



Hyrja (input), dalja (output) dhe gjendja

Hyrja (input), dalja (output) dhe gjendja

Objektivat:

- ⑥ Të zbatohet hyrje interaktive, ku një aplikacion mund të ketë interaksion me shfrytëzuesin dhe të kërkojë input kur është e nevojshme për kompjutim.
- ⑥ Të shfrytëzohet inheritimi për të disenjuar dritare grafike që afishojnë output në forma teksti, ngjyrash dhe figurash.
- ⑥ Të tregohet si një objekt mund të „mbajë mend“ „gjendjen“ e kompjutimit me anë të variablave të fushave.

Hyrja (input) interaktive

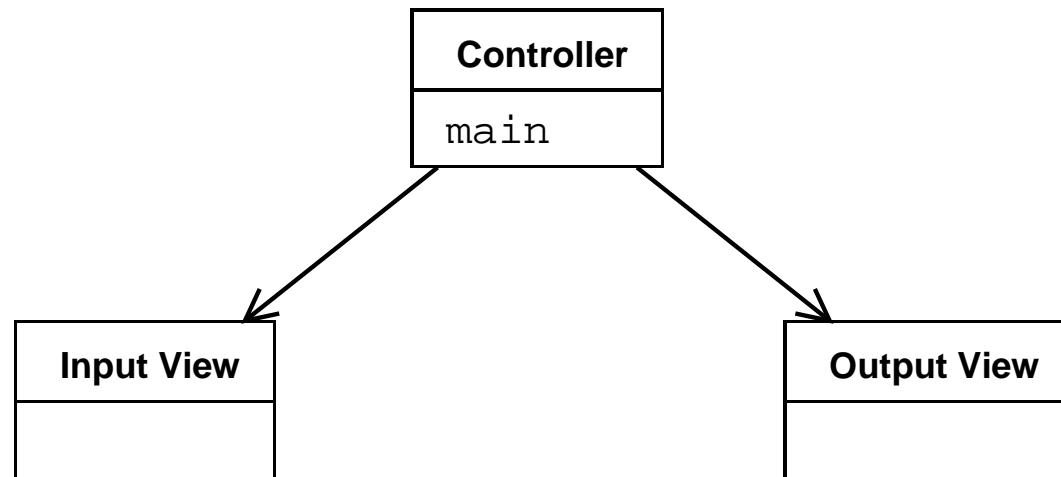


Figura 1. Një arkitekturë me input-view

Hyrja (input) interaktive – Vazhdim

Input view: komponentet e një aplikacioni të cilat pranojnë të dhëna hyrëse.

Output view: komponentet e një aplikacioni të cilat afishojnë të dhëna dalëse.

Hyrje interaktive: të dhënat hyrëse të cilat shfrytëzuesi i jep gjatë ekzekutimit të programit.

Dialog: dritare e cila afishon një mesazh të shkurtër dhe/ose pranon hyrje interaktive, p.sh., duke rradhitur shfrytëzuesi tekst ose duke shtypur një buton. Kur shfrytëzuesi të ketë përfunduar interaksionin me të, dialogu zhduket.

Hyrja (input) interaktive – Vazhdim

Hapat kryesorë për të shfrytëzuar klasën `JOptionPane` për gjenerim të dialogut për lexim hyrjeje:

1. Kërko në pakon `javax.swing` klasën `JOptionPane`:

```
import javax.swing.*;
```

2. Krijë dialogun në displej:

```
String input
```

```
= JOptionPane.showInputDialog(POROSIA);
```

ku *POROSIA* është një string i cili do të afishohet përbrenda dialogut.

Hyrja (input) interaktive – Vazhdim

```
import java.text.*;
import javax.swing.*;

/** FahrenheitToCelsius konverton vlerën e temperaturës
 *  input: gradët Fahrenheit, int
 *  output: gradët Celsius, double
 */
public class FahrenheitToCelsius
{ public static void main(String[] args)
  { String input = JOptionPane.showInputDialog("Gradë Fahrenheit:");
    int f = new Integer(input).intValue();
    double c = (5.0/9.0) * (f - 32);
    DecimalFormat formatter = new DecimalFormat("0.0");
    System.out.println("Për gradë Fahrenheit " + f + ",");
    System.out.println("gradë Celsius " + formatter.format(c));
  }
}
```

Hyrja (input) interaktive – Vazhdim

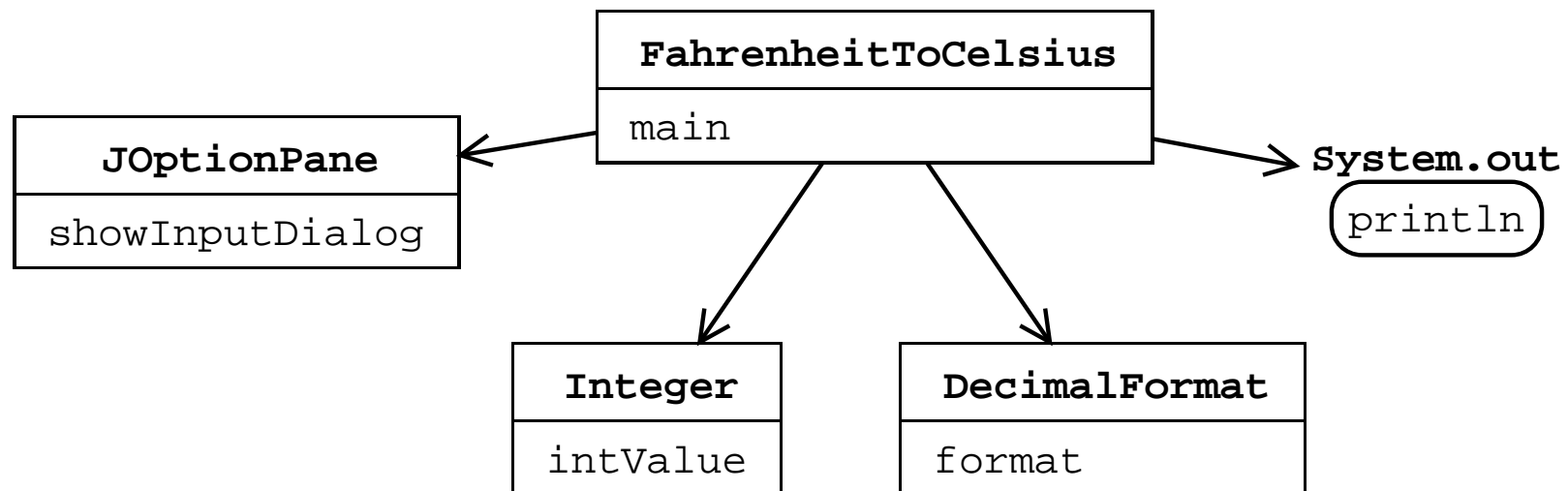


Figura 2. Arkitektura e aplikacionit për konvertimin e temperaturës

Dialogu dalës

Krijimi i një dialogu të mesazhit në displej:

```
JOptionPane.showMessageDialog(null, MESAZHI);
```

ku *MESAZHI* është stringu i cili do të afishohet në dialogun.

`null`: vlerë e posaçme në Java e cila shënon „nuk ka vlerë“. Përdorimi i `null` si operand në shprehje aritmetike ose dërgimi kësaj i një mesazhi gjeneron një përjashtim (run-time error).

Dialogu dalës – Vazhdim

```
import java.text.*;
import javax.swing.*;

/** FahrenheitToCelsius konverton vlerën e temperaturës
 *  input: gradët Fahrenheit, int
 *  output: gradët Celsius, double
 */
public class FahrenheitToCelsius
{ public static void main(String[] args)
  { String input = JOptionPane.showInputDialog("Gradë Fahrenheit:");
    int f = new Integer(input).intValue();
    double c = (5.0/9.0) * (f - 32);
    DecimalFormat formatter = new DecimalFormat("0.0");
    JOptionPane.showMessageDialog(null, f + " Fahrenheit janë "
                                + formatter.format(c) + " Celsius");
  }
}
```

Dalja (output) grafike



Output view për dalje grafike janë dritaret grafike, të cilat afishojnë tekst, figura, ngjyra, etj.

Dritaret grafike janë objekte.

Klasat e dritareve grafike shkruhen duke zgjeruar me inheritim (trashigimi) klasë nga bibliotekat e Java (nga të cilat mund të konstruktohen dritare boshe).

Panelat dhe kornizat e tyre

Frame (*korniza*): emërtimi në Java për dritare grafike.

Kornizat ndërtohen nga class `JFrame`, e cila ndodhet në pakon `javax.swing`.

```
import javax.swing.*;
/** FrameTest krijon një kornizë */
public class FrameTest
{
    public static void main(String[] args)
    {
        JFrame sampleFrame = new JFrame();
        System.out.println("Ku është korniza?");
    }
}
```

Panelat dhe kornizat e tyre – Vazhdim

```
import javax.swing.*;
/** FrameTest krijon dhe afishon një kornizë */
public class FrameTest
{ public static void main(String[] args)
  { JFrame sampleFrame = new JFrame();
    sampleFrame.setSize(300, 200);
    sampleFrame.setVisible(true);
    System.out.println("Korniza u shfaq!");
  }
}
```

Përshtatja e panelave me trashigimi

Panel: objekti në Java mbi të cilin vizatohet. Për të vizatuar mbi panel shfrytëzohet teknika e quajtur trashigimi (inheritance).

1. Duke u nisur nga `class JPanel`, shkruhet klasa e veçantë, p.sh., `class TestPanel`. Përbrenda kësaj klase përfshihen instruksionet për vizatim.
2. Konstruktohet një objekt nga `class TestPanel`, konstruktohet një kornizë nga `class JFrame`, dhe futet objekti panel në objektin kornizë.
3. Vëhet madhësia dhe dukshmëria e kornizës.

Përshtatja e panelave me trashigimi – Vazhdim

```
import java.awt.*;
import javax.swing.*;
public class TestPanel extends JPanel
{ public void paintComponent(Graphics g)
  { // ... instruktionet të cilat vizatojnë mbi panelin
  }
}
```

- ⑥ Nga `public class TestPanel` mund të konstruktohen objekte panela.
- ⑥ Fraza `extends JPanel` bën që `class TestPanel` ta zgjerojë/trashigojë `class JPanel`. `TestPanel` quhet nënklasë, `JPanel` quhet mbiklasë.
- ⑥ Instruktionet shitesë përbrenda `class TestPanel` përmbahen në metodën e re `paintComponent`. Metoda `paintComponent` startohet dhe ristartohet automatikisht.
- ⑥ Penda grafike `g` është objekti i cili vizaton mbi panelin. Penda grafike shfrytëzon klasa nga biblioteka `java.awt`.

Përshtatja e panelave me trashigimi – Vazhdim

```
import java.awt.*;
import javax.swing.*;
/** TestPanel krijon një panel që afishon një string */
public class TestPanel extends JPanel
{ /** paintComponent vizaton panelin
  * @param g - penda grafike */
  public void paintComponent(Graphics g)
  { g.setColor(Color.red);
    g.drawString("Tungjatjeta!", 30, 80);
  }
}
```

Përshtatja e panelave me trashigimi – Vazhdim

```
import javax.swing.*;
/** FrameTest krijon dhe afishon një kornizë,
 * dhe fut një panel në të. */
public class FrameTest3
{ public static void main(String[] args)
  { TestPanel samplePanel = new TestPanel();
    JFrame sampleFrame = new JFrame();
    sampleFrame.getContentPane().add(samplePanel);
    sampleFrame.setSize(300, 200);
    sampleFrame.setVisible(true);
    System.out.println("Korniza u shfaq e shkruar!");
  }
}
```


Përshtatja e panelave me trashigimi – Vazhdim

Trashigimia (*inheritance*): teknikë për lidhjen e dy klasave ashtu që njëra është „përshtatje“ ose zgjerim i tjetrës. Në qoftë se klasa C2 trashigon (zgjeron) klasën C1

```
class C2 extends C1
```

atëherë çdo objekt i konstruktuar nga `class C2` do të ketë tërë strukturën e definuar përbrenda `class C1` si dhe strukturën përbrenda `class C2`.

Nënklasa dhe *mbiklasa*: C2 është nënklasë e C1, kurse C1 është mbiklasë e C2.

Përshtatja e panelave me trashigimi – Vazhdim

FrameTest

```
main(...)  
{ samplePanel == a1  
  sampleFrame == a3  
  ...  
}
```

a2 : Graphics

```
setColor(...) { ... }  
drawString(...) { ... }  
...
```

[Objekti Graphics konstruohet
automatikisht me TestPanel.]

a1 : TestPanel

```
paintComponent(g == a2 )  
{ ... }  
Instruksionet nga klasa JPanel
```

a3 : JFrame

```
setSize(...) { ... }  
setVisible(...) { ... }  
...
```

Përshtatja e panelave me trashigimi – Vazhdim

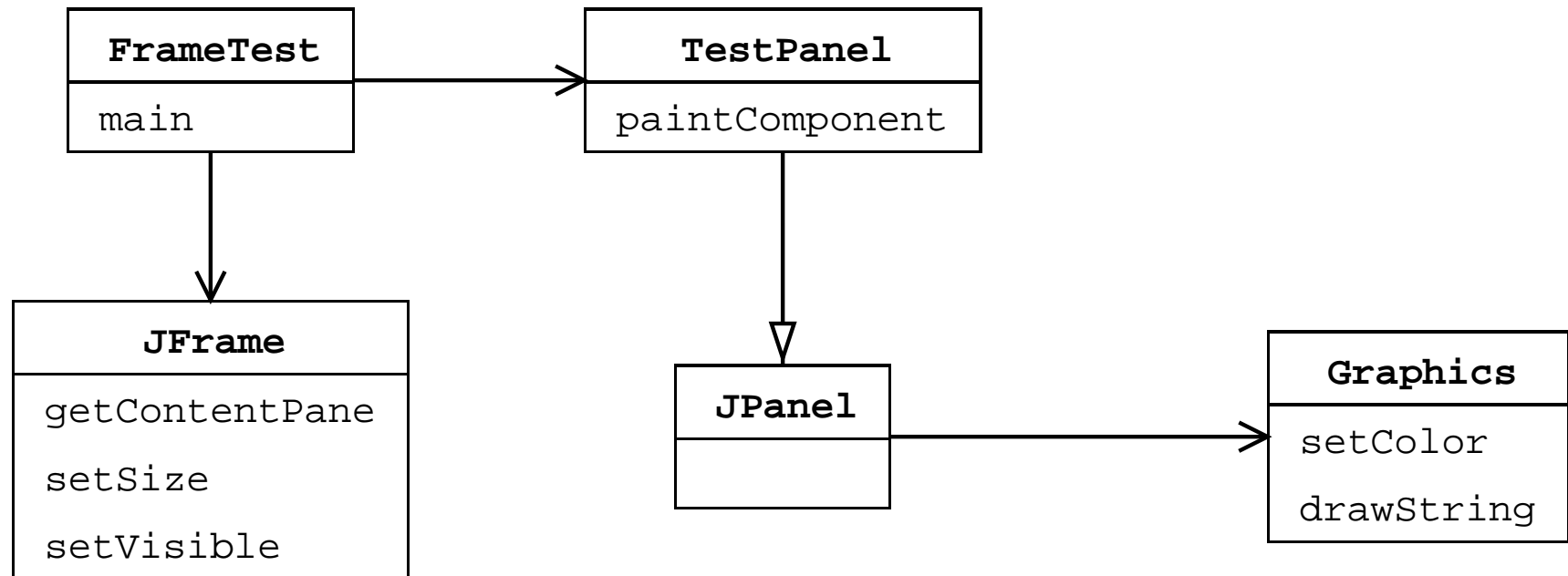


Figura 3. Arkitektura e aplikacionit me TestPanel

Përshtatja e panelave me trashigimi – Vazhdim

```
import java.awt.*;
import javax.swing.*;
/** MyPanel krijon një panel me ngjyra */
public class MyPanel extends JPanel
{ /** paintComponent vizaton mbi panelin
    * @param g -- penda grafike */
    public void paintComponent(Graphics g)
    { g.setColor(Color.red);
      int left = 100;
      int top = 70;
      int width = 90;
      int height = 60;
      g.drawRect(left, top, width, height);
      int diameter = 40;
      g.fillOval(left + width - diameter, top, diameter, diameter);
    }
}
```

Përshtatja e panelave me trashigimi – Vazhdim

```
import java.awt.*;
import javax.swing.*;
/** FrameTest2 afishon një dritare me ngjyra */
public class FrameTest2
{ public static void main(String[] args)
  { JFrame myFrame = new JFrame();
    myFrame.getContentPane().add(new MyPanel());
    myFrame.setTitle("Vizatuesi im i kornizave");
    myFrame.setBackground(Color.white);
    int width = 300;
    int height = 200;
    myFrame.setSize(width, height);
    myFrame.setVisible(true);
  }
}
```

Përshtatja e panelave me trashigimi – Vazhdim



Ngjyrosje e dritares: vizatimi i tekstit, ngjyrave dhe figurave në dritaren grafike; kryhet përmes metodës `paintComponent`.

Pendë grafike: objekti i shfrytëzuar nga metoda `paintComponent` për ngjyrosjen e dritares.

Piksel: njëra nga „pikat“ e imazhit; shfrytëzohet si njësi matëse për distanca përbrenda dritares grafike.

Formati dhe metodat për vizatim

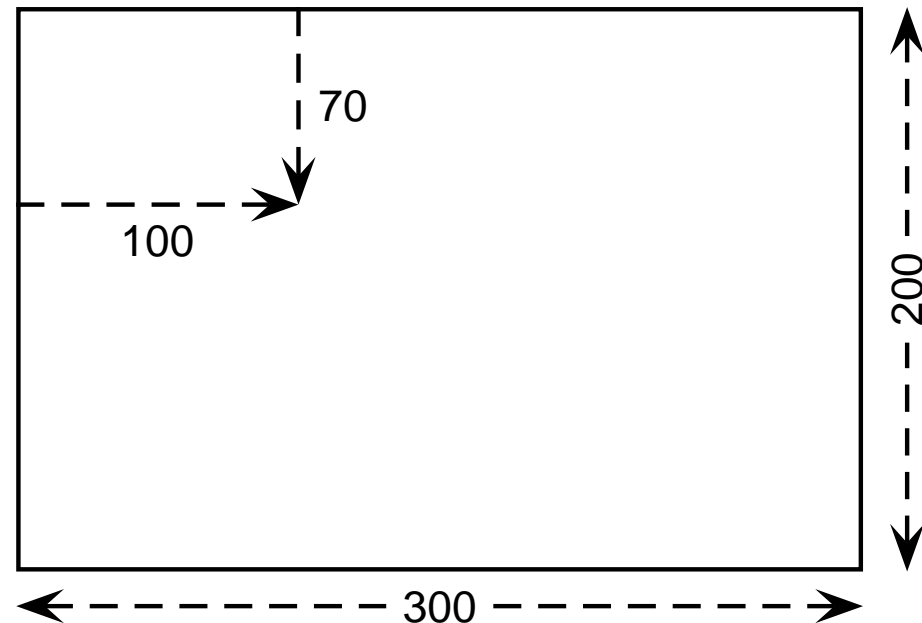


Figura 4. Alokimi i pikselit me koordinata (100,70)

Formati dhe metodat për vizatim – Vazhdim

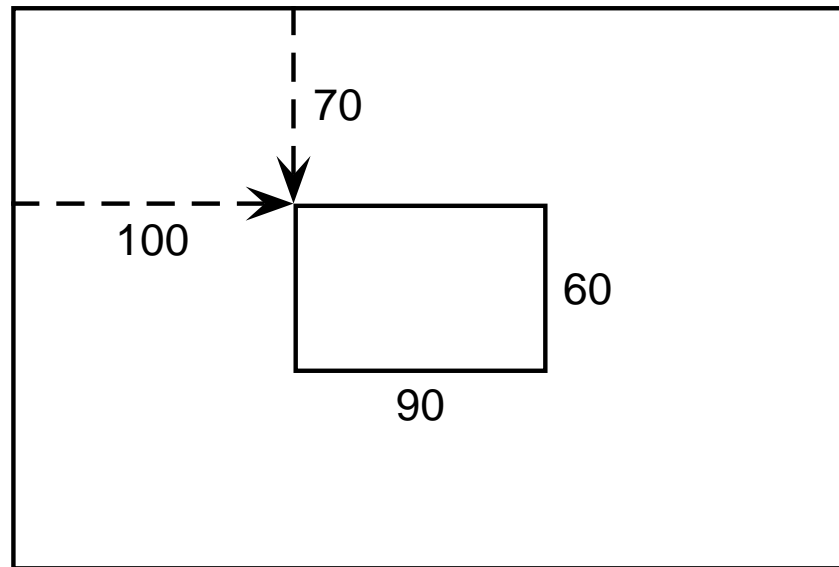


Figura 5. `g.drawRect(100, 70, 90, 60)`

Formati dhe metodat për vizatim – Vazhdim

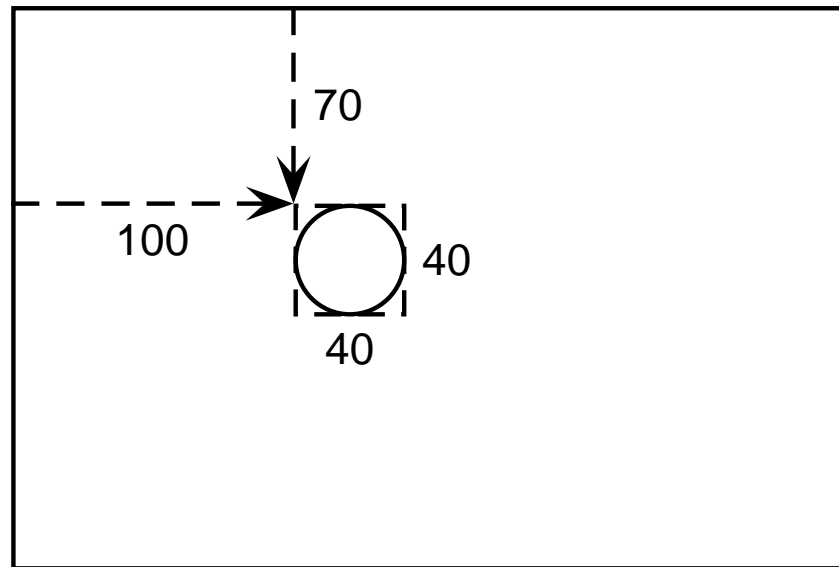


Figura 6. `g.drawOval(100, 70, 40, 40)`

Formati dhe metodat për vizatim – Vazhdim

Metoda	Semantika
<code>setColor(c)</code>	Mbush pendën me ngjyrën
<code>drawLine(x1, y1, x2, y2)</code>	Vizaton segmentin
<code>drawString(s, x, y)</code>	Afishon stringun të pozicionuar sipas koordinatave
<code>drawRect(x, y, width, height)</code>	Vizaton drejtkëndëshin
<code>fillRect(x, y, width, height)</code>	Mbush drejtkëndëshin
<code>drawOval(x, y, width, height)</code>	Vizaton elipsën
<code>fillOval(x, y, width, height)</code>	Mbush elipsën
<code>fillArc(x, y, width, height, startAngle, thickness)</code>	Vizaton pjesën e harkut të elipsës

Tabela 1. Disa metoda të pendës grafike (class Graphics)

Konstruktor metodat dhe objekti

this

Metodë konstruktor: metodë përbrenda një klase që ekzekutohet kur të konstruktohet një objekt nga klasa. Emri i konstruktorit është i njëjtë sikurse emri i klasës në të cilën paraqitet.

`this`: vlerë e posaçme në Java e cila shënon „këtë objekt“. Zbatohet për vetëreferencim të objekteve.

Konstruktor metodat dhe objekti

this – Vazhdim

```
import java.awt.*;
import javax.swing.*;
import java.util.*;
/** ClockWriter vizaton një orë */
public class ClockWriter extends JPanel
{ public ClockWriter()
  { int width = 200;
    JFrame clockFrame = new JFrame();
    clockFrame.getContentPane().add(this);
    clockFrame.setTitle("Ora");
    clockFrame.setBackground(Color.white);
    clockFrame.setSize(width, width);
    clockFrame.setVisible(true);
  }
}
```

Konstruktor metodat dhe objekti

this – Vazhdim

```
/** paintComponent vizaton orën
 * @param g - penda grafike */
public void paintComponent(Graphics g)
{
    GregorianCalendar time = new GregorianCalendar();
    int minutesAngle = 90 - time.get(Calendar.MINUTE) * 6;
    int hoursAngle = 90 - time.get(Calendar.HOUR) * 30;
    int left = 50;
    int top = 50;
    int diameter = 100;
    g.setColor(Color.black);
    g.drawOval(left, top, diameter, diameter);
    g.setColor(Color.red);
    g.fillArc(left + 5, top + 5, diameter - 10, diameter - 10,
              minutesAngle, 5);
    g.setColor(Color.blue);
    g.fillArc(left + 25, top + 25, diameter - 50, diameter - 50,
              hoursAngle, -8);
}
```

Konstruktor metodat dhe objekti

this – Vazhdim

```
/** main: për testim */  
public static void main(String[] args)  
{ new ClockWriter(); }  
}
```

Konstruktor metodat dhe objekti

this – Vazhdim

Fraza

```
new ClockWriter()
```

ka semantikën vijuese:

- ⑥ Konstruktohet një `ClockWriter` objekt në memorien qendrore;
- ⑥ Objektit të ri i dërgohet mesazh për të ekzekutuar instruksionet në konstruktorin e tij, të quajtur poashtu `ClockWriter`.

Objektet me gjendje: Variablat e fushave

Variabël e fushës (*fushë*): variabël e deklaruar në klasë pavarësisht nga metodat e klasës. Fusha mund të referencohet dhe t'i ndahet vlerë nga të gjitha metodat në atë klasë.

Variabël lokale: variabël e deklaruar përbrenda një metode. Mund të referencohet vetëm nga urdhërat përbrenda asaj metode.

Objektet me gjendje: Variablat e fushave – Vazhdim

```
import java.awt.*;
import javax.swing.*;
/** FieldExample: Afishon numrin e ngjyrosjeve të dritares */
public class FieldExample extends JPanel
{ private int count; // numri i ngjyrosjeve të dritares
  /** FieldExample: konstrukton dritaren */
  public FieldExample()
  { count = 0; // dritarja akoma nuk është ngjyrosur
    JFrame myFrame = new JFrame();
    myFrame.getContentPane().add(this);
    myFrame.setTitle("Shembull fushe");
    int height = 200;
    myFrame.setSize(3 * height / 2, height);
    myFrame.setVisible(true);
  }
}
```

Objektet me gjendje: Variablat e fushave – Vazhdim

```
/** paintComponent vizaton numrin e ngjyrosjeve
 * @param g - penda grafike */
public void paintComponent(Graphics g)
{ count = count + 1; // ose count++
  g.setColor(Color.black);
  int margin = 25;
  int lineHeight = 20;
  int firstLine = 40;
  int baseline = firstLine + lineHeight * count;
  g.drawString("E ngjyrosur " + count + " herë", margin, baseline);
}
/** main: për testim */
public static void main(String[] args)
{ new FieldExample(); }
}
```

Objektet me gjendje: Variablat e fushave – Vazhdim

FieldExample

```
main(...)  
{ ... > }
```

a1 : FieldExample

```
private int count == 0  
paintComponent( g == a2 )  
{ count = count + 1;  
  ...  
}
```

a2 : Graphics

```
setColor(...) { ... }  
drawString(...) { ... }  
...
```

a3 : JFrame

```
setSize(...) { ... }  
setVisible(...) { ... }  
...
```

Objektet me gjendje: Variablat e fushave – Vazhdim

Fushat gjithashtu mund të përmbajnë edhe objekte (adresa të tyre).

```
// ... import -- njësoj sikurse më parë
public class ClockWriter extends JPanel
{ private GregorianCalendar time = new GregorianCalendar();
  public ClockWriter()
  { // ... Njësoj sikurse më parë
  }
  public void paintComponent(Graphics g)
  { int minutesAngle = 90 - time.get(Calendar.MINUTE) * 6;
    int hoursAngle = 90 - time.get(Calendar.HOUR) * 30;
    // ... Njësoj sikurse më parë
  }
  // ... main -- njësoj sikurse më parë
}
```

Objektet me gjendje: Variablat e fushave – Vazhdim

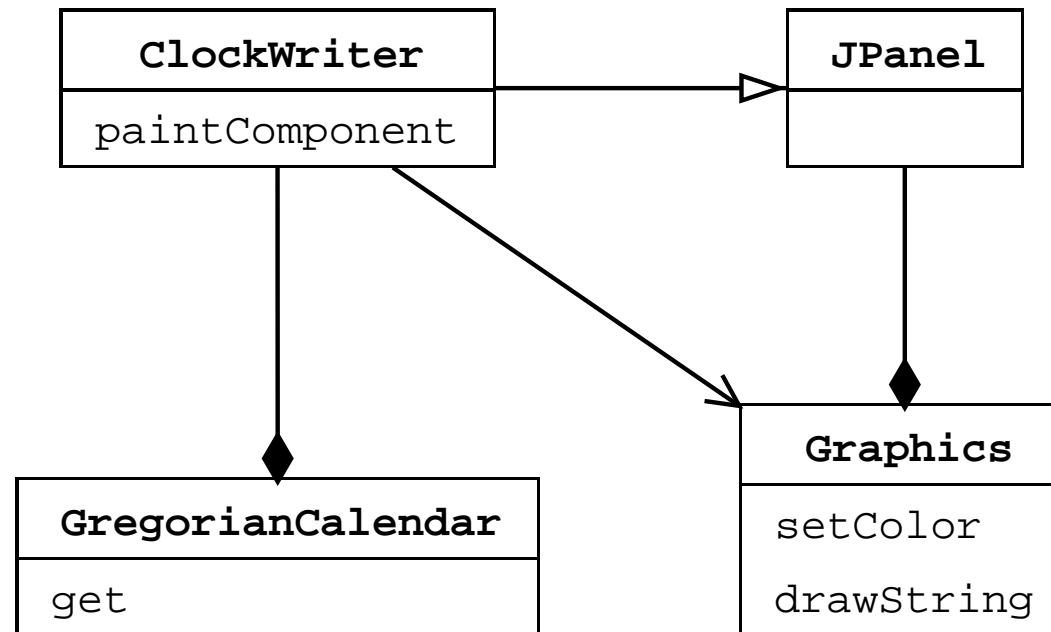


Figura 7. Arkitektura e `ClockWriter`

Objektet me gjendje: Variablat e fushave – Vazhdim

Relacioni *është*: në qoftë se klasa C2 zgjeron klasën C1, themi se C2 është C1.

Relacioni *shfrytëzon*: në qoftë se klasa C2 dërgon mesazhe objekteve të konstruktura nga klasa C1, themi se C2 shfrytëzon C1.

Relacioni *ka*: në qoftë se klasa C2 posedon fushë vlera e së cilës është objekt i klasës C1 të cilin ajo e ka konstruktuar, themi se C2 ka C1.

Objektet me gjendje: Variablat e fushave – Vazhdim

Semantika e fushave:

- ⑥ Qeliza e fushës krijohet kur edhe vetë objekti.
- ⑥ Fusha mund të inicializohet me vetë krijimin, në instruksionin e deklarimit. Në qoftë se fusha nuk inicializohet gjatë deklarimit, atëherë preferohet që të inicializohet përbrenda metodës konstruktor.
- ⑥ Vlera në qelizën e një fusheje mbahet edhe kur nuk ekzekutohet asnjë metodë e objektit.
- ⑥ Emrat e fushave nuk duhet të rideklarohen përbrenda ndonjë metode.

Zbatimi i fushave për memorimin e hyrjes dhe përgjegjjeve

```
import java.awt.*;
import javax.swing.*;
import java.text.*;
/** Konverton vlerën e temperaturës. */
public class FahrenheitToCelsiusWriter extends JPanel
{ private int fahrenheit;
  private double celsius;
  private int leftMargin = 20;
  private int lineHeight = 20;
```


Zbatimi i fushave për memorimin e hyrjes dhe përgjegjjeve – Vazhdim

```
/** Konstruktori. */  
public FahrenheitToCelsiusWriter()  
{ String input = JOptionPane.showInputDialog("Gradë Fahrenheit:");  
  fahrenheit = new Integer(input).intValue();  
  celsius = (5.0/9.0) * (fahrenheit - 32);  
  int frameHeight = 200;  
  JFrame f = new JFrame();  
  f.getContentPane().add(this);  
  f.setTitle("Celsius në Fahrenheit");  
  f.setBackground(Color.white);  
  f.setSize(3 * frameHeight / 2, frameHeight);  
  f.setVisible(true);  
}
```

Zbatimi i fushave për memorimin e hyrjes dhe përgjegjjeve – Vazhdim

```
/** Vizaton porositë.  
 * @param g penda grafike */  
public void paintComponent(Graphics g)  
{ g.setColor(Color.red);  
  int firstLine = lineHeight * 4;  
  g.drawString(fahrenheit + " Fahrenheit janë",  
               leftMargin, firstLine);  
  DecimalFormat formatter = new DecimalFormat("0.0");  
  g.drawString(formatter.format(celsius) + " Celsius",  
               leftMargin, firstLine + lineHeight);  
}  
/** main: për testim */  
public static void main(String[] args)  
{ new FahrenheitToCelsiusWriter(); }  
}
```

Testimi i programit i cili shfrytëzon hyrje

Varësisht nga lloji i test shembujve, stili i testimit mund të jetë:

1. Testim i „kutisë së bardhë“.
2. Testim i „kutisë së zezë“.