

# Теория и технология программирования

## Программирование на языке Java

---

### Раздел 6. Апплеты

Глухих Михаил Игоревич, к.т.н., доц.

[mailto: glukhikh@mail.ru](mailto:glukhikh@mail.ru)

# Определение

---

- Апплеты - программы, работающие в среде другой программы (интернет-браузера, browser)
  - Обычные GUI-приложения запускаются операционной системой
  - Апплеты запускаются браузером

# Принцип работы браузера

---

- ❑ Браузер соединяется с сервером по указанному адресу
- ❑ Загружает оттуда HTML-страницу
- ❑ Отображает ее в своем окне
- ❑ В процессе отображения с того же сервера могут загружаться элементы страницы:
  - изображения
  - звуки
  - видео
  - апплеты
  - ...

# Что такое HTML-страница?

---

- ❑ Файл, написанный на языке HTML
- ❑ HTML - HyperText Markup Language - язык разметки гипертекста - показывает, где, что и в каком формате следует отобразить
- ❑ HTML **НЕ ЯВЛЯЕТСЯ** языком программирования
- ❑ Один из многочисленных учебников - <http://html.manual.ru>.

# Пример HTML-файла

---

- **<html>**
- **<title>**Здравствуй, мир!**</title>**
- **<body>**
- **<p>**Это пример HTML-страницы**</p>**
- **<p>**А здесь у нас  
  **<a href="applet.html">**  
  страница с апплетом  
  **</a>**
- **</p>**
- **</body>**
- **</html>**

# Пример страницы с апплетом

---

- **<html>**
- **<title>**Страница с апплетом**</title>**
- **<body>**
- **<p>**Здесь находится простой апплет**</p>**
- **<applet code="simpleapplet.SimpleApplet.class"**  
**width="300" height="100">**
- **</applet>**
- **<p><a href="board.html">**Здесь**</a>**
- будет апплет посложнее
- **</p>**
- **</body>**
- **</html>**

# Что в результате

---

- ❑ Браузер читает  
... `applet code="simpleapplet.SimpleApplet.class"`
- ❑ И ищет в директории `simpleapplet`
- ❑ файл с Java-классом `SimpleApplet.class`
- ❑ относящийся к Java-пакету `simpleapplet`
  
- ❑ Панель с размерами `width="300" height="100"`
- ❑ внутри окна браузера
- ❑ становится корневым контейнером апплета

# Пример апплета

---

```
□ public class SimpleApplet extends JApplet {  
□     @Override  
□     public void paint(Graphics g) {  
□         super.paint(g);  
□         g.drawString(  
           "Это - самый простой апплет",  
           20, 20);  
□     }  
□ }
```



# Устройство JApplet

---

- ❑ Похоже на устройство JFrame - также имеется контейнер `ContentPane`, все компоненты добавляются в него
- ❑ Могут возникать проблемы с перерисовкой
- ❑ Решить их можно как и в JFrame - путем добавления панели

# Перегружаемые функции

---

- ❑ `public void init()` -  
вызывается сразу после загрузки апплета
- ❑ `public void destroy()` -  
вызывается перед выгрузкой апплета
- ❑ `public void start()` -  
вызывается при появлении апплета на экране
- ❑ `public void stop()` -  
вызывается при исчезновении апплета с экрана

# Апплет посложнее

---

- Панель, по которой можно перемещать круглые фишки
- См. пример

# HTML-файл

---

- ❑ **<html>**
- ❑ **<title>**Страница со сложным апплетом**</title>**
- ❑ **<body>**
- ❑ **<p align=right>**
- ❑ Здесь находится сложный апплет
- ❑ **</p>**
- ❑ **<applet code="boardapplet.BoardApplet.class"**
- ❑ **width="800" height="300" align=right**
- ❑ **archive="BoardApplet.jar">**
- ❑ **</applet>**
- ❑ **</body>**
- ❑ **</html>**

# Протоколирование

---

- ❑ `static private String logStr = "";`
- ❑ `static private int counter = 0;`
- ❑ `private int id = 0;`
  
- ❑ `private void log(String msg) {`
- ❑  `logStr = logStr + msg + id + " ";`
- ❑  `if (statusLabel != null)`
- ❑  `statusLabel.setText(logStr);`
- ❑ `}`
- ❑ *// А почему бы не писать в файл?*

# Стандартные ограничения

---

- Нет доступа к файловой системе
- Нет доступа к внешним устройствам
- Нельзя запускать и останавливать приложения
- Нельзя связываться по сети с посторонними сайтами
- ...

# Функция init

---

```
□ public void init() {  
□     counter++; // Увеличение счетчика апплетов  
□     id=counter; // Установка собственного номера  
□     log("init");  
□     boardPanel = new BoardPanel();  
□     statusLabel = new JLabel();  
□     statusLabel.setText(logStr);  
□     statusLabel.setBorder(new LineBorder(Color.BLACK, 1));  
□     add(boardPanel, BorderLayout.CENTER);  
□     add(statusLabel, BorderLayout.SOUTH);  
□     log("^" + boardPanel.getWidth() + "x" +  
□         boardPanel.getHeight() + "^");  
□ }
```

# Функция start

---

```
 public void start() {  
     // ...  
     // Почему не в init()?  
     for (int i=0; i<4; i++)  
         boardPanel.addChip();  
     repaint();  
 }
```



# Выбор мышью

---

```
□ private void onMousePressed(MouseEvent e) {  
□     if (e.getButton() != MouseEvent.BUTTON1)  
□         return;  
□     for (Chip chip: chips) {  
□         if (chip.contains(e.getPoint())) {  
□             chosenChip = chip;  
□             return;  
□         }  
□     }  
□     chosenChip = null;  
□ }
```

# Перетаскивание мышью

---

```
❑ private void onMouseDragged(MouseEvent e) {  
❑     if (chosenChip==null) return;  
❑     Point p=e.getPoint();  
❑     Point oldCenter = chosenChip.getCenter(); // Старое положение  
❑     if (!chosenChip.moveTo(p)) return; // Было ли движение  
❑     if (!chosenChip.inBorder(getBounds())) { // Проверка границ  
❑         chosenChip.moveTo(oldCenter);  
❑         return;  
❑     }  
❑     for (Chip chip: chips) { // Проверка пересечений  
❑         if (chosenChip!=chip && chip.crossed(chosenChip)) {  
❑             chosenChip.moveTo(oldCenter);  
❑             return;  
❑         } }  
❑     repaint();  
❑ }
```

# Инициализация всплывающего меню

---

```
□ private void initMenu() {  
□     popupMenu = new JPopupMenu();  
□     addChipMenu = new  
        JMenuItem("Добавить фишку");  
□     addChipMenu.addActionListener(...);  
□     popupMenu.add(addChipMenu);  
□     clearChipsMenu = new  
        JMenuItem("Очистить поле");  
□     clearChipsMenu.addActionListener(...);  
□     popupMenu.add(clearChipsMenu);  
□ }
```

# Активация всплывающего меню

---

- @Override
- `public void processMouseEvent`  
    `(MouseEvent e) {`
- `if (e.isPopupTrigger())`
- `popupMenu.show(`  
           `e.getComponent(),`  
           `e.getX(), e.getY());`
- `else`
- `super.processMouseEvent(e);`
- `}`

# Изображение и звук в апплетах

---

## □ Загрузка:

- Image image =  
getImage(getCodeBase(), filename);

- AudioClip audioClip =  
getAudioClip(getCodeBase(), filename);

## □ getCodeBase() - URL-путь к html-файлу

## □ getDocumentBase() - URL-путь к директории с апплетом

# Отрисовка изображения

---

```
□ public void paint(Graphics g) {  
□     super.paint(g);  
□     int width = image.getWidth(observer);  
□     int height = image.getHeight(observer);  
□     if (width < 1 || height < 1)  
□         return;  
□     for (int x=1; x<getWidth()-width; x+=width) {  
□         for (int y=1; y<getHeight()-height; y+=height) {  
□             g.drawImage(image, x, y, width, height,  
□                 observer);  
□         }  
□     }  
□ }
```

# Проигрывание звука

---

- ❑ `clip.play()` - однократное проигрывание
- ❑ `clip.stop()` - остановка проигрывания
- ❑ `clip.loop()` - зацикливание проигрывания

# Итоги

---

- ❑ Механизмы запуска апплетов
- ❑ Особенности функционирования апплетов
- ❑ Воспроизведение изображений и звука