

# АПЛЕТИ

ЛЕКЦИОНЕН КУРС “ПРОГРАМИРАНЕ НА JAVA”



# ДВА ВИДА JAVA-ПРОГРАМИ

- **Приложения:**

Директно изпълнение с Java-Interpreter  
(`java Hallo`)

- **Аплети:**

Стартиране с помощта на Web-Browser  
(Firefox, Mozilla, Microsoft Internet Explorer)

# СТАРТИРАНЕ НА JAVA-АПЛЕТИ ПРЕЗ WEB-BROWSER

→ HTML разширение

- **Детайли:**

- Компилирани Java-програми се свързват във Web-страници
- Web-Browser познава Java-Interpreter
  - Може да го извика
- Java-аплетите могат да бъдат заредени през мрежата и директно извикани от Browser
  - Виртуални .class-програми
  - Не .java-първичен код
- Аплетите работят в графичен интерфейс
  - Напр., Internet Explorer-интерфейс

# HTML-ДОКУМЕНТИ: СЪСТАВНИ ЕЛЕМЕНТИ

- **Текст**

(с атрибути: цветове, подчертаване, надебеляване, големина)

- **Указатели към други документи:**

```
<a href="http:// delc.fmi.uni-pu.bg">
```

- **Картини:**

```

```

- **Програми:** изпълняват се като част на Website

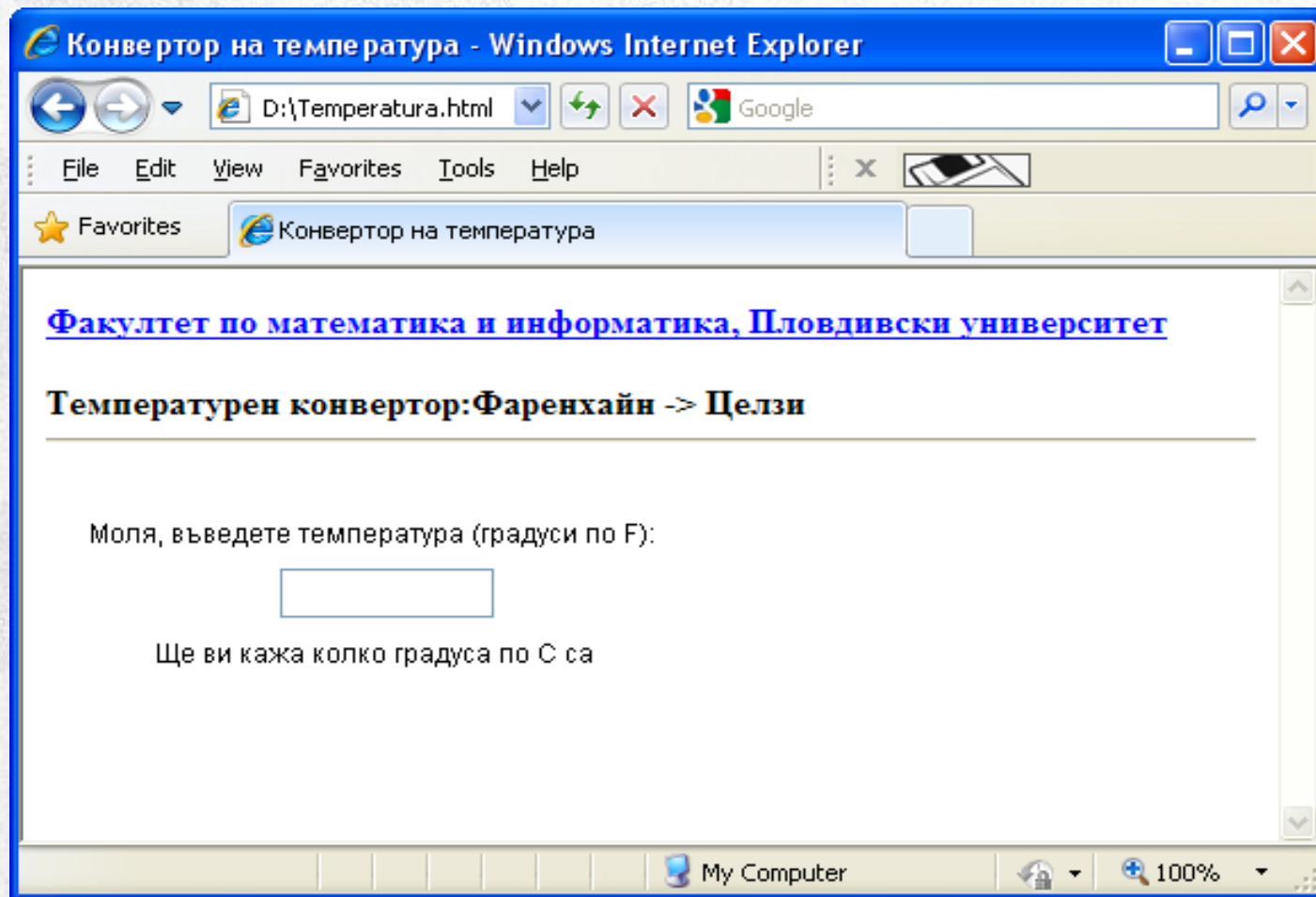
```
<applet  
    CODE="TempApplet.class" WIDTH=300 HEIGHT=100>  
</applet>
```

# ПРИМЕР: TEMPAPPLET.HTML

```
<html>
  <head>
    <title>Конвертор на температура</title>
  </head>
  <body>
    <P>
      <b>
        <a href=
          "http:// www.fmi-plovdiv.org">
            Факултет по математика и информатика, Пловдивски университет
          </a>
        </b>
      <P>
        <b>Температурен конвертор:Фаренхайн -> Целзи</b>
      <hr>
      <P>
        <applet CODE="TempApplet.class" WIDTH=300 HEIGHT=100>
      </applet>
    </body>
  </html>
```

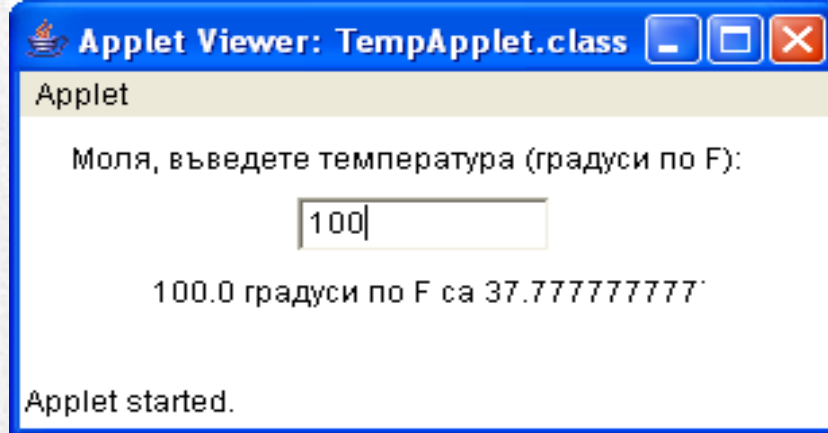
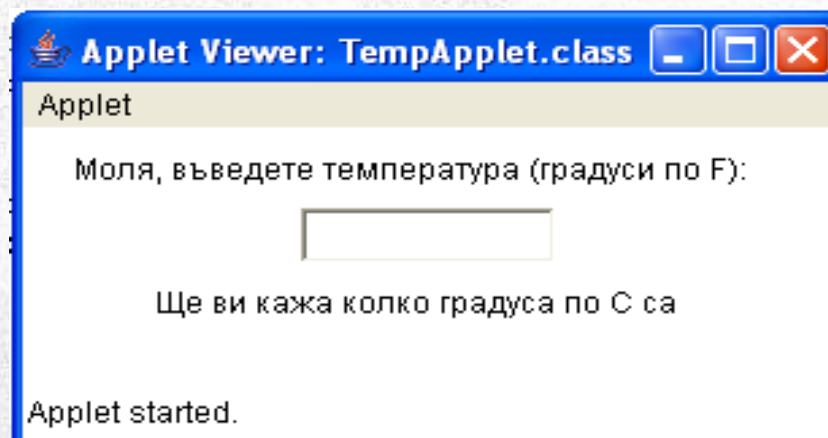


# IE: TEMPARPLET.HTML ЗАРЕДЕН



# APPLETVIEWER: ДИРЕКТНО АКТИВИРА АПЛЕТИТЕ

```
> appletviewer TempApplet.html
```



# TEMPAPPLET.JAVA

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class TempApplet extends Applet
    implements ActionListener {

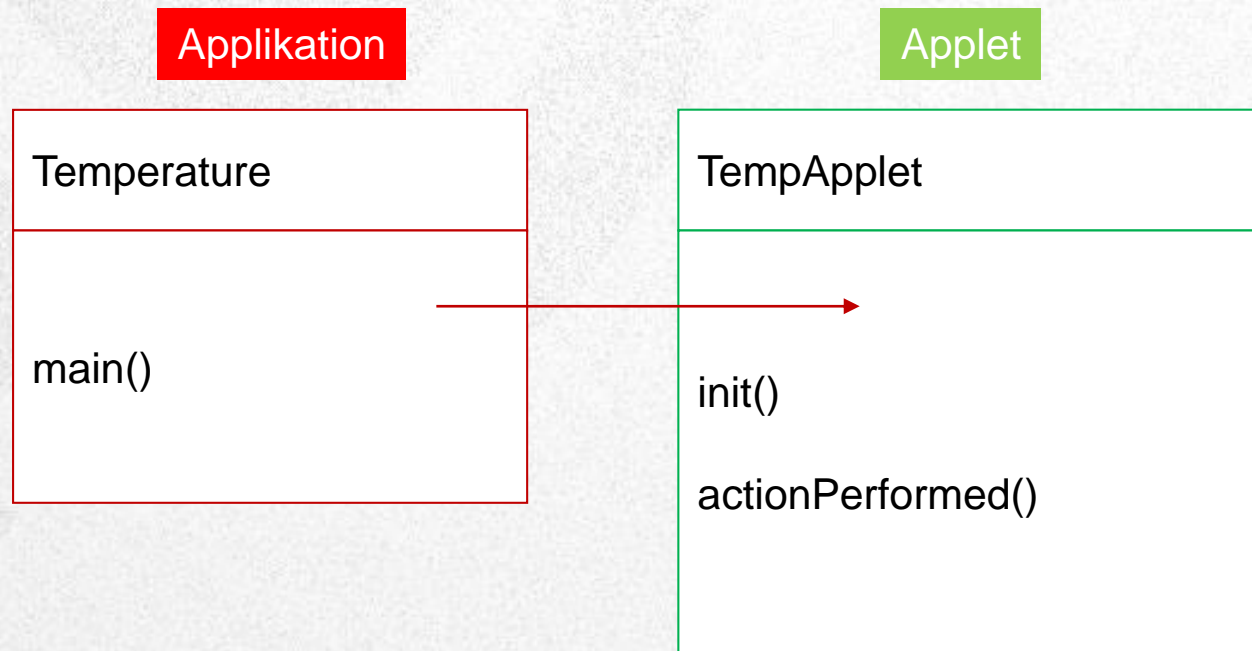
    // Convert from Fahrenheit to Centigrade
    TextField tFahr;
    Label lCent;
    public void init() {
        // Create the TextField and the Label
        tFahr = new TextField(10);
        lCent = new Label( "Ще Ви кажа  
колко градуса по C са");
        // Lay out the three Components
        add(new Label("Моля, въведете  
температура  
temperature (градуси по F):  
"));
        add(tFahr);
        add(lCent);
        // Register the Component Listener
        tFahr.addActionListener(this);
    }
```

```
// Respond to Action Event:
// typing in the tFahr TextField
public void actionPerformed (ActionEvent e) {

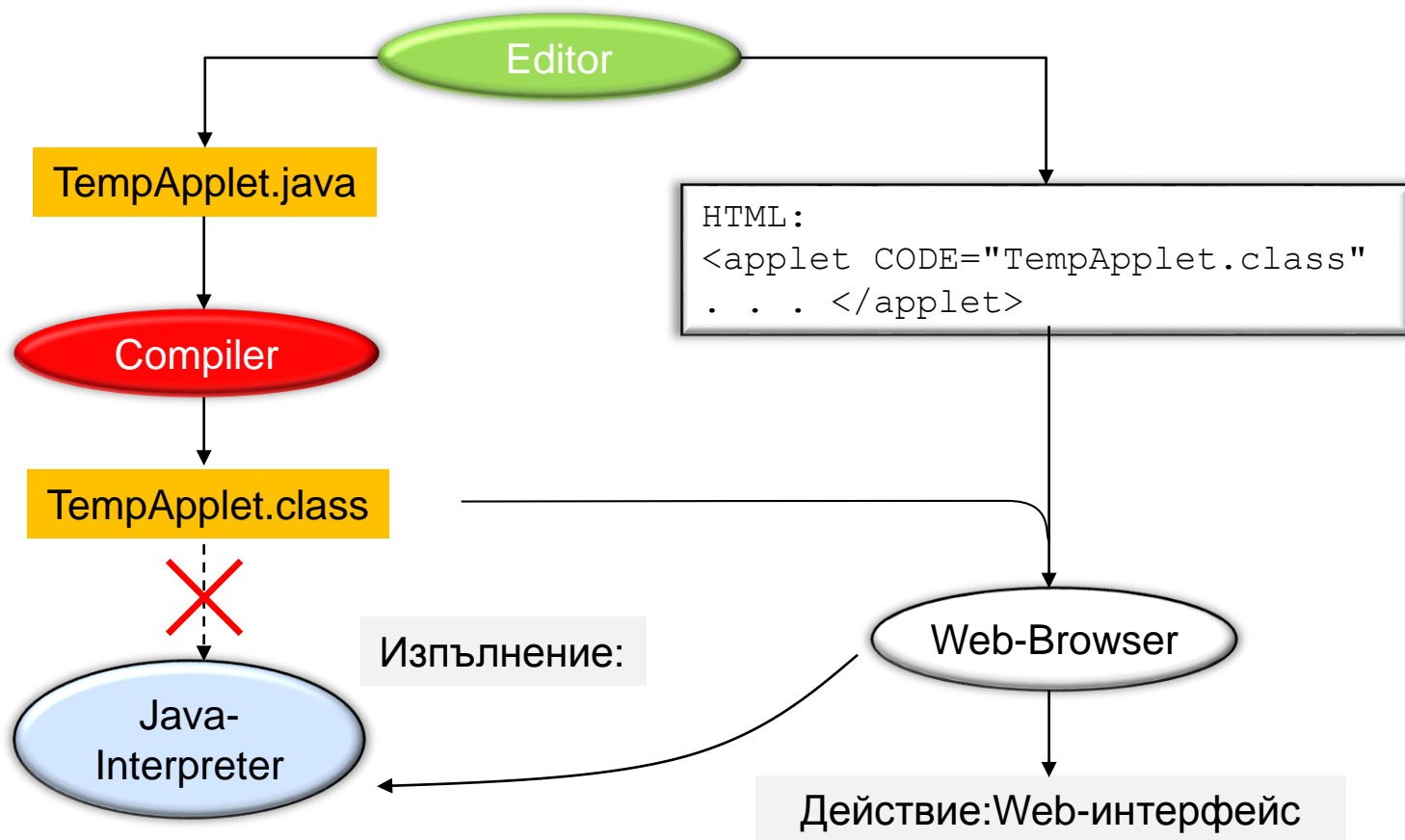
    double fahr = 0.0,
    cent = 0.0;
    fahr = Integer.parseInt(tFahr.getText());
    cent = 5.0 * (fahr - 32) / 9.0;
    lCent.setText(fahr + " градуси по F са " +
cent");
    }
}
```



# СРАВНЕНИЕ: ПРИЛОЖЕНИЕ - АПЛЕТ



# РАЗРАБОТВАНЕ НА ПРОГРАМИ: АПЛЕТИ



# ОБЯСНЕНИЕ НА TEMPERATURE-APPLET

1. Импорт
2. API-Структура: API - Пакети - Класове
3. Клас Applet
4. Инициализация на прозореца
5. Реагиране на събития
6. Локални & глобални променливи

# ИМПОРТ

```
import java.awt.*;
```

Програмиране на графични потребителски интерфейси (awt = abstract windowing toolkit)

```
import java.applet.*;
```

Applet-class

```
import java.awt.event.*;
```

Обработка на събития

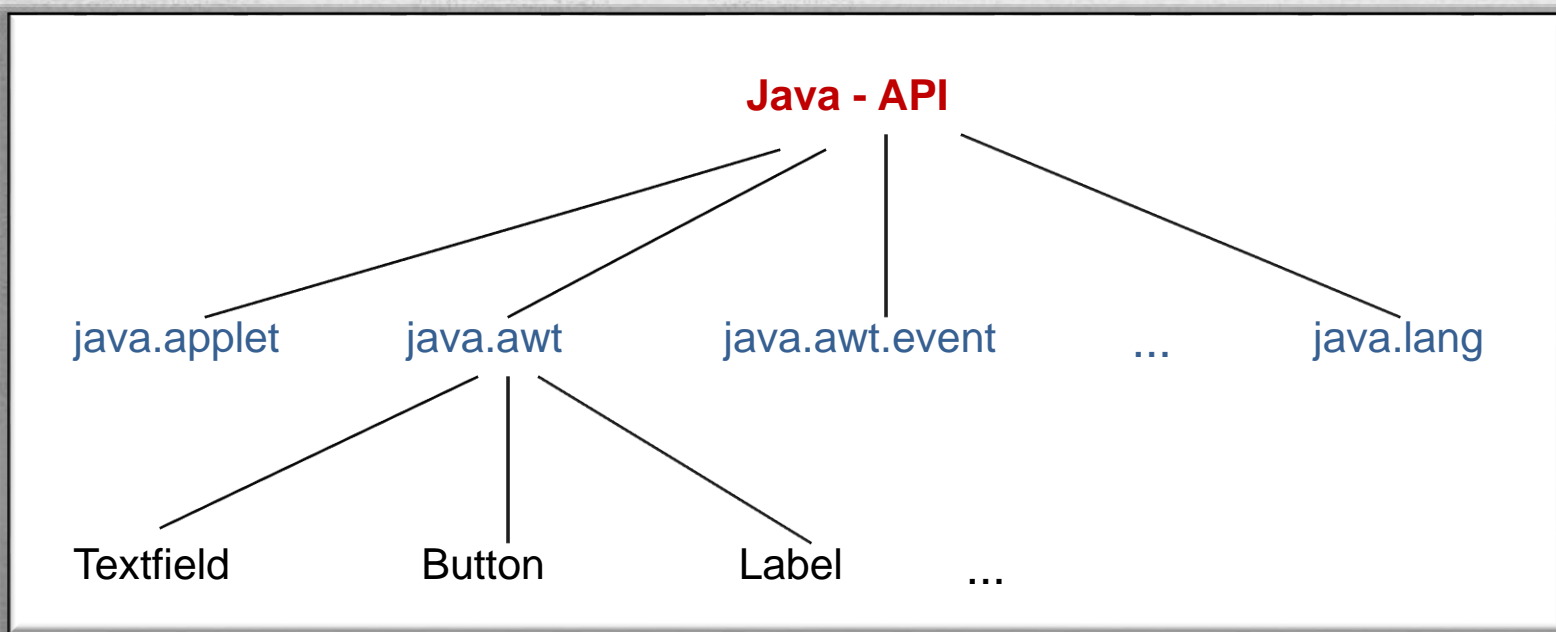
```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class TempApplet extends Applet
    implements ActionListener {

    // Convert from Fahrenheit to Centigrade
    TextField tFahr;
    Label lCent;

    public void init() { ...}
    public void actionPerformed (...){...}
}
```

# ОРГАНИЗАЦИЯ НА JAVA-API



Пакети:  
МНОЖЕСТВО ОТ КЛАСОВЕ

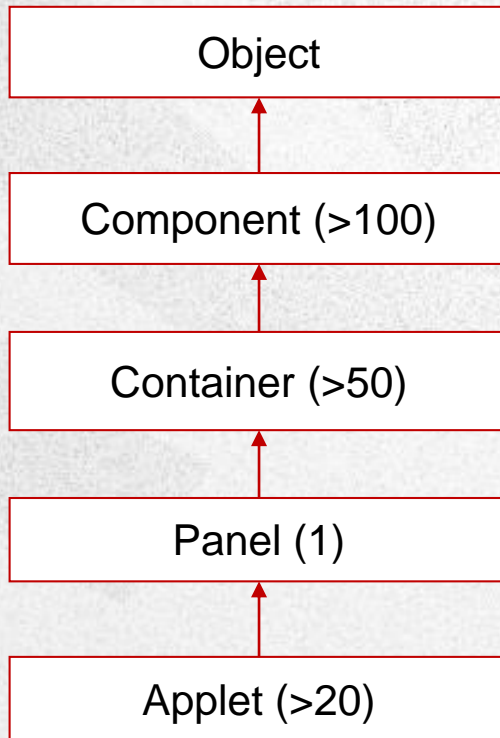
Класове:  
софтуерни компоненти

Име на пакет (напр. java.awt.event) отразява име на директория:  
→ /java/awt/event



# API: APPLET

## Брой на методите



```
import java.awt.*;  
import java.applet.*;  
import java.awt.event.*;
```

```
public class TempApplet extends Applet  
    implements ActionListener {
```

```
    // Convert from Fahrenheit to Centigrade  
    TextField tFahr;  
    Label lCent;
```

```
    public void init() {  
        ...
```

# КЛАС APPLET: ИЗВИКВАНЕ НА МЕТОДИ

TempApplet

init()

Старт на програмата: аналогично на main() при приложенията

actionPerformed()

Реакция на Enter-клавиш (събитие):  
- чете Temperature-поле  
- извежда преизчислената стойност

# ИНИЦИАЛИЗИРАНЕ НА ПРОЗОРЕЦА

Applet-прозорецът се извежда на екрана (начално състояние)

```
public void init ( ) {  
  
    tFahr = new Textfield (10);  
    lCent = new Label (" I'll tell...");  
  
    add(new Label ("Please ..."));  
    add(tFahr);  
    add(lCent);  
  
    tFahr.addActionListener(this);  
}
```

1. Създава (още невидими):  
- Активно текстово поле (вход)  
- Label-текстово поле (изходен текст)

2. Компонентите се пласират в прозореца на аплета

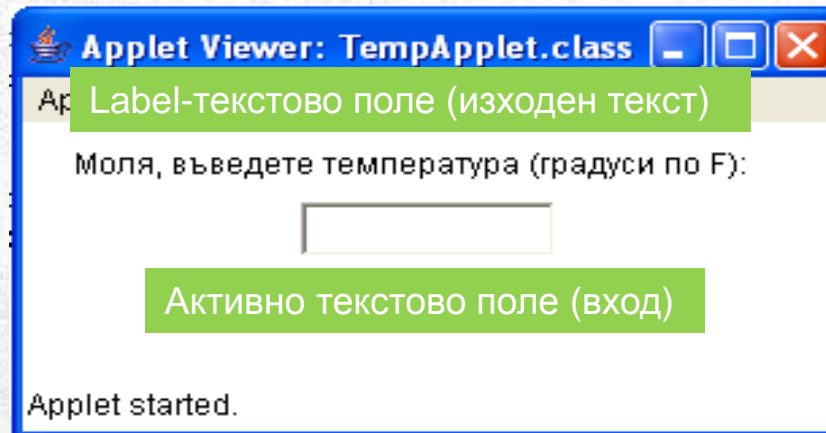
3. Текстовото поле tFahr трябва да се 'наблюдава': при enter-клавиш се извиква actionPerformed(...)

Съставни елементи на един графичен интерфейс: > 50 класа, като напр.: Textfield, Button, Label, Checkbox, Scrollbar ...

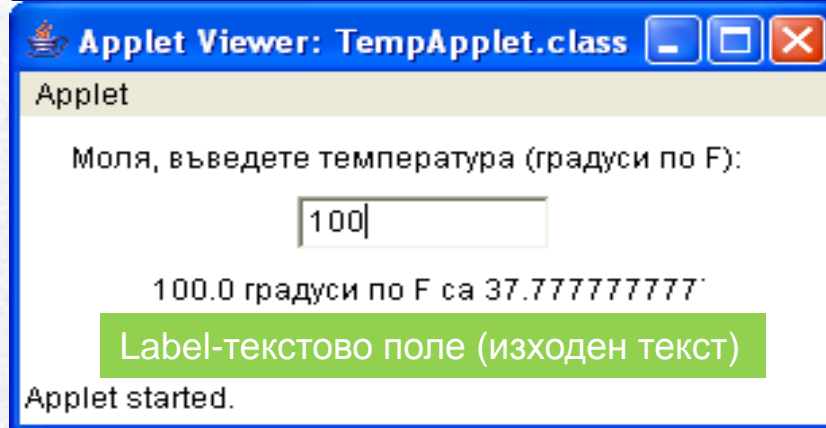
# ДЕЙСТВИЕ НА TEMPAPPLET

```
> appletviewer TempApplet.html
```

След init():



След actionPerformed():



# РЕАКЦИЯ НА СЪБИТИЯ: ENTER-КЛАВИШ

```
public void actionPerformed (...) {  
  
    double fahr = 0.0;  
    double cent = 0.0;  
  
    fahr = Integer.parseInt(tFahr.getText());  
  
    cent = 5.0 * (fahr - 32) / 9.0;  
  
    lCent.setText (  
        fahr + " deg F is " + cent + " deg C ");  
}
```

Чете от текстово поле tFahr и трансформира в Integer

Изход на резултата в Label-полето lCent

Съставни низове



# ПРОМЕНЛИВИ: ГЛОБАЛНИ В КЛАСОВЕ – ЛОКАЛНИ В МЕТОДИ

```
public class TempApplet ... {  
  
    TextField tFahr ;  
    Label lCent ;  
  
    public void init ( ) {  
        ...  
        add (tFahr);  
        add(lCent);  
    }  
  
    public void actionPerformed (...) {  
        double fahr = 0.0;  
        double cent = 0.0;  
  
        fahr = Integer.parseInt (tFahr.getText());  
    }  
}
```

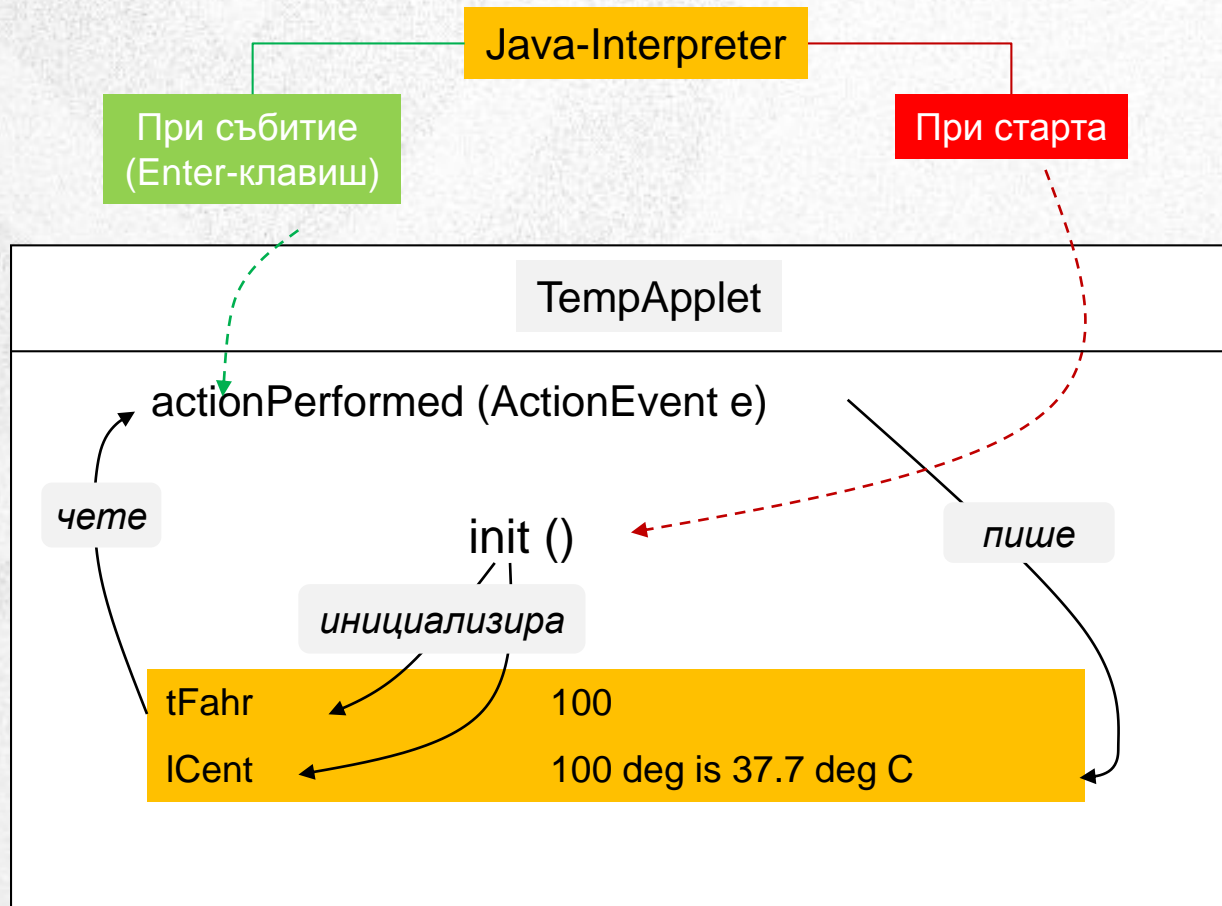
глобални:

- видими в целия клас
- продължителност на живот = "цялото" време

локални:

- видими само в метода
- продължителност на живот = времето за обработка на метода

# ОБЕКТ ОТ КЛАС TEMPAPPLET: ИЗВИКВАНЕ НА МЕТОДИ



# ОБРАБОТКА НА АПЛЕТИ: СТАРТ

```
class TempApplet extends Applet ... {  
    init() -----  
    actionPerformed(...) ←  
}
```

4. действие:  
диал.элементи  
tFahr, lCent  
на экрана

```
TempApplet.html:  
<applet ... "TempApplet.class" ...>
```

1. чете

Web-Browser:  
Netscape др.

2. извиква

Java-  
Interpreter

3. извиква

# ОБРАБОТКА НА АПЛЕТИ: СЪБИТИЕ

```
class TempApplet extends Applet ... {  
    init()  
    actionPerformed(...)  
}
```

1. Събитие: Enter-клавиш и др.

2. Hardware: регистрира събитието

3. ... Съобщава събитието на операционната система

4. Операционна система: съобщава събитието на  
Java-Interpreter

**Java-  
Interpreter**

Създава събитие *е*

Извиква `actionPerformed(e)`

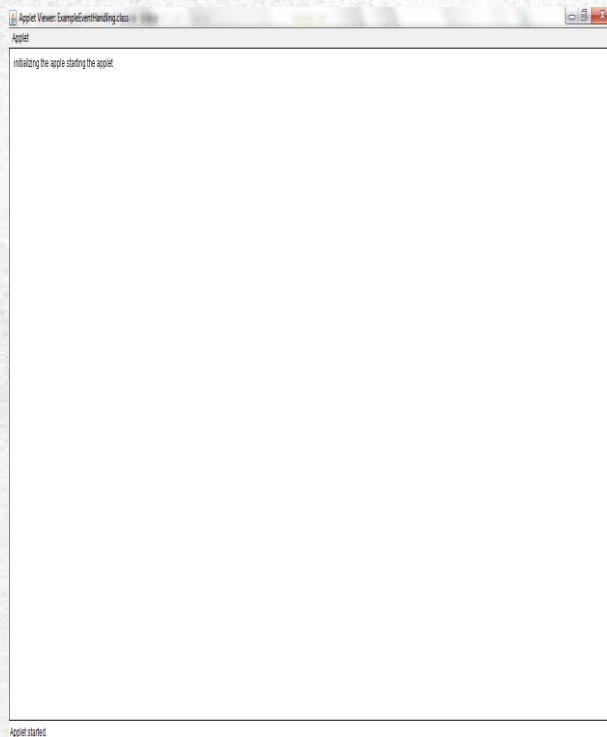
# ПРИМЕР ЗА АПЛЕТ

```
import java.awt.event.MouseListener;
import java.awt.event.MouseEvent;
import java.applet.Applet;
import java.awt.Graphics;
public class ExampleEventHandling extends Applet implements MouseListener {

    StringBuffer strBuffer;
    public void init() {
        addMouseListener(this);
        strBuffer = new StringBuffer();
        addItem("initializing the apple ");
    }
    public void start() { addItem("starting the applet "); }
    public void stop() { addItem("stopping the applet "); }
    public void destroy() { addItem("unloading the applet"); }
    void addItem(String word) { System.out.println(word); strBuffer.append(word); repaint(); }
    public void paint(Graphics g) {
        g.drawRect(0, 0, getWidth() - 1, getHeight() - 1);
        g.drawString(strBuffer.toString(), 10, 20); }
    public void mouseEntered(MouseEvent event) { }
    public void mouseExited(MouseEvent event) { }
    public void mousePressed(MouseEvent event) { }
    public void mouseReleased(MouseEvent event) { }
    public void mouseClicked(MouseEvent event) { addItem("mouse clicked! "); }
}
```

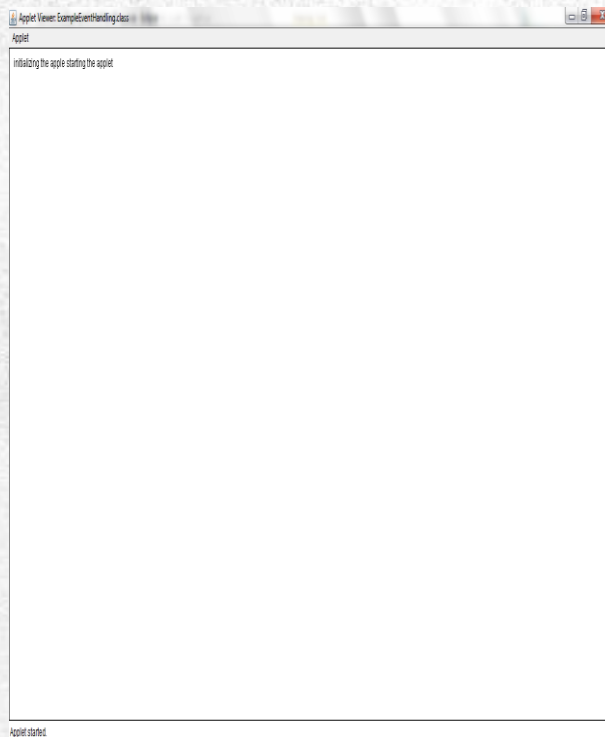


# АПЛЕТ



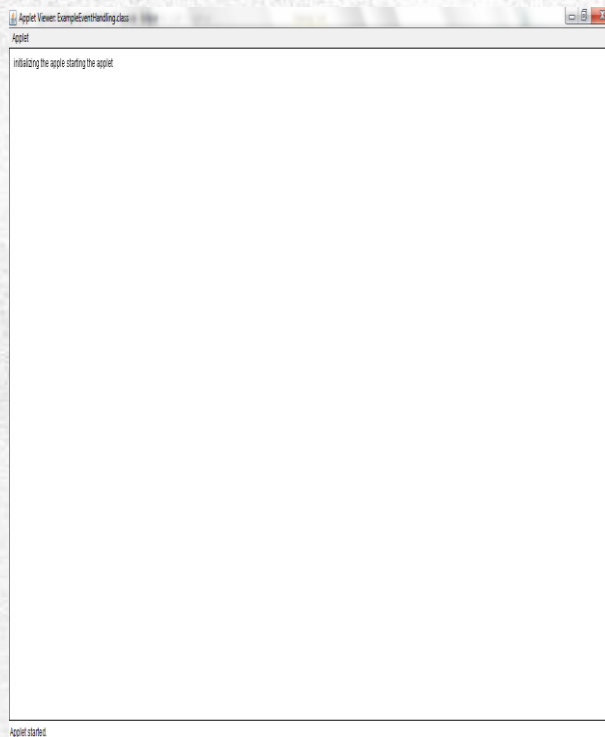
initializing the apple  
starting the applet

# АПЛЕТ



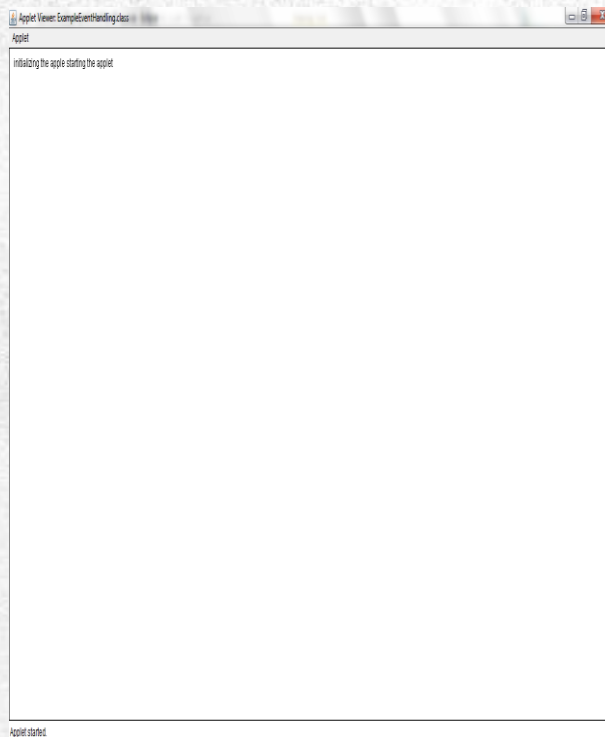
initializing the apple  
starting the applet  
mouse clicked!

# АПЛЕТ



initializing the apple  
starting the applet  
mouse clicked!  
mouse clicked!

# АПЛЕТ



initializing the apple  
starting the applet  
mouse clicked!  
mouse clicked!  
mouse clicked!



# АПЛЕТ

initializing the apple  
starting the applet  
mouse clicked!  
mouse clicked!  
mouse clicked!  
stopping the applet  
unloading the applet



# БИБЛИОТЕКИ ЗА ГРАФИЧЕН ПОТРЕБИТЕЛСКИ ИНТЕРФЕЙС (GUI)

- AWT (Abstract Window Toolkit) – Java 1.0
  - Различни недостатъци
- JFC (Java Foundation Classes) – Java 2
  - Swing – GUI частта
  - Графични компоненти
    - JavaBeans
  - Събитиен модел

# РАМКИ НА ПРИЛОЖЕНИЯ

- Елементите в библиотеките се групират съобразно функционалността си
- Различни типове библиотеки:
  - Използват се така както са създадени, без да се променят
  - Проектирани като изграждащи блокове за създаване на нови класове
  - Рамки на приложения
    - Подпомагат изграждането на приложения, осигурявайки класове, създаващи основното поведение, от което се нуждае всяко приложение
    - За да се адаптира за конкретното приложение те се наследяват и се предефинират необходимите методи
    - Аpletите са рамки на приложения

# МЕТОДИ

- `init()`
  - Извиква се автоматично за първоначална инициализация
  - Вкл. разположението на компонентите
  - Винаги се предефинира
- `start()`
  - Извиква се всеки път, когато аpletът влезе в зоната на видимост на уеб браузъра
  - Позволява на аплета да започне нормалните си операции
  - Извиква се след `init()`
- `stop()`
  - Извиква се всеки път, когато аpletът излезе от зоната на видимост на уеб браузъра
  - Извиква се точно преди `destroy()`
- `destroy()`
  - Окончателно освобождаване на ресурсите, когато аpletът не се използва вече



БЛАГОДАРЯ ЗА ВНИМАНИЕТО!

КРАЙ “АПЛЕТИ”

