhje ndërmjet dy ndryshoreve me emrin e njëjtë s.

### 2. Shkruaji trupat që mungojnë për metodat e kësaj klase; shikoje aplikacionin që pason nëse të duhet ndihmë.

```
import javax.swing.*;
/** KlaseEmer e mban mend nje emer dhe afishon te dhena per te. */
public class KlaseEmer
{ private String emri; // emri qe mbahet mend

    /** Konstruktori KlaseEmer e inicializon nje objekt KlaseEmer
     * @param n - emri qe do te mbahet mend */
    public KlaseEmer(String n)
    { ... }

    /** afishoEmrin e afishon emrin e mbajtur mend nga ky objekt */
    public void afishoEmrin()
    { ... }

    /** afishoGjatesine e afishon gjatesine e emrit te mbajtur mend.
     * (Udhezim: perdore metoden length nga Tabela 5, Kapitulli 3.) */
    public void afishoGjatesine()
    { ... }
}
```

```

public class TestoKlasenEmer
{
    public static void main(String[] args)
    {
        KlaseEmer emri_im = new KlaseEmer("Filan Fisteku");
        System.out.print("une quhem ");
        emri_im.afishoEmrin();
        System.out.print("emri im i ka kaq karaktere: ");
        emri_im.afishoGjatesine();
    }
}

```

#### Përgjigje:

```

/** Konstruktori KlaseEmer e inicializon nje objekt KlaseEmer
 * @param n - emri qe do te mbahet mend */
public KlaseEmer(String n)
{
    emri = n;
}

/** afishoEmrin e afishon emrin e mbajtur mend nga ky objekt */
public void afishoEmrin()
{
    System.out.println(emri);
}

/** afishoGjatesine e afishon gjatesine e emrit te mbajtur mend.
 * (Udhezim: perdore metoden length nga Tabela 5, Kapitulli 3.) */
public void afishoGjatesine()
{
    System.out.println(emri.length());
}

```

### 5.3.1 Format e Parametrave

#### 1. Qe një klasë ndihmëse:

```

public class KlaseAritmetike
{
    private int baza;

    public KlaseAritmetike(int b)
    {
        baza = b;
    }

    public void afishoShumezimin(String etiketa, double d)
    {
        System.out.println(etiketa + (baza * d));
    }
}

```

a) Çka afishon aplikacioni vijues? (Vizatoni shtigjet (gjurmët) e ekzekutimit nëse nuk jeni të sigurtë rreth pasimit të parametrave.)

```

public class TestoKlasenAritmetike
{
    public static void main(String[] args)
    {
        KlaseAritmetike k = new KlaseAritmetike(2);
        k.afishoShumezimin("3", 4.5 + 1);
        int i = 4;
        k.afishoShumezimin("A", i);
        k.afishoShumezimin("A", i-1);
    }
}

```

b) Shpjego gabimet në këtë aplikacion:

```
public class Gabimet
{
    public static void main(String[] args)
    {
        KlaseAritmetike k = new KlaseAritmetike();
        afishoShumezimin("A", 5);
        int s = 0;
        k.afishoShumezimin(s, 2 + s);
        k.afishoShumezimin(1, "A");
        k.afishoShumezimin("A", 9, 10);
        k.afishoDicka();
    }
}
```

Përgjigje:

a)

311.0

A8.0

A6.0

(Invokimi) Thirrja e parë i ka parametrat e vërtetë (aktualë) "3" dhe 5.5, që shkakton që stringut "3" t'i shtohet 11.0 (dmth. baza \* 5.5). Thirrja e dytë i ka parametrat e vërtetë "A" dhe 4; thirrja e tretë i ka parametrat "A" dhe 3.

b) Qe gabimet e gjetura nga Java-kompajleri:

- `KlaseAritmetike k = new KlaseAritmetike();` konstruktori e ka numrin e gabuar të argumenteve
- `afishoShumezimin("A", 5);` thirrjes (invokimit) i mungon emri i objektit pranues
- `k.afishoShumezimin(s, 2 + s);` parametri i dytë aktual është `String`, ndërsa kërkohet `double`
- `k.afishoShumezimin(1, "A");` parametri i parë është numër i plotë, ndësa kërkohet `String`, dhe parametri i dytë është `String`, ndërsa kërkohet `double` - renditja e parametrave është me rëndësi.
- `k.afishoShumezimin("A", 9, 10);` është dhënë sasi e gabuar e parametrave aktualë
- `k.afishoDicka();` është thirrur metodë e cila nuk është fare në klasën përkatëse `KlasëAritmetike`.

## 2. Shkruajni metodat që mungojnë për këtë klasë:

```

/** ShumeEPergjithshme i ndihmon nje femije t'i mbledhe disa numra */
public class ShumeEPergjithshme
{ private int shuma; // shuma e krejt numrave te mbledhur deri tash

    /** Konstruktori ShumeEPergjithshme e inicializon objektin */
    public ShumeEPergjithshme()
    { shuma = 0; }

    /** shtojaShumes ia shton edhe nje numer shumes se pergjithshme
     * @param num - numri i plote qe do t'i shtohet shumes */
    ...

    /** afishoShumen e afishon shumen aktuale te krejt numrave te shtuar
     * deri tash */
    ...

public class MbledhiDoNumra
{ public static void main(String[] args)
  { ShumeEPergjithshme llogaritesi = new ShumeEPergjithshme();
    llogaritesi.shtojaShumes(16);
    llogaritesi.shtojaShumes(8);
    llogaritesi.afishoShumen();
    llogaritesi.shtojaShumes(4);
    llogaritesi.afishoShumen();
  }
}

```

## 3. Metoda `paintComponent` e klasës që e ndërton orën, `VizatuesIOres` (te 4.3.1), mund të rishkruhet ashtu që ta përdorë këtë metodë:

```

/** vizatojeOren e vizaton nje ore me kohen momentale
 * @param oret - numri momental i oreve, numer i plote midis 1 e 12
 * @param minutat - numri momental i minutave,
 *                 numer i plote midis 0 e 59
 * @param x - këndi i sipërm i majtë ku duhet të paraqitet ora
 * @param y - këndi i sipërm i djathtë ku duhet të paraqitet ora
 * @param diametri - diametri i orës
 * @param g - penda grafike qe perdoret per ta vizatuar oren */
public void vizatojeOren(int oret, int minutat, int x, int y,
                        int diametri, Graphics g)

```

Shkruaje këtë metodë, fute në klasën `VizatuesIOres`, dhe rishkruaje `paintComponent-in` e klasës për ta thirrur metodën.

Pastaj, bëje `paintComponent-in` t'i vizatojë  $dy$  ora - një për kohën tënde dhe një për kohën aktuale në Sidnej (+9 orë, krahasuar me Prishtinë).

Vërejtje: Nëse ky ushtrim kërkon tepër, lexoje njësinë tjetër pastaj kthehu ta ripunosh ushtrimin.

Përgjigje: Klasa VizatuesIOres rregullohet si vijon:

```
import java.awt.*;
import javax.swing.*;
import java.util.*;
/** VizatuesIOres3 e vizaton nje ore ne nje panel. */
public class VizatuesIOres3 extends JPanel
{ public VizatuesIOres3()
    { int gjeresia = 200; // gjeresia e ores
      // konstruktoje kornizen e ketij paneli:
      JFrame korniza_e_ores = new JFrame();
      // dhe fute kete panel (panelin _this_) ne kornizen e vet:
      korniza_e_ores.getContentPane().add(this);
      // tregoje kornizen:
      korniza_e_ores.setTitle("Ora");
      korniza_e_ores.setSize(gjeresia, gjeresia);
      korniza_e_ores.setVisible(true);
    }

    /** paintComponent e vizaton oren me kohen korrekte.
     * @param g - penda grafike qe e ben vizatimin */
    public void paintComponent(Graphics g)
    { int gjeresia = 200;
      g.setColor(Color.white);
      g.fillRect(0, 0, gjeresia, gjeresia); // ngjyrose prapavijen
      int skaji_majtas = 50;
      int lart = 50;
      int diametri = 100;
      GregorianCalendar koha = new GregorianCalendar();
      vizatojeOren(koha.get(Calendar.HOUR), koha.get(Calendar.MINUTE),
                   skaji_majtas, lart, diametri, g);
    }

    /** vizatojeOren e vizaton nje ore me kohen momentale
     * @param oret - numri momental i oreve, numer i plote midis 1 e 12
     * @param minutat - numri momental i minutave, numer i plote midis 0 e 59
     * @param x - këndi i sipërm i majtë ku duhet të paraqitet ora
     * @param y - këndi i sipërm i djathtë ku duhet të paraqitet ora
     * @param diametri - diametri i orës
     * @param g - penda grafike qe perdoret per ta vizatuar oren */
    public void vizatojeOren(int oret, int minutat, int x, int y,
                             int diametri, Graphics g)
    { int kendi_i_minutave = 90 - (minutat * 6);
      int kendi_i_oreve = 90 - oret * 30;
      // vizatoje oren si rreth te zi:
      g.setColor(Color.black);
      g.drawOval(x, y, diametri, diametri);
      // vizatoje akrepin e minutave te kuq, 10 piksela me te vogel,
      // me gjeresi prej 5 shkallesh
      g.setColor(Color.red);
      g.fillArc(x + 5, y + 5, diametri - 10, diametri - 10,
                kendi_i_minutave, 5);
      // vizatoje akrepin e oreve te kalter, 50 piksela me te vogel,
      // me gjeresi prej -8 shkallesh
      g.setColor(Color.blue);
      g.fillArc(x + 25, y + 25, diametri - 50, diametri - 50,
                kendi_i_oreve, -8);
    }
}
```

```

/** Metoda main e monton oren ne kornizen e vet.
 * Metoda eshte futur ketu per qellime te testimit. */
public static void main(String[] args)
{ new VizatuesIOres3(); }
}

```

Për t'i vizatuar dy ora: të Prishtinës dhe të Sidnejit, i bëjmë këto ndryshime:

```

7   { int gjeresia = 400;  // gjeresia e kornizes

14   korniza_e_ores.setSize(gjeresia, gjeresia/2);

21   { int gjeresia = 400;

30   int distanca = 200; // distanca ndermjet dy orave
31   vizatojeOren((koha.get(Calendar.HOUR)+9) % 24, koha.get(Calendar.MINUTE),
32               skaji_majtas + distanca, lart, diametri, g);

```

## 5.4 Case Study: Kornizë Dalëse me Destinim të Përgjithshëm

### 1. Qe klasa Shkrues:

```

import java.awt.*; import javax.swing.*;
/** Shkrues e krijon nje dritare grafike qe e afishon nje fjali */
public class Shkrues extends JPanel
{ private int gjeresia; // gjeresia e kornizes
  private int lartesia; // lartesia e kornizes
  private String fjalia = ""; // e mban fjaline qe do te afishohet
  private int x_pozita; // x-pozita e fjalise
  private int y_pozita; // y-pozita e fjalise

  /** Konstruktori Shkrues e konstrukton Panelin
   * @param gj - gjeresia e dritares
   * @param l - lartesia e dritares */
  public Shkrues(int gj, int l)
  { gjeresia = gj;
    lartesia = l;
    x_pozita = gjeresia / 5; // caktoje poziten e fjalise
    y_pozita = lartesia / 2;
    JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("Shkrues");
    korniza_ime.setSize(gjeresia, lartesia);
    korniza_ime.setVisible(true);
  }

  /** paintComponent e vizaton panelin
   * @param g - 'penda grafike' qe i vizaton elementet */
  public void paintComponent(Graphics g)
  { g.setColor(Color.red);
    g.drawString(fjalia, x_pozita, y_pozita);
  }
}

```

```

/** shkruajeFjaline e afishon nje string te ri ne dritare
 * @param s - fjalia qe do te afishohet */
public void shkruajeFjaline(String s)
{
    fjalia = s;
    this.repaint(); // terthorazi e detyron metoden paintComponent
                    // qe te ekzekutohet
}
/** vendoseFjaline e riafishon fjaline ekzistuese ne pozite te re
 * @param x_i_ri - pozita e re filluese horizontale
 * @param y_i_ri - pozita e re filluese vertikale */
public void vendoseFjaline(int x_i_ri, int y_i_ri)
{
    x_pozita = x_i_ri;
    y_pozita = y_i_ri;
    this.shkruajeFjaline(fjalia); // detyro rishkruarjen e fjalise ekzistuese
}
}

```

Shpjegoje të sjellurit e këtij aplikacioni që e përdor klasën Shkrues:

```

import javax.swing.*;
public class ShembullTjeter
{
    public static void main(String[] args)
    {
        Shkrues shkruesi = new Shkrues(300, 200);
        String s = JOptionPane.showInputDialog("Ju lutem shkruani dicka:");
        shkruesi.shkruajeFjaline(s);
        shkruesi.vendoseFjaline(0, 190);
        shkruesi.shkruajeFjaline(s + s);
    }
}

```

**Përgjigje:** Aplikacioni e afishon fjalinë e parë të futur në pozitën 60, 100 (300/5, 200/2). Fjalia e dytë e futur paraqitet në pozitën e njëjtë dhe pastaj menjëherë rivendoset në 0, 190. Pothuajse menjëherë pas kësaj, fjalia afishohet dy herë në 0, 190. Për sytë e përdoruesit, krejt çka shihet është afishimi i parë dhe i fundit.

**2.** Shkruaje një aplikacion që kërkon nga përdoruesi ta shkruajë një numër të plotë, e llogarit rrënjën katrore të atij numri, dhe e përdor klasën Shkrues për ta afishuar numrin dhe rrënjën katrore të tij. Rrënja të afishohet me saktësi prej 6 vendesh decimale.

**Përgjigje:**

```

import java.text.*; import javax.swing.*;
public class RrenjaKatrore
{
    public static void main(String[] args)
    {
        String s = JOptionPane.showInputDialog(
            "Ju lutem shkruane nje numer te plote:");
        int num = new Integer(s).intValue();
        double d = Math.sqrt(num);
        String pergjigjja = new DecimalFormat("0.000000").format(d);
        new Shkrues(300, 200).shkruajeFjaline("rrenja katrore e " + num
            + " eshte " + pergjigjja);
    }
}

```



**3. Shtoja këtë metodë të re klasës Shkrues:**

```

/** shkruajeFjalineEDyte e afishon nje fjali, t, përfundi fjalise
 * se pare ne dritare.
 * @param t - fjalia e dyte qe do te afishohet */
public void shkruajeFjalineEDyte(String t)

```

Kur e shton këtë metodë, a duhet ta rindreqësh shkruajeFjaline? rivendoseFjaline?

**Përgjigje:**

```

import java.awt.*; import javax.swing.*;
/** Shkrues2 e krijon nje dritare grafike qe i afishon dy fjali */
public class Shkrues2 extends JPanel
{ private int gjeresia; // gjeresia e kornizes
  private int lartesia; // lartesia e kornizes

  private String fjalia = ""; // e mban fjaline qe do te afishohet
  private String fjalia2 = ""; // fjalia e dyte qe do te afishohet

  private int x_pozita; // x-pozita e fjalise
  private int y_pozita; // y-pozita e fjalise
  private int shtesa = 20; // ku afishohet fjalia e dyte nen te paren

  /** Konstruktori Shkrues e konstrukton Panelin
   * @param gj - gjeresia e dritares
   * @param l - lartesia e dritares */
  public Shkrues2(int gj, int l)
  { ... si më parë ... }

  /** paintComponent e vizaton panelin
   * @param g - 'penda grafike' qe i vizaton elementet */
  public void paintComponent(Graphics g)
  { g.setColor(Color.red);
    g.drawString(fjalia, x_pozita, y_pozita);
    g.drawString(fjalia2, x_pozita, y_pozita + shtesa);
  }

  /** shkruajeFjaline e afishon nje string te ri ne dritare
   * @param s - fjalia qe do te afishohet */
  public void shkruajeFjaline(String s)
  { ... si më parë ... }

  /** vendoseFjaline e riafishon fjaline ekzistuese ne pozite te re
   * @param x_i_ri - pozita e re filluese horizontale
   * @param y_i_ri - pozita e re filluese vertikale */
  public void vendoseFjaline(int x_i_ri, int y_i_ri)
  { ... si më parë ... }

  /** shkruajeFjalineEDyte e afishon nje fjali, t,
   * nen fjaline e pare ne dritare.
   * @param t - fjalia e dyte qe do te afishohet */
  public void shkruajeFjalineEDyte(String t)
  { fjalia2 = t;
    this.repaint();
  }
}

```

**4. Shkruaje një klasë që e plotëson këtë specifikacion:**

TekstShkrues i afishon deri ne tre rreshta te tekstit ne dritare grafike

***Konstruktori:***

```
TekstShkrues(int gj, int l)
```

E konstrukton dritaren me gjerësi `gj` piksela dhe lartësi `l` piksela dhe e afishon një dritare me tre rreshta të zbrazët

***Metodat:***

<code>afisho1(String s)</code>	Ia shton <code>s</code> fundit të Rreshtit 1 dhe i afishon vlerat aktuale të të tre rreshtave të tekstit.
<code>ricakto1(String s)</code>	E shndërron Rreshtin 1 të tekstit në <code>s</code> dhe i afishon vlerat aktuale të të tre rreshtave
<code>afisho2(String s)</code>	Ia shton <code>s</code> fundit të Rreshtit 2 dhe i afishon vlerat aktuale të të tre rreshtave të tekstit.
<code>ricakto2(String s)</code>	E shndërron Rreshtin 1 të tekstit në <code>s</code> dhe i afishon vlerat aktuale të të tre rreshtave
<code>afisho3(String s)</code>	Ia shton <code>s</code> fundit të Rreshtit 1 dhe i afishon vlerat aktuale të të tre rreshtave të tekstit.
<code>ricakto3(String s)</code>	E shndërron Rreshtin 1 të tekstit në <code>s</code> dhe i afishon vlerat aktuale të të tre rreshtave

**Testoje klasën me këtë aplikacion:**

```
import javax.swing.*;
public class TestoTekstShkruesin
{ public static void main(String[] args)
  { TekstShkrues shkrues = new TekstShkrues(300, 200);
    String s = JOptionPane.showInputDialog("Ju lutem shkruani ca tekst:");
    shkrues.afisho1(s);
    s = JOptionPane.showInputDialog("Provoni edhe nje here:");
    shkrues.afisho1(s);
    s = JOptionPane.showInputDialog("Edhe nje here:");
    shkrues.afisho3(s);
    s = JOptionPane.showInputDialog("Hera e fundit:");
    shkrues.ricakto1(s);
  }
}
```

Pastaj, rishkruaje klasën `CelsiusNëFahrenheit`, ashtu që ta përdorë një `ShkruesTeksti` për ta afishuar daljen e vet.

Përgjigje:

```
import java.awt.*;
import javax.swing.*;
/** TekstShkrues i paraqet deri ne tre rreshta tekst ne nje dritare grafike */
public class TekstShkrues extends JPanel
{ private int gjeresia; // gjeresia e kornizes
  private int lartesia; // lartesia e kornizes

  private String fjalial1 = "";
  private String fjalial2 = "";
  private String fjalial3 = "";
  private int x_pozita; // x-pozita e fjalive
  private int y_pozita; // y-pozita e fjalive
  private int shtesa = 20; // distanca ndermjet fjalive

  /** Konstruktori TekstShkrues e krijon dritare dhe e ben te dukshme
   * @param gj - gjeresia e dritares
   * @param l - lartesia e dritares */
  public TekstShkrues(int gj, int l)
  { gjeresia = gj;
    lartesia = l;
    x_pozita = gjeresia / 5; // caktoje poziten e fjalise
    y_pozita = lartesia / 2;
    JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("TekstShkrues");
    korniza_ime.setSize(gjeresia, lartesia);
    korniza_ime.setVisible(true);
  }

  /** paintComponent e vizaton dritaren
   * @param g - penda grafike qe i vizaton elementet brenda dritares */
  public void paintComponent(Graphics g)
  { g.setColor(Color.red);
    g.drawString(fjalial1, x_pozita, y_pozita);
    g.drawString(fjalial2, x_pozita, y_pozita + shtesa);
    g.drawString(fjalial3, x_pozita, y_pozita + (2 * shtesa));
  }

  public void afishol(String s)
  { fjalial1 = fjalial1 + s;
    this.repaint();
  }

  public void ricaktol(String s)
  { fjalial1 = s;
    this.repaint();
  }

  // metodat tjera jane te ngjashme me dy te mesipermet
}
```

## 5.5 Rezultatet e Metodave: Funksionet

### 1. Çka afishon ky aplikacion?

```
public class Numerues2
{ private int numri;

    public Numerues2(int i)
    { numri = i; }

    public int numeroA()
    { numri = numri + 1;
      return numri;
    }

    public double numeroB()
    { return numri + 1.5; }

    public String numeroC()
    { return "*" + numri; }
}

public class TestoNumeruesin2
{ public static void main(String[] args)
  { Numerues2 n = new Numerues2(3);
    int x = n.numeroA();
    System.out.println(x + " " + n.numeroB());
    System.out.println(n.numeroC() + n.numeroA());
  }
}
```

### Përgjigje:

```
4 5.5
*45
```

### 2. Qe funksioni celsiusNeFahrenheit:

```
/** celsiusNeFahrenheit i përkthen gradët Celsius në Fahrenheit
 * @param c - gradët në Celsius
 * @return (kthen) gradët përkatëse (ekuivalente) në Fahrenheit */
public double celsiusNeFahrenheit(double c)
{ double f = ((9.0 / 5.0) * c) + 32;
  return f;
}
```

Vendose këtë funksion në klasën ShnderruesITemperatures. Shtoja edhe këtë metodë klasës:

```
/** fahrenheitNeCelsius i shndërron gradët Fahrenheit në Celsius
 * @param f - gradët në Fahrenheit, double
 * @return (kthen) ekuivalentin e shkallëve në Celsius, double */
```

(Udhëzim: përdore algjebrën për ta llogaritur formulën shndërruese.)

Pastaj, testojë klasën me këtë aplikacion:

```
public class ShnderroTemperaturat
{ public static void main(String[] args)
  { ShnderruesITemperatures llogaritesi = new ShnderruesITemperatures();
    int temp = new Integer(args[0]).intValue(); // merre hyrjen si arg. prog.
    double ftemp = llogaritesi.celsiusNeFahrenheit(temp);
    System.out.println(temp + "C eshte " + ftemp + "F");
    System.out.println("Verifiko: " + ftemp + "F eshte "
      + llogaritesi.fahrenheitNeCelsius(ftemp) + "C");
    double ctemp = llogaritesi.fahrenheitNeCelsius(temp);
    System.out.println(temp + "F eshte " + ctemp + "C");
    System.out.println("Verifiko: " + ctemp + "C eshte "
      + llogaritesi.celsiusNeFahrenheit(ctemp) + "F");
  }
}
```

Përgjigje:

```
public class ShnderruesITemperatures
{ public ShnderruesITemperatures() { }

  public double celsiusNeFahrenheit(double c)
  { return ((9.0 / 5.0) * c) + 32; }

  public double fahrenheitNeCelsius(double f)
  { return (5 * (f - 32)) / 9.0; }
}
```

**3. Shkruaj funksionet që iu përgjigjen secilit specifikacion vijues:**

a)

```
/** kilometraNeMila e konverton nje sasi kilometrash ne mila
 * duke e perdorur formulen: Mila = 0.62137 * Kilometra
 * @param k - sasia e kilometrave
 * @return - kthen sasine korresponduese te milave */
public double kilometraNeMila(double k)
```

Fute funksionin në një aplikacion që e lexon një vlerë të kilometrave nga përdoruesi dhe e afishon vlerën në mila.

b)

```
/** totaliIShtuarI e llogarit totalin e shtuar qe rezulton
 * nga kapitali fillestar, k, shkalla e interesit, i,
 * dhe kohezgjatja prej n periudhave shtuese, duke e perdorur kete formule:
 * totali = k((1 + (i/n))^n (^ dmth. "ne fuqine")
 * @param k - kapitali fillestar, sasi e dollareve, centeve (p.sh. 150.67)
 * @param i - shkalla e interesit per periudhen shtuese,
 * ne trajte decimale (ose si thyese) (p.sh. 0.01 eshte 1%)
 * @param n - periudhat perberese (p.sh., nese shtuarja behet ne muaj,
 * ateherë dy vite bejne 24 periudha te shtuarjes) */
public double totaliIShtuarI(double k, double i, int n)
```

c)

```
/** plotpjesetohetMeNente e provon se a plotpjesetohet argumenti i vet
 * me 9
 * @param arg - argumenti qe do te provohet
 * @return - kthen true, nese plotpjesetohet me 9, perndryshe kthen false
 */
public boolean plotpjesetohetMeNente(int arg)
```

ç)

```
/** rroga e llogarit rrogen javore te nje nepunesi
 * @param emri - emri i nepunesit
 * @param oret - oret e punuara gjate javes
 * @param cmimi_i_ores - cmimi i ores
 * @return - kthen nje tekst (string) qe perbehet nga emri, i vijuar
 * nga "Euro" dhe rroga */
public String rroga(String emri, int oret, double cmimi_i_ores)
```

Përgjigje:

a)

```
public double kilometraNeMila(double k)
{ return (0.62137 * k); }
```

b)

```
public double totaliIShtuarI(double k, double i, int n)
{ return ( k * Math.pow(1 + (i/n), n) ); }
```

c)

```
public boolean plotpjesetohetMeNente(int arg)
{ return ((arg % 9) == 0); }
```

ç)

```
public String rroga(String emri, int oret, double cmimi_i_ores)
{ double rroga = oret * cmimi_i_ores;
  return (emri + ": Euro " + rroga);
}
```

4. Qe një aplikacion që do të kishte dobi nga një funksion. Rishkruaje aplikacionin ashtu që ta ketë një funksion.

```
/** Syprinat i afishon syprinat e tre rratheve */
public class Syprinat
{ public static void main(String[] args)
  { System.out.println("Per rrezen 4, syprina = " + (Math.PI * 4*4));
    System.out.println(Math.PI * 8*8);
    System.out.println((Math.PI * 19*19) + " eshte syprina per rrezen 19");
  }
}
```

(Vërejtje: `Math.PI` është emri i Java-s për konstanten matematikore, `Pi`.)

Përgjigje:

```

/** Syprinat2 i afishon syprinat e tre rratheve */
public class Syprinat2
{ public static void main(String[] args)
  { Formula f = new Formula();
    System.out.println("Per rrezen 4, syprina = " + f.syprinaE(4));
    System.out.println(f.syprinaE(8));
    System.out.println(f.syprinaE(19) + " eshte syprina per rrezen 19");
  }
}

public class Formula
{ public Formula() { }

  public double syprinaE(int rrezja)
  { return (Math.PI * rrezja * rrezja); }
}

```

**5.6 Metodat Private****1. Qe klasa VizatuesIVezeveNeStek:**

```

import java.awt.*;
import javax.swing.*;
/** VizatuesIVezeveNeStek i paraqet tri veze, te stekuara (nje mbi nje) */
public class VizatuesIVezeveNeStek extends JPanel
{ private int gjeresia_e_kornizes;
  private int lartesia_e_kornizes;
  // madhesite (gjeresia) e tri vezeve te stekuara:
  private int madhesia_e_vezes1;
  private int madhesia_e_vezes2;
  private int madhesia_e_vezes3;

  /** Konstruktori VizatuesIVezeveNeStek i stekon tri veze 3-me-2 ne dritare
   * @param gjeresia - gjeresia e panelit
   * @param lartesia - lartesia e panelit
   * @param madhesia1 - gjeresia e vezes se poshtme
   * @param madhesia2 - gjeresia e vezes se mesme
   * @param madhesia3 - gjeresia e vezes se siperme */
  public VizatuesIVezeveNeStek(int gjeresia, int lartesia,
                                int madhesia1, int madhesia2, int madhesia3)
  { gjeresia_e_kornizes = gjeresia;
    lartesia_e_kornizes = lartesia;
    madhesia_e_vezes1 = madhesia1;
    madhesia_e_vezes2 = madhesia2;
    madhesia_e_vezes3 = madhesia3;
    JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("VizatuesIVezeveNeStek");
    korniza_ime.setSize(gjeresia_e_kornizes, lartesia_e_kornizes);
    korniza_ime.setVisible(true);
  }
}

```

```

/** paintComponent e mbush dritaren me vezet
 * @param g - penda grafike */
public void paintComponent(Graphics g)
{ g.setColor(Color.yellow);
  g.fillRect(0, 0, gjeresia_e_kornizes, lartesia_e_kornizes); // prapavija
  // leshoje vezen e pare ne fund te kornizes:
  int vezal_lart = vizatojeNjeVeze(lartesia_e_kornizes,
                                  madhesia_e_vezes1, g);

  // steko dy vezet e mbetura mbi te:
  int veza2_lart = vizatojeNjeVeze(vezal_lart,
                                   madhesia_e_vezes2, g);

  int veza3_lart = vizatojeNjeVeze(veza2_lart,
                                   madhesia_e_vezes3, g);
}

/** vizatojeNjeVeze e vizaton nje veze ne proporcion 3-me-2
 * (lartesia e vezes eshte sa dy-te-tretat e lartesisë se saj)
 * @param fundi - pozita e fundit te vezes
 * @param gjeresia - gjeresia e vezes
 * @param penda - penda grafike qe do ta vizatoje vezen
 * @return y-poziten e skajit te siperm te vezes se vizatuar. */
private int vizatojeNjeVeze(int fundi, int gjeresia, Graphics penda)
{ int lartesia = (2 * gjeresia) / 3;
  int skaji_lart = fundi - lartesia;
  int skaji_majtas = (gjeresia_e_kornizes - gjeresia) / 2;
  penda.setColor(Color.pink);
  penda.fillOval(skaji_majtas, skaji_lart, gjeresia, lartesia);
  penda.setColor(Color.black);
  penda.drawOval(skaji_majtas, skaji_lart, gjeresia, lartesia);
  return skaji_lart;
}

/** Testoje dritaren: */
public static void main(String[] args)
{ int gjeresia_totale = 300;
  int lartesia_totale = 200;
  new VizatuesIVezeveNeStek(gjeresia_totale, lartesia_totale, 50, 90, 140);
}
}

```

**Shkruaje një aplikacion testues që i konstrukton këto variacione të klasës**

VizatuesIVezeveNeStek

- a) new VizatuesIVezeveNeStek(300, 200, 30, 0, 500);
- b) new VizatuesIVezeveNeStek(300, 200, 30, -10, 30);
- c) new VizatuesIVezeveNeStek(300, 200, -300, 300, 30);

**Mund të të duket e dobishme ta maksimizosh dritaren për t'i parë disa rezultate.**

**Përgjigje:**

```

a)
public class TestoVezetNeStek1
{ public static void main(String[] args)
  { new VizatuesIVezeveNeStek(300, 200,
                              30, 0, 500);
  }
}

```



**2. Shtoja metodat publike** `caktoMadhesineEVezes1(int madhesia),`  
`caktoMadhesineEVezes2(int madhesia),`  
**dhe** `caktoMadhesineEVezes3(int madhesia)` **klasës** `VizatuesIVezeveNeStek`.

Natyrisht, secila metodë e ricakton madhësinë e vezës përkatëse të paraqitur dhe e rivizaton dritaren. Testoje klasën e modifikuar me këtë aplikacion:

```
import javax.swing.*;
public class StekoDisaVeze
{ public static void main(String[] args)
  { VizatuesIVezeveNeStek2 vizatuesi
    = new VizatuesIVezeveNeStek2(300, 200, 0, 0, 0);
    String s = JOptionPane.showInputDialog(
      "Shkruaje madhesine e vezes se fundit:");
    vizatuesi.caktoMadhesineEVezes1(new Integer(s).intValue());

    s = JOptionPane.showInputDialog("Shkruaje madhesine e vezes se mesme:");
    vizatuesi.caktoMadhesineEVezes2(new Integer(s).intValue());

    s = JOptionPane.showInputDialog("Shkruaje madhesine e vezes se siperme:");
    vizatuesi.caktoMadhesineEVezes3(new Integer(s).intValue());
  }
}
```

**Përgjigje:** Pa e ndërruar asnjë pjesë të kodit të klasës `VizatuesIVezeveNeStek`, shtoji këto metoda:

```
/** caktoMadhesineEVezes1 e cakton madhesine e vezes se poshtme
 * @param madhesia - madhesia (gjeresia) e vezes */
public void caktoMadhesineEVezes1(int madhesia)
{ madhesia_e_vezes1 = madhesia; repaint(); }

public void caktoMadhesineEVezes2(int madhesia)
{ madhesia_e_vezes2 = madhesia; repaint(); }

public void caktoMadhesineEVezes3(int madhesia)
{ madhesia_e_vezes3 = madhesia; repaint(); }
```

**3. Qe klasa** `Shkrues3`, e cila është modifikim i klasës `Shkrues`:

```
import java.awt.*; import javax.swing.*;
/** Shkrues3 e krijon nje dritare grafike qe e afishon nje fjali */
public class Shkrues3 extends JPanel
{ private int gjeresia; // gjeresia e kornizes
  private int lartesia; // lartesia e kornizes
  private String fjalia = ""; // e mban fjaline qe do te afishohet
  private int x_pozita = 50; // x-pozita e fjalise
  private int y_pozita = 80; // y-pozita e fjalise

  /** Konstruktori Shkrues3 e konstrukton Panelin
   * @param gj - gjeresia e dritares
   * @param l - lartesia e dritares */
  public Shkrues3(int gj, int l)
  { ... si te klasa Shkrues ... }
```

```

/** paintComponent e vizaton panelin
 * @param g - 'penda grafike' qe i vizaton elementet */
public void paintComponent(Graphics g)
{ ndertoKufirin(g);
  g.setColor(Color.red);
  g.drawString(fjalia, x_pozita, y_pozita);
}

/** ndertoKufirin e nderton kufirin e kornizes.
 * @param penda - penda grafike e perdorur per ta ngjyrosur kufirin */
private void ndertoKufirin(Graphics penda)
{ penda.setColor(Color.blue);
  // ngjyrose krejt dritaren me te kalter:
  penda.fillRect(0, 0, gjeresia, lartesia);
  int madhesia_e_kufirit = 20;
  int gjeresia_e_drejtkeneshit_qendror
    = gjeresia - (2 * madhesia_e_kufirit);
  int lartesia_e_drejtkeneshit_qendror
    = lartesia - (2 * madhesia_e_kufirit);
  penda.setColor(Color.white);
  // vizatoje drejtkeneshin qendror te bardhe:
  penda.fillRect(madhesia_e_kufirit, madhesia_e_kufirit,
                  gjeresia_e_drejtkeneshit_qendror,
                  lartesia_e_drejtkeneshit_qendror);
}

/** shkruajeFjaline e afishon nje string te ri ne dritare
 * @param s - fjalia qe do te afishohet */
public void shkruajeFjaline(String s)
{ ... si te klasa Shkrues ... }

/** vendoseFjaline e riafishon fjaline ekzistuese ne pozite te re
 * @param x_i_ri - pozita e re filluese horizontale
 * @param y_i_ri - pozita e re filluese vertikale */
public void vendoseFjaline(int x_i_ri, int y_i_ri)
{ ... }

/** main per testim */
public static void main(String[] args)
{ Shkrues3 shkruesi = new Shkrues3(300, 200);
  shkruesi.shkruajeFjaline("Sa kornize e bukur!");
}
}

```

Modifikoje metodën `ngjyrosKufirin` ashtu që ta vizatojë një rreth të mbushur të bardhë, me diametër sa lartësia e dritares, në qendër – përpara prapavijës së verdhë.

### Përgjigje.

```

private void ndertoKufirin(Graphics penda)
{ penda.setColor(Color.yellow);
  // ngjyrose krejt dritaren me te verdhe:
  penda.fillRect(0, 0, gjeresia, lartesia);
  int diametri = lartesia;
  penda.setColor(Color.white);
  // vizatoje rrethin qendror te bardhe:
  penda.fillOval(gjeresia/2 - diametri/2, 0, diametri, diametri);
}

```

**4. Kjo klasë mund të përfitojë nga një metodë private; fute atë metodë.**

```
import java.awt.*;
import javax.swing.*;
/** Rrathet i vizaton tre rrathe koncentrike (bashkeqendrore) */
public class Rrathet extends JPanel
{ public Rrathet()
  { JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("TekstShkrues");
    korniza_ime.setSize(200, 200);
    korniza_ime.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { int x_poz = 100; // x-pozita e qendres se rrethit
    int y_poz = 100; // y-pozita e qendres se rrethit
    int diametri = 60; // diametri i rrethit qe vizatohet
    g.setColor(Color.black);
    int rrezja = diametri / 2;
    g.drawOval(x_poz - rrezja, y_poz - rrezja, diametri, diametri);
    diametri = diametri + 20;
    rrezja = diametri / 2;
    g.drawOval(x_poz - rrezja, y_poz - rrezja, diametri, diametri);
    diametri = diametri + 20;
    rrezja = diametri / 2;
    g.drawOval(x_poz - rrezja, y_poz - rrezja, diametri, diametri);
  }
}
```

Përgjigje:

```
import java.awt.*;
import javax.swing.*;
/** Rrathet2 i vizaton tre rrathe koncentrike (bashkeqendrore) */
public class Rrathet2 extends JPanel
{ private int diametri = 60; // diametri i rrethit me te madh
  int x_poz = 100;          // x-pozita e qendres se rrethit
  int y_poz = 100;          // y-pozita e qendres se rrethit

  public Rrathet2()
  { JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("TekstShkrues");
    korniza_ime.setSize(200, 200);
    korniza_ime.setVisible(true);
  }

  public void paintComponent(Graphics g)
  { g.setColor(Color.black);
    vizatoRrethin(diametri, g);
    vizatoRrethin(diametri / 2, g);
    vizatoRrethin(diametri / 4, g);
  }

  public void vizatoRrethin(int d, Graphics g)
  { int rrezja = d / 2;
    g.drawOval(x_poz - rrezja, y_poz - rrezja, d, d);
  }

  /** main - per testim */
  public static void main(String[] args)
  { new Rrathet2(); }
}
```

## **VI Struktura e Kontrollit: Urdhërat Kushtëzues dhe Arkitektura e Softuerit**

### **(6.1 Rrjedha e Kontrollit dhe Struktura e Kontrollit)**

### **6.2 Struktura Kushtëzuese e Kontrollit**

1. Çka do të afishojnë këta shembuj?

```
a) double d = 4.14;
    int i = 3;
    if ( i == d )
        { System.out.println("Baraz"); }
    else { i = (int)d; }
    if ( i != 3 )
        { System.out.println("Jo baraz"); }

b) public static void main(String[] args)
    { System.out.println( f(2, 3) );
      int i = 2;
      System.out.println( f(i+1, i) );
    }

    private static String f(int x, int y)
    { String përgjigjja;
      if ( x <= y )
          { përgjigjja = "më i vogël"; }
      else { përgjigjja = "jo më i vogël"; }
      return përgjigjja;
    }
```

Përgjigje:

```
a) Jo baraz
b) më i vogël
    jo më i vogël
```

2. Të themi se ndryshorja  $x$  është inicializuar si `int x = 1`. Për secilin nga urdhërat vijues, gjeji gabimet sintaksore që do t'i detektojë Java-kompajleri. (Nëse jeni të pasigurtë për gabimet, shkruani urdhërat në një program testues dhe provoni t'i kompajloni.)

```
a) if ( x ) { } else { x = 2; }
b) if x>0 { x = 2 }
c) if ( x = 0 ) { x = 2; }; else {}
```

Përgjigje:

- a) Testi i urdhërit kushtëzues duhet të jetë i tipit `boolean`, por `x` e ka tipin `int`.  
 b) Përreth shprehjes `x > 0` kërkohen kllapat. Poashtu, `x = 2` duhet të përfundojë me pikëpresje.  
 c) Shprehja `x = 0` nuk është korrekte - do të duhej të shkruhet `x == 0`. Poashtu, pikëpresja para fjalës kyçe `else` duhet të hiqet.

**3. Përdori urdhërat kushtëzues për t'i shkruar trupat e këtyre metodave:**

- a) 

```
/** afishoÇift e afishon se a është argumenti i vet çift apo tek:
 * Afishon "ÇIFT" kur argumenti është çift.
 * Afishon "TEK" kur argumenti është tek.
 * @param a - argumenti */
public void afishoÇift(int a)
```
- b) 

```
/** minus e zbret argumentin e vet të dytë nga i pari,
 * duke u siguruar që rezultati është jonegativ
 * @param arg1 - argumenti i parë, duhet të jetë jonegativ
 * @param arg2 - argumenti i dytë
 * @return (kthen) (arg1 - arg2), nëse arg1 >= arg2;
 * return (kthen) -1, nëse arg1 është negativ ose nëse arg2 > arg1 */
public int minus(int arg1, int arg2)
```

(Udhëzim: vendose një urdhër kushtëzues brenda një kushtëzuesi.)

- c) 

```
/** mod e bën pjesëtimin e plotë të argumentëve të vet
 * @param arg1 - i pjesëtueshmi
 * @param arg2 - pjesëtuesi
 * @return (arg1 / arg2), duke u siguruar që arg2 është jozero;
 * return 0 dhe afisho një porosi të gabimit, nëse arg2 == 0 */
public int mod(int arg1, int arg2)
```

Përgjigje:

- a)
- ```
public void afishoÇift(int a)
{ if ( (a % 2) == 0 ) // a eshte cift?
  { System.out.println("ÇIFT");
  }
  else // përndryshe, është tek
  { System.out.println("TEK");
  }
}
```
- b)
- ```
public int minus(int arg1, int arg2)
{ int përgjigjja = -1;
  if (arg1 >= 0)
  { if (arg1 >= arg2)
    { përgjigjja = arg1 - arg2; }
  }
  return përgjigjja;
}
```

### 6.2.1 Urdhërat Kushtëzues të Vendosur Përbrenda

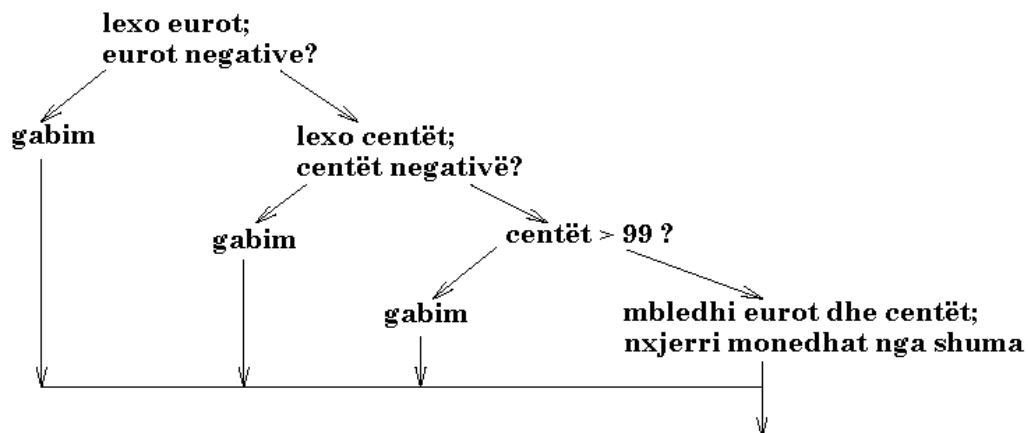
1. Çka do të afishojë ky shembull?

```
int i = 1;
if ( i < 0 )
    { System.out.println("a"); }
else { System.out.println("b");
      if ( i == 1 )
          { System.out.println("c"); }
        System.out.println("d");
      }
System.out.println("e");
```

Përgjigje:

b  
c  
d  
e

2. Përdore figurën vijuese për t'i sajuar rastet e testimit ashtu që secila rrjedhë e kontrollit nëpër `PërimtoPërsëri` ekzekutohet nga të paktën një rast i testimit. Testoje aplikacionin tënd me rastet tua të testimit.



Përgjigje: Një bashkësi e testeve do të mund të ishte: -2 0, 2 -2, 2 102, dhe 2 2.

## 3. a) Shkruaje këtë metodë; përdori kushtëzuesit e futur përbrenda

```

/** konvertONëSekonda e konverton (përkthen) një sasi të orëve, minutave
 * në kohë ekuivalente në sekonda.
 * @param orët - orët, numër i plotë jonegativ
 * @param minutat - minutat, numër i plotë në rangun 0..59
 * @return (kthen) kohën në sekonda;
 * nëse ka argumente të këqija, kthen -1 */
public int konvertONëSekonda(int orët, int minutat)

```

- b) Pastaj, rishkruaje metodën e mëparshme *pa* kushtëzues të futur përbrenda - përdore një ndryshore `boolean` për ta mbajtur mend statusin e konvertimit.

Përgjigje:

a)

```

import javax.swing.*;
...
public int konvertONëSekonda(int orët, int minutat)
{ int sekondat = -1;
  if ( orët < 0 )
  { JOptionPane.showMessageDialog(null,
    "gabim: orë negative: " + orët);
  }
  else { if ( minutat < 0 )
    { JOptionPane.showMessageDialog(null,
      "gabim: minuta negative: " + minutat);
    }
    else { if ( minutat > 59 )
      { JOptionPane.showMessageDialog(null,
        "gabim: minutat: " + minutat);
      }
      else { sekondat = ((orët*60) + minutat) * 60; }
    }
  }
  return sekondat;
}

```



b)

```

import javax.swing.*;
...
public int konvertONëSekonda2(int orët, int minutat)
{ boolean mirë = true; // e mban mend a janë të dhënat hyrëse të pranueshme
  int sekondat = -1;
  if ( orët < 0 )
  { JOptionPane.showMessageDialog(null,
    "gabim: orë negative: " + orët);
    mirë = false;
  }
  if ( minutat < 0 )
  { JOptionPane.showMessageDialog(null,
    "gabim: minuta negative: " + minutat);
    mirë = false;
  }
  if ( minutat > 59 )
  { JOptionPane.showMessageDialog(null,
    "gabim: minutat: " + minutat);
    mirë = false;
  }
  if ( mirë )
  { sekondat = ((orët * 60) + minutat) * 60; }
  return sekondat;
}

```

4. Aplikacionet në Figurat 2 dhe 4 (në libër) nuk sillen saktësisht njëjtë. Gjeje një rast testues që e demonstroi këtë, dhe shpjego se çfarë ndërrimesh duhet t'i bëhen Figurës 4 ashtu që të sillet saktësisht njëjtë si Figura 2 me të gjitha hyrjet e mundshme.

#### Përgjigje:

Shqyrtojmë argumentet, -2 abc; Figura 2 e paraqet një gabim me argumentin e parë dhe nuk bën më llogaritje; Figura 4 e gjuan një përjashtim (exception) dhe përfundon menjëherë.

Për ta bërë Figurën 4 të sillet në mënyrë identike me Figurën 2, urdhëri i tretë i metodës main duhet të ndërrohet, të themi, `int centët = -1`, dhe ndarja e vlerës, `centët = new Integer(args[1]).intValue()`, duhet të zhvendoset brenda krahut-then të urdhërit të dytë kushtëzues.

### **(6.2.2 Problemet Sintaksore me Kushtëzues)**

## 6.3 Veprimet Relacionale (Operatorët Logjikë)

1. Llogariti përgjigjet për secilën nga shprehjet vijuese. Supozo se `int x = 2` dhe `double y = 3.5`.

- a) `(x > 1) && ((2*x) <= y)`
- b) `!(x == 1)`
- c) `(x >= 0 && x <= 1) || (1 <= y)`
- ç) `x > 0 && x < 10 && (y == 3)`

Përgjigje:

- a)
  - `(x > 1) && ((2*x) <= y)`
  - `=> (2 > 1) && ((2*2) <= y)`
  - `=> true && ((2*2) <= y)`
  - `=> true && ((2*2) <= y)`
  - `=> true && (4 <= y)`
  - `=> true && (4 <= 3.5)`
  - `=> true && false`
  - `=> false`
- b)
  - `!(x == 1) => !(2 == 1) => !false => true`
- c)
  - `(x >= 0 && x <= 1) || (1 <= y)`
  - `=> (2 >= 0 && x <= 1) || (1 <= y)`
  - `=> (true && x <= 1) || (1 <= y)`
  - `=> (true && 2 <= 1) || (1 <= y)`
  - `=> (true && false) || (1 <= y)`
  - `=> false || (1 <= y)`
  - `=> false || (1 <= 3.5)`
  - `=> false || true`
  - `=> true`
- ç)
  - `x > 0 && x < 10 && (y == 3)`
  - `=> 2 > 0 && x < 10 && (y == 3)`
  - `=> true && x < 10 && (y == 3)`
  - `=> true && 2 < 10 && (y == 3)`
  - `=> true && true && (3.5 == 3)`
  - `=> true && false`
  - `=> false`

2. Nëse është dhënë kjo metodë,

```
public int minus(int arg1, int arg2)
{
    int përgjigjja;
    if ( arg1 < 0 || arg2 > arg1 )
    {
        përgjigjja = -1;
    }
    else { përgjigjja = arg1 - arg2; }
    return përgjigjja;
}
```

çfarë rezultate kthehen nga `minus(3, 2)?` `minus(2, 3)?` `minus(-4, -5)?` `minus(4, -5)?`

Përgjigje:

```

minus(3, 2) kthen 1
minus(2, 3) kthen -1
minus(-4, -5) kthen -1
minus(4, -5) kthen 9

```

**3. Për secilën nga këto metoda, shkruaje një trup që përmban vetëm një urdhër kushtëzues. (Testi i kushtëzuesit i përdor veprimet relacionale.)**

a)

```

/** ështëIThjeshtëIVogël tregon se a është argumenti numër i thjeshtë
 * më i vogël se 10
 * @param i - argumenti
 * @return (kthen) true, nëse argumenti është nr. i thjeshtë, < 10 ;
 * return (kthen) false, përndryshe. */
public boolean ështëIThjeshtëIVogël(int i)

```

b)

```

/** pjesëto e bën pjesëtimin e dy argumenteve të veta
 * @param x - vlerë jonegative
 * @param y - vlerë jo e barabartë me zero
 * @return (kthen) (x / y), nëse vlejnë kushtet e mësipërme për x e y;
 * return (kthen) 0, përndryshe. */
public double pjesëto(double x, double y)

```

Përgjigje:

a)

```

public boolean ështëIThjeshtëIVogël(int i)
{ boolean përgjigjja;
  if ( i == 2 || i == 3 || i == 5 || i == 7 )
    { përgjigjja = true; }
  else { përgjigjja = false; }
  return përgjigjja;
}

```

**Vërejtje:** trupi i metodës mund të zëvendësohet me këtë urdhër të vetëm:

```

return ( i == 2 || i == 3 || i == 5 || i == 7 );

```

b)

```

public double pjesëto(double x, double y)
{ double përgjigjja = 0;
  if ( x >= 0 && y != 0 )
    { përgjigjja = x / y; }
  return përgjigjja;
}

```

## 6.4 Zbatimi i Urdhërave Kushtëzues

1. Testoje `ora12she` me secilën nga këto kohë: 9,45; 23,59; 0,01; 50,50; -12,-12; 24,0.

Përgjigje: Ta zëmë se `ora12she` gjendet në klasën `KonvertuesIKohës`:

```
public class TestoKohën
{ public static void main(String[] args)
  { KonvertuesIKohës k = new KonvertuesIKohës();
    String koha = k.ora12she(9, 45);
    System.out.println(koha);
    // Në të vërtetë, është më thjesht të testohet secili rast
    // me nga një urdhër:
    System.out.println(k.ora12she(23, 59));
    System.out.println(k.ora12she(0, 01));
    System.out.println(k.ora12she(50, 50));
    System.out.println(k.ora12she(-12, -12));
    System.out.println(k.ora12she(24, 0));
  }
}
```

2. Përdore `ora12she` në një aplikacion që e merr kohën momentale (duke e përdorur një objekt `GregorianCalendar`), e konverton në string (duke e përdorur `ora12she`), dhe e afishon stringun në dritaren komanduese.

Përgjigje:

Ta zëmë se `ora12she` gjendet në klasën `KonvertuesIKohës`:

```
import java.util.*;
public class TestoKohën2
{ public static void main(String[] args)
  { KonvertuesIKohës k = new KonvertuesIKohës();
    GregorianCalendar koha = new GregorianCalendar();
    String s =
      k.ora12she(koha.get(Calendar.HOUR_OF_DAY),
        koha.get(Calendar.MINUTE));
    System.out.println(s);
  }
}
```

**3. Shkruaje një metodë që e plotëson këtë specifikim:**

```
/** përkthejeNotën i konverton poenat numerikë në notë-shkronjë.
 * @param poenat - poena numerikë që duhet të jenë në rangun 0..100
 * @return (kthen) një notë-shkronjë bazuar në këtë shkallë:
 *      100..90 = "A"; 89-80 = "B"; 79..70 = "C"; 69..60 = "D"; 59..0 = "F" */
public String përkthejeNotën(int poenat)
```

Përdore metodën në një aplikacion që e lut përdoruesin për t'i shkruar poenat numerikë dhe e afishon notën-shkronjë përkatëse.

**Përgjigje:**

```
public String përkthejeNotën(int poenat)
{ String nota = "F";
  if ( poenat >= 90 )
  { nota = "A"; }
  else { if ( poenat >= 80 )
        { nota = "B"; }
        else { if ( poenat >= 70 )
                { nota = "C"; }
                else { if ( poenat >= 60 )
                        { nota = "D"; }
                        // përndryshe, nota mbetet "F"
                }
        }
  }
  return nota;
}
```

**4. Përmirësoje aplikacionin PërimtoPërsëri ashtu që**

a) përgjigjet prej zero monedhash nuk afishohen. Për shembull, t'imat për 0 Euro e 7 centë duhet të afishojnë vetëm

```
pesa = 1
njëshe = 2
```

b) nëse nevojitet vetëm një lloj i monedhës për të bërë t'ima, atëherë përdoret emër njëjës (dhe jo shumës) për emërtim, p.sh. për 0 Euro e 91 centë, aplikacioni afishon

```
1 gjysmë
2 njëzeta
1 njësh
```

**(6.5 Ndryshimi i rrjedhës së kontrollit)****(6.6 Urdhëri Switch)****(6.7 Komponentet Model dhe Kontrollues)**

## 6.8 Menaxher Kontosh të Bankës (Case Study)

1. Qe disa ushtrime të vogla lidhur me testimin:

- a) Shkruaje një testues-kontrollues që e krijon një objekt `LlogariEBankës`, i depoziton 1000, i tërheq 700, dhe pyet për balansin. Afishoje balansin në dritaren komanduese.
- b) Shkruaje një testues-kontrollues që e krijon një objekt `LexuesIBankës` dhe kërkon nga objekti ta lexojë një komandë. Kontrolluesi e kopjon kodin e komandës dhe sasinë në dritaren komanduese.
- c) Zgjerije testin që e bërë në `LlogariEBankës` për ta përfshirë edhe një objekt `ShkruesIBankës`. Përdore metodën `tregoTransaksionin` të objektit `ShkruesIBankës` për t'i afishuar rezultatet e depozitit dhe tërheqjes.
- ç) Përfundimisht, shkruaje një testues që e përdor një objekt `LexuesIBankës` për ta lexuar një komandë të depozitimit, e vendos depozitin në një objekt `LlogariEBankës`, dhe i tregon një objekti `ShkruesIBankës` që ta afishojë rezultatin.

Përgjigje:

- a)

```
public class TestoLlogarine
{
    public static void main(String[] args)
    {
        LlogariEBankes llog = new LlogariEBankes(0);
        llog.depozito(1000);
        boolean mire = llog.terhiqi(700);
        System.out.println("balansi = " + llog.merreBalansin());
    }
}
```
- b)

```
public class TestoLexuesin
{
    public static void main(String[] args)
    {
        LexuesIBankes l = new LexuesIBankes();
        char c = l.lexoKomanden("Shkruaje nje komande: ");
        System.out.println("Komanda ishte: " + c);
        int i = l.lexojeSasine();
        System.out.println("Sasia ishte: " + i);
    }
}
```

c)

```
import javax.swing.*;
public class TestoLlogarineShkruesin
{ public static void main(String[] args)
  { LlogariEBankes llog = new LlogariEBankes(0);
    ShkruesIBankes sh = new ShkruesIBankes("Shkruesi", llog);
    llog.depozito(1000);
    sh.tregojeTransaksionin("depozit", 1000);
    JOptionPane.showMessageDialog(null, "Shtypni OK");
    boolean mire = llog.terhiqi(700);
    sh.tregojeTransaksionin("terheqje", 700);
  }
}
```

ç)

```
public class TestoKrejt
{ public static void main(String[] args)
  { LlogariEBankes llog = new LlogariEBankes(0);
    LexuesIBankes l = new LexuesIBankes();
    ShkruesIBankes sh = new ShkruesIBankes("Shkruesi", llog);
    String s = "" + l.lexoKomanden("Shkruani dicka:");
    int i = l.lexojeSasine();
    llog.depozito(i);
    sh.tregoTransaksionin(s, i);
  }
}
```

## 2. Shtoja këtë metodë klasës LlogariEBankës:

```
/** depozitoInteresin e rrit llogarinë me një pagesë të interesit,
 * të llogaritur si (shkalla_e_interesit * balansi_momental)
 * @param shkalla - shkalla e interesit, vlerë ndërmjet 0.0 dhe 1.0
 * @return (kthen) sasinë e depozituar në llogari;
 * nëse depoziti nuk mund të kryhetm kthen 0 */
public int depozitoInteresin(double shkalla)
```

dhe shtojë komandën hyrëse, I dd.d, që e lejon përdoruesin që ta rrisë balansin momental për dd.d%.

Përgjigje: Metoda e re i shtohet modelit:

```
public class LlogariEBankes
{ ...
  public int depozitoInteresin(double shkalla)
  { int sasia = 0;
    if ( shkalla < 0.0 || shkalla > 1.0 )
    { System.out.println(
      "Gabim te LlogariEBankes: shkalle e parregullt e interesit");
    }
    else { sasia = (int)(shkalla * balansi);
      depozito(sasia);
    }
    return sasia;
  }
}
```

Kontrolluesi rregullohet që të lexojë:

```
public class KontrolluesILlogarise
{
    ...
    public void perpunoTransaksionet()
    {
        ...
        else if ( komanda.equals("I") )
        {
            double shkalla = lexuesi.lexoShkallenEInteresit();
            sasia = llogaria.depozitoInteresin(shkalla);
            shkruesi.tregoTransaksionin("Depozit i interesit: ", sasia);
        }
        ...
    }
}
```

Dhe kjo e kërkon një metodë të re, `lexoShkallënEInteresit`, në pamjen hyrëse:

```
public class LexuesIBankes
{
    ...
    /** lexoShkallenEInteresit e lexon nje shkalle te interesit perqindje
     * @return shkallen, te konvertuar ne thyese (numer me presje) */
    public double lexoShkallenEInteresit()
    {
        ... } // programoje këtë të duket ngjashëm me lexoSasinë
    }
}
```

**3. Konstruktori për klasën `LlogariEBankës` e lejon një llogari të re të bankës të fillojë me vlerë jonegative. Ndryshoje (modifikoje) kontrolluesin, class `MenaxherILlogarisë`, ashtu që në fillim, e pyet përdoruesin për vlerën fillestare që do ta vendosë në llogarinë e re të krijuar.**



### 6.8.10 Objekte të Shumëfishta nga e Njëjta Klasë

1. Ripunoje UdhëheqësILlogarisë2 ashtu që mund t'i transferojë fondet nga llogaria e parë në llogarinë e dytë. Komanda e re, >, e përcjellur nga një sasi, e tërheq sasinë nga llogaria e parë dhe e depoziton në llogarinë e dytë. Pastaj, shtojë komandën, <, e cila i transferon paratë nga llogaria e dytë në të parën.

#### Përgjigje:

```
public class KontrolluesILlogarise2
{
    ...
    public void perpunoTransaksionet()
    {
        ...
        case '>':
        {
            int sasia = lexuesi.lexoSasine();
            boolean mire = llogaria_primare.terhiqi(sasia);
            if ( mire )
            {
                llogaria_sekondare.depozito(sasia);
                shkruesi_primari.tregoTransaksionin("Transfer: Euro ", sasia);
                shkruesi_sekondar.tregoTransaksionin("Depozit: Euro ", sasia);
            }
            else { shkruesi_primari.tregoTransaksionin("transferi jovalid", sasia);
                break;
            }
        }
        ...
    }
}
```

Ngjashëm shkruhet kodi edhe për rastin '<'.

## (6.9 Testimi i Metodave dhe Klasave)

## Shtojcë: Këshilla Praktike për Programim

(artikuj origjinalë nga Ridvan Bunjaku)

### Emërtimi në Gjuhën Shqipe i Emrave në Kod

**Ë**SHTË ME RËNDËSI që emërtimi i entiteteve të programit (ndryshoret, fushat, metodat, klasët) të bëhet në mënyrë të përshtatshme. Kjo e lehtëson programimin. Poashtu e lehtëson edhe lexueshmërinë e kodit dhe e rrit qartësinë e tij. Nëse programi ka emra të përshtatshëm të entiteteve, më vonë është shumë lehtë t'i kthehemi që ta zgjerojmë e avansojmë edhe më tej.

Po çka, në të vërtetë, domethënë t'i emërojmë entitetet në mënyrë të përshtatshme? Së pari po përmendim një standard të zakonshëm të emërimit që përdoret në literaturën e programimit të mirë. Ky standard mundëson që t'i dallojmë fare lehtë në kod ndryshoret, metodat (funksionet) dhe klasët ndërmjet vete. Në pak fjalë, sipas këtij standardi:

- klasët emërtohen ashtu që fillojnë me shkronjë të madhe dhe vazhdojnë me shkronja të vogla. Nëse emri i klasës përbëhet prej më shumë se një fjale, atëherë fjala dallohet ashtu që fillohet me shkronjë të madhe. P.sh.

`Katror, TopLevizes, ModelIKartes, LlogariEBankes etj.`

- metodat, funksionet emërtohen ashtu që fillojnë me shkronjë të vogël dhe vazhdojnë poashtu me shkronjë të vogël. Nëse emri ka më shumë se një fjale, atëherë fjala tjetër dallohet duke filluar me shkronjë të madhe. Dhe, natyrisht, pas emrit vijnë kllapat me argumentet përkatëse (nëse ka). P.sh.

`depozito(), afishoBalansin(), vizatojeTopin(),  
gjejeHipotenuzen(kateta_1, kateta_2)`

Përfundim prej kësaj rregulle bën vetëm konstruktori i klasës i cili, siç e dimë, e ka emrin e njëjtë me të klasës së vet, pra ai fillon me shkronjë të madhe. P.sh.

`Katror(brinja), TopLevizes(x, y, rrezja), ModelIKartes(ngjyra),  
LlogariEBankes(200)`

- ndryshoret në program fillojnë me shkronjë të vogël dhe shkruhen tërësisht me shkronja të vogla. Nëse emri përbëhet prej dy ose më shumë fjalësh, distanca midis fjalëve shkruhet me 'vizë poshtë', dmth. me '\_'. P.sh.

`brinja_a, siperfaqja_e_katrorit, gjeresia, lartesia`

Në kodin që është i shkruar sipas këtyre rregullave, drejtpërdrejt mund të përcaktohet se çka është 'filan emri':

- Nëse fillon me shkronjë të madhe dhe nuk ka kllapa, atëherë është klasë.
- Nëse fillon me shkronjë të madhe por ka kllapa, atëherë është konstruktor.
- Nëse fillon me shkronjë të vogël dhe ka kllapa, është metodë, funksion. Këtu ndihmon edhe ajo se kur emri përbëhet dy a më tepër fjalëve, fjala tjetër fillon me të madhe.
- Nëse emri është krejt me shkronja të vogla dhe distancat eventuale ndërmjet fjalëve janë '\_', atëherë kjo është ndryshore, variabël, ose fushë.

Si shembull për ilustrimin e këtyre rregullave, do ta shqyrtojmë programin vijues:

```
import java.awt.*;
import javax.swing.*;
/** Vizatim i vizaton tri veze nje-mbi-nje */
public class Vizatim extends JPanel
{ int gjeresia = 200;
  int lartesia = gjeresia * 2 / 3;

  /** Konstruktori Vizatim e konstrukton objektin */
  public Vizatim()
  { JFrame korniza_ime = new JFrame();
    korniza_ime.getContentPane().add(this);
    korniza_ime.setTitle("Vizatimi (c)");
    korniza_ime.setSize(gjeresia, lartesia);
    korniza_ime.setVisible(true);
  }

  /** paintComponent e kryen vizatimin: 3 veze nje mbi nje
   * @param g - penda grafike */
  public void paintComponent(Graphics g)
  { g.setColor(Color.white);
    g.fillRect(0, 0, gjeresia, lartesia);
    g.setColor(Color.yellow);
    int qh = gjeresia / 2; // qendra horizontale
    int qv = lartesia / 2; // qendra vertikale
    int r1 = gjeresia / 10; // gjysme-diametri i pare (horizontal)
    int r2 = lartesia / 10; // gjysme-diametri i dyte (vertikal)
    g.fillOval(qh - r1, qv + r2, 2*r1, 2*r2);
    g.fillOval(qh - r1, qv - r2, 2*r1, 2*r2);
    g.fillOval(qh - r1, qv - 3*r2, 2*r1, 2*r2);
  }

  /** main per ta konstruktuar objektin */
  public static void main(String[] args)
  { new Vizatim(); }
}
```

Këtu, klasë është `Vizatim`, me konstruktorin e vet `Vizatim()`, pastaj `JFrame` dhe `Graphics`, që janë përkufizuar në klasat e importuara (me urdhërat `import`). Vërejmë se klasa dallohet nga konstruktori i vet me atë se konstruktori i ka kllapat, ndërsa klasa nuk i ka.

Metoda janë `paintComponent(Graphics g), main(String[] args), getContentPane(), add(Component this), setTitle(String titulli), setSize(int gjeresia, int lartesia), setVisible(boolean duket),`  
`pastaj setColor(Color c), fillRect(int x1, int y1, int gjeresia, int lartesia)`  
`dhe fillOval(int x1, int y1, int gjeresia, int lartesia).`

Ndërsa ndryshore janë: `gjeresia, lartesia, korniza_ime, qh, qv, r1 dhe r2.`

Vërejmë se nuk kemi nevojë ta 'lodhim kryet' shumë për ta kuptuar se `gjeresia` është ndryshore e jo, bie fjala, metodë apo klasë. (Po të ishte metodë, do të kishte kllapa: `gjeresia()`, po të ishte klasë do të fillonte me shkronjë të madhe: `Gjeresi`).

## Disa Specifika të Gjuhës Shqipe Rreth Emërtimit

Gjuhët programuese zakonisht janë dizajnuar ashtu që të jetë e përshtatshme të shkruhet kod në gjuhën angleze. Anglishtja edhe ashtu ka strukturë të thjeshtë gramatikore: nuk ka rasa. Emri zakonisht e ka vetëm një trajtë. Kjo gjë nuk ndodh në gjuhën shqipe, ku emri i ka pesë rasa. Në shqip, emri ka edhe trajtë - mund të jetë i shquar: topi, ose i pashquar: top. (Në anglisht trajta e shquar realizohet me 'the', por emri nuk pëson ndryshim.)

Këtu do të japim disa këshilla praktike se si të emërtohen entitetet në kod ashtu që kodi të jetë i kuptueshëm edhe kur përdorim emra shqip për identifikatorë.

### Klasët

Klasët emërtohen si emra në trajtë të pashquar. P.sh.

`Top, Letër, Vizatues, LlogariEBankës`

Përdorimi i këtillë i jep kuptim kodit. P.sh. kodi

`LlogariEBankës llogaria_ime;`

ka kuptim dhe interpretohet kështu: `llogaria_ime` është `LlogariEBankës`.

Më e paqartë do të ishte,

`LlogariaEBankës llogaria_ime;`

Tash s'po kuptohet qartë se cila është `llogaria` (ndryshorja), e cila është klasa (tipi i ndryshores).

## Metodat

Metodat emërtohen si folje në mënyrën urdhërore. P.sh.

```
llogariteRrogen(), vizatojeKubin(5),
shkruajeFjalinë("Liza në botën e çudirave"),
përgjithësisht: kryejePunën(...)
```

Kjo rregull është e përshtatshme sepse metodat, funksionet nënkuptojnë veprim, aksion që duhet ta bëjë kompjuteri. Folja është ajo që e përshkruan veprimin, ndërsa mënyra urdhërore i përshtatet asaj se ne e urdhërojmë kompjuterin ta kryejë një funksion apo metodë.

## Ndryshoret, fushat

Ndryshoret, fushat emërtohen si emra në trajtë të shquar. P.sh.

```
gjatësia, gjerësia, bilansi, ngjyra
```

Këtu trajta e shquar është më e përshtatshme sesa trajta e pashquar. P.sh. do të ishte e palogjikshme të shkruajmë

```
gjatësi = 5;
```

Në këtë mënyrë, kodi i shkruar me identifikatorë në gjuhën shqipe do të ishte i kuptueshëm që në shikim të parë.

Shembuj të këqinj të përdorimit të emrave:

### Klasë:

```
Rrethi
bilans
Vizatoje
```

### Gabimi:

(është në trajtë të shquar)  
(shkronja e parë e vogël)  
(është folje, duhet emër)

### Si duhet:

```
Rreth
Bilans
Vizatues
```

### Metoda:

```
llogaritjaERroges() (është emër, duhet folje)
gjeje_mesin() (distancë me '_')
```

```
llogariteRrogen()
gjejeMesin()
```

### Ndryshore:

```
Rrezja (fillon me të madhe)
qendraERrethit (fjala tjetër fillon me të madhe)
ngjyrë (është në trajtë të pashquar)
```

```
rrezja
qendra_e_rrethit
ngjyra
```

**Metodat standarde `get` dhe `set`**

Shpesh nëpër kode nevojitet të mirret vlera e ndonjë fushe (të dhëne private). Kjo në kodet standarde anglisht bëhet me metodat `get`. Ndërsa kur kërkohet t'i jipet një vlerë ndonjë fushe, përdoren metodat `set`. P.sh.

```
getColor(), getBalance(), getSize()
```

Shqip këto metoda do të emërtoheshin:

```
merreNgjyren(), merreBalansin(), merreMadhesine()
```

Nganjëherë hasim edhe në kësi trajta të ngjashme:

```
colorOf(), nameOf(), numberOf()
```

të cilat shqip do të mund të shkruheshin si:

```
ngjyraE(), emriI(), numriI()
```

Për metodat `set`, ndoshta fjala më e përshtatshme shqip do të ishte: cakto. Domethënë,

```
setSize(...), setName(), setBalance()
```

në shqip do të ishin:

```
caktoMadhesine(...), caktoEmrin(...), caktoBalansin(...)
```

Ja një shembull i një klase i shkruar sipas këtij sistemi të emërimit:

```
/** TopLevizet e modelon nje top qe leviz */
public class TopLevizet
{ private int x_poz; // x-pozita e qendres se topit
  private int y_poz; // y-pozita e qendres se topit
  private int rrezja; // rrezja e topit

  private int x_shpejtesia = +5; // shpejtesia horizontale;
  private int y_shpejtesia = +2; // shpejtesia vertikale;

  private Kutia permbajtesi; // permbajtesi ku leviz topi

  /** Konstruktori TopLevizet e konstrukton topin.
   * @param x_fillues - qendra (hori.) e pozites filluese te topit
   * @param y_fillues - qendra (vert.) e pozites filluese te topit
   * @param r - rrezja e topit
   * @param kutia - permbajtesi ne te cilin udheton topi */
```

```
public TopLevizes(int x_fillues, int y_fillues, int r, Kuti kutia)
{
    x_poz = x_fillues;
    y_poz = y_fillues;
    rrezja = r;
    permbajtesi = kutia;
}

/** xPozita e kthen poziten horizontale aktuale te topit */
public int xPozita()
{ return x_poz; }

/** yPozita e kthen poziten vertikale aktuale te topit */
public int yPozita()
{ return y_poz; }

/** rrezjaE e kthen rrezen e topit */
public int rrezjaE()
{ return rrezja; }

/** leviz e leviz topin
 * @param njesi_kohe - sasia e kohes prejse eshte levizur topi
 * heren e fundit */
public void leviz(int njesi_kohe)
{
    x_poz = x_poz + (x_shpejtesia * njesi_kohe);
    if ( permbajtesi.neKontaktHorizontal(x_poz) )
        { x_shpejtesia = -x_shpejtesia; } // ktheje drejtimin
                                           // horizontal mbrapa
    y_poz = y_poz + (y_shpejtesia * njesi_kohe);
    if ( permbajtesi.neKontaktHorizontal(y_poz) )
        { y_shpejtesia = -y_shpejtesia; } // ktheje drejtimin
                                           // vertikal mbrapa
}

public static void main(String[] args)
{
    Kuti kutia = new Kuti(50); // me madhesi 50 me 50 (piksela)
    TopLevizes topi_im = new TopLevizes(25, 25, 10, kutia);
                                // rrezja eshte 10 piksela
    while ( true )
    {
        topi_im.leviz(1); // 1 njesi te kohes se kaluar
        System.out.println("x = " + topi_im.xPozita()
            + "; y = " + topi_im.yPozita());
    }
}
}
```

Vërejtje: Shembulli është i lidhur me klasën `Kuti`.

## Nivelet në Kod dhe Hierarkia

Programi i shkruar mirë është i lexueshëm në atë masë që të shihet algoritmi drejtpërdrejt nga kodi. Shkruarja e programeve të bukura, të lexueshme, të qarta dhe të kuptueshme është art më vete në programim. Këtu do ta prezentojmë një sistem të nivelizimit të kodit që e bën atë të duket mirë e qartë.

Për t'i dalluar pjesët e programit (klasët, metodat, urdhërat) ndërmjet vete, kodi organizohet në nivele.

Niveli i parë është niveli i klasës. Klasët shkruhen në fillim të rreshtit, pa u futur fare 'brenda', dmth. pa u shtyrë fare djathtas:

```
/** TopLevizes e modelon nje top qe leviz */
public class TopLevizes
{
}
}
```

Niveli i dytë, që është një Tab (ose dy karakterë) brenda, është niveli i metodave dhe i fushave të klasës (ndryshoreve globale të klasës):

```
/** TopLevizes e modelon nje top qe leviz */
public class TopLevizes
{ private int x_poz; // x-pozita e qendres se topit
  private int y_poz; // y-pozita e qendres se topit
  ...
  /** Konstruktori TopLevizes e konstrukton topin... */
  public TopLevizes()
  {

  }

  /** rrezjaE e kthen rrezen e topit */
  public int rrezjaE()
  {

  }
}
```



Niveli i tretë është edhe një Tab (ose dy karakterë) djathtas. Ky është niveli i urdhërave të metodave, dmth. i trupave të metodave:

```
/** TopLevizet e modelon nje top qe leviz */
public class TopLevizet
{ private int x_poz;  // x-pozita e qendres se topit
  private int y_poz;  // y-pozita e qendres se topit
  ...
  /** Konstruktori TopLevizet e konstrukton topin... */
  public TopLevizet(int x_fillues, int y_fillues, int r, Kuti kutia)
  { x_poz = x_fillues;
    y_poz = y_fillues;
    rrezja = r;
    permbajtesi = kutia;
  }

  /** rrezjaE e kthen rrezjen e topit */
  public int rrezjaE()
  { return rrezja; }
}
```

Qe një grafikoni që e ilustron këtë organizim:

```
|----- (niveli i klasës)
|
public class NdonjeKlase
{ |----- (niveli i metodave)
| |
| public NdonjeKlase()
| { |--- (niveli i trupave te metodave)
| | | ...
| | |---
| | }
| |
| public void ndonjeMetode()
| { |--- (niveli i trupave te metodave)
| | | ...
| | | ...
| | |---
| | }
| |
| |-----
| }
|-----
```

## Urdhërat e Përbërë

Urdhëra të përbërë këtu i quajmë ata urdhëra që në vete mund të ngërthejnë bllok urdhërash. Të tillë janë: `if`, `while`, `for`. Ideja e përgjithshme është që fjala kyçe e urdhërit (`if`, `while`, përkatësisht `for`) të mos ketë urdhëra të trupit të vet nën të, por që trupi të jetë i shtyer djathtas mjaft. Me fjalën 'trup', këtu e quajmë bllokun e urdhërave, domethënë urdhërat që ndodhen brenda kllapave gjarpërore.

### Urdhëri if

```
if ( ... )
{ |
  |
  ...
  |
}
```

### Urdhëri if...else...

```
if ( ... )
{ |
  |
  ...
  |
}
else { |
      |
      ...
      |
}
```

### Urdhëri while

```
while ( ... )
{ |
  |
  ...
  |
}
```

### Urdhëri for

```
for ( ...; ...; ... )
{ |
  |
  ...
  |
}
```

Rregulla e njëjtë duhet të ruhet edhe kur kemi urdhër të përbërë përbrenda një urdhëri tjetër të përbërë:

```

if ( ... )
    { while ( ... )
      { |
        |
        ...
        |
      }
    }
else { for ( ...; ...; ... )
      { |
        |
        ...
        |
      }
    }

```

Ngjashëm veprohet për të gjitha kombinimet e urdhërave. Tash është më e qartë pse përdoren vetëm dy karakterë për t'i dalluar nivelet. Po të përdorim më shumë, për shembull tre ose katër karakterë, urdhërat e brendshëm na shkojnë shumë djathtas dhe na dalin prej pamjes. Dy karakterë mjaftojnë për ta dalluar nivelin nga niveli. Po përsërisim edhe njëherë se kodi i shkruar bukur është i qartë, vetë-shpjegues për të gjithë ata që e shikojnë, duke e përfshirë edhe vetë autorin e tij.

## Nxjerrja e ASCII kodeve

Kodi vijues, kur ekzekutohet, e nxjerr tabelen ASCII në rangun 32..255.

```

public class ASCII
{ public static void main(String[] args)
  { for (int i = ' '; i<=255; i++)
    { System.out.print((char)i + "=" + (i<100?" ":"") + i + " | ");
      if ( (i - 1) % 10 == 0 )
        { System.out.println(); }
    }
  }
}

```

Prej aty mund t'i lexoni vlerat për 'ë', 'Ë', 'ç' dhe 'Ç':

```

ë = Alt + 235
Ë = Alt + 203
ç = Alt + 231
Ç = Alt + 199

```

## Fjalët Kyçe të Gjuhës Programuese Java

<code>abstract</code>	<code>int</code>
<code>boolean</code>	<code>interface</code>
<code>break</code>	<code>long</code>
<code>byte</code>	<code>native</code>
<code>case</code>	<code>new</code>
<code>cast</code>	<code>null</code>
<code>catch</code>	<code>operator</code>
<code>char</code>	<code>outer</code>
<code>class</code>	<code>package</code>
<code>const</code>	<code>private</code>
<code>continue</code>	<code>protected</code>
<code>default</code>	<code>public</code>
<code>do</code>	<code>rest</code>
<code>double</code>	<code>return</code>
<code>else</code>	<code>short</code>
<code>extends</code>	<code>static</code>
<code>final</code>	<code>super</code>
<code>finally</code>	<code>switch</code>
<code>float</code>	<code>synchronized</code>
<code>for</code>	<code>this</code>
<code>future</code>	<code>throw</code>
<code>generic</code>	<code>throws</code>
<code>goto</code>	<code>transient</code>
<code>if</code>	<code>try</code>
<code>implements</code>	<code>var</code>
<code>import</code>	<code>void</code>
<code>inner</code>	<code>volatile</code>
<code>instanceof</code>	<code>while</code>



\* \* \* \* \* artridesign \* \* \* \* \*

