

Visoka tehnička škola Niš

Studijski program:

Savremene računarske tehnologije

Internet programiranje
(12)

Apleti

Prof. dr Zoran Veličković, dipl. inž. el.

Decembar, 2018.

Apleti (1)

- ❑ **APLETI** (engl. Applets) predstavljaju **SPECIFIČNU KLASU** **Java** programa sa **DRUGAČIJOM** arhitekturom od do sada proučavanih aplikacija.
- ❑ **APLETI** su **MALE APLIKACIJE** razvijene u programskom jeziku **Java** sa sledećim **specifičnostima**:
 - Smeštene su na **SERVERU**;
 - Preuzimaju se sa servera **PUTEM MREŽE** (Interneta);
 - **AUTOMATSKI** se instaliraju **NA KLIJENTU**;
 - **IZVRŠAVAJU** se kao **DEO** Web stranice.
- ❑ Još se može se reći da **APLETI** predstavljaju posebnu **KLASU** u javi realizovanu u **GRAFIČKOM OKRUŽENJU**.
- ❑ **APLETI** koriste Javinu **grafičku biblioteku** **AWT** (engl. **A**bstract **W**indow **T**oolkit) i **IZVEDENI** su iz klase **Applet**.
- ❑ Klasa **Applet** se može naći u paketu: **java.applet.Applet**.

Apleti (2)

- U klasi **Applet** definisane su sledeće **METODE** za **UPRAVLJANJE IZVRŠAVANJEM APLETA**:
 - **init()**,
 - **start()**,
 - **stop()**,
 - **destroy()**,
 - **getContext()**.
- Pored pomenutih metoda, klasa **Applet** poseduje i metode za **UČITAVANJE i PRIKAZIVANJE SLIKA** i drugih multimedijalnih sadržaja:
 - **Image** getImage(URL url),
 - **void** resize(int širina, int visina),
 - **getAudioClip**(URL url, String ime sekvence),
 - **play**(URL url) - metode **UČITAVANJE i REPRODUKCIJU AUDIO SEKVENCI**

Apleti (3)

- ❑ Takođe, u paketu java.applet su definisana i **TRI INTERFEJSA**:
 - AppletContext;
 - AudioClip;
 - AppletStub.
- ❑ Apleti imaju **OGRANIČENI PRISTUP RESURSIMA KLIJENATA**, tako da se mogu izvršavati **BEZ BOJAZNI** od virusne infekcije.
- ❑ **OKRUŽENJE** u kome se **IZVRŠAVA** aplet naziva se KONTEJNEROM APLETA i potpuno je odgovoran za njegovo:
 - **preuzimanje** i
 - **životni ciklus** apleta.
- ❑ Već je napomenuto da su apleti nešto DRUGAČIJE STRUKTUIRANI u odnosu na **standardne Java aplikacije**.
- ❑ Primer izgleda strukture Java Apleta je dat na jednostavnom primeru apleta **SimpleApplet**.

SimpleApplet

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

Uvoze se **SVE** klase iz grafičkog AWT paketa

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

Uvozi se paket **applet** koji sadrži klasu **Applet**

Dekleracija klase **SimpleApplet**

```
public class SimpleApplet extends Applet {
```

```
    public void paint(Graphics g) {
```

Nasleđivanje klase **Applet**

```
        g.drawString("A Simple Applet", 20, 20);
```

Aplet **nema** metodu **Main()** !

Dekleracija metode **paint()** iz AWT paketa, **SimpleApplet** je **REDEFINIŠE**. Metoda **paint()** se poziva **SVAKI PUT** kada aplet treba da osveži prikaz.

Objekt **g** tipa **Graphics** opisuje **GRAFIČKO OKRUŽENJE** u kome se aplet izvršava.

Metoda **drawString()** je član klase **Graphics** i iscrtava **ZNAKOVNI NIZ** na x,y lokaciji.

Apleti i HTML

```
// fajl: PokreniApp.html
```

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

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

```
/*
```

```
<applet code="SimpleApplet" width="200" height="60"> </applet>
```

```
*/
```

```
public class SimpleApplet extends Applet {
```

```
    public void paint(Graphics g) {
```

```
        g.drawString("A Simple Applet", 20, 20);
```

```
    }
```

```
}
```

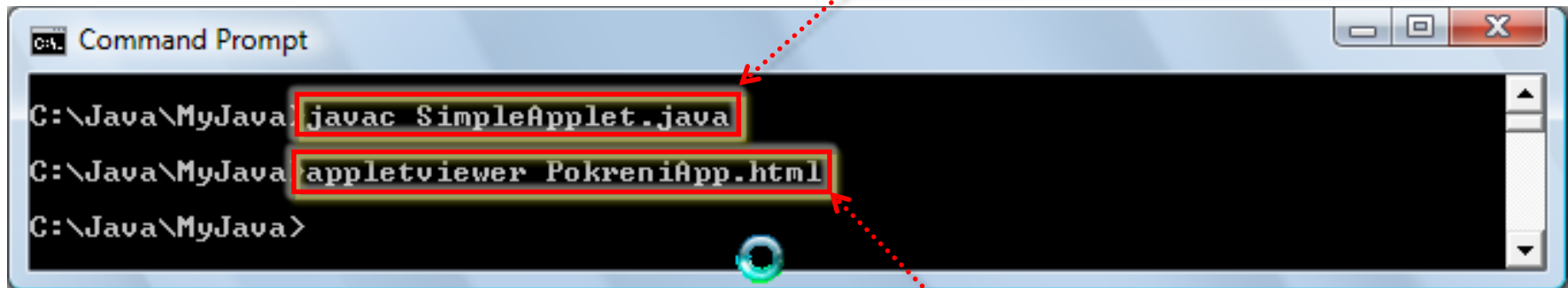
HTML kod koji aplicira
aplet **SimpleApplet**

HTML kod je
pod komentarom
i koristi ga
appletviewer

Izvršenje apleta se odvija uz pomoć čitača **Weba** ili uz pomoć programa za prikazivanje apleta: **appletviewer-a**.

čitač Apleta: Appletviewer

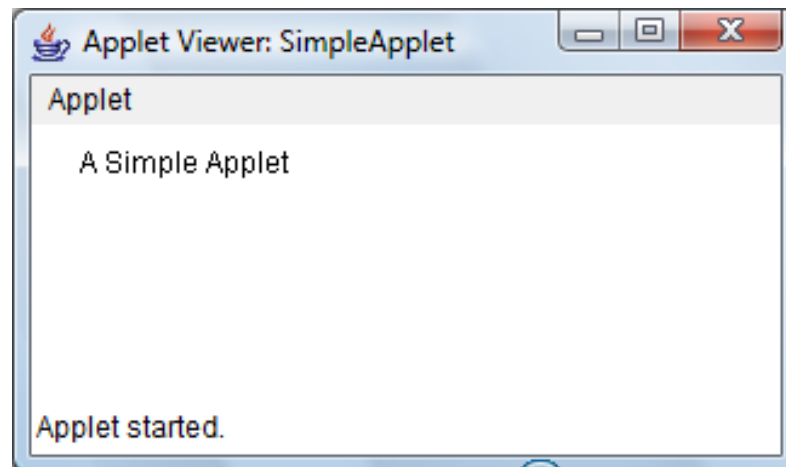
Kompajliranje apleta **SimpleApplet**



```
C:\Java\MyJava> javac SimpleApplet.java
C:\Java\MyJava> appletviewer PokreniApp.html
C:\Java\MyJava>
```

The screenshot shows a Windows Command Prompt window with a black background and white text. The title bar reads 'C:\. Command Prompt'. Three lines of text are visible: 'C:\Java\MyJava> javac SimpleApplet.java', 'C:\Java\MyJava> appletviewer PokreniApp.html', and 'C:\Java\MyJava>'. The first two lines are highlighted with red rectangular boxes. A red dotted arrow points from the text 'Kompajliranje apleta SimpleApplet' to the first box. Another red dotted arrow points from the text 'Startovanje izvršenja apleta SimpleApplet u appletvieweru koji je implementiran u html fajlu PokreniApp.html' to the second box.

Startovanje izvršenja apleta **SimpleApplet** u **appletvieweru** koji je implementiran u html fajlu **PokreniApp.html**



Rezultat izvršenja apleta: grafički prozor apleta **SimpleApplet**


Apleti i događaji

- ❑ Iz pokazanog primera se može uočiti da se **aplet** izvršava u **GRAFIČKOM PROZORU** (za razliku od kompajlera koji se izvršava sa komandne linije).
- ❑ Apleti **nasleđuju** iz klase **Applet** sve potrebne **METODE ZA RAD U GRAFIČKOM OKRUŽENJU** i radu sa **PROZOROM APLETA**.
- ❑ Apletima uglavnom **UPRAVLJAJU DOGAĐAJI** koji se mogu uporediti sa **prekidnim rutinama** kod mikroprocesora.
- ❑ **DOGAĐAJI APLETA** se mogu posmatrati na sličan način kao i **DOGAĐAJI** koji potiču od **tastature** ili **miša**.
- ❑ **APLET ČEKA DOGAĐAJ**, a o njegovom nastanku ga obaveštava **AWT** (grafičko okruženje) pozivom procedure za **OBRADU DOGAĐAJA**.
- ❑ Po **obavljenom poslu**, aplet **vraća** kontrola **AWT**-u.
- ❑ Sam **Aplet** samo **KRATKO VREME ZDRŽAVA KONTROLU** nad izvršavanjem.
- ❑ Konkretni aplet treba da **REDEFINIŠE OSNOVNI SKUP METODA** za rad sa **apletom**.

Apleti i osnovne metode

- U klasi Applet definisane su sledeće **četiri metode**:
 - init(),
 - start(),
 - stop(),
 - destroy().
- Takođe, u apletima se koristi metoda paint() koja je definisana u klasi Component iz **AWT-a**.
- **PRIMER** koji je dat u nastavku samo prikazuje **REDOSLED** kojim AWT poziva metode predefinisanom **STRUKTUROM APLETA**.
- Kada aplet ZAPOČNE rad, AWT poziva metode sledećim - **UNAPRED** utvrđenim **REDOM**:
 1. init(),
 2. start(),
 3. paint().

Životni ciklus Apleta

- Kada se aplet **ZAVRŠI**, pozivaju se metode sledećim **UTRĐENIM REDOM**:
 1. **stop()**,
 2. **destroy()**.
- Za **AŽURIRANJE GRAFIČKIH PODATAKA** u prozoru apleta se koristi metoda **repaint()**, koja poseduje **ČETIRI KONSTRUKTORA**:
 1. void **repaint()**,
 2. void **repaint**(int levo, int gore, int širina, int visina),
 3. void **repaint**(max_Odlaganje),
 4. void **repaint**(max_Odlaganje, int levo, int gore, int širina, int visina).
- Metoda **Showstatus()**, prikazuje **PORUKU** u **STATUSNOM PROZORU ČITAČA** Weba ili programa za prikazivanje apleta.
- Struktura apleta je prikazana na primeru apleta **AppletSkel**.

Struktura Apleta – (1)

```
import java.awt.*;
import java.applet.*;
/*
<applet code="AppletSkelet" width="300" height="100"></applet>
*/

public class AppletSkelet extends Applet {
    // Prvi poziv
    public void init() {
        // Ostatak koda za inicijalizaciju
    }

    /* Metoda start() se poziva posle init() metode.
       Takođe poziva se svaki put kada se applet aktivira.*/
    public void start() {
        // Kod za start ili nastavak izvršavanja
    }
}
```

Uvoze se klase za podršku apleta

Nasleđivanje klase Applet

1. poziv init() metode

2. poziv start() metode

Struktura Apleta – (2)

// Poziva se kada je aplet stopiran

```
public void stop() {
```

3. poziv `stop()` metoda

```
// suspenzija izvršavanja
```

```
}
```

/* Poziva se kada je aplet završen. Ova metoda se poslednja izvršava. */

```
public void destroy() {
```

4. poziv `destroy()` metoda

```
// obavljanje završetka aktivnosti
```

```
}
```

// Poziva se kada se prozor apleta mora restorirati

```
public void paint(Graphics g) {
```

5. poziv `paint()` metoda

```
// ponovno prikazivanje – osvežavanje sadržaja prozora
```

```
// Java kod zadužen za ponovno iscrtavanje prozora
```

```
}
```

```
}
```

Parametarizovani apleti

- ❑ Setite se, oznaka **Appleta** u HTML-u omogućava **PROSLEĐIVANJE PARAMETARA** apletu, oznaka **param**.
- ❑ Evo kako smo **PROSLEĐIVALI PARAMETRE** apletu iz **HTML** koda:

`<param name = "fontName" value = "Courier">`

- ❑ Za **OČITAVANJE PARAMETARA** upućenih Java apletu **iz** HTML-a koristi se metoda **getParametar()**.
- ❑ Zapamtite, ova metoda **vraća vrednost** u obliku **ZNAKOVNOG NIZA!**
- ❑ Dakle, **logičke** i **numeričke** vrednosti se moraju **PRVO PRETVORITI** iz znakovnog niza u njihov **interni format**.
- ❑ Pogledajmo sledeći slajd sa primerom **PROSLEĐIVANJA PARAMETARA iz HTML koda**.

Apleti i param oznaka (1)

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

HTML kôd - dopisano samo radi izvršenja u vreme razvoja apleta

```
<applet code="ParamDemo" width="300" height="80">
```

```
<param name = fontName value = "Courier" >
```

```
<param name = fontSize value = "14" >
```

```
<param name = leading value = "2">
```

```
<param name = accountEnabled value = "true">
```

Parametri koji se prosleđuju apletu ParamDemo

```
</applet>
```

```
*/
```

IMENA i VREDNOSTI parametara koja se prenose u aplet

```
public class ParamDemo extends Applet {
```

Aplet ParamDemo

```
String fontName;
```

```
int fontSize;
```

```
float leading;
```

```
boolean active;
```

Dekleracija parametara apleta koji će biti preuzeti iz HTML-a

Apleti i param oznaka (2)

// Inicijalizacija stringa koji treba da se prikaže

```
public void start() {
```

```
    String param;
```

```
    fontName = getParameter("fontName");
```

```
    if (fontName == null)
```

```
        fontName = "Not Found";
```

```
    param = getParameter("fontSize");
```

```
    try {
```

```
        if (param != null)
```

// nije pronađen

```
            fontSize = Integer.parseInt(param);
```

```
        else
```

```
            fontSize = 0;
```

```
    } catch (NumberFormatException e) {
```

```
        fontSize = -1;
```

```
    }
```

Preuzimanje parametra **fontName**

Provera vrednosti parametra

Preuzimanje parametra **fontSize**

Konverzija tipa string u integer

Provera vraćene vrednosti

Apleti i param oznaka (3)

```
param = getParameter("leading");  
try {  
    if (param != null) // parametar nađen ?  
        leading = Float.valueOf(param).floatValue();  
    else  
        leading = 0;  
} catch (NumberFormatException e) {  
    leading = -1;  
}
```

```
param = getParameter("accountEnabled");  
if(param != null)  
    active = Boolean.valueOf(param).booleanValue();  
}
```

Apleti i param oznaka (4)

// Prikaz parametara, redeinisanje metode paint()

```
public void paint(Graphics g) {  
    g.drawString("Font name: " + fontName, 0, 10);  
    g.drawString("Font size: " + fontSize, 0, 26);  
    g.drawString("Leading: " + leading, 0, 42);  
    g.drawString("Account Active: " + active, 0, 58);  
}  
}
```

Interfejs AppletContext (1)

- ❑ Već je pomenuto, u paketu **Java.applet** su definisana **TRI INTERFEJSA**:
 1. **AppletContext** (dobavlja podatke o okruženju),
 2. **AppletStub** (daje podršku Web čitačima) i
 3. **AudioClip** (sa metodama za manipulaciju audio sadržajem: play(), loop() i stop()).
- ❑ **AppletContext** je **INTERFEJS** koji omogućava da se **preuzmu podaci iz/o okruženja** u kome se aplet izvršava.
- ❑ **Metode** koje ovaj interfejs definiše su:
 - Za applet: **getApplet**(String imeApleta),
 - Za AudioClip: **getAudioClip**(URL url),
 - Za sliku: **getImage**(URL url),
 - Za ulazni strim: **getStream**(String ključ),
 - Za Učitavanje datoteke: void **showDocument**(URL url),
 - Za Učitavanje datoteke void **showStatus**(String str), ...

Interfejs AppletContext (2)

- ❑ Zbog bezbednosnih razloga, Java dozvoljava da aplet uči podatke **SAMO IZ FOLDERA** u kome se nalazi **HTML datoteka** ili **klasa apleta**.
- ❑ Imena ovih foldera mogu se dobiti sledećim metodama:
 - getDocumentBase() i
 - getCodeBase().
- ❑ Ove metode pripadaju velikoj klasi URL.
- ❑ Za **UČITAVANJE DATOTEKE** koristi se metoda showDocument(URL url) iz interfejsa AppletContext.
- ❑ Kada se dobije **KONTEKST** u kome se aplet izvršava može se prikazati **BILO KOJI DOKUMENT** metodom showDocument(URL url).

Interfejs AppletContext (3)

/* Korišćenje konteksta apleta: getCodeBase(), showDocument() za prikaz HTML fajla.

*/

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

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

```
import java.net.*;
```

```
/* <applet code="ACDemo" width="300" height="50"> </applet> */
```

```
public class ACDemo extends Applet{
```

```
    public void start() {
```

```
        AppletContext ac = getAppletContext();
```

```
        URL url = getCodeBase();        // čitanje url-a ovog apleta
```

Interfejs AppletContext (4)

```
try {  
    ac.showDocument (new URL(url+"Test.html"));  
}  
catch (MalformedURLException e) {  
    showStatus("URL not found");  
}  
}  
}
```

PRIKAZIVANJE dokumenta Test.html iz prethodno dobavljenog foldera

- Datoteka **Test.html** koja se prikazuje **MORA BITI** u **ISTOM FOLDERU** gde se nalazi i sam aplet!

Interfejs AudioClip

- ❑ **AudioClip** je **INTERFEJS** koji definiše sledeće metode:
 - **play()** - reprodukuje sekvencu od početka
 - **stop()** - prekida reprodukciju
 - **loop()** - reprodukcija u kontinualnoj formi
- ❑ **UČITAVANJE AUDIO SEKVENCE** se obavlja metodom **getAudioClip()**, dok se reprodukcija dobavljene audio sekvence obavlja metodom **play()**.
- ❑ **INTERFEJS AppletStub** obezbeđuje saradnju apleta i čitača.
- ❑ Prikazivanje na konzoli se često koristi u procesu debugovanja.
- ❑ Ako se u apletu pozove metoda **System.out.println()**, podaci se **NE ŠALJU** u prozor apleta.
- ❑ Za potrebe debugovanja treba koristiti **drow-String()**.