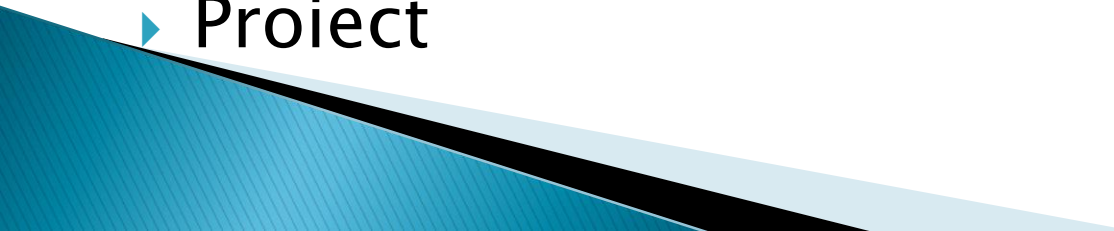


Java - S07

Alin Zamfiroiu

Java – S07

Continut:

- ▶ Equals
 - ▶ Final: attribute, metode si clase;
 - ▶ Exceptii
 - ▶ Java FX
 - ▶ Proiect
- 

07 – equals

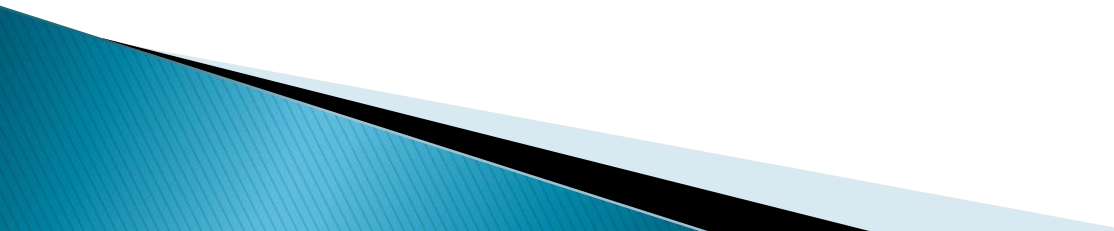
- ▶ Se definește clasa Persoana cu atributele:
 - nume;
 - sex (M sau F);
 - varsta.
- ▶ Pentru această clasă se suprascrive metoda equals.
- ▶ De unde este equals?

07– Clase, metode si attribute finale

- ▶ Atributul sex al clasei Persoana este declarat final.
- ▶ Clasa este declarata finala.

```
public final class Persoana {  
  
    private String nume;  
    private final String sex;  
    private int varsta;  
}
```

07– Clase, metode si attribute finale

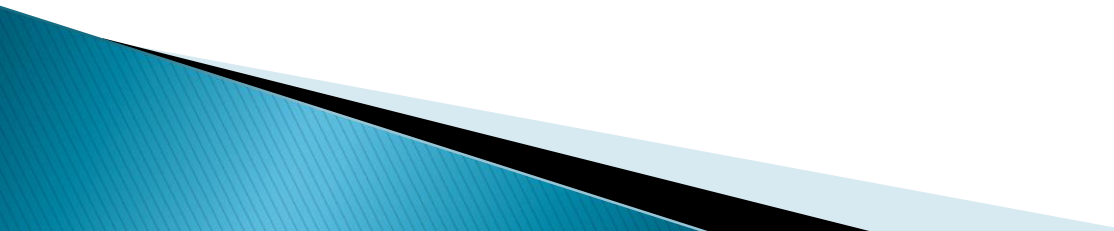
- ▶ Sa se realizeze setter si getter pentru atributul sex.
 - ▶ Sa se realizeze clasa Student care mosteneste clasa Persoana.
 - ▶ Ce este o metoda finala?
- 

07 – Exceptii

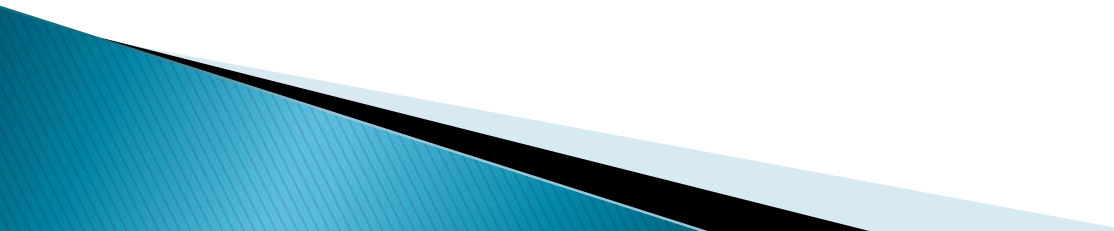
- ▶ In cadrul setterului de varsta, daca parametrul primit este mai mic decat 0, se arunca o exceptie.

```
public void setVarsta(int varsta) {  
    if(varsta>0)  
        this.varsta = varsta;  
    else  
        throw new IllegalArgumentException();  
}
```

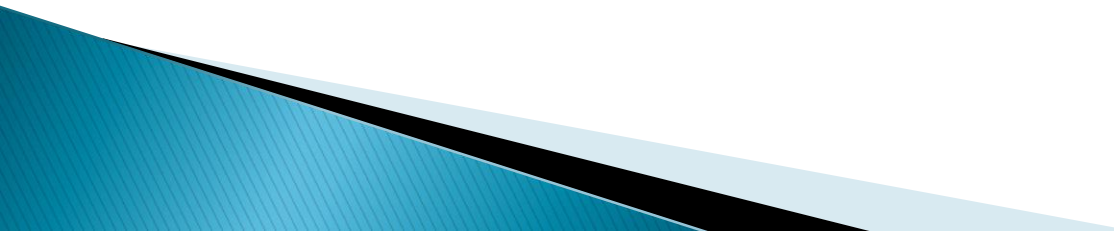
07 – Test

- ▶ Joi seara sau vineri dimineata?
 - ▶ Exemplu cu testul Persoana.
 - ▶ Indurare! Indurare! Indurare!
- 

07 – Java FX

- ▶ Scene & Stage
 - ▶ Label
 - ▶ Button
 - ▶ Text Field
 - ▶ CheckBox
 - ▶ ComboBox
- 

07 – Scene & Stage

- ▶ Se creeaza un proiect cu JRE8 si SDK8.
 - ▶ Se adauga biblioteca jfxrt.jar.
 - ▶ Se creeaza o clasa care extinde Application.
 - ▶ In main doar se apeleaza metoda launch
 - ▶ In cadrul metodei start se creeaza un VBox, ca radacina, si o scena care primeste ca parametrii root-ul si prosibilele dimensiuni.
- 

07 – Scene & Stage

```
@Override
public void start(Stage stage) throws Exception {
    VBox root=new VBox();
    Scene scena=new Scene(root,500,500);
    stage.setScene(scena);
    stage.show();
}
```

- ▶ Scena nou creată se adaugă stage-ului primit ca parametru de către metoda start().

07-Label

- ▶ In cadrul metodei start se creeaza un label si se adauga radacinii.

```
Label label=new Label("Acesta este un text static");  
root.getChildren().add(label);
```

- ▶ setText, setTooltip, setFont, setRotate, setTranslate..., setScale... .

07 – Button

- ▶ Se creeaza un buton si se adauga radacinii.
- ▶ Butonul are proprietatea de adaugare a unei actiuni: `setOnAction`, care primeste un `EventHandler`.

```
Button buton=new Button("Apasa");
buton.setOnAction(new EventHandler() {
    @Override
    public void handle(Event arg0) {
        label.setText("Ai apasat!");
    }
});
root.getChildren().add(buton);
```

07 – Button

- ▶ Daca vrem sa adaugam o imagine pe buton trebuie sa initializam un obiect de tip Image si pe buton avem metoda setGraphic cu un new ImageView.

```
Button buton=new Button("Apasa");  
Image imagine=new Image(getClass().getResourceAsStream("pic.jpg"));  
buton.setGraphic(new ImageView(imagine));  
root.getChildren().add(buton);
```

- ▶ Imaginea trebuie sa fie in directorul bin/numele_pachetului.
- ▶ setOnMouse....

07 – Text Filed

- ▶ Se creeaza un obiect de tip TextField si se adauga la root.

- ▶ setOnKeyPressed
- ▶ setOnKeyReleased
- ▶ setEditable
- ▶ setFont
- ▶ setPromptText

```
TextField tf=new TextField();
Button buton=new Button("Apasa");
root.getChildren().addAll(tf, buton);

buton.setOnAction(new EventHandler() {
    @Override
    public void handle(Event arg0) {
        label.setText(tf.getText());
    }
});
```

07-CheckBox

- ▶ Se creeaza doua obiecte de tip CheckBox. Se seteaza textul celor doua si se afiseaza in radacina.
- ▶ setAllowIndeterminate
- ▶ setIndeterminate
- ▶ isIndeterminate
- ▶ setDisable

```
CheckBox cb1=new CheckBox();
CheckBox cb2=new CheckBox();
Button btn=new Button("Selectie");
cb1.setText("Selecteaza 1");
cb2.setText("Selecteaza 2");
root.getChildren().addAll(cb1,cb2,btn);
btn.setOnAction(new EventHandler() {
    @Override
    public void handle(Event arg0) {
        if(cb1.isSelected())
            label.setText("A selectat 1");
        if(cb2.isSelected())
            label.setText("A selectat 2");
    }
});
```

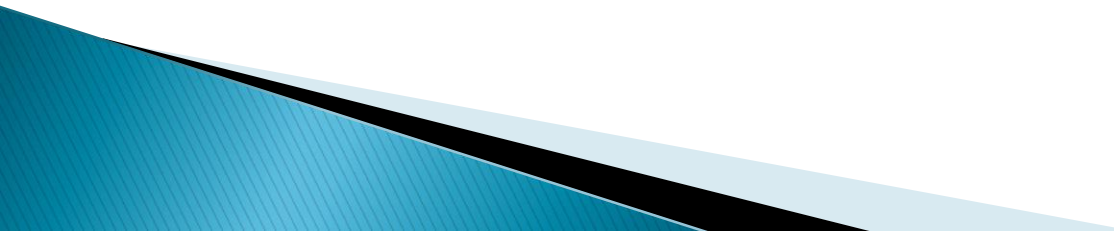
07 – ComboBox

- ▶ Se creeaza un obiect de tip ComboBox. Se adauga item-urile care pot fi selectate si se adauga la root.

```
ComboBox cbb=new ComboBox<>();  
cbb.getItems().addAll("Optiune 1", "Optiune 2", "Optiune 3");  
cbb.setEditable(true);  
  
root.getChildren().add(cbb);
```

- ▶ setEditable, getValue.

Proiect

- ▶ Fiecare isi alege un domeniu atractiv.
 - ▶ Se realizeaza o aplicatie care rezolva o problema din acel domeniu.
 - ▶ In cadrul aplicatiei este masurat comportamentul utilizatorului si salvate in fisiere, informatii despre modul de utilizare a aplicatiei de catre utilizator.
- 

Proiect

- ▶ Cerinte proiect individual JAVA:
 - ▶ – minim patru clase;
 - ▶ – minim o interfata;
 - ▶ – minim o clasa abstracta;
 - ▶ – minim o clasa derivata;
 - ▶ – minim 2 colectii generice;
 - ▶ – salvarea de informatii in fisiere text (citire/scriere);
 - ▶ – salvarea datelor in fisiere binare (citire/scriere);
 - ▶ – rularea de thread-uri;
 - ▶ – comunicarea prin Sockets;
 - ▶ – salvarea si restaurare datelor intr-o baza de date;
 - ▶ – scrierea si citirea de fisiere XML / JSON
 - ▶ – interfata grafica Java FX.