­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Лабораторна робота №10

з курсу «Об’єктно-орієнтоване програмування»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 14

Виконав студент гр. КНз-2

Чалий Михайло

­­

Львів 2014

# Програмування аплетів в Java. Обробка подій в Java.

## Мета роботи

Навчитися програмувати аплети в Java, навчитися обробляти події в Java.

## Теоретичні відомості

Аплети(applets) - це програми, що працюють в середовищі іншої програми - браузера. Аплету не потрібне вікно верхнього рівня - їм служить вікно браузера. Вони не запускаються JVM — їх завантажує браузер, котрий сам запускає JVM для виконання аплету. Ці особливості відбиваються на написанні програми аплета.

З точки зору мови Java, аплет — це всяке розширення класу Applet, котрий, в свою чергу, розширяє клас Panel. Таким чином, аплет - це панель спеціального виду, контейнер для розміщення компонентів з додатковими властивостями і методами. Менеджером розміщення компонентів по замовчуванню, як і в класі Panel, служить FІowLayout. Класс Applet знаходиться в пакеті java.applet, в якому крім нього єсть тільки три інтерфейси, реалізовані в браузері. Треба відмітити, що не всі браузери реалізують ці інтерфейси повністю.

Оскільки JVM не запускає аплет, відпадає необхідність в методi main(), його немає в аплетах. В аплетах рідко зустрічається конструктор. Справа в тому, що при запуску першого створюється його контекст. Під час виконання конструктора контекст ще не сформований, тому не всі початкові значення вдається визначити в конструкторі. Початкові дії, зазвичай виконувані в конструкторі і методі main(), в аплеті записуються в метод init() класу Applet. Цей метод автоматачно запускається виконуючою системою Java браузера зразу ж післе завантаження аплета. Ось як він виглядає у вихідному коді класу Applet:

public void init(){}

Не густо! Метод init() не має аргументів, не повертає значення і повинен перевизначатися в кожному аплеті — підкласі класу Applet. Зворотні дії — завершення роботи, звільнення ресурсів — записуються при необхідності в метод destroy(), також виконуваний автоматично при вивантаженні аплету. В класі Applet єсть пуста реалізація цього методу.

Крім методів init() і destroy() в класі Applet присутні ще два пустих методи, виконувані автоматично. Браузер повинен звертатися до методу start() при кожній появі аплету на екрані і звертатися до методу stop()*,* коли аплет зникаєт з eкрана. В методі stop() можна визначити дії, зупиняючі роботу аплета, в методі start() — відновлюючі її. Треба зразу ж відмітити, що не всі браузери звертаються до цих методів як повинно. Так, перший із розглянутих нижче аплетів HelloWorld.html мені не вдалося запустити браузером Internet Explorer, прийшлося скористатися власним Java браузером appletviewer, як пояснено далі.

Роботу указаних методів можна пояснити простим житійським прикладом. Приїхавши весною на дачу, ви прокладаєте водопровідні труби, прикручуєте крани, протягуєте шланги - виконуєте метод init() для своєї зрошувальної системи. Після цього, приходячи на ділянку, включаєте крани — запускаєте метод start(), а виходячи, виключаєте їх — виконуєте метод stop(). Нарешті, восени ви розбираєте зрошувальну систему, відкручуєте крани, просушуєте і укладуєте водопровідні труби - виконуєте метод destroy().

## Завдання

При клацанні мишею по зображенню в графічному вікні воно міняється на інше зображення такого ж розміру, а при повторному клацанні відновлюється первинне зображення.

## Реалізація

Лістінг 1. PlayImage.java

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** java.applet.\*;

**public** **class** PlayImage **extends** Applet {

// При клацанні мишею по зображенню в графічному вікні воно

// міняється на інше зображення такого ж розміру,

// а при повторному клацанні відновлюється первинне зображення.

**private** **static** **final** **long** *serialVersionUID* = -5756110082683888874L;

**private** Image img1;

**private** Image img2;

**private** **boolean** drawSecond;

**public** **void** init(){

img1 = getImage(getDocumentBase(), "img1.jpg");

img2 = getImage(getDocumentBase(), "img2.jpg");

setSize(500, 281);

addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent arg0) {

toggleImage();

}

});

}

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

Image current = drawSecond ? img2 : img1;

**int** width = current.getWidth(**this**);

**int** height = current.getHeight(**this**);

g.drawImage(current, 0, 0, width, height, **this**);

}

**private** **void** toggleImage(){

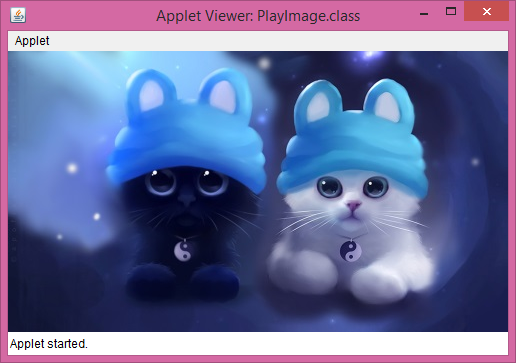
drawSecond = !drawSecond;

**this**.repaint();

}

}

## Результат



## Висновки

Навчитився програмувати аплети в Java, навчитився обробляти події в Java.