

Google Web Toolkit

Elek Márton
Drótposta kft.

Tartalom

A GWT általában

Java forrás, Web mód, Hosted mód

Speciális szolgáltatások

History kezelés, távoli eljárás hívás,

Külső függvénykönyvtárak

GWT-Ext, EXT GWT, Gwittir, HTMLTemplate, GWT-

Log

Tapasztalatok, jövő

GWT 1.5

Célok

To radically improve
the web experience for users

by enabling developers
to use existing Java tools

to build no-compromise **AJAX**
for any modern browser

GWT

Java forrás



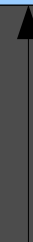
GWT fordító



JavaScript



Szerver oldal
PHP



HTML

JavaScript

Miért jó ez nekünk I.

Miért Java? Meglévő jól bevált eszközeinket szeretnénk használni.

Bármilyen Java IDE-t használhatunk (kódszinezés, kiegészítés, dokumentáció)

Tesztelhető (egységtesztek)

Debugolható!

Szabványosan dokumentálható (Javadoc)

Forrás analizátorok, teszt lefedettség mutatók, stb.

Újrahasznosítható és terjeszthető JAR fájlakon keresztül

Miért jó ez nekünk II.

Fordítás

Fordítás közben optimalizáció

Gyorsabban töltődik be, mint egy kézzel írt JavaScript

Kisebb, kompaktabb, cachelhető kód

Automatikusan támogatja a különböző böngészőket
(IE, Firefox, Mozilla, Safari, és Opera)

Hibák egy része fordítási időben kiderül

```
log.debug("Nem találom a partnert " + p.getId() + " !");
```

Web mód

Java forrás



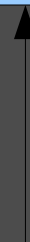
GWT fordító



JavaScript



Szerver oldal



HTML

JavaScript

Hosted mód

Java forrás



GWT Hosted mód képernyő

Szerver oldal,



GWT for

HTML

JavaScript



Hosted mód / web mód

Hosted mód

fejlesztéshez

speciális Java/Mozilla
alapú GUI-ban futtatja a
kódot

Debuggolható
bármilyen Java
debugger képes IDE-vel

Refresh esetén
újrafordít

Web mód

éles üzembe

Bármilyen web
böngészőben fut

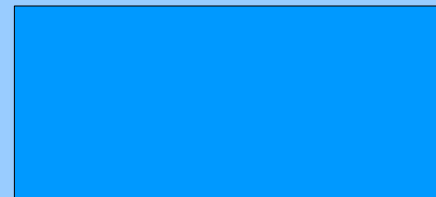
Példa alkalmazás

Pelda.java

Pelda.gwt.xml

Szerver oldal

index.html



Egy egyszerű példa: a Java kód

```
public class Hello implements EntryPoint {  
  
    public void onModuleLoad() {  
  
        Button b = new Button("Click me");  
        RootPanel.get().add(b);  
  
    }  
}
```

Egy egyszerű példa: a Java kód

```
public class Hello implements EntryPoint {  
  
    public void onModuleLoad() {  
        Button b = new Button("Click me",  
            new ClickListener() {  
                public void onClick(Widget sender) {  
                    Window.alert("Hello, Ajax");  
                }  
            });  
        RootPanel.get().add(b);  
    }  
}
```

Egy egyszerű példa: a HTML

```
<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
  </head>
  <body>
    <script type="text/javascript" language='javascript'
src='hu.conf.web.Pelda.nocache.js'></script>
  </body>
</html>
```

Egy egyszerű példa: az xml

```
<?xml version="1.0" encoding="UTF-8"?>
<module>
  <source path='client' />

  <inherits name='com.google.gwt.user.User' />

  <entry-point class='hu.conf.web.client.Pelda' />
</module>
```

Tartalom

A GWT általában

Java forrás, Web mód, Hosted mód

•**Speciális szolgáltatások**

History kezelés, távoli eljárás hívás,
Külső függvénykönyvtárak

GWT-Ext, EXT GWT, Gwittir, HTMLTemplate, GWT-
Log

Tapasztalatok, jövő

GWT 1.5

JavaScript Native Interface

A Java kódból hívhatunk tiszta JavaScriptet:

```
public static native void alert(String msg) /*-{  
    $wnd.alert(msg);  
}-*/;
```

Hagyományos JavaScriptből hozzáférünk a generált
Javascripthez

```
@hu.conf.web.PartnerService::getPartner(Ljava/lang/String;  
g;)(s);
```


Távoli eljárás hívás

Java forrás



GWT fordító



JavaScript



Szerver oldal



HTML

JavaScript

Távoli eljárás hívás II.

- GWT RPC

 - Interfészeken keresztül közös felület a szerver és a kliens oldalon

 - Aszinkron modell

 - A szerializáció nem a mi dolgunk

 - Tömörített adatátvitel

- JSON

 - Az aszinkron lekérés eredményéből JSON objektumot csinálunk

 - Nem típusos adattárolás

Obfuszkálás és optimalizáció

- kiseb
- tömörebb
- gyorsabb

-style OBFUSCATED

-style PRETTY (emberi olvasásra)

-style DETAILED (hibakeresésre, elemzésre)

Kódgenerálási stílus: DETAILED

```
_ = hu_conf_web_client_Pelda.prototype = new  
_java_lang_Object();
```

```
_.java_lang_Object_typeId$ = 0;
```

```
function  
hu_conf_web_client_Pelda$1_onClick__Lcom_google_gwt_user  
_client_ui_Widget_2(sender){  
com_google_gwt_user_client_Window_alert__Ljava_lang_Stri  
ng_2('Hello, Ajax');  
}
```

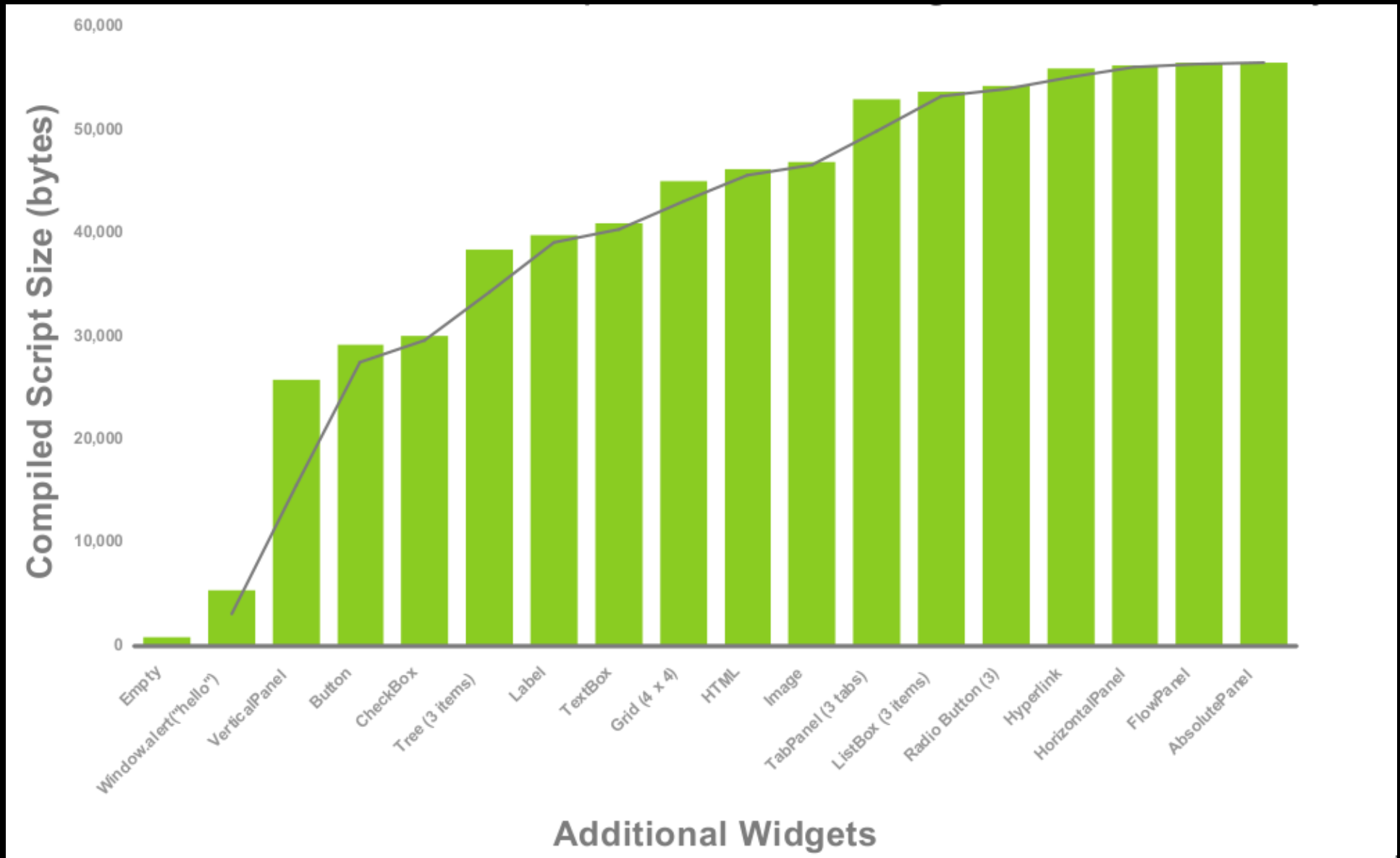
Javascript stílus: PRETTY

```
_ = Pelda.prototype = new Object_0();  
_.typeId$ = 0;  
function onClick(sender){  
    alert('Hello, Ajax');  
}
```

Obfuszkált stílus: OBF

```
_ =jh.prototype=new ci();_.tI=0;function mh(a){Fc('Hello,  
Ajax');}
```

A generált kód mérete



History kezelés

- Előzménysort tölthetünk fel Stringekkel

`view/partner/12`

`edit/partner/12`

- Ezek között navigálhatunk előre és hátra
- Változás után az aktuális értéket megkapja a kezelő osztály, aki betölti a String alapján az előző/következő állapotot.

Tartalom

A GWT általában

Java forrás, Web mód, Hosted mód

Speciális szolgáltatások

History kezelés, távoli eljárás hívás,

• **Külső függvénykönyvtárak**

GWT-Ext, EXT GWT, Gwittir, HTMLTemplate, GWT-

Log

Tapasztalatok, jövő

GWT 1.5

GWT-Ext vs. Ext GWT

- Ext GWT

régebbi néven MyGWT

Tisztán **Java alapú** widget könyvtár, az ExtJS könyvtárhoz hasonló widgetekkel

Layout rendszer

- GWT-EXT

Csak egy burkuló Java felület

A felszín alatt az Ext JS JavaScript könyvtár kódját hívja meg

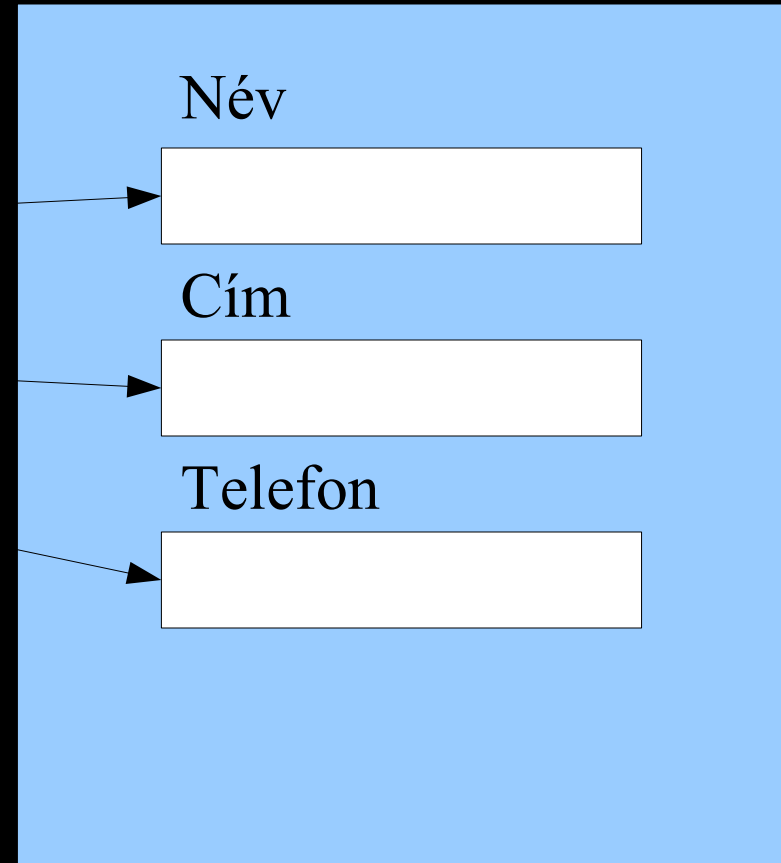
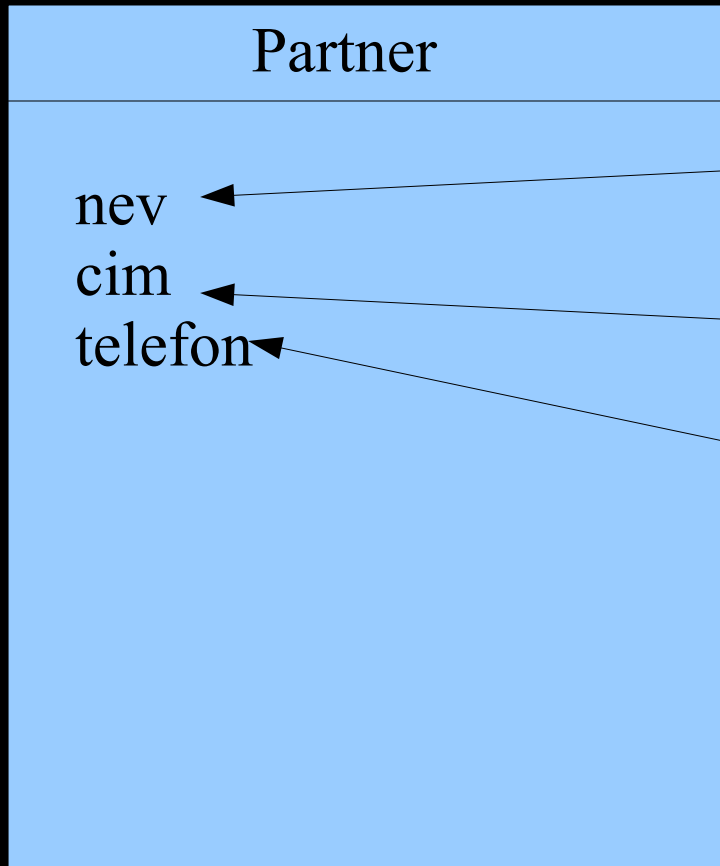
Sokkal gazdagabb widget kínálat

- Mindkettő licensze LGPL-ről GPL-re változott.

Gwittir

- Általános MVC keretrendszer, ami néhány egyéb szolgáltatást is nyújt
- Adat másolás (binding)
- Introspekció
- Animációs lehetőségek
- Fejlettebb billentyűzetkezelés
- Loggolás

Gwittir binding



Gwittir introspekció

•Introspekció

```
PropertyDescriptor bd =  
Introspector.INSTANCE.getDescriptor(Partner.class);
```

```
Property property = bd.getProperty("cim");
```

```
String cim =  
property.getAccessorMethod().invoke(partner, new  
Object[]);
```

•Binding

```
Binding b = new Binding( partner, "nev", form , "nev");  
b.bind();
```

HTMLTemplate

Header.html

```
<div><span>Itt egy gomb<span>  
    <input type="button" id="search"/>  
</div>
```

Header.java

```
HTMLWidget widget = new HTMLWidget("Header.html");  
widget.add("search", new Button("Keresés"));
```

- Külső designer is könnyen belenyúlhat a megjelenítésbe.
- A html sablonokat újra és újra letölti, nem cache-el.

GWT-log

Loggolás

Firebug konzolba

Felugró HTML DIV elembe

Szerver oldalra

Standard output

Hosted mód konzolja

Szintek már vannak, de osztály szintű loggerek még nincsenek

•Tartalom

A GWT általában

Java forrás, Web mód, Hosted mód

Speciális szolgáltatások

History kezelés, távoli eljárás hívás,

Külső függvénykönyvtárak

GWT-Ext, EXT GWT, Gwittir, HTMLTemplate, GWT-

Log

•Tapasztalatok, jövő

GWT 1.5

Hátrányok, nehézségek

- Java 1.5 hiánya (annotációk, generikus típusok)
- Viszonylag lassú fordítási idő
- Optimalizáció

Arrays.sort

- A keretrendszerek még elég alap funkcionalitással bírnak
- nem lehet lefordított részprojektet lefordítva újrahasznosítani

A jövő

- Jelenlegi verzió: 1.4
- Következő verzió 1.5 (Jelenleg Milestone 2)

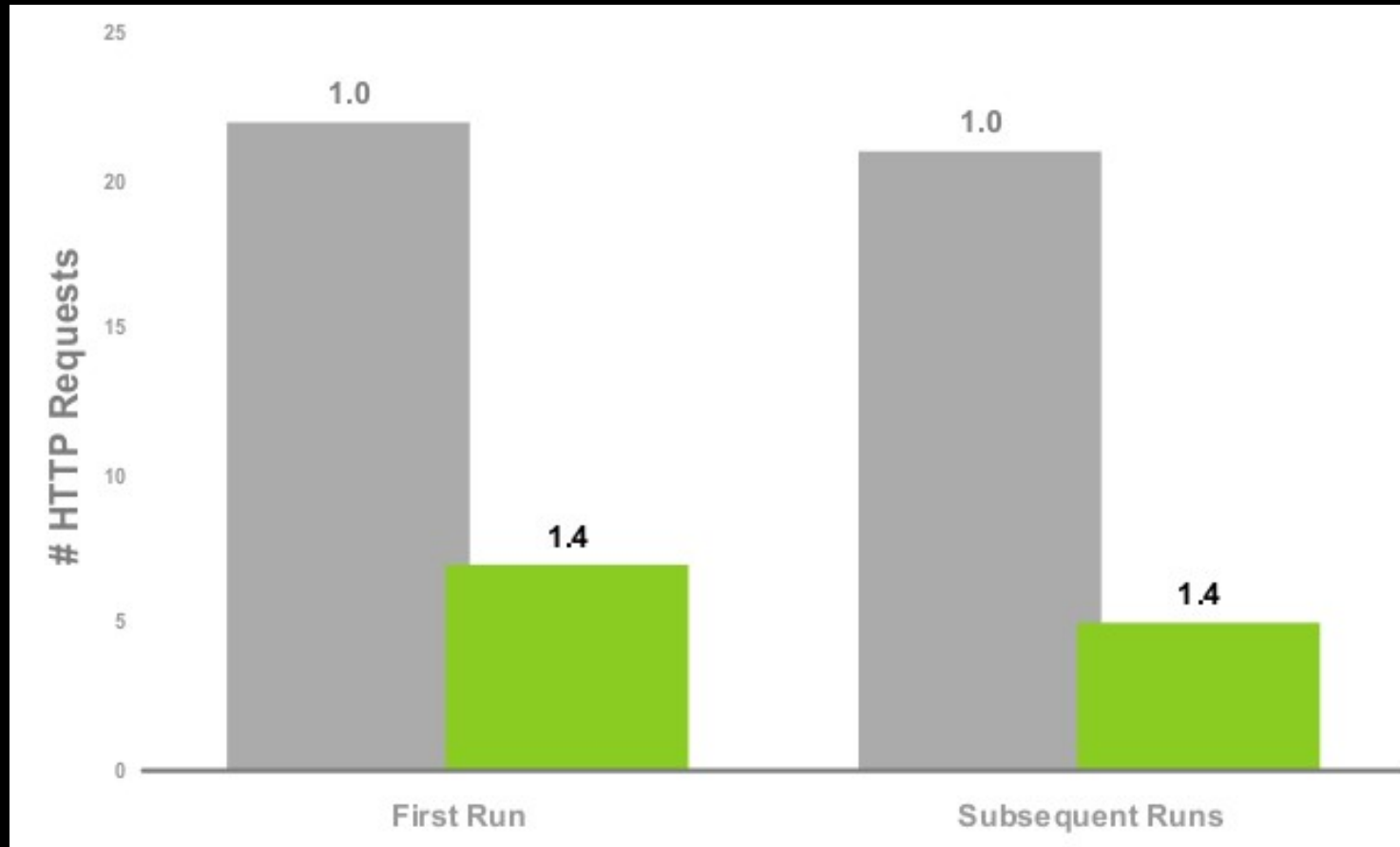
Java 1.5 hiánya -> lesz Java 1.5

Lassú fordítási idő -> Modulárisabb kód

Kevésbé kiforrott keretrendszerek

Optimalizáció -> Gyorsabb lesz, jobb natív interfész

Optimalizáció



Összefoglalás

- professzionális fejlesztő eszközök
- széles körben elterjedt
- jó platform támogatás
- az 1.4-es változattól kezdve (>1 év) már éles környezetben is használható
- az 1.5-ös további látványos fejlődést hozhat

Kérdések / elérhetőségek

marton.elek@mail.drotposta.hu

<http://jum.hu> -- Java User Group

<http://jhacks.anzix.net> -- Java Wiki

Élő alkalmazások

- Google Checkout
- Google Base
- <http://www.queplix.com/>
- <http://myhippocampus.com>
- <http://www.threerings.net/whirled/>
- <http://www.crazybob.org/twubble/>

Arrays.sort I.

```
private static void mergeSort(Object[] src,
                               Object[] dest,
                               int low,
                               int high,
                               int off) {
    int length = high - low;

    // Insertion sort on smallest arrays
    if (length < INSERTIONSORT_THRESHOLD) {
        for (int i=low; i<high; i++)
            for (int j=i; j>low &&
                ((Comparable) dest[j-1]).compareTo(dest[j])>0; j--)
                swap(dest, j, j-1);
        return;
    }

    // Recursively sort halves of dest into src
    int destLow = low;
    int destHigh = high;
    low += off;
    high += off;
    int mid = (low + high) >>> 1;
    mergeSort(dest, src, low, mid, -off);
    mergeSort(dest, src, mid, high, off);
}
```

Arrays.sort II.

```
arr.sort()
```