Tutorial Eclipse

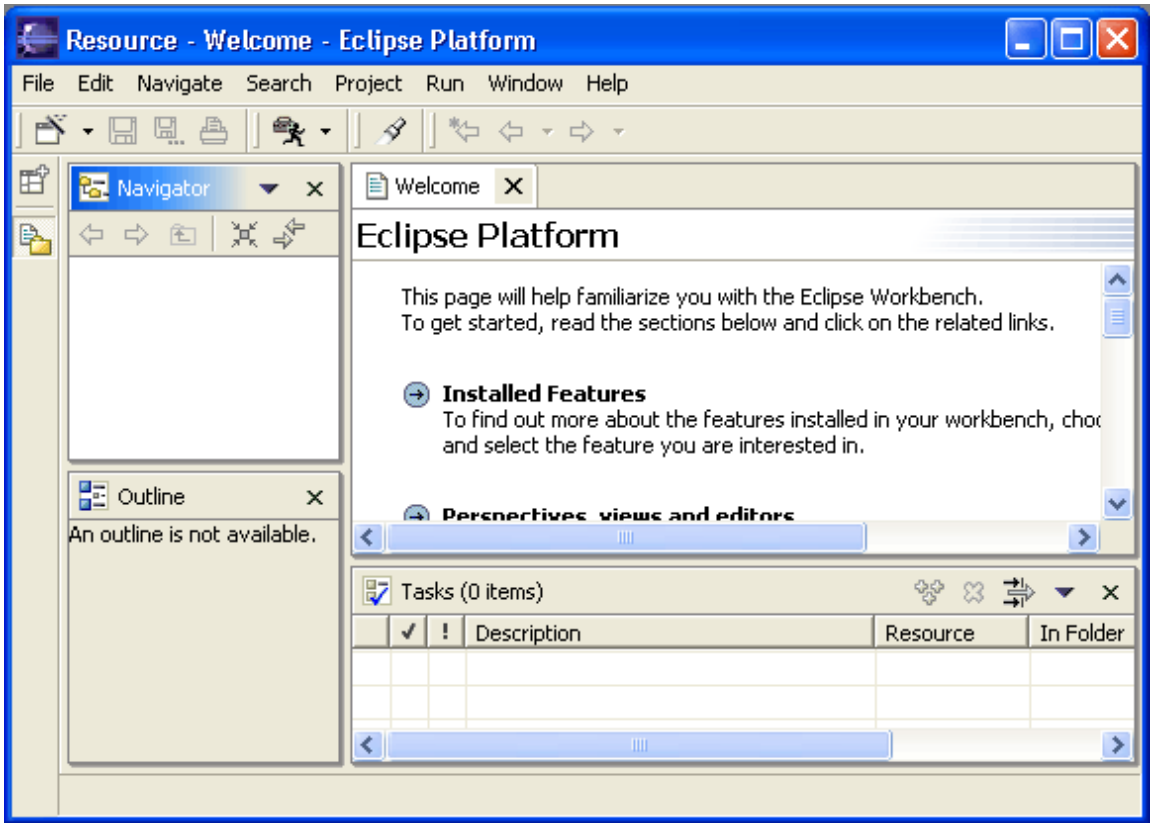
Ginanjar Utama
ginanjar_utama@yahoo.com

Workbench – Tempat Bekerja

Tempat bekerja dalam eclipse disebut dengan workbench. Dalam jendela workbench akan ditampilkan beberapa perspective. Pada saat awal Resource perspective yang ditampilkan, selanjutnya kita akan berada pada posisi terakhir kita meninggalkan eclipse. Sebuah perspective mengandung beberapa view (contoh Navigator) dan editor. Lebih dari satu jendela Workbench dapat dibuka bersamaan.

Pada sebelah kiri dari jendela ada icon shortcut untuk pindah perspective atau menampilkan perspective baru. Nama dari perspective yang aktif ditunjukkan pada judul jendela dan iconnya yang tampak melesak ke dalam.

Dengan melihat judul dari jendela Workbench maka kita tahu bahwa kita sedang berada pada Resource perspective. Navigator, Tasks, dan Outline view juga terbuka, bersamaan dengan editor pada halaman Welcome. Silahkan mencoba menelusuri halaman tersebut untuk mengetahui kegunaan eclipse lebih jauh lagi atau memodifikasi workbench.



Memasang Plugin Baru: JDT Example Plug-ins

Untuk menambah feature baru di eclipse, kita hanya perlu memasang plugin baru. Caranya juga sangat mudah, biasanya tinggal diekstract saja ke direktori pluginsnya eclipse.

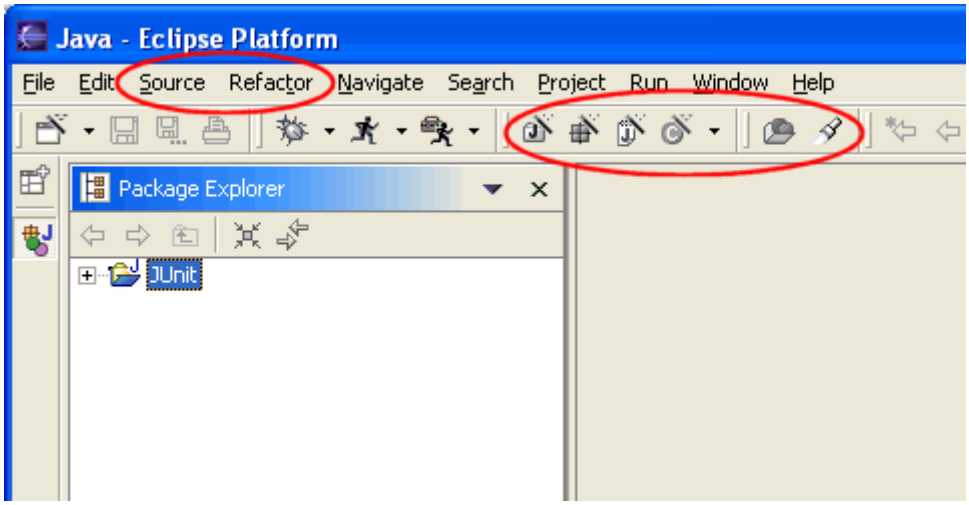
Plugin JDT example menyumbangkan New Wizard yang secara otomatis membuka project percontohan JUnit pada workbench anda. Pada tutorial ini, kita akan secara manual membuat project baru supaya tahu langkah-langkahnya. Tapi sebelumnya anda perlu mendownload danmemasang plugin tersebut (jika anda belum melakukannya).

1. Pergi ke http://www.eclipse.org/downloads/ dan temukan release eclipse yang kita gunakan.
2. Pindah ke bagian Example Plug-ins dan ikuti perintah pemasangan.
3. Jika workbench sedang berjalan, matikan saja. Plug-ins tidak boleh dipasang pada saat workbench hidup.
4. Extract isi dari file Zip ke dalam direktori eclipse (e.g. c:\eclipse).
5. Jalankan workbench.

Membuat project Java yang pertama.

Pada bagian ini, kita akan membuat sebuah project Java baru, dengan menggunakan JUnit sebagai project percontohan. JUnit adalah open source unit testing framework untuk Java. Lihat lebih dalam di http://www.junit.org/ untuk informasi mengenai JUnit.

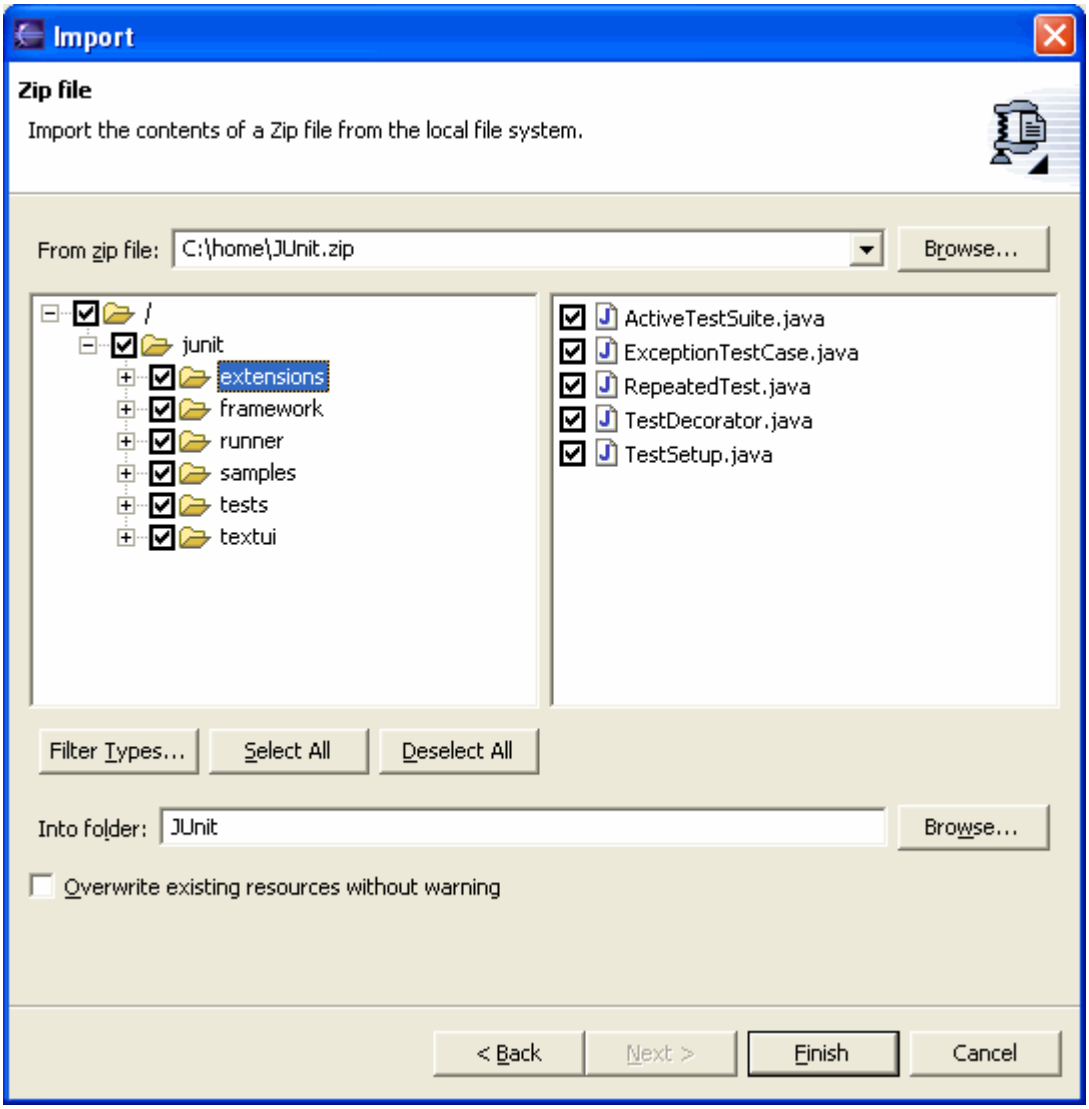
1. Pilih menu item File > New > Project.... untuk membuka wizard New Project.
2. Pada panel kiri halaman wizard, pilih Java, dan di sebelah kanan, pilih Java Project. Kemudian klik Next. Pada halaman berikutnya, ketik "JUnit" pada field Project name dan klik Finish. Satu perspective Java terbuka di dalam workbench dengan project Java yang baru di dalam Package Explorer. Ketika perspective Java aktif, pilihan menu baru dan tombol-tombol khusus Java muncul dalam toolbar workbench. Bergantung pada view atau editor yang sedang aktif, tombol dan pilihan menu yang baru juga tersedia.



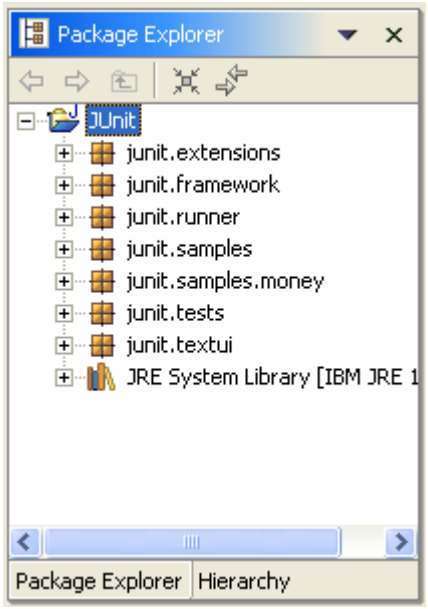
3. Dalam Package Explorer, pastikan project JUnit terpilih. Pilih item menu File > Import....
4. Pilih Zip file, kemudian klik Next.
5. Klik tombol Browse di sebelah field Zip file dan pilih <eclipseInstallPath>/plugins/org.eclipse.jdt.ui.examples.projects/archive/junit/junit37src.jar.

Catatan: Asumsinya JDT example plug-ins telah berhasil dipasang.

6. Dalam Import wizard, di bawah daftar hirarki pilih Select All. Anda dapat memperluas dan memilih elemen-elemen apa yang ingin diimpor. Dalam tutorial ini kita pilih semua elemen.



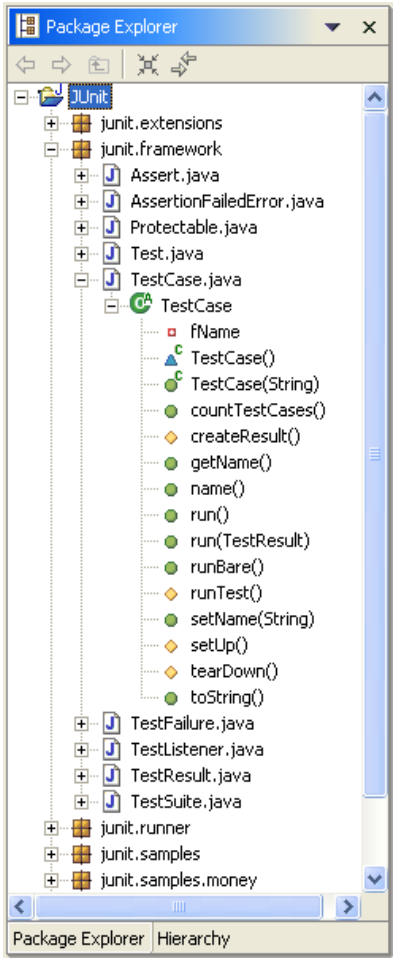
7. Pastikan project JUnit muncul pada Folder tujuan, lalu klikFinish. Pada saat diimpor maka resource langsung di build secara otomatis. Setting ini bisa diubah pada halaman preferensi dari Workbench. Kita timpa .classpath yang lama
8. Dalam viewPackage Explorer, buka project JUnit untuk melihat package-packagenya.



Menggunakan Package Explorer

elemen-elemen Java di dalam project JUnit project.

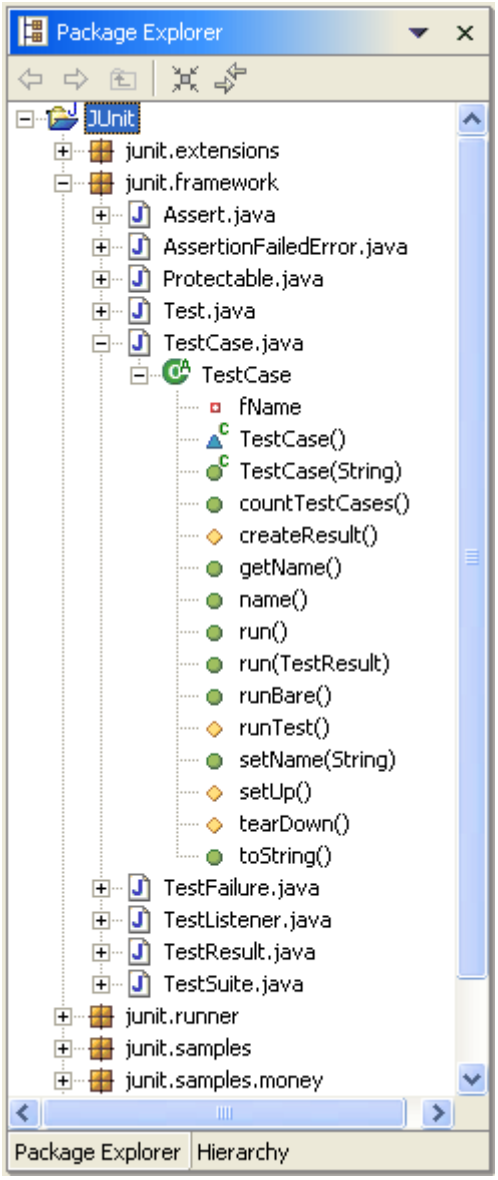
1. Perluas package junit.framework untuk melihat file-file Java files yang ada di dalamnya.
2. Perluas file TestCase.java. Perhatikan bahwa Package Explorer menampilkan garis besar dari source code. Deklarasi import, public type, member dari class (field and method) muncul dalam struktur.



Melihat elemen Java

Pada bagian ini, kita akan melihat isi dari project JUnit.

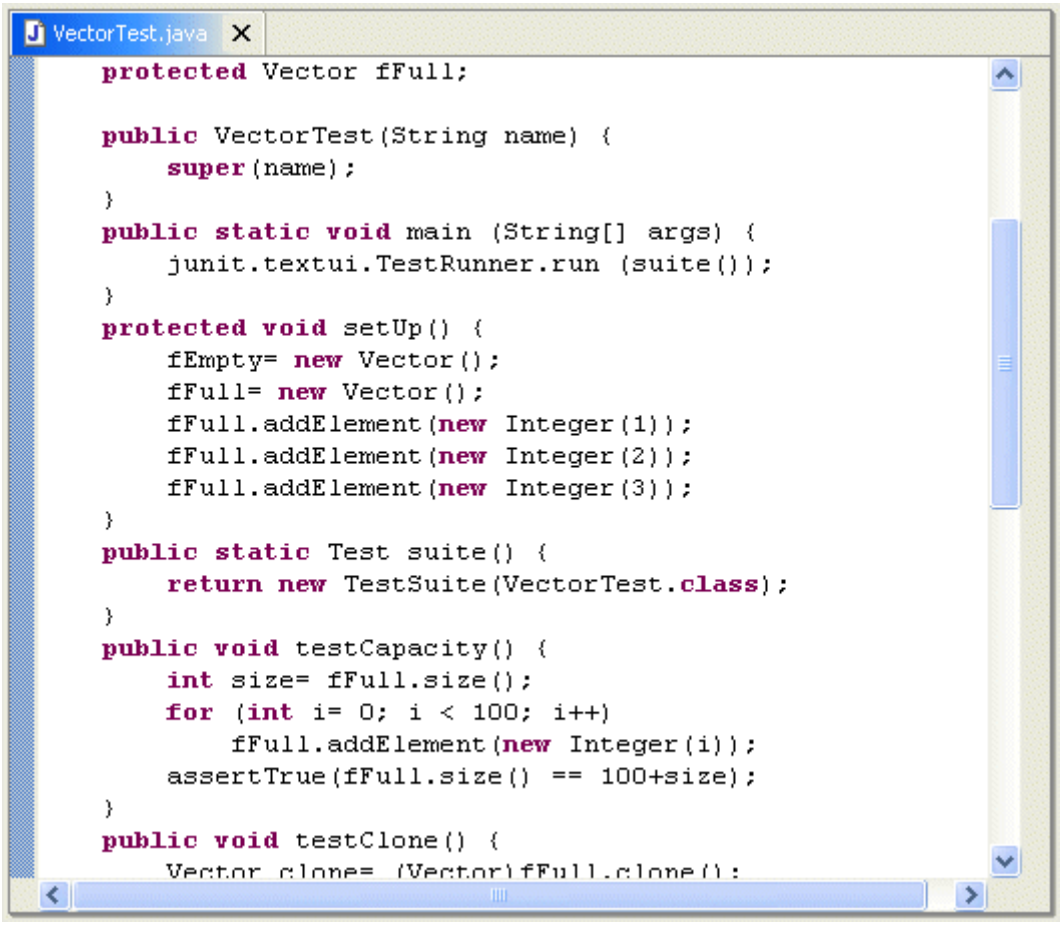
1. Buka package junit.framework untuk melihat file-file Java di dalam package tersebut.
2. Kembangkan file Java TestCase.java.



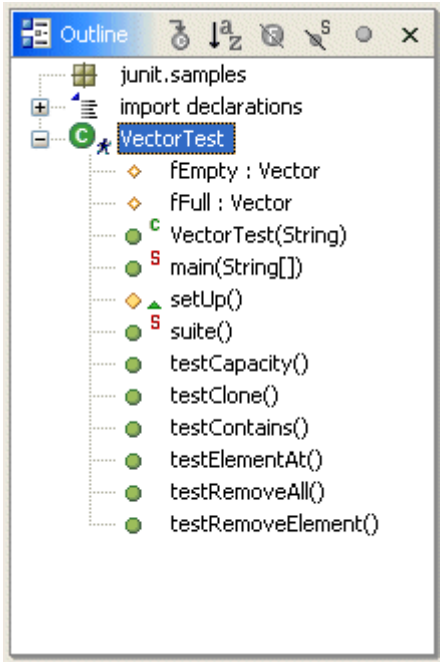
Memakai editor

Kita akan belajar dasar-dasar mengedit file Java.

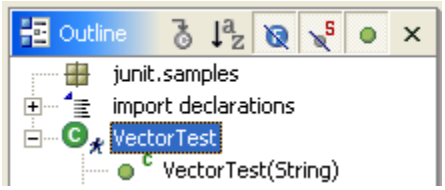
1. Perluas package *junit.samples* dan pilih file *VectorTest.java*. Buka editor untuk *VectorTest.java* dengan klik ganda. Umumnya kita dapat membuka editor hanya dengan mengklik ganda pada tempat yang ingin kita buka. Contohnya untuk membuka langsung method *testClone* dalam *VectorTest.java* klik ganda pada nama method dalam Package Explorer.
2. Perhatikan warna dari source code. Beberapa hal mempunyai warna yang berbeda:
   - Komentar umum
   - Komentar Javadoc
   - Keyword
   - String.



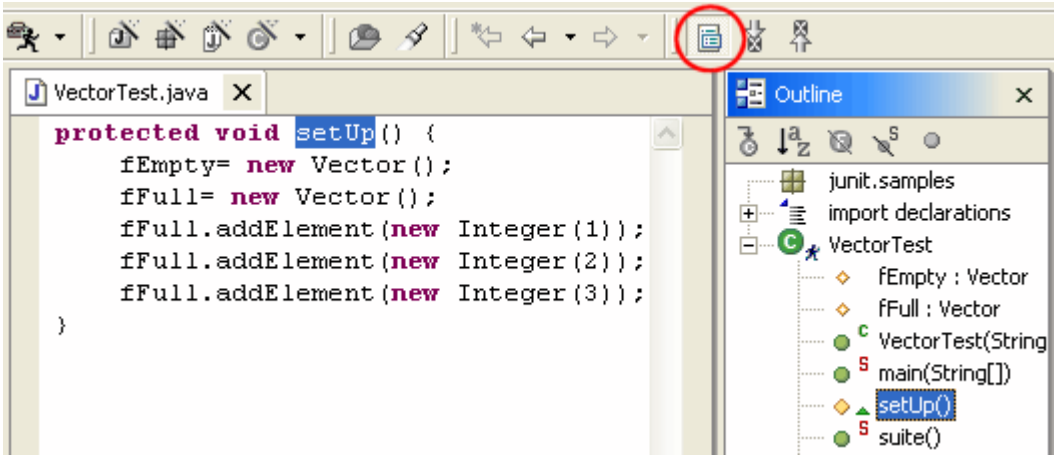
3. Lihat pada Outline view. Ia menampilkan garis besar dari file Java tersebut termasuk pernyataan package, import, field, method dan type. Icon-icon digunakan untuk menjelaskan banyak hal seperti apakah suatu elemen itu static, abstract, atau final. Atau apakah ia override dari base class atau interface.



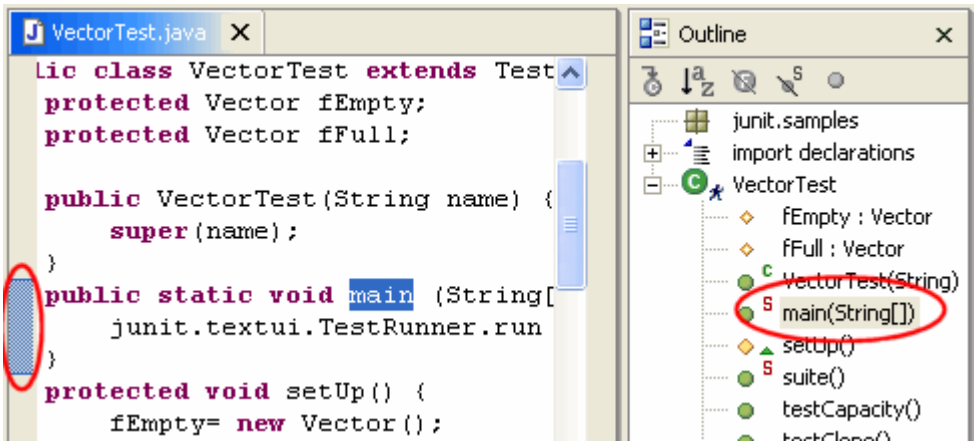
4. Tekan tombol **Hide Fields**, **Hide Static Members**, dan **Hide Non-Public Members** pada toolbar Outline view untuk menyaring tampilan.



5. Kita dapat mengedit kode dengan melihat keseluruhan teks atau hanya melihat satu teks elemen saja. Toolbar mempunyai tombol **Show Source of Selected Element Only**, yang membuat kita hanya melihat elemen terpilih saja dalam Java editor. Contoh di bawah ini, hanya method *setUp()* yang ditampilkan.

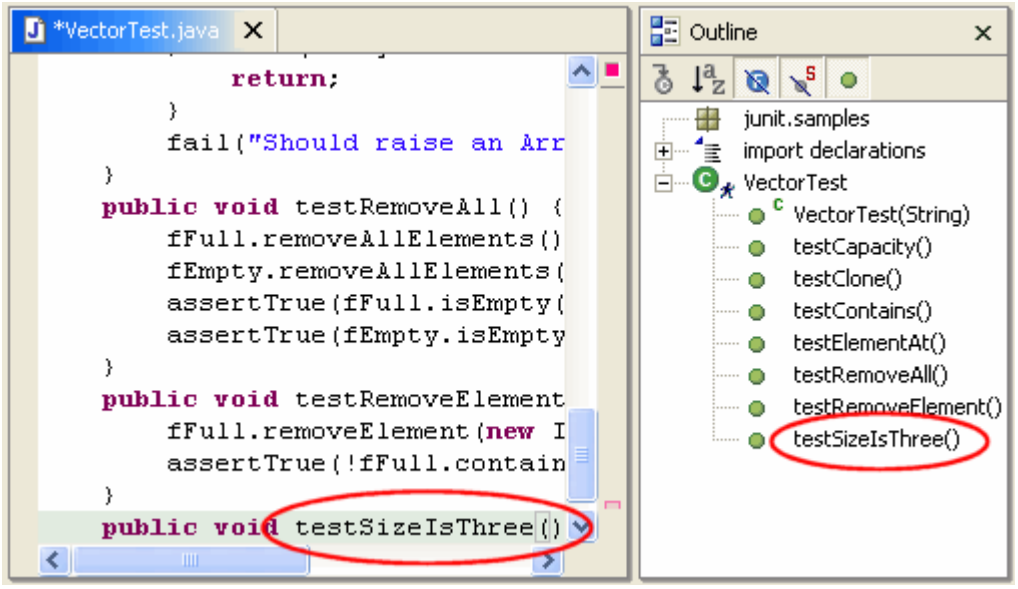


6. Tekan tombol **Show Source of Selected Element Only** lagi untuk melihat seluruh naskah lagi. Kemudian pilih sembarang elemen dalam Outline view, perhatikan pada sebelah kiri editor akan ada indikator yang menunjukkan rentang elemen dalam naskah tersebut.

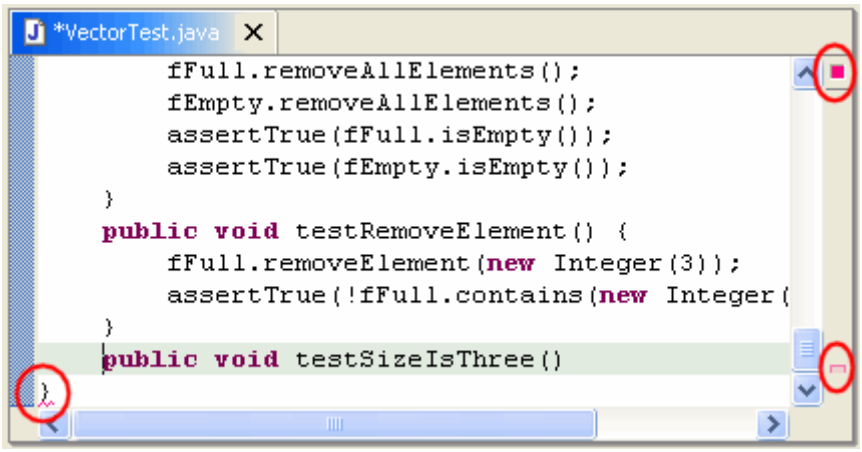


Menambahkan method baru

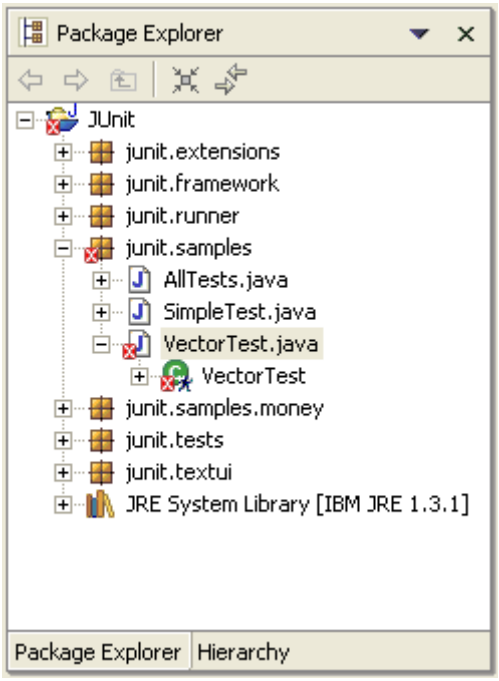
1. Mulai menambahkan sebuah method dengan mengetikkan diakhir file *VectorTest.java* (dan sebelum kurung penutup):
   public void testSizeIsThree()
   Begitu kita mengetikkan nama method, maka method yang baru tampak di Outline view.



Selain itu, sebuah pemberitahuan error (kotak merah) muncul di sisi kanan editor. Jika kita menggerakkan ujung mouse dia atas tanda itu, maka kita akan diberitahu *Unmatched bracket*; *Syntax error on token "}", "{" expected*, yang menunjukkan kesalahan kita. Perilaku ini dapat diatur melalui halaman preferensi **Java > Editor > Annotations**.



2. Klik tombol **Save**. Unit kompilasi dikompilasi secara otomatis dan tanda error juga muncul di Package Explorer view, dalam Tasks view dan di mana lagi coba cari... .



3. Lengkapi method yang baru dengan menambahkan:

```
{
 assertTrue(fFull.size() == 3);
```

   Perhatikan, kurung penutup telah otomatis dimasukkan.
4. Simpan file. Perhatikan indikator menghilang ketika pasangan kurung yang hilang sudah ditambahkan.

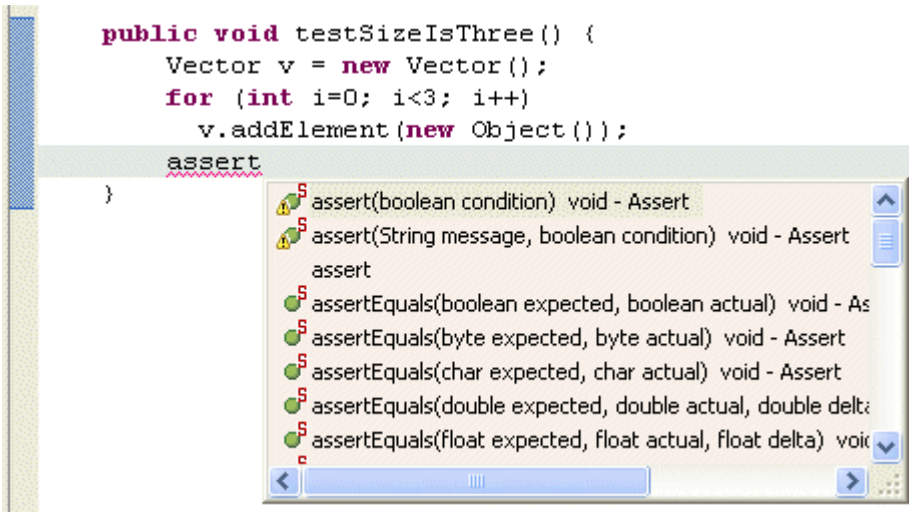
Menggunakan content assist

Pada bagian ini kita akan menggunakan *content assist* untuk menyelesaikan penulisan method baru. Buka file *junit.samples.VectorTest.java* file dan pilih method testSizeIsThree() dalam Outline view.

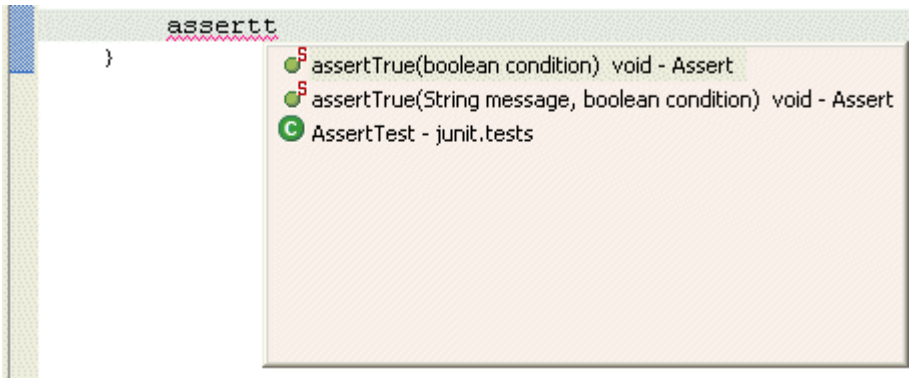
1. Tambahkan baris berikut pada akhir method:

   ```
   Vector v = new Vector();
   for (int i=0; i<3; i++)
     v.addElement(new Object());
   assert
   ```

2. Dengan cursor yang berada pada akhir kata assert, tekan Ctrl+Space untuk mengaktifkan content assist. Lihat pilihan-pilihan yang diajukan.



3. Sewaktu jendela content assist masih terbuka, ketik huruf 't' setelah assert (tanpa ada spasi). Daftar tadi dipersempit dan hanya menunjukkan baris yang dimulai dengan 'assertt'. Pilih dan tahan di atas pihan anda, Javadoc help akan muncul apabila tersedia.



4. Pilih assertTrue(boolean) dari daftar dan tekan Enter. Kode assertTrue(boolean) ditambahkan ke dalam method.

5. Complete the line so that it reads as follows:

   assertTrue(v.size() == fFull.size());

6. Simpan file.

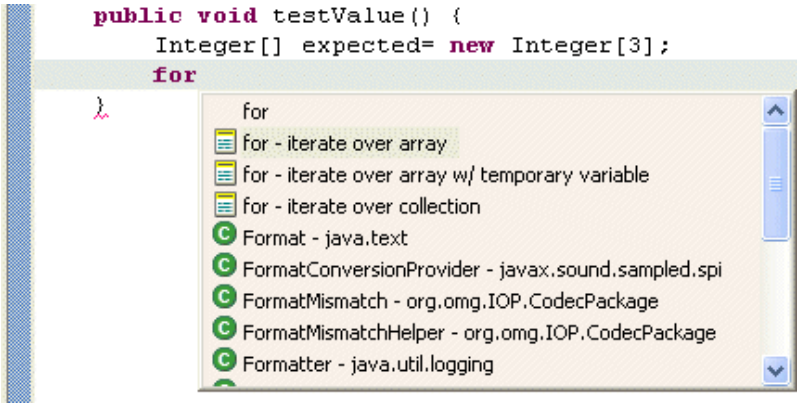
Menggunakan template

Sekarang kita akan belajar menggunakan content assist untuk membuat template dari sebuah struktur kalang. Buka file *junit.samples.VectorTest.java*

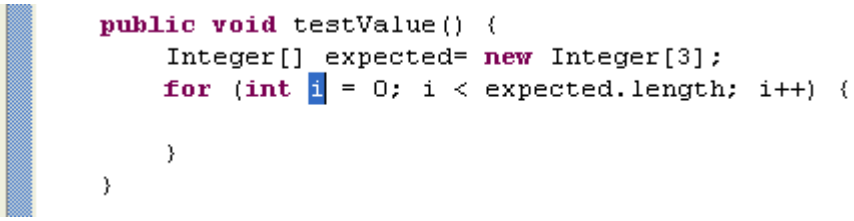
1. Mulai menambahkan method baru dengan mengetikkan:

   ```
   public void testValues() {
       Integer[] expected= new Integer[3];
       for
   ```

2. Sewaktu kursor berada di ujung for, tekan Ctrl+Space untuk mendapatkan content assist. Kita akan mendapatkan daftar pilihan dan jika kita menahannya untuk beberapa saat akan muncul template contohnya.



3. Kita pilih baris for - iterate over array dan tekan Enter untuk memasukkan ke source code.



4. Selanjutnya kita ganti nama index variable dari *i* ke *e*. Cukup tekan e, karena index variable otamatis terpilih. Amati bahwa nama dari index variable berubah di semua tempat.

```
public void testValue() {
    Integer[] expected= new Integer[3];
    for (int e = 0; e < expected.length; e++) {

    }
}
```

5. Menekan kunci tab memindahkan kursor ke variabel selanjutnya dari code template, yaitu array *expected*. Karena kita tidak ingin menggantinya, tekan tab lagi dan meninggalkan template yang sudah dibuat.

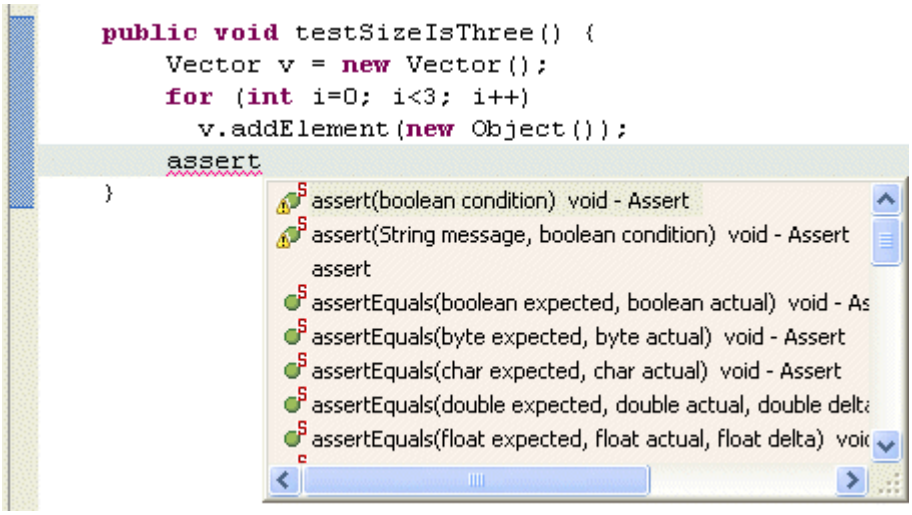
```
public void testValue() {
    Integer[] expected= new Integer[3];
    for (int e = 0; e < expected.length; e++) {

    }
}
```

6. Sempurnakan kalang for sebagai berikut:

```
for (int e= 0; e < expected.length; e++) {
    expected[e]= new Integer(e + 1);
}
Integer[] actual= to
```

7. Saat kursor di akhir to, tekan Ctrl+Space untuk mengaktifkan content assist. Pilih toarray - convert collection to array dan tekan Enter untuk memastikan pilihan atau klik ganda.



Template di masukkan ke dalam editor dan type berlatar biru dan terpilih.

```
public void testValue() {
    Integer[] expected= new Integer[3];
    for (int e = 0; e < expected.length; e++) {
        expected[e]= new Integer(e + 1);
    }
    Integer[] actual= (type[]) collection.toArray(new type[collection.size()]);
}
```

8. Timpa dengan mengetikkan Integer. Type dari array constructor berubah berbarengan.
9. Tekan Tab untuk pindah ke collection dan timpa dengan mengetikkan fFull.

```
public void testValue() {
    Integer[] expected= new Integer[3];
    for (int e = 0; e < expected.length; e++) {
        expected[e]= new Integer(e + 1);
    }
    Integer[] actual= (Integer[]) fFul.toArray(new Integer[fFul.size()]);
}
```

10. Tambah baris-baris berikut untuk menyelesaikan methodnya:

```
assertEquals(expected.length, actual.length);
for (int i= 0; i < actual.length; i++)
 assertEquals(expected[i], actual[i]);
}
```

11. Simpan file tersebut.