

PROGRAMARE ORIENTATĂ PE OBIECTE

GENGE BÉLA

**LABORATOR 7**  
**Moștenirea**

# Moștenirea

- Definiția din curs...

# Desenarea componentelor

- În laboratorul anterior desenarea s-a realizat prin crearea unui nou obiect Graphics și desenarea pe JFrame.
- Problema: obiectele desenate NU sunt salvate de JFrame, iar la reîmprospătarea ferestrei (e.g., resize) se vor șterge.

# Desenarea componentelor

- În AWT: desenarea se realiza în metoda `paint()`.
  - Crearea unor componente personalizate necesita suprascrierea acestei metode!
- În Swing: metoda `paint()` apelează la rândul ei o serie de alte metode:
  - `paintComponent()`.
  - `paintBorder()`.
  - `paintChildren()`.
- În Swing: suprascrierea metodei `paint()` poate să conducă la:
  - Nedesenarea conturului.
  - Nedesenarea componentelor copii.

# Desenarea componentelor

- În Swing se recomandă:
  - Suprascrierea metodei `paintComponent()`.

```
@Override  
protected void paintComponent(Graphics g) {
```

- Exemplu de creare a unui buton personalizat:

```
class MyButton extends JButton {  
    @Override  
    protected void paintComponent(Graphics g) {  
        g.setColor(Color.RED);  
        g.fillRect(0, 0, getWidth(), getHeight());  
    }  
}
```

# Desenarea componentelor

- Observație: desenarea conturului la intrarea mouse-ului în suprafața
- Acest lucru este posibil datorită menținerii metodei “paint()”.
- Metoda invalidate(): cauzează redesenarea (asincron) componenta curentă și toate componentele copii.
- Metoda repaint(): cauzează redesenarea (asincron) componentei curente. Există variante pentru declanșarea redesenării unei anumite zone.

# Task 1

- Creați un buton personalizat MyButton prin moștenirea clasei JButton și suprascrierea metodei paintComponent().
- Adăugați butonul la o fereastră JFrame. Dimensiunea butonului să fie 100x50.
- Constructorul MyButton să primească:
  - Textul ce trebuie afișat în buton.
  - Două obiecte Color: culoare1 și culoare2.
- Să se implementeze interfața MouseListener (adăugați mouse listener în constructor).
- Mouse în afara butonului: culoarea butonului să fie culoare1.
- Mouse în suprafața butonului: culoarea butonului să fie culoare2

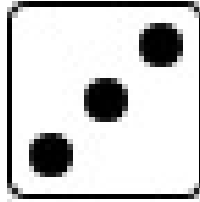
# Task 2

- Să se afișeze textul la mijlocul suprafeței butonului.
- Estimarea dimensiunii desenate a textului se poate realiza prin codul următor:
  - `int width = g.getFontMetrics().stringWidth(s);`

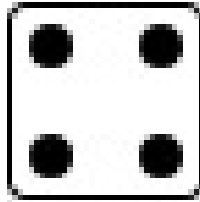


# Task 3

- Să se creeze un buton cu următorul aspect:



- La intrarea mouse-ului în suprafața butonului să afișeze următorul conținut:



- La apăsarea mouse-ului să afișeze:

