

# Лекция 14.2

## Java Аплети

# Основни теми

- Различия между приложения на Java и аплети на Java.
- Синтаксис за аплети на Java.
- HTML тагове за вмъкване на аплети в уеб страници.
- Основни методи на всеки аplet на Java.

- 20.1 Въведение**
- 20.2 Примерни аплети в JDK**
- 20.3 Пример : Извеждане на текст в графичен режим**
  - 20.3.1 Изпълнение на аplet с appletviewer**
  - 20.3.2 Изпълнение на аplet с уеб търсачка**
- 20.4 Основни етапи в изпълнение на аplet**
- 20.5 Инициализиране на данните на аплета с метод `init`**

**Задачи**

**Литература:**

*Java How to Program, Sixth Edition, глава 20*

## 20.1 Въведение

- Аплети

- Java приложения, които могат да се изпълняват в контекста на уеб страници. Това са **класове**- пишат се по същите правила, както и останалите Java класове
- Използват се специални тагове от **HyperText Markup Language** (HTML) за вмъкването на аплети в уеб страници
- Когато браузера зареди уеб страница, съдържаща аplet, аплета се прочита и изпълнява от браузера
- Търсачката, която изпълнява аплета се нарича “**applet container**”
- За тестване, аpletите могат да се изпълняват извън уеб страница като се използва **appletviewer**

## 20.2 Примерни приложения в JDK

- **JDK** включва няколко демо аплети
  - Обикновено се намират *demo* директорията
    - Подразбираща се директория в Windows инсталацията на JDK :  
C:\Program Files\Java\jdk1.5.0\demo
  - Могат да се намерят на
    - [java.sun.com/j2se/5.0/](http://java.sun.com/j2se/5.0/)

Пример	Описание
<b>Animator</b>	<b>Изпълнява анимации.</b>
<b>ArcTest</b>	<b>Рисува дъги.</b>
<b>BarChart</b>	<b>Рисува диаграми.</b>
<b>Blink</b>	<b>Извежда мигащ текст.</b>
<b>CardTest</b>	<b>Демонстрира GUI компоненти и начини за разположение.</b>
<b>Clock</b>	<b>Рисува часовник.</b>
<b>DitherTest</b>	<b>Демонстрира щриховка.</b>
<b>DrawTest</b>	<b>Демонстрира работа с мишка при рисуване.</b>
<b>Fractal</b>	<b>Рисува фрактали.</b>
<b>GraphicsTest</b>	<b>Демонстрира графични способности.</b>

**Fig. 20.1 | Някои от демо версиите за аплети на JDK.**

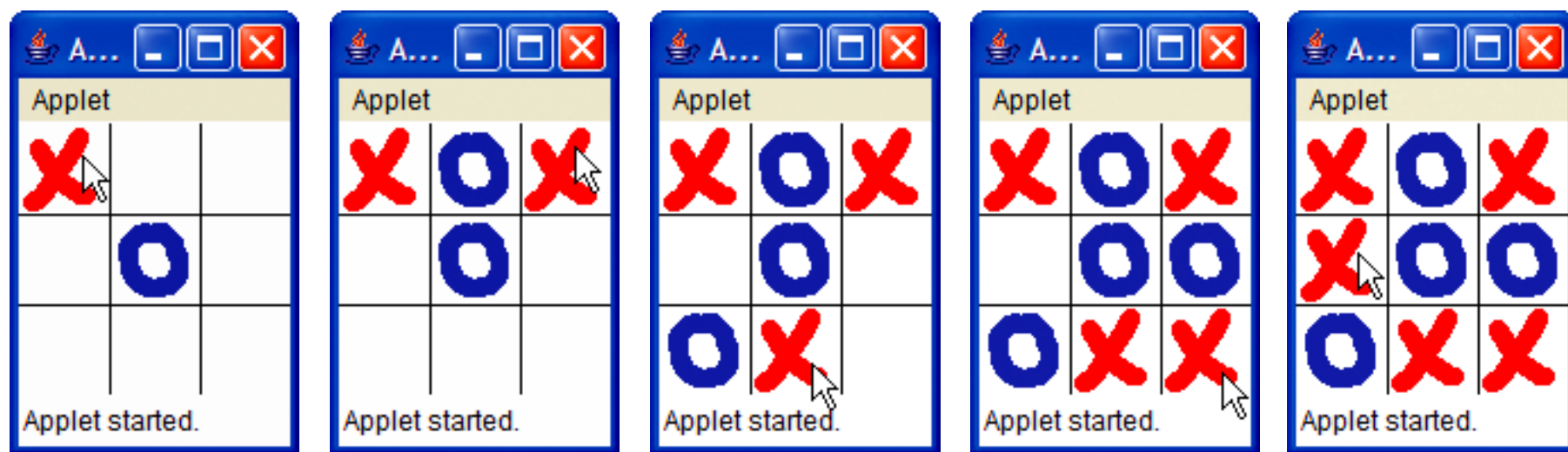
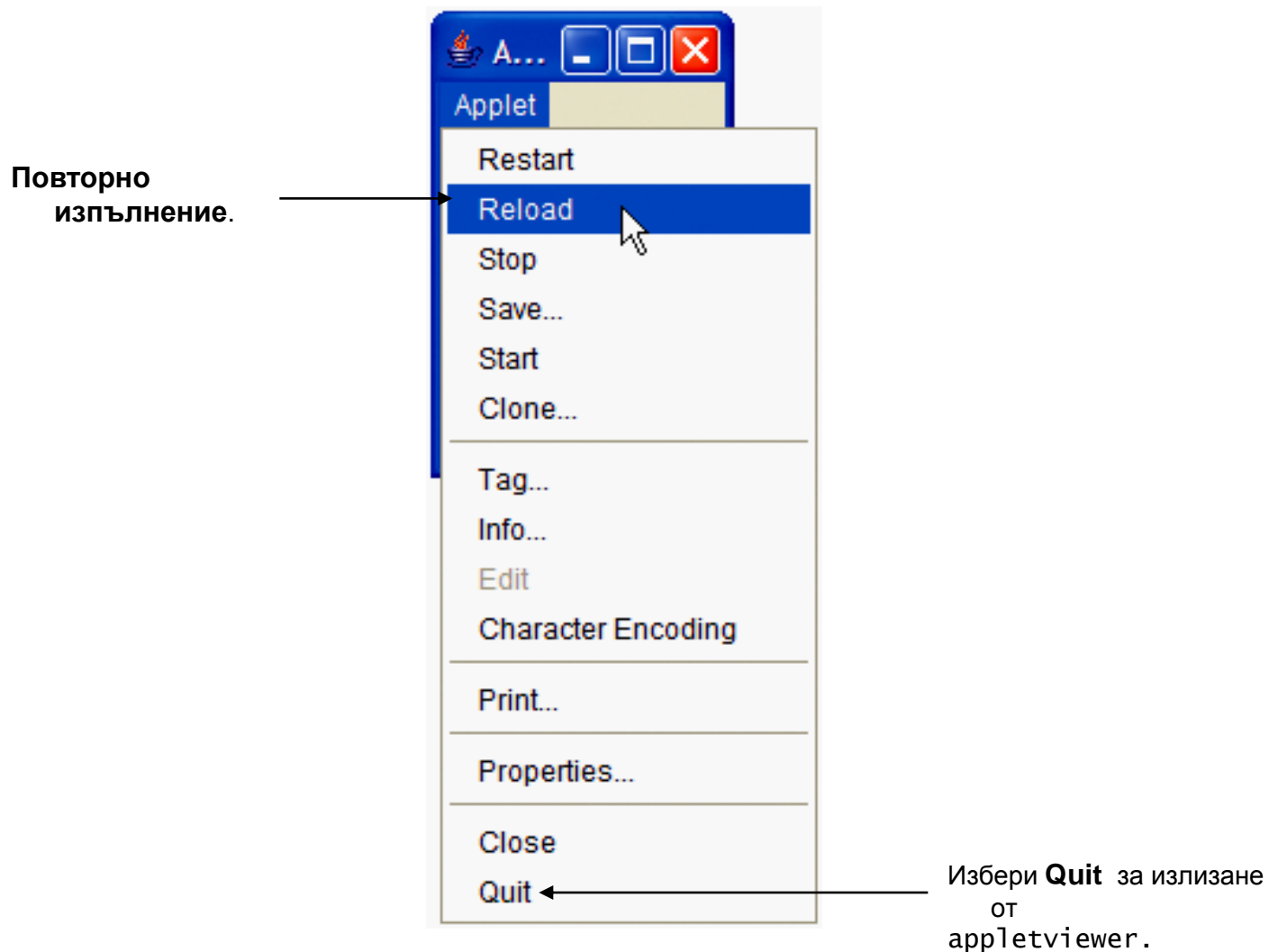


Fig. 20.2 | ТiсТасТое аплет и примерно изпълнение.



**Fig. 20.3** | Меню при изпълнение на аплет с **appletviewer**.

## 3.2 Изпълнение на аплети

- Посредством `appletviewer`
  - На командния ред, напишете командите

```
cd c:\j2sdk1.x.x\demo\applets
cd appletDirectoryName
```
  - Необходим е HTML за изпълнение на аплета
  - Например, въведете командата

```
appletviewer example1.html
```

    - `appletviewer` –а зарежда уеб страницата, където е вмъкнат аплета
    - Браузърът прочита HTML файла, интерпретира таговете на аплета, зарежда аплета и го изпълнява
  - Изпълнението на аплета се повтаря и съответно прекратява с командите **Reload** и **Quit** от менюто на `appletviewer`

```
1 <html>
2 <applet code = "AdditionApplet.class" width = "300" height = "65">
3 </applet>
4 </html>
```



Load **AdditionApplet**

Пример за HTML таговете необходими за вмъкване на аплета **AdditionApplet.class** в уеб страница



## 20.3 Пример: Рисуване на *String*

- Да се научим да пишем аплети
  - По същество това са същите Java класове, с които работихме досега
  - Различават се по това, че са производни класове на JApplet и има нужда да се предефинират специфични методи, които позволяват на браузера да изпълнява аплета
  - Браузери могат да изпълняват само такива Java класове, които са производни на JApplet - преносимост
- Примерен аплет
  - Извежда текст в графичен режим  
"Welcome to Java Programming!"
  - Демонстрира основен **синтаксис на аплет** и **аплет HTML тагове**

## 20.3 Пример: Рисуване на *String*

- Писане на *applet* клас
  - Всеки “applet container” (браузер) може да изпълнява само обекти на класове, които са `public` и са производни на клас `JApplet`
  - Това гарантира, че браузера може да изпълнява методите `init`, `start`, `paint`, `stop` и `destroy` на клас `JApplet`
    - най- често някои от тези методи са предефинирани в производния клас на `JApplet`
    - При откриване на аplet таг в уеб страницата, браузерът създава обект от този клас и извиква неговите методи `init`, `start` и `paint`

## Outline

WelcomeApplet  
.java

```
1 // Fig. 20.6: WelcomeApplet.java
2 // A first applet in Java.
3 import java.awt.Graphics; // program uses class Graphics
4 import javax.swing.JApplet; // program uses class JApplet
5
6 public class WelcomeApplet extends JApplet
7 {
8     // draw text on applet's background
9     public void paint( Graphics g )
10    {
11        // call superclass version of method paint
12        super.paint( g );
13
14        // draw a String at x-coordinate 25 and y-coordinate 25
15        g.drawString( "Welcome to Java Programming!", 25, 25 );
16    } // end method paint
17 } // end class WelcomeApplet
```

импортира **Graphics** и  
**JApplet** класовете

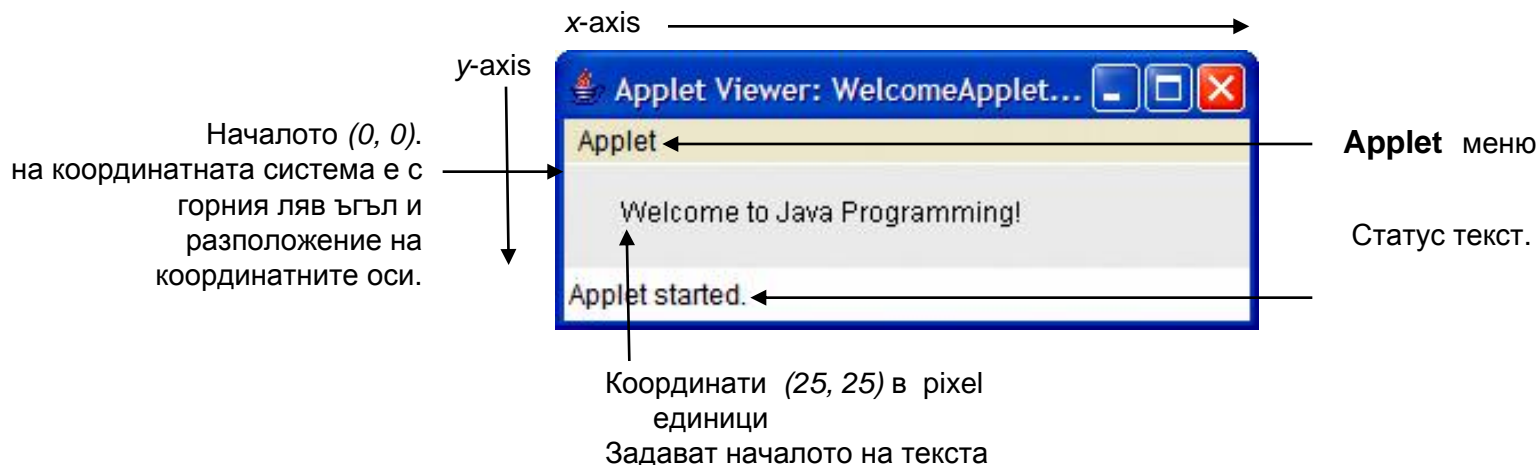
Клас **WelcomeApplet** е  
произведен на class  
**JApplet**

Извиква се **paint** на  
базовия клас

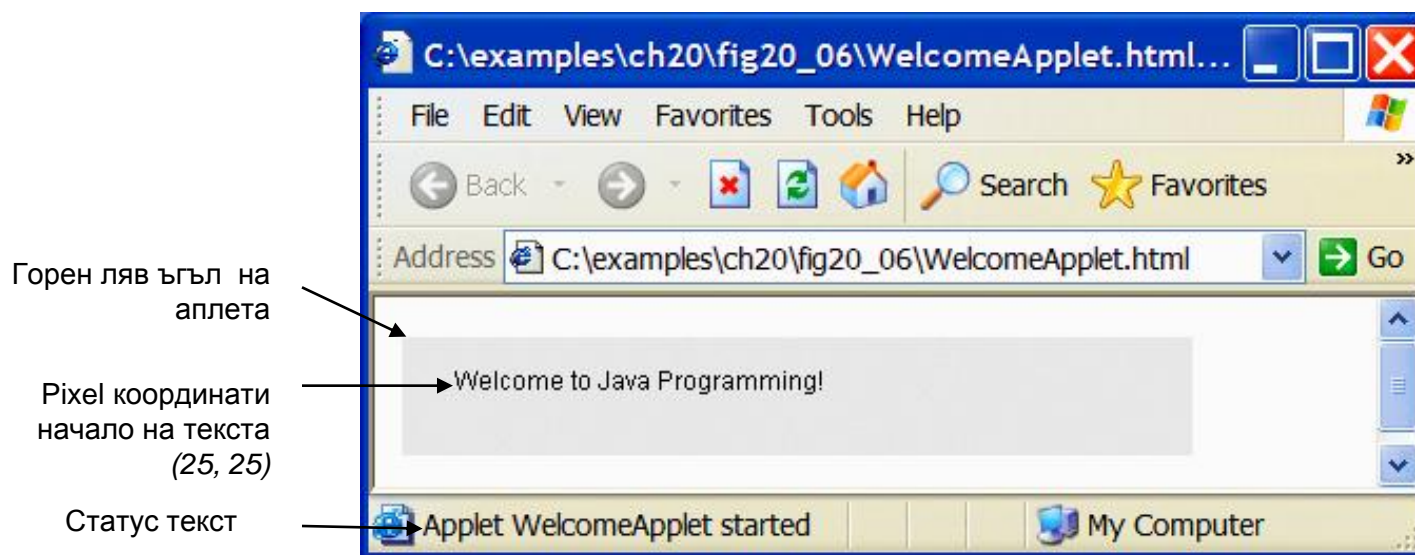
Използва се метода **drawString** на  
**Graphics** обект за извеждане на текста  
**Welcome to Java Programming!** в  
графичен формат



## welcomeApplet изпълнение в appletviewer



## welcomeApplet при изпълнение в IE



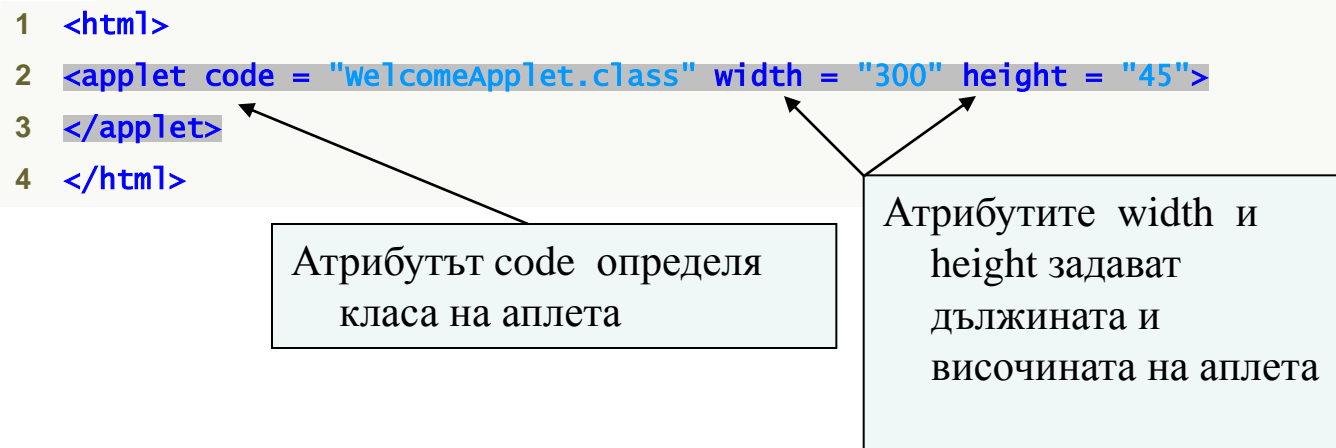
**Fig. 20.7 | Примерно изпълнение на аплета.**

## 20.3 Пример: Рисуване на *String*

- Предефиниране на метода **paint** за рисуване
  - При изпълнение на аплета, браузерът изпълнява метода **paint** като му подава аргумент **Graphics** обект който позволява да се изпълнят графични операции на рисуване
  - Методът **paint** не може да се изпълнява директно..
  - Ред 12 извиква версията на **paint** в базовия клас, както е онаследена от клас **JApplet**. Това трябва да е първата команда в **paint** метода на производния клас **Пропускането** ѝ води до **грешно изобразяване** на графичните компоненти.

## 20.3.1 Пример: Рисуване на *String* изпълнение в `appletviewer`

- Аплетите могат да се изпълняват само в *applet container- уеб браузер*
  - Преди да се изпълни аплета трябва да се създаде HTML документ, който определя аплета в този документ
    - HTML файлове имат разширение “.html” или “.htm”
    - HTML документите се форматират с тагове
      - Всеки HTML таг започва с `<` и завършва с `>`
      - Между двете скоби се изписва името на тага, евентуално атрибути и стойности на тези атрибути
  - За изпълнението на аплета `WelcomeApplet` в `appletviewer`
    - Изпълнете в директорията, съдържаща аплета и HTML файла му `appletviewer WelcomeApplet.html`
    - `appletviewer` разбира само HTML таговете `<applet>` и `</applet>` и игнорира всички други тагове



**Fig. 20.8 | welcomeApplet.html – тагове и задължителни атрибути.**



## 20.3.2 Пример: Рисуване на *String* изпълнение в Web Browser

- За изпълнение в Internet Explorer:
  - Изберете **Open...** от **File** menu
  - Изберете **Browse...**
  - Намерете HTML документа, описващ аплета за изпълнение
  - Изберете HTML документа
  - Изберете **Open**
  - Изберете **OK**

Метод	Кога се изпълнява и с какви задачи
-------	------------------------------------

**public void init()**

Изпълнява се веднъж- при зареждане на аплета в браузера. Служи за инициализация на аплета- данни, създаване на GUI компоненти, зареждане на изображения и пр. .

**public void start()**

Изпълнява се след като **init** завърши изпълнение, а също и всеки път когато потребителят повторно се върне към уеб страницата на аплета, без да е затварял браузера преди това. Този метод служи за привеждане на компонентите на аплета в началното им състояние например започване на анимация от начало, изчистване на графичната област и пр..

**Fig. 20.9 | JApplet основни фази в изпълнението на аplet. ( 1 от 3.)**

Метод	Кога се изпълнява и с какви задачи
-------	------------------------------------

**public void paint( Graphics g )**

Изпълнява се след методите **init** и **start**. Методът **paint** се изпълнява и всеки път когато има нужда от прерисуване на аплета- промяна на размера на прозореца на браузера, припокриването му с друг прозорец и пр. Типично, този метод получава като аргумент **Graphics** обект от контейнера на аплета и позволява да се изпълнят необходимите графични операции.

**public void stop()**

Този метод се изпълнява, когато потребителя премине към друга уеб страница- понеже потребителят на този етап може да се върне отново към аплета, то методът **stop** временно прекратява изпълнението на аплета, за да не се губят ресурси за цялостното му зареждане при възобновяване на изпълнението .

**Fig. 20.9 | JApplet основни фази в изпълнението на аplet. (2 от 3.)**

Метод	Кога се изпълнява и с какви задачи
-------	------------------------------------

<code>public void destroy()</code>	
------------------------------------	--

**Тозе метод се изпълнява при затваряне на браузера Служи да освобождаване на всички ресурси заемани от аплета.**

**Fig. 20.9 | JApplet основни фази в изпълнението на аplet ( 3 от 3.)**

## Обичайна грешка при програмиране 20.2

---

**Деклариране на методите `init`, `start`, `paint`, `stop` или `destroy` по различен начин от описаните води до погрешно изпълнение на аплета**

## 20.5 Инициализиране на данни с `init`

- аplet `AdditionApplet`
  - Пресмята сумата на две числа и извежда резултата като `String` текст в графичната област на аплета
    - Сумата се съхранява в данна на `class AdditionApplet`
    - Това позволява клас достъп до тази данна от методите `init` и `paint`

## Outline

AdditionApplet  
.java

(1 от 3)

```

1 // Fig. 20.10: AdditionApplet.java
2 // Adding two floating-point numbers.
3 import java.awt.Graphics;           // program uses class Graphics
4 import javax.swing.JApplet;         // program uses class JApplet
5 import javax.swing.JOptionPane;     // program uses class JOptionPane
6
7 public class AdditionApplet extends JApplet
8 {
9     private double sum; // sum of values entered by user
10
11     // initialize applet by obtaining values from user
12     public void init()
13     {
14         String firstNumber; // first string entered by user
15         String secondNumber; // second string entered by user
16
17         double number1; // first number to add
18         double number2; // second number to add
19
20         // obtain first number from user
21         firstNumber = JOptionPane.showInputDialog(
22             "Enter first floating-point value" );
23
24         // obtain second number from user
25         secondNumber = JOptionPane.showInputDialog(
26             "Enter second floating-point value" );
27

```

Декларираме данна  
**sum** от тип **double**

**init** методът се изпълнява  
първ (след конструктора на  
класа, ако има такъв)



## Outline

Сумиране и инициализиране на **sum**

AdditionApplet  
.java

(2 от 3)

```
28 // convert numbers from type String to type double
29 number1 = Double.parseDouble( firstNumber );
30 number2 = Double.parseDouble( secondNumber );
31
32 sum = number1 + number2; // add numbers
33 } // end method init
34
35 // draw results in a rectangle on applet's background
36 public void paint( Graphics g )
37 {
38     super.paint( g ); // call superclass version of method paint
39
40     // draw rectangle starting from (15, 10) that is 270
41     // pixels wide and 20 pixels tall
42     g.drawRect( 15, 10, 270, 20 );
43
44     // draw results as a String at (25, 25)
45     g.drawString( "The sum is " + sum, 25, 25 );
46 } // end method paint
47 } // end class AdditionApplet
```

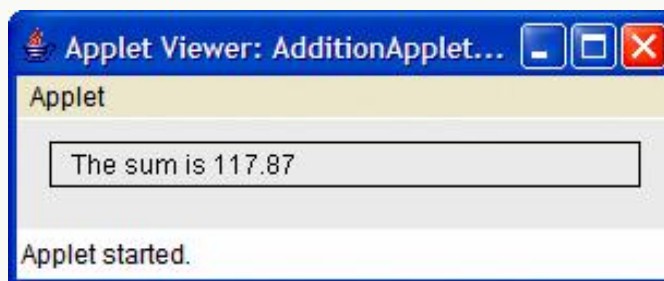
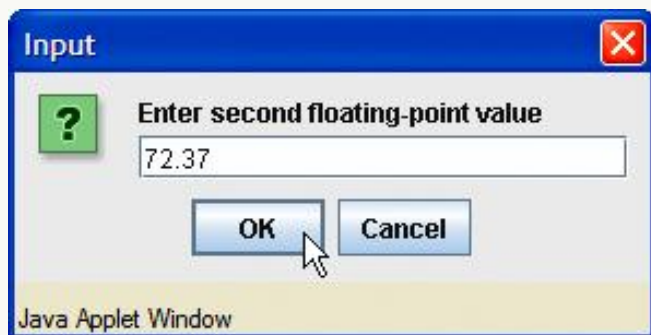
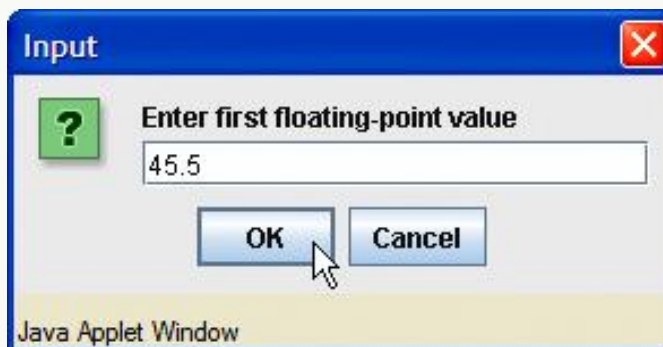
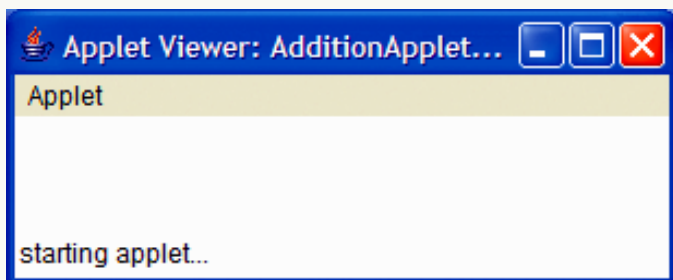
Изпълнява  
**drawString** за  
извеждане на **sum**  
в графичен  
формат



## Outline

### AdditionApplet .java

(3 от 3)



```
1 <html>
2 <applet code = "AdditionApplet.class" width = "300" height = "65">
3 </applet>
4 </html>
```

HTML тагове за аплета **AdditionApplet**

**Fig. 20.11** | `AdditionApplet.html` указва на браузера да зареди и изпълни `class AdditionApplet` като част от веб страницата.



## 20.6 Използване на параметри с аплети

```
<HTML>
<HEAD>
  <TITLE>Getting colors from user to draw a rectangle</TITLE>
</HEAD>
<BODY>
  <APPLET CODE="AppletWithInput.class" WIDTH=500 HEIGHT=500>
    <PARAM NAME="Red" VALUE="1.0">
    <PARAM NAME="Green" VALUE="0.7">
    <PARAM NAME="Blue" VALUE="0.7">
  </APPLET>
</BODY>
</HTML>
```

**Имената на параметрите за зависими от малки и големи букви!**

## 20.6 Използване на параметри с аплети

```
public void init() {  
    float MyRed = Float.parseFloat(getParameter("Red"));  
    float MyGreen = Float.parseFloat(getParameter("Green"));  
    float MyBlue = Float.parseFloat(getParameter("Blue"));  
    MyColor = new Color(MyRed, MyGreen, MyBlue);  
}
```

# Задачи

1. Напишете *JApplet* който рисува окръжност с даден радиус и нека окръжността се допира до осите  $x$  и  $y$ - на координатната система. Нарисувайте също линия  $AB$  ортогонална на оста  $x$ - и започваща от зададена  $x$  координата и с дължина диаметъра на окръжността.
  - Намерете пресечните точки на линията  $AB$  с окръжността
  - Нарисувайте малки окръжности около точките на пресичане и изведете до тях координатите на точките на пресичане  $rcle$ .
  - Нарисувайте окръжността и линията в различни цветове (червено и синьо)
  - Радиусът на окръжността и  $x$  координатата на линията  $AB$  да се въвеждат като параметри в таговета на аплета от *HTML* документа съдържащ аплета.