# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №2**

**«Побудова аплетів з графічними зображеннями»**

**з дисципліни «Програмне забеспечення мережевих технологій»**

Виконав:

слухач групи ПЗС-11

Гринчук Тарас

Прийняв:

ст. викл.Сенів М.М.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2014

**Тема роботи**: Побудова аплетів з графічними зображеннями.

**Мета роботи:** Освоїти елементи програмування мовою JAVA при розробленні аплетів, а також засвоїти класи і методи бібліотек java.awt, java.applet.

## 1. Теоретичні відомості

Java-аплет - прикладна програма, найчастіше всього, написана мовою програмування Java у формі байт - коду. Java-аплети виконуються в браузері з використанням віртуальної Java машини(JVM), або в Sun's AppletViewer, автономному засобі для випробування аплетів. Java- аплети були впроваджені в першій версії мови Java в 1995 році. Java-аплети звичайно пишуться на мові програмування Java, але можуть бути написані і на інших мовах , які компілюються в байт -код Java , таких , як Jython.

Аплети використовуються для надання інтерактивних можливостей веб-додатків, які не можуть бути надані HTML. Так як байт-код Java платформо-незалежний, Java- аплети можуть виконуватися за допомогою плагінів браузерами багатьох платформ, включаючи Microsoft Windows, UNIX, Apple Mac OS і GNU / Linux. Такі програми з відкритим вихідним кодом, як applet2app, можуть бути використані для перетворення аплета в самостійні програми на Java або виконувані файли Linux і Windows.

Java Servlet іноді неофіційно порівнюють з серверним апплетом, але він відрізняється своєю мовою, функціями і кожної з описаних тут характеристик аплетів.

Java-аплети в більшості браузерів виконуються в «пісочниці», запобігаючи їх доступ до локальних даних. Код аплету завантажується з веб-сервера і браузер або вставляє аплет у веб-сторінку, або відкриває нове вікно з власним користувача інтерфейсом аплету. Аплет може бути відображений на веб-сторінці шляхом використання застарілого HTML елемента applet, або рекомендованого елемента object. Цим визначається місце розташування і джерело аплету.

Головний клас аплета розширює клас java.applet.Applet або, якщо створюється Swing аплет, javax.swing.JApplet. Клас повинен перевизначити методи створення користувальницького інтерфейсу всередині себе (Applet є нащадком Panel, який, у свою чергу, є нащадком Container).

Переваги використання Java-аплетів:

* кроссплатформенність;
* аплет може працювати на «всіх» встановлених до цього часу версіях Java, а не тільки з останньою версією, проте, якщо аплет вимагає останню версію JRE, то клієнт буде змушений чекати більш тривалої завантаження;
* апплет підтримується більшістю браузерів;
* він кешується в більшості браузерів, а тому буде швидко завантажуватися при поверненні на веб-сторінку; але аплет може зберігатися в кеші і створювати проблеми після виходу нових версій;
* він може мати повний доступ до машини, на якій виконується, якщо користувач згоден на це;
* він може запуститися з порівнянною (але зазвичай повільніше) швидкістю на інших компільованих мовах, таких як C++, але у багато разів швидше, ніж JavaScript;
* він може перенести роботу з сервера до клієнта, обслуговцючи інтернет-рішення з великим числом користувачів / клієнтів.

## 2. Хід роботи

**Варіант 5.** Створити аплет, у вікні якого після подвійного кліку виникає безперевно рухається текст. Наступний подвійний клік змінює текст. Клавіша “delete” очищає вікно.

В середовищі netbeans створимо новий проект наберемо текст JAVA-аплета:

import java.applet.\*;

import java.awt.\*;

import java.util.\*;

public class MyApplet extends Applet {

private int x, y; //поточні координати

private int tokenIndex; //індекс поточного слова

private String[] token; //масив слів

private boolean started; //true - виводимо текст, false - пустий екран

private Timer timer; //таймер

private MyTask t; //завдання таймера

@Override

public void init() { //початкове заповнення властивостей аплета

started = false;

x = 20; y = 20;

tokenIndex = -1;

//розбиваємо стрічку на масив слів

token = "Hello! This is my first Java-applet!".split(" ");

//створюємо таймер

timer = new Timer();

t = new MyTask(this);

//викликаємо кожних 200 мсек.

timer.schedule(t, 0, 200);

}

@Override

public void paint(Graphics g) {

//задамо кольори фону та тексту

setBackground(Color.BLUE);

g.setColor(Color.YELLOW);

//текст не виводимо

if(!started) return;

//виводимо поточне слово у поточних координатах

g.drawString(token[tokenIndex], x, y);

}

@Override

public boolean mouseDown(Event evt, int x, int y) {

//якщо подвійний клік

if(evt.clickCount > 1) {

//дозволяємо виводити текст

started = true;

repaint();

//якщо вивели останнє слово в реченні - починаємо з першого

if(++tokenIndex >= token.length) tokenIndex = 0;

}

return true;

}

@Override

public boolean keyDown(Event evtObj, int key) {

//натиснули клавішу DELETE

if(key == Event.DELETE) {

//зупинка таймера

t.cancel();

//переініціалізація властивостей аплета

init();

repaint();

}

return true;

}

//переміщення тексту у вікні

public void Move() {

if(!started) return;

/\* збільшуємо координати, споочатку по осі Y,

\* а потім - Х, з вказаним кроком та контролем границь

\*/

y += 10;

if (y > 200) {

y = 20;

x += 15;

}

if (x > 200) x = 20;

//перемальовуємо вікно

repaint();

}

}

//клас завдання таймера

class MyTask extends TimerTask {

//ссилка на наш аплет

private MyApplet app;

//конструктор

public MyTask(MyApplet app) {

super();

this.app = app;

}

//при завершенні певного проміжку часу - перемалбовуємо текст аплета

@Override

public void run() {

app.Move();

}

}

Відредагуємо файл MyApplet.html, який було автоматично створено середовищем netbeans до наступного вигляду:

<HTML>

<HEAD>

<TITLE>Л/р №2</TITLE>

</HEAD>

<BODY>

<H3><HR WIDTH="100%">Побудова аплетів з графічними зображеннями<HR WIDTH="100%"></H3>

<P>

<APPLET codebase="classes" code="org/me/hello/MyApplet.class" width=350 height=200></APPLET>

</P>

<HR WIDTH="100%"><FONT SIZE=-1><I>Автор: Тарас Гринчук</I></FONT>

</BODY>

</HTML>

Запустимо даний документ у браузері (рис. 2.1):

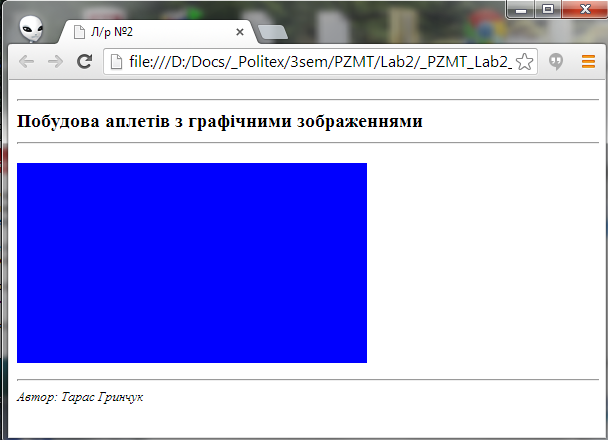


Рис. 2.1. Вигляд html-документа у браузері

Зробимо подвійний клік на формі вікна (рис. 2.2):

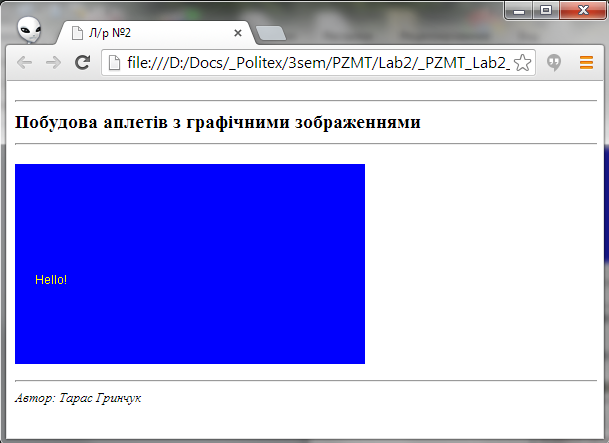


Рис. 2.2

Як бачимо з’явився біжучий текст. Зробимо ще декілька подвійних натискань мишою (рис. 2.3):

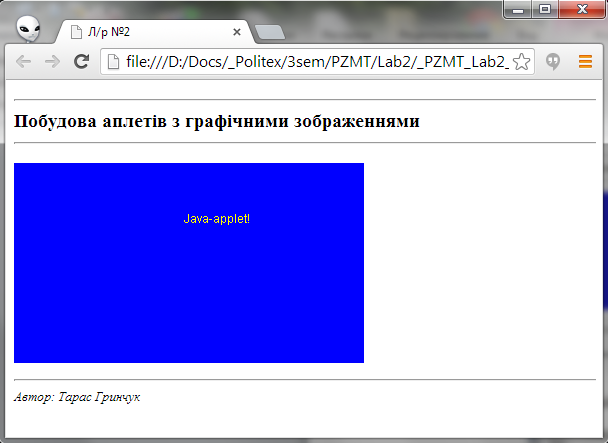


Рис. 2.3

Текст змінюється. Натиснемо на клавіатурі клавішу delete і побачимо, що екран очиститься, як на рис. 2.1. Якщо зараз зробити подвійний натиск на вікні – то текст знову почне переміщатися з початкової позиції.

## ВИСНОВКИ

На даній лабораторній роботі я вивчив основні принципи побудови JAVA-програм, освоїв елементи програмування мовою JAVA при розробленні аплетів, а також засвоїв класи і методи бібліотек java.awt, java.applet.