

PART 1

Eclipse IDE Eğitimi

Bu eğitim Eclipse'in Java IDE olarak kullanımını anlatır. Eğitim; Eclipse'in kurulumunu, Java programların oluşturulmasını ve Eclipse'in kullanım kolaylıklarını anlatır. Bu eğitim Eclipse 4.2 (juno) üzerinedir.

1. Eclipse Nedir?

Bir çok insan Eclipse'i Java için entegre geliştirme ortamı (IDE) olarak bilir. Günümüzde %65 pazar payıyla öncü geliştirme ortamıdır.

Eclipse projeleri Eclipse Vakfı tarafından yönetilmektedir. Eclipse Vakfı kar amacı gütmeyen, üyeleri tarafından desteklenen, Eclipse açık kaynak kodlarını sağlayan ve hem açık kaynak topluluğunu ilerleten hem de tamamlayıcı ürünler ve servislerini içinde barındıran bir kuruluştur.

Eclipse IDE ek yazılım bileşenleriyle genişletilebilir. Eclipse bu yazılım bileşenlerine *plug-in* adını verir. Bazı açık kaynaklı projeler ve şirketler Eclipse'i genişletmiştir.

Eclipse ayrıca genel amaçlar için oluşturulacak uygulamalar için taban olarak da kullanılabilir. Bu uygulamalar *Eclipse Zengin İstemci Platformu*(*Eclipse Rich Client Platform*(*Eclipse RCP*)) uygulamaları olarak da bilinir.

2. Genel Kullanıma Açık Eclipse Lisansı

Genel kullanıma açık Eclipse lisansı (Eclipse Public Licence(EPL)); Eclipse Vakfı tarafından kendi yazılımları için kullandığı açık kaynaklı lisans yazılımıdır. EPL lisanslı programlar kullanılabilir, değiştirilebilir, kopyalanabilir, ücretsiz dağıtılabilir ve EPL yazılım lisansına sahip olanlar bunu kapalı kaynak kodlu programlarda kullanabilirler. Bu konuda sadece değişiklik yapanlar EPL kodunu yayınlamak zorundadır.

3. Eclipse'in Yklenmesi

3.1. Eclipse'in Java Gereksinimleri

Eclipse iin Java Runtime'ın ykl olması gerekmektedir.Eclipse 4.2 'nin alıřması iin en dřk versiyon olarak Java 5 gerekmektedir. Eclipse IDE kendi iinde Java derleyicisini bulundurur. Eclipse dıřından bir Java kaynak kodu alıřtırılması halinde veya st seviye geliřtirme senaryolarında; rneėin web programlamaları iin; otomatik derleme kullanılacaksa *Java Development Tools* gereklidir.

3.2 Java'nın Yklenmesi

Java sizin bilgisayarınızda hali hazırda ykl olabilir. Bunu konsol ekranını aıp (eėer Windows kullanıyorsanız: Win+R, komut satırına *cmd* yazın ve *Enter* tuřuna basın) ařaėıdaki komutu yazarak ėrenebilirsiniz.

```
java -version
```

Eėer Java dzgn olarak yklenmiřse, Java'nın yklenme bilgilerine gz atmalısınız. Eėer komut satırı "program bulunamadı" (program could not be found) řeklinde bir bilgi ile dnyorsa, Java'yı yklemelisiniz. Java'yı indirebileėeėiniz baėlantı ařaėıdadır."

<http://java.com>

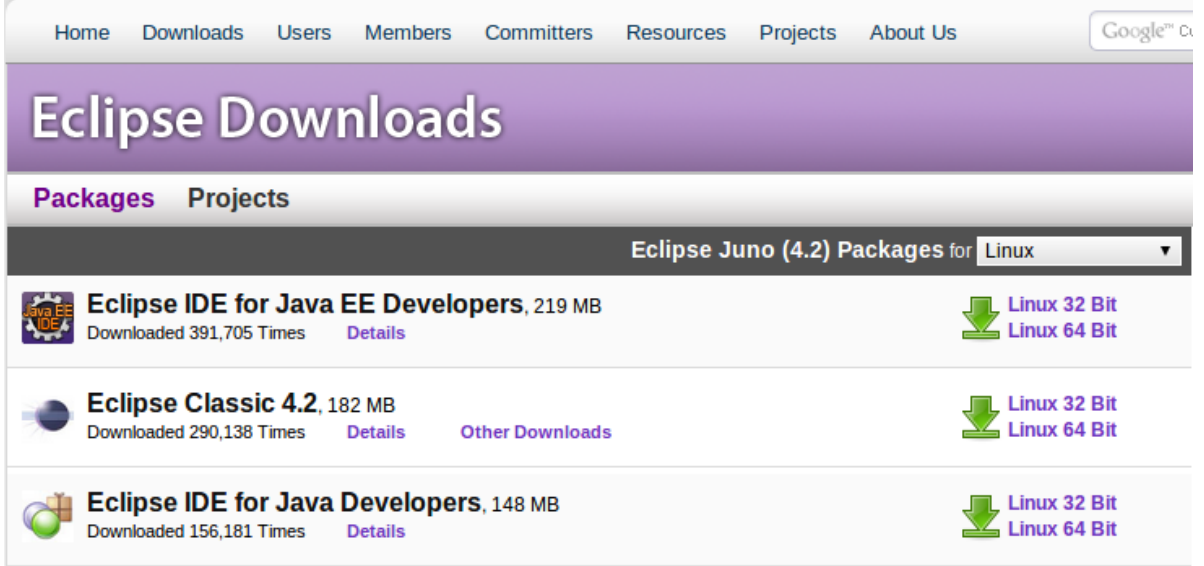
3.3. Eclipse'in İndirilmesi

Eclipse IDE bileřenlerden oluřmaktadır ve bu bileřenleri yapılandırmak mmkndr. Eclipse.org tipik kullanımlar iin nceden paketlenmiř Eclipse'i bulundurmaktadır. *Eclipse IDE for Java Developers* seeneėi standart Java kodlarını geliřtirmek iin tasarlanmıřtır.

Ařaėıdaki baėlantıdan **Eclipse IDE for Java Developers** paketini indirin. Bilgisayarınız iin doėru versiyonu indirdiėinizden emin olun(32 bit veya 64 bit).

<http://www.eclipse.org/downloads>

Aşağıdaki ekran görüntüsü Eclipse'in Linux sistemler için olanıdır. Paketin yanındaki linki seçerek indirme işlemine başlayabilirsiniz.Örneğin; Linux 64 Bit



3.4. Eclipse'in Yüklenmesi

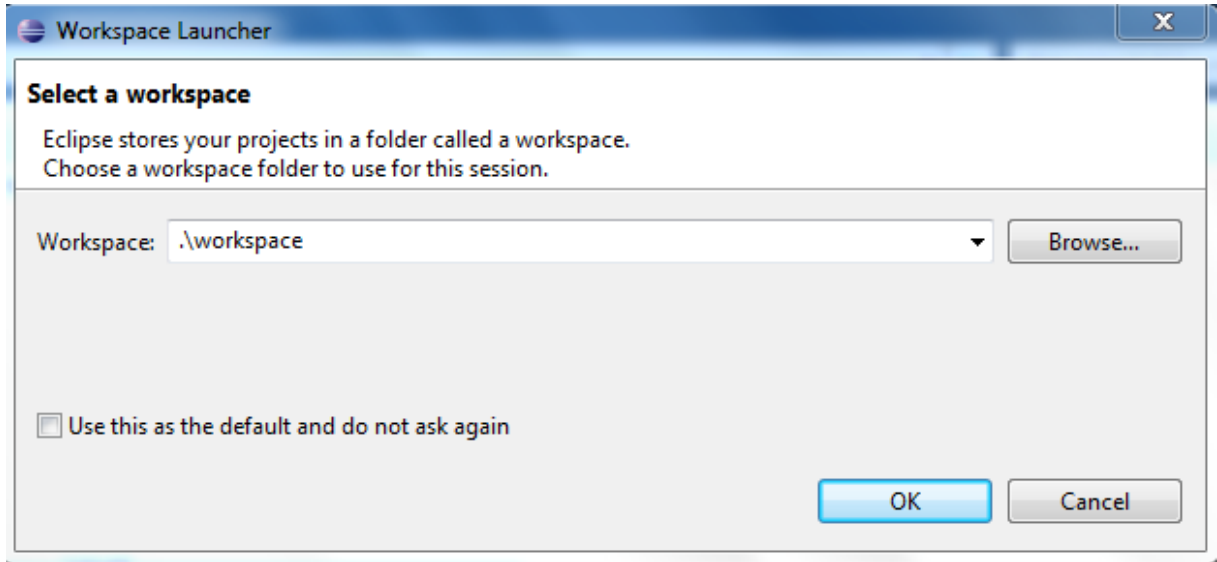
İndirme tamamlandıktan sonra zip dosyasını yerel dizinlerde açın. Birçok işletim sistemi kendi dosya sisteminde zip dosyalarını açabilirler, örnek olarak Windows 7'de farenin sağ tuşuna tıklayarak "Hepsini çıkart(Extract all)" seçeneği seçilmelidir.

İçinde boşluk karakterini bulundurmeyen bir dizin seçin, çünkü Eclipse bu konuda bazen problem yaşayabiliyor. Zip dosyasını açtıktan sonra Eclipse; hiç bir yükleme işlemine gerek olmadan çalışmaya hazırdır.

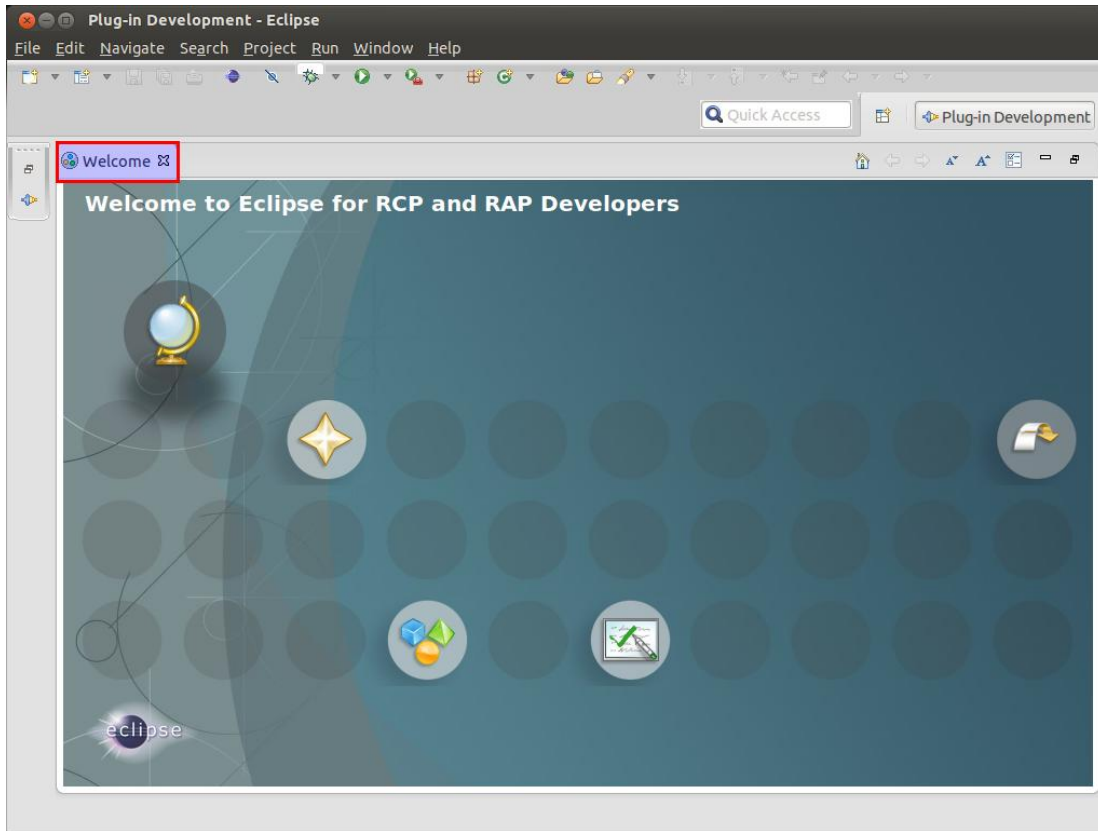
4.Başlarken

4.1. Eclipse'in Başlatılması

Eclipse'i çalıştırmak için indirdiğiniz dizinin içinde, Windows'ta *eclipse.exe* Linux ve Mac OS'de ise *eclipse* dosyasını çift tıklayın. Sistem sizin karşınıza *çalışma ortamını(workspace)* çıkartacaktır. Çalışma ortamı sizin geliştirme yapacağınız yerdir. Boş bir dizini seçip OK butonunu seçin.



Eclipse başlayacak ve size “Welcome” sayfasını açacak. Bu sayfayı kapatın. Burayı “Welcome” yazısının yanındaki çarpı işaretini (X) seçerek kapatın.

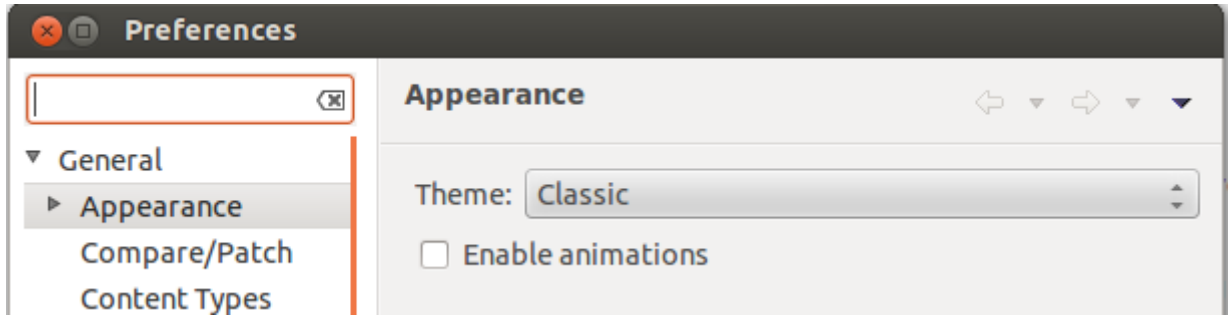


4.2. Görünüm

Eclipse'in görünümü değiştirilebilir. Eclipse kendi içinde bir kaç adet tema barındırır ancak bunları arttırabilirsiniz. Bu eğitimin eki en popüler temaları listelemektedir.

Window → Preferences → General → Appearance.

Tema seçimi Eclipse'in görünümünü değiştirmenize olanak sağlar. Animasyonları devre dışı bırakmak Eclipse'in daha hızlı çalışmasını sağlayacaktır.



Eclipse'in yeni görünümünün çalışması için; yeniden başlatılması gerekir.

5. Eclipse'in Önemli Terimleri

Eclipse; *Perspective*, *View* ve *Editor*'ü içinde bulundurur. *View* ve *Editor*; *Perspective*'in altında gruplanmıştır.

5.1. Workspace

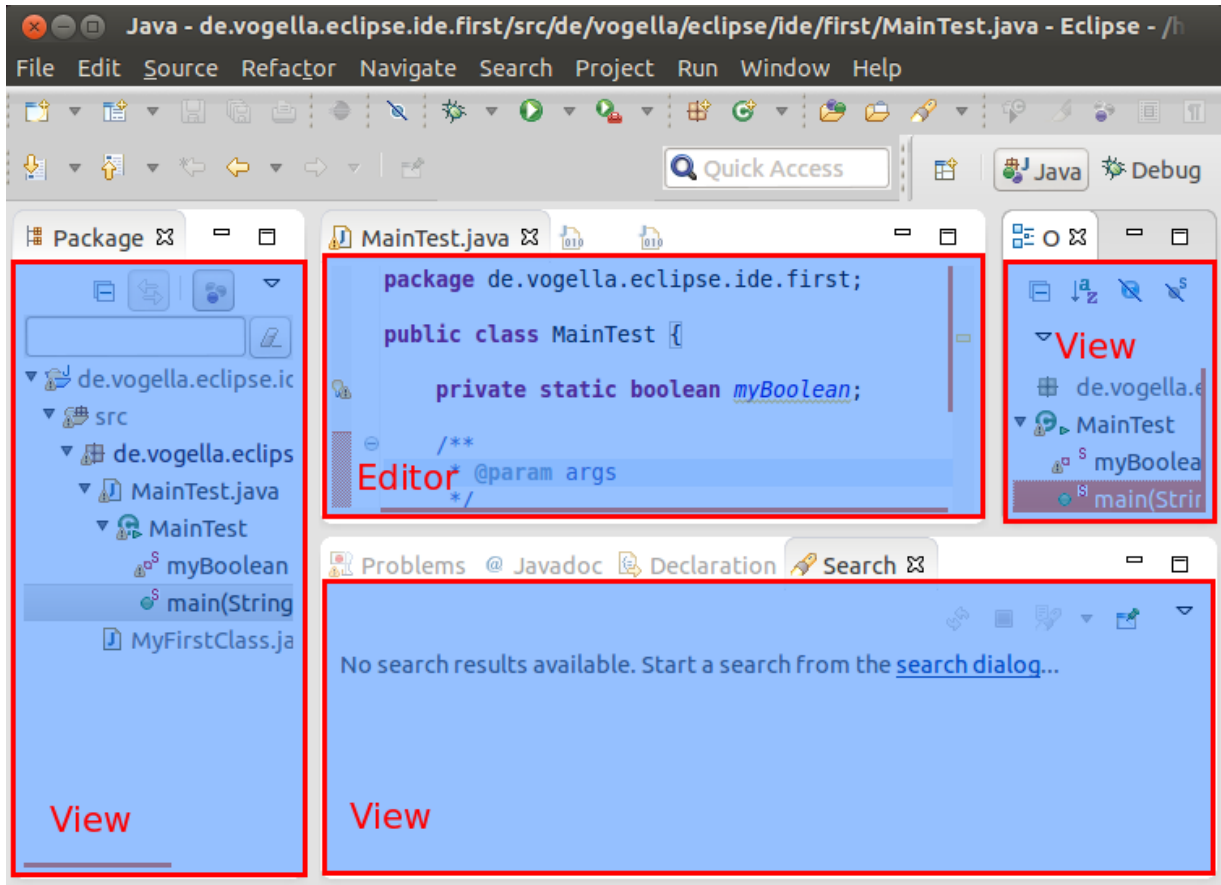
Workspace; proje dosyalarının hard diskte tutulduğu konumdur. Projeleriniz, resimleriniz ve diğer yaptıklarınız çalışma ortamında tutulur ve saklanır.

Çalışma ortamınızı Eclipse başlarken veya menüden seçebilirsiniz.

(File → Change Workspace → Other)

5.2. View ve Editor Bölümleri

Part'lar sizin veriyi yönlendirdiğiniz ve değiştirdiğiniz kullanıcı arayüzleridir.

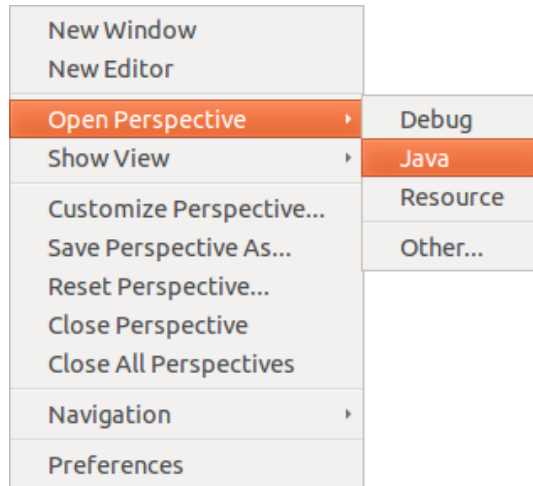


View; veri üstünde çalışmak için kullanılır. Bu veriler hiyerarşik durumda bulunabilir. Eğer veri view üzerinden değiştirildiyse, bu değişiklik direkt olarak ilgili veri yapılarına uygulanır. View'a bir örnek Java Packet Explorer verilebilir. Java Packet Explorer, Eclipse projelerinizin dosyalarını incelemenize olanak sağlar. Eğer Java Packet Explorer içindeki bir veriyi değiştirirseniz; örnek olarak ismini değiştirmek; o dosyanın ismi direkt olarak dosya sisteminde de değişir.

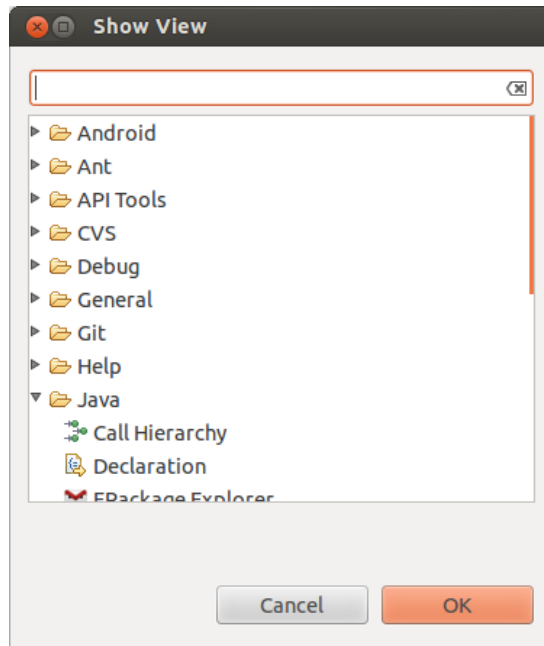
5.3. Perspective

Perspective; içinde Part gruplarının bulunduğu görsel bir taşıyıcıdır. Eclipse Part'ları, menüyü ve farklı geliştirme görevlerini düzenlemek için Perspective kullanır. Açık Editorler, Perspective'ler arasında paylaşılır. Örneğin; Java Perspective'inde spesifik bir sınıf için editör açıkken Debug Perspective'ine geçildiğinde, Editor açık bekler. Perspective'ler arasında geçiş yapmak için;

Window → Open Perspective → Other



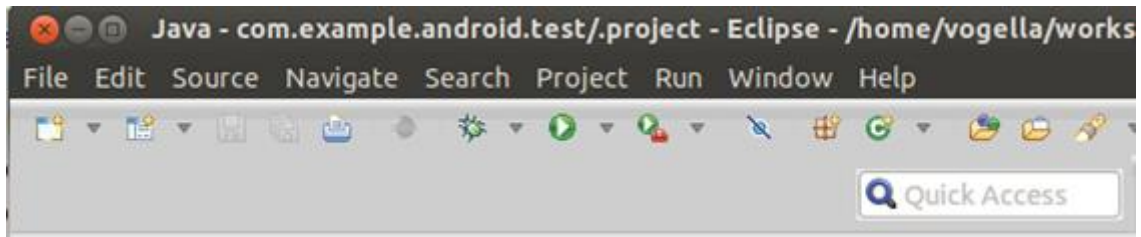
Düzeni ve Perspective içindeki içeriği Part'ları açıp kapatarak veya yeniden düzenleyerek değiştirebilirsiniz.



6. Eclipse Java Perspective

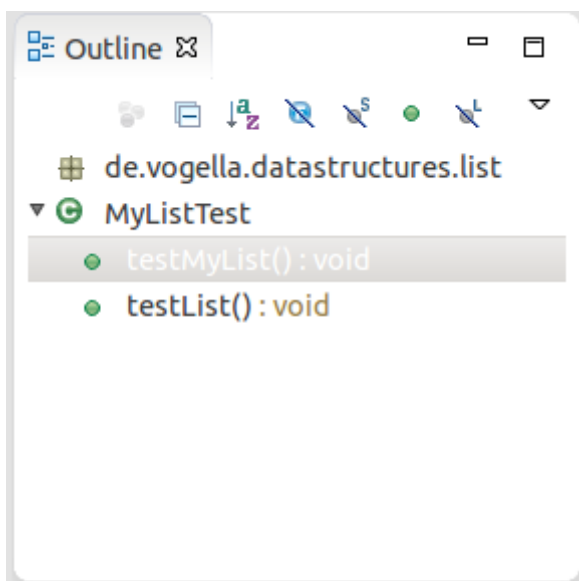
6.1. Toolbar

Uygulama araç çubuğu genel olarak yapılan eylemleri bulundurur. Buna örnek olarak; Java kaynağı yaratma veya çalışan Java uygulamaları verilebilir. Ayrıca açık perspective'ler arasında geçiş yapma olanağı da sunar.



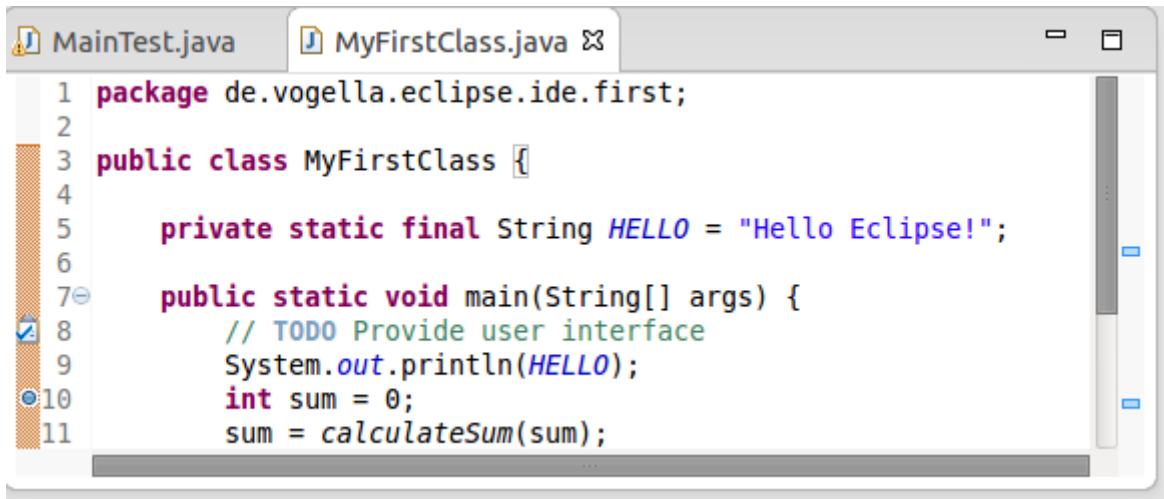
6.2. Outline view

Outline; seçilmiş Java kaynak dosyalarının yapısını gösterir



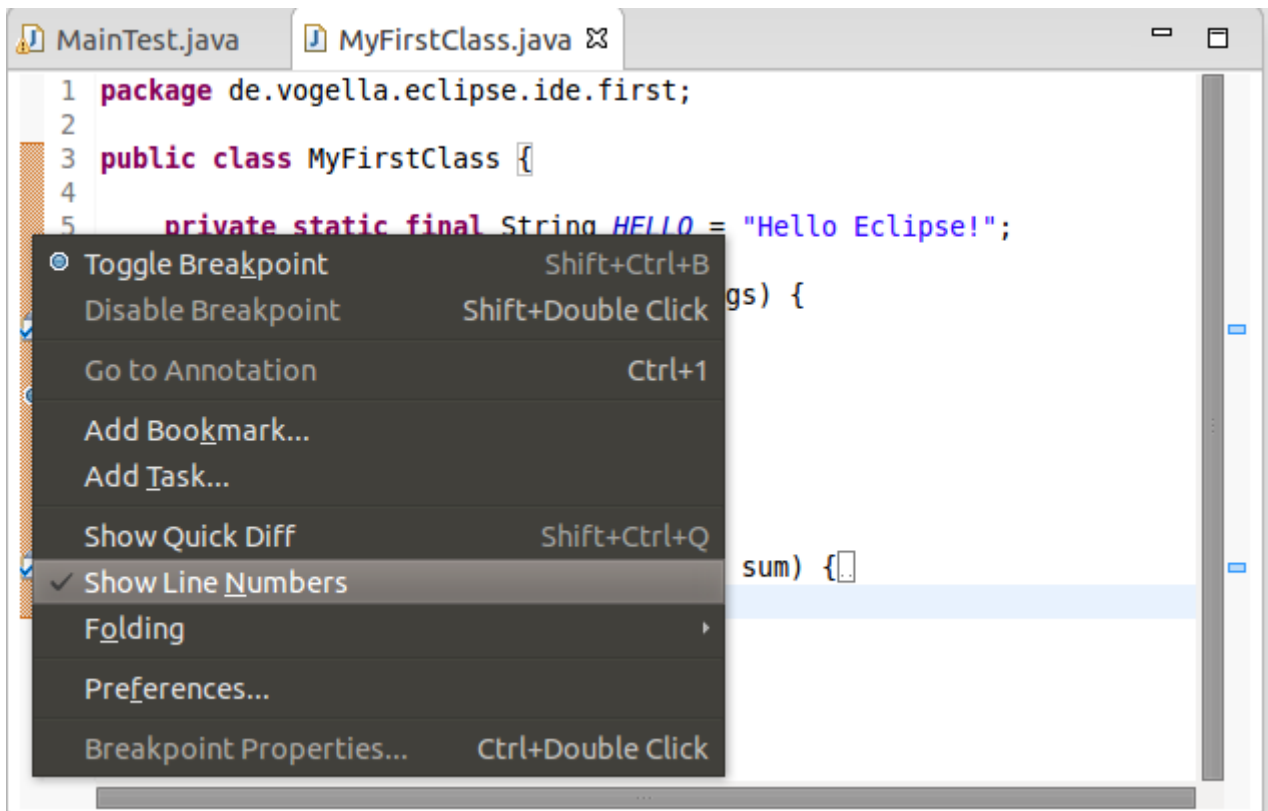
6.3. Java editor

Java Editörü; Java kaynak dosyalarını değiştirmek için kullanılır. Her farklı kaynak dosyası ayrı editörde açılır.



```
1 package de.vogella.eclipse.ide.first;
2
3 public class MyFirstClass {
4
5     private static final String HELLO = "Hello Eclipse!";
6
7     public static void main(String[] args) {
8         // TODO Provide user interface
9         System.out.println(HELLO);
10        int sum = 0;
11        sum = calculateSum(sum);
12    }
13 }
```

Eğer editörün sol kolonu seçilirse, özellikleri ayarlanabilir; örneğin satır sayılarının gösterilmesi gibi.



```
1 package de.vogella.eclipse.ide.first;
2
3 public class MyFirstClass {
4
5     private static final String HELLO = "Hello Eclipse!";
6
7     public static void main(String[] args) {
8         // TODO Provide user interface
9         System.out.println(HELLO);
10        int sum = 0;
11        sum = calculateSum(sum);
12    }
13 }
```

7. İlk Java Programının Oluşturulması

Burada Eclipse kullanılarak küçük bir Java programının nasıl oluşturulduğu anlatılacaktır. Programlama dünyasının bir geleneği olarak, ilk programda konsol ekranına “Merhaba Dünya” yazdırılır. Biz bu geleneği kendimize uyarlayarak “Merhaba Eclipse” yazdıracağız.

7.1. Projeyi Oluřturma

Bu kısım isimlendirmeyi anlatmaktadır. Proje ismi paket ismiyle aynı olmalıdır. Menüden seçerek ;

File → New → Java project.

Proje ismi olarak `de.vogella.eclipse.ide.first` ismini seçin. “*Create separate folders for sources and class*” seçeneğini işaretleyin.



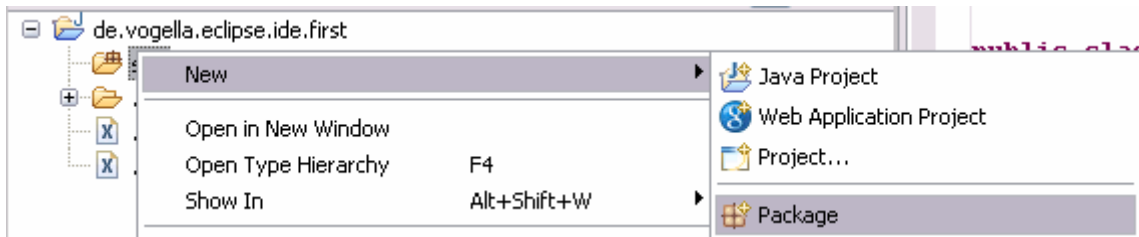
Finish butonunu seçerek projeyi oluřturun. Yeni proje bir klasör olarak oluřturulacak ve o řekilde görüntülenecektir. `de.vogella.eclipse.ide.first` klasörünü açarak içeriğini inceleyin.

7.2. Paket Oluřturma

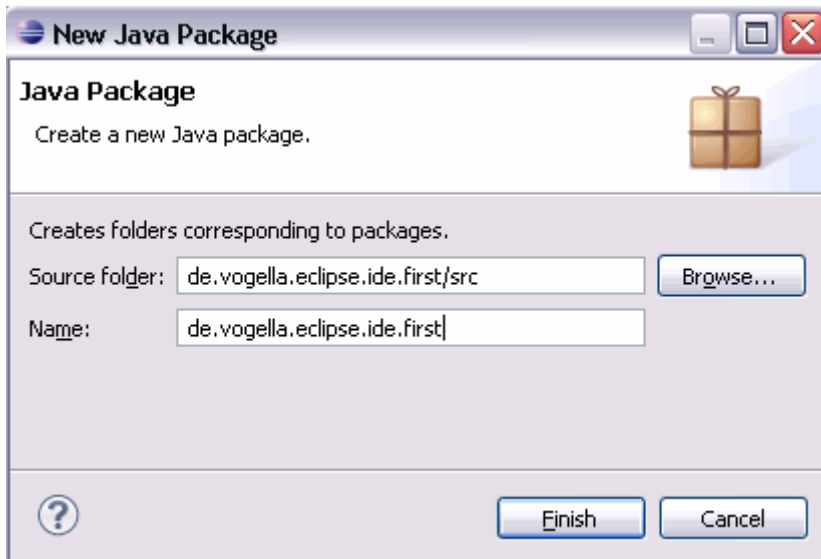
Bir sonraki ařamada yeni bir *paket* oluřturacaksınız. İsimlendirmenin en doęru yöntemi; proje ve bir üst seviyedeki paketin isimleriyle yaratacaęınız paketlerin isimlerinin aynı olmasıdır.

`de.vogella.eclipse.ide.first` paketini oluřturmak için, `src` klasörünün üzerinde saę tuřu seçin.

`src` → *New* → *Package*



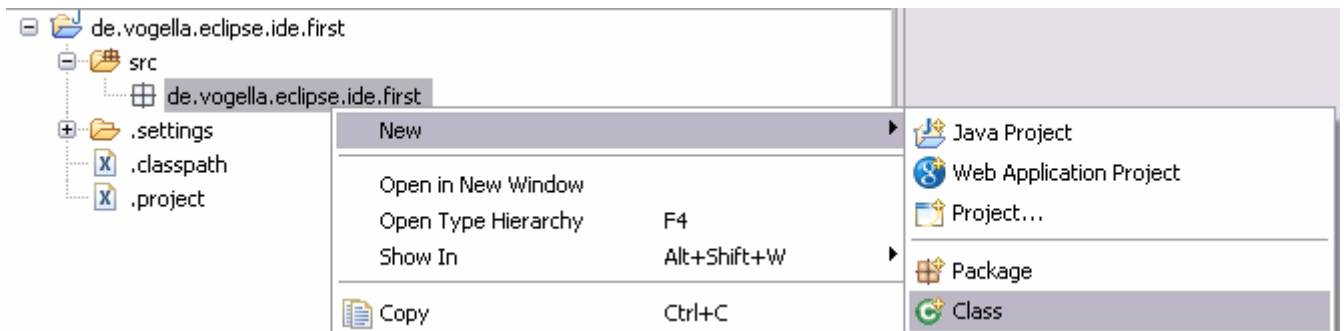
Açılan pencerede paketin ismini girip işlemi tamamlayın.



7.3. Java Sınıfı Oluřturma

Java sınıfı oluřturmak için bulunduęunuz pakette saę tuřu seçin;

New → *Class*.



İsim olarak **MyFirstClass** verin ve **public static void main (String[] args)** seçeneğini işaretleyin



Finish butonunu seçin. Bu işlem yeni dosyayı oluşturur ve Java kaynak kodları için Editorü açar. Oluşturduğunuz sınıfı aşağıdaki gibi değiştirin.

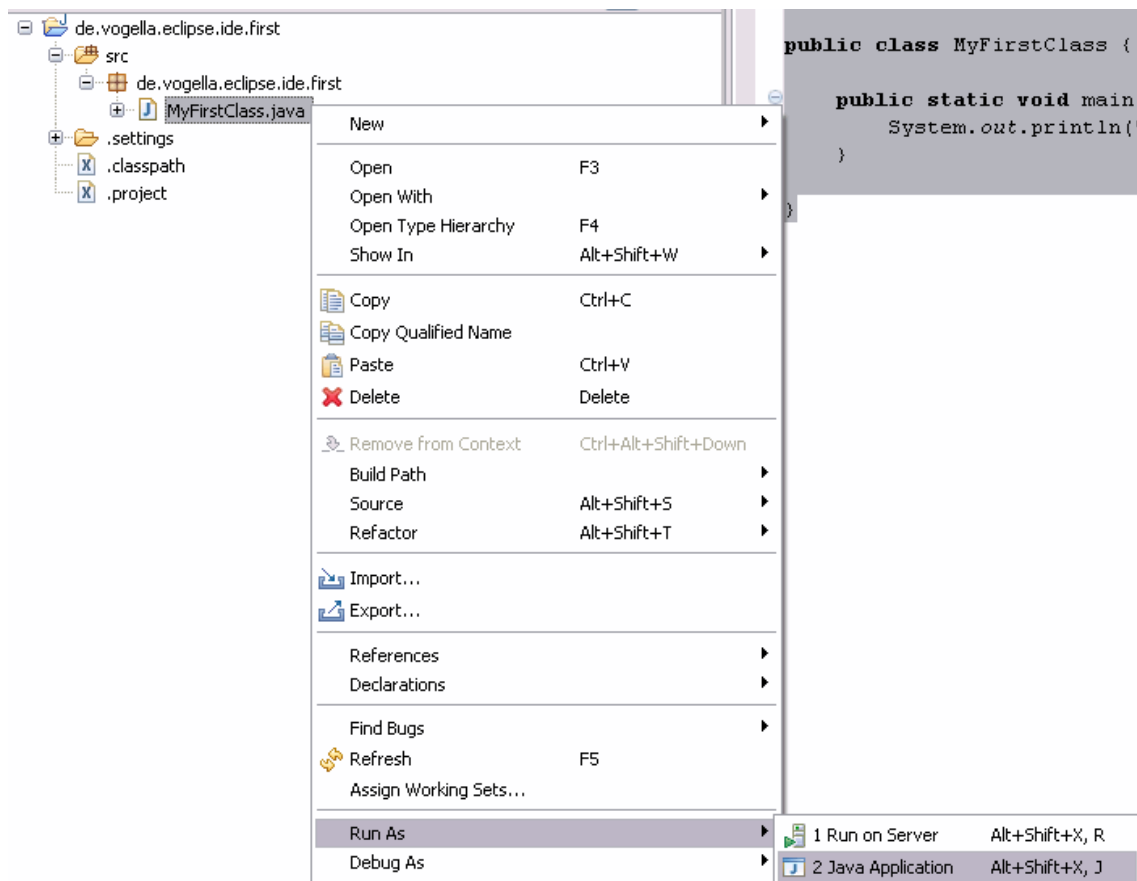
```
package de.vogella.eclipse.ide.first;  
public class MyFirstClass {
```

```
public static void main(String[] args) {  
    System.out.println("Hello Eclipse!");  
}  
  
}
```

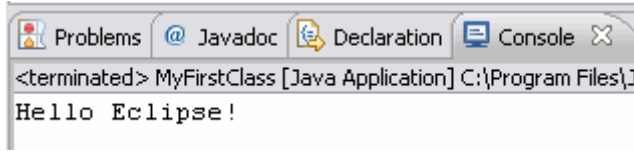
Ayrıca bu pencereyi kullanarak yeni paketler oluşturabilirsiniz. Eğer bu pencereye yeni paketler girerseniz otomatik olarak oluşturulacaktır.

7.4. Projenizi Eclipse’de Çalıştırın

Şimdi kodunuzu çalıştırın. İster Java Packet Explorer üzerinde sağ tuşu seçerek, isterseniz Java sınıfında sağ tuşu seçerek yapabilirsiniz; *Run-as → Java application*.



Eclipse Java programınızı konsol görünümünde çalıştıracaktır.

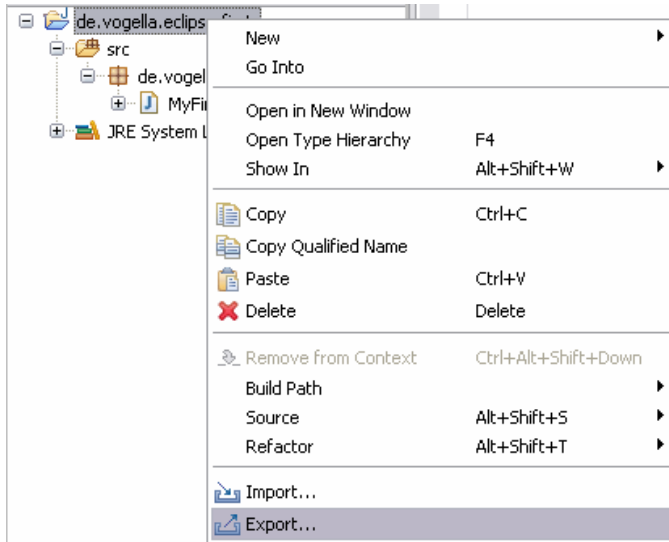


Tebrikler. İlk Java projenizi, paketinizi ve Java sınıfınızı Eclipse üzerinde çalıştırdınız.

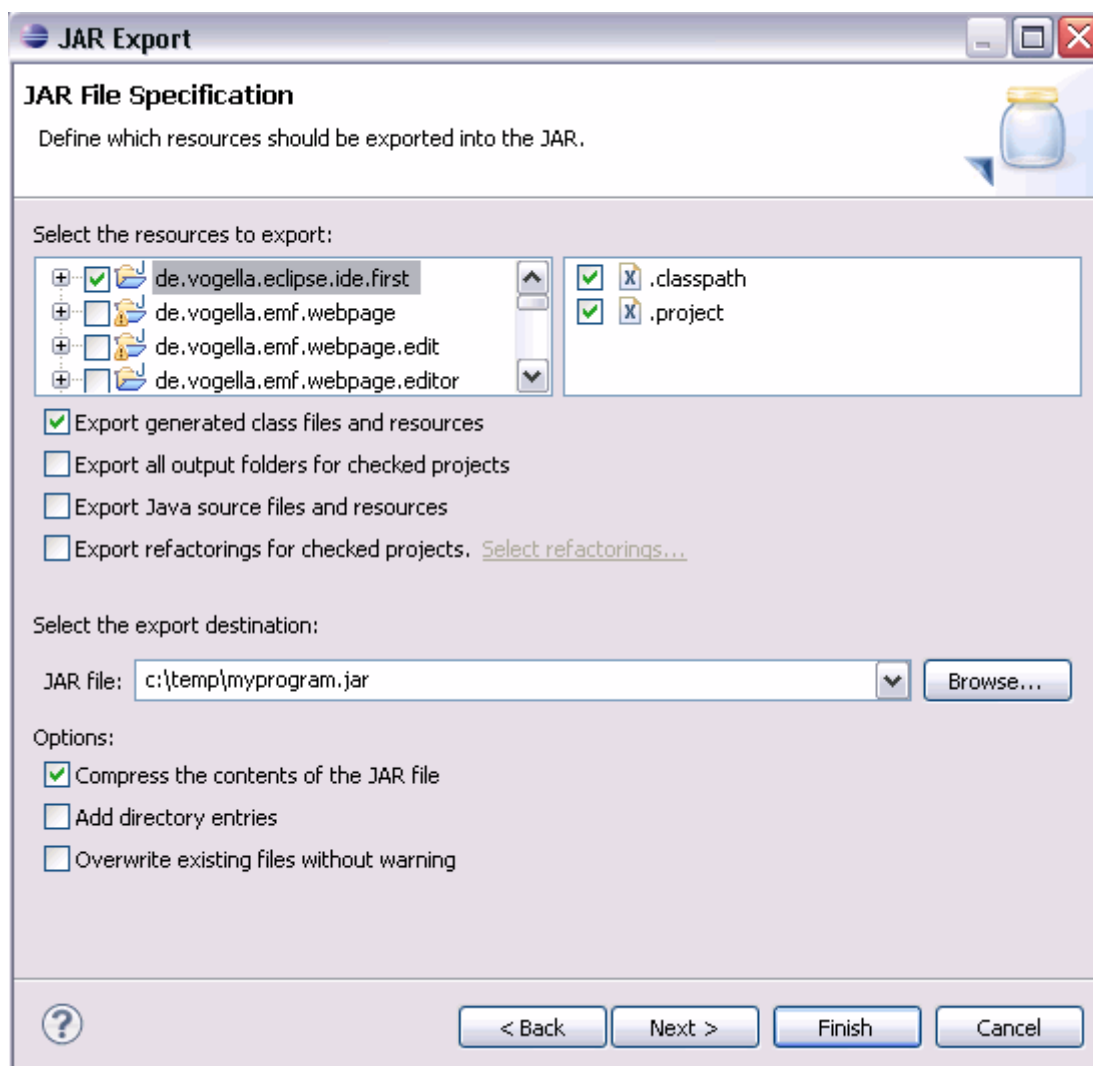
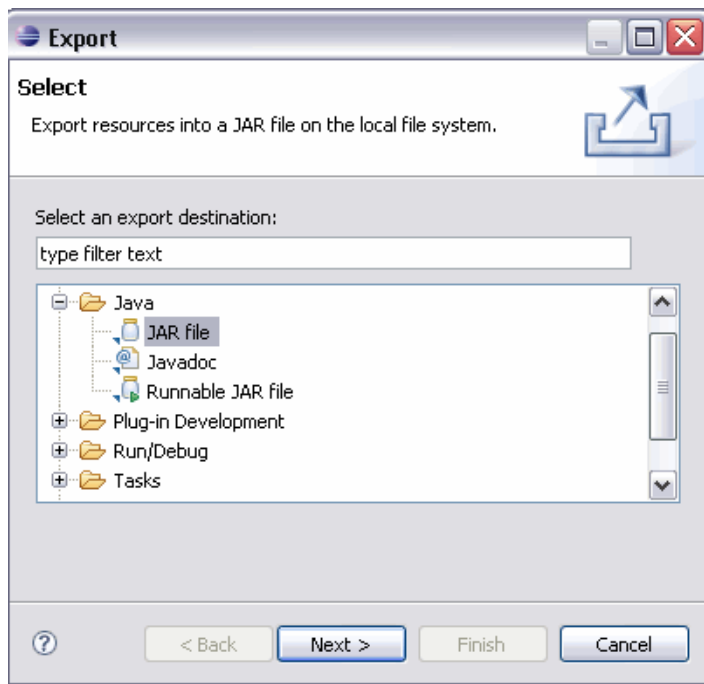
8. Java Programlarını Eclipse Dışında Çalıştırmak

8.1. Jar Dosyası Oluşturmak

Java programınızı Eclipse dışında çalıştırmak için dosyayı .jar şeklinde dışarı aktarmalısınız. .Jar dosyası Java uygulamaları için standart formattır.



Jar dosyasını seçin. Next butonunu seçin. Projenizi seçin, oluşturulacak dosya konumunu ve ismini belirtin. Burada `myprogram.jar` şeklinde isimlendirildi.



Finish butonunu seçin. Bu işlem seçtiğiniz klasöre .jar dosyasını oluşturacaktır.

8.2. Programın Eclipse Dışında Çalıştırılması

Komut satırını açın.

cd "dosya yolu"

komutuyla jar dosyasını oluşturduğunuz konuma gelin. Aşağıdaki komutla programınızı çalıştırın. "Merhaba Eclipse" mesajını komut satırında göreceksiniz.

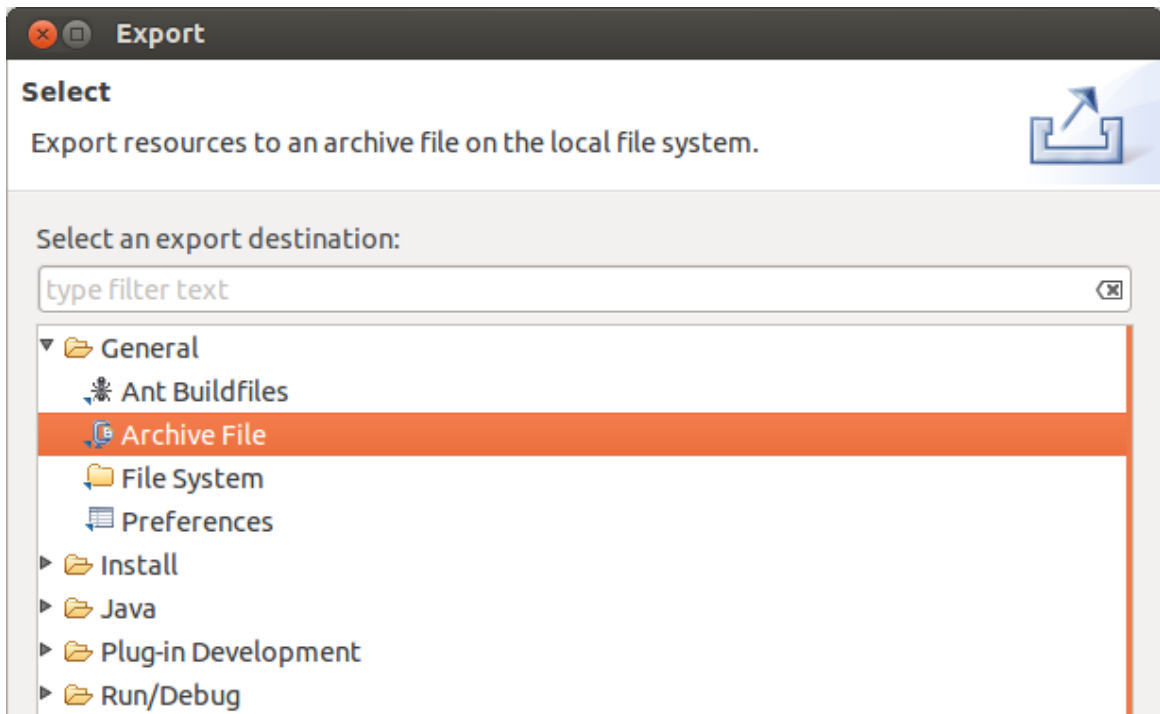
```
java -jar myprogram.jar
```

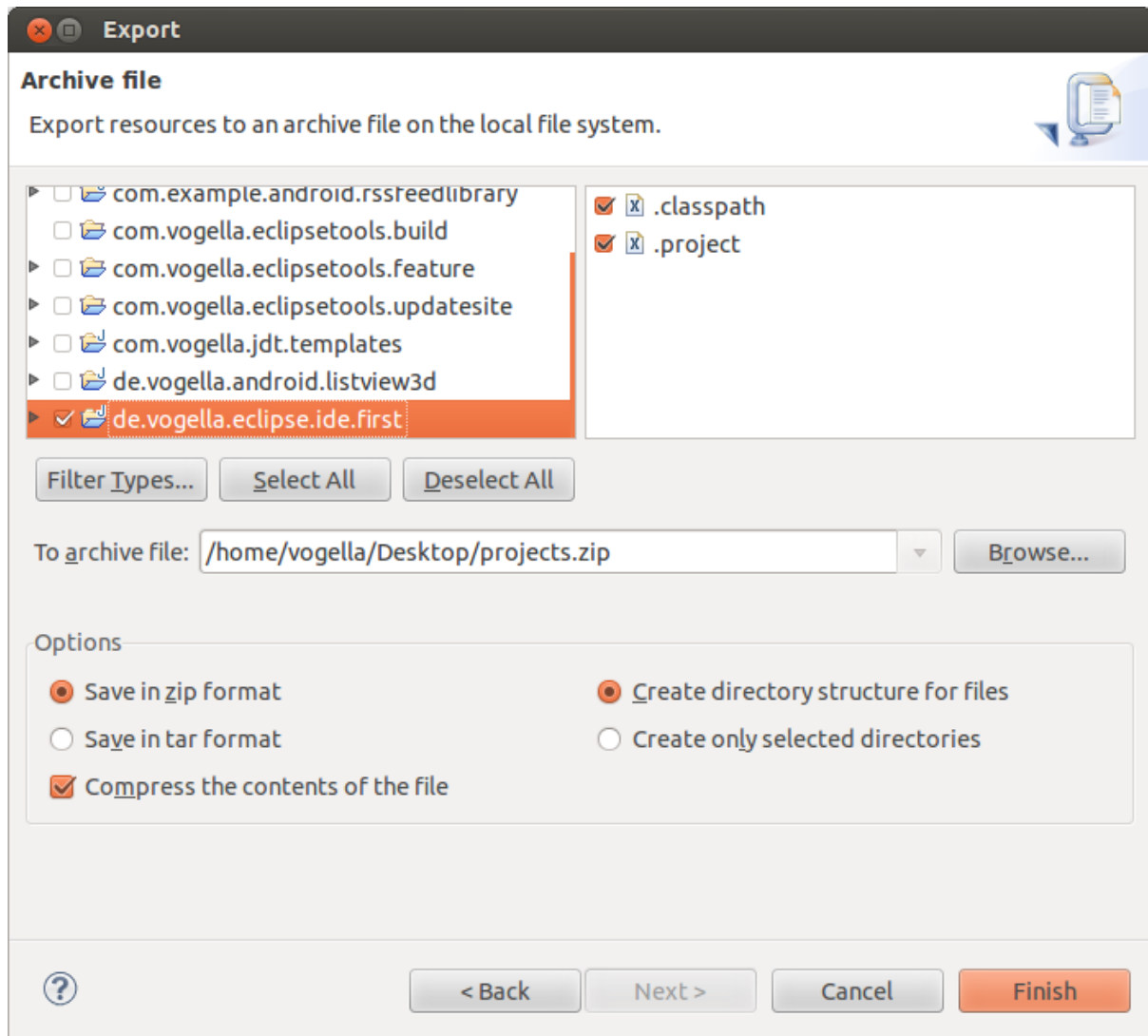
9. Projeleri İçe ve Dışa Aktarmak.

9.1. Projeleri Dışa Aktarmak

Eclipse'de projelerinizi içe ve dışa aktarabilirsiniz. Bu sayede projelerinizi başkalarıyla paylaşabilir ve hali hazırda bulunan projeleri kullanabilirsiniz.

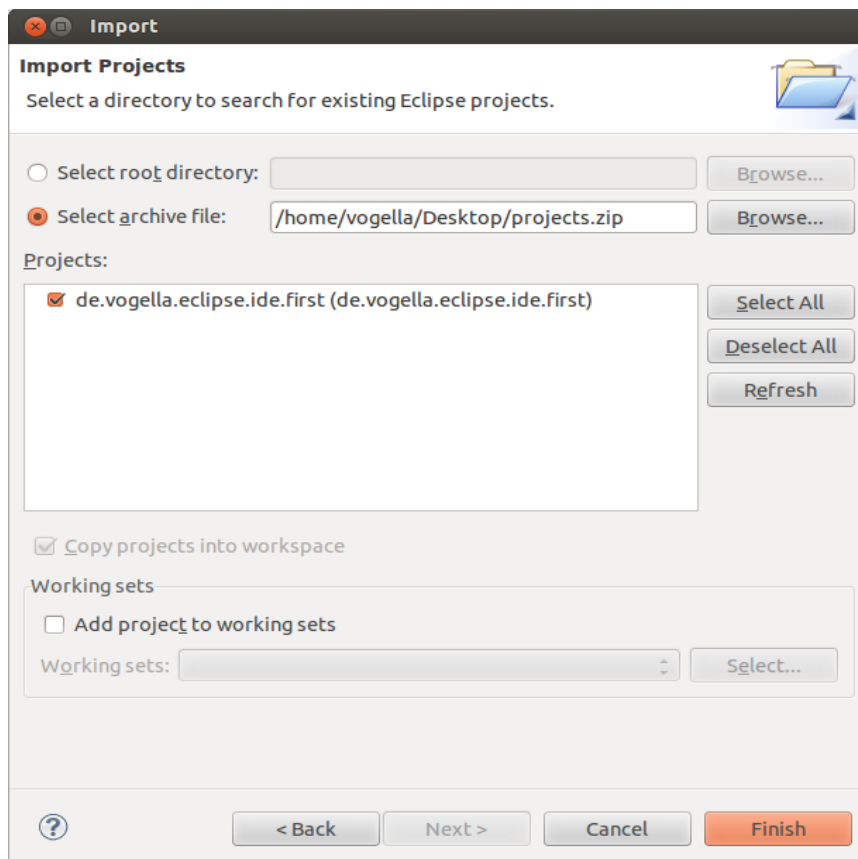
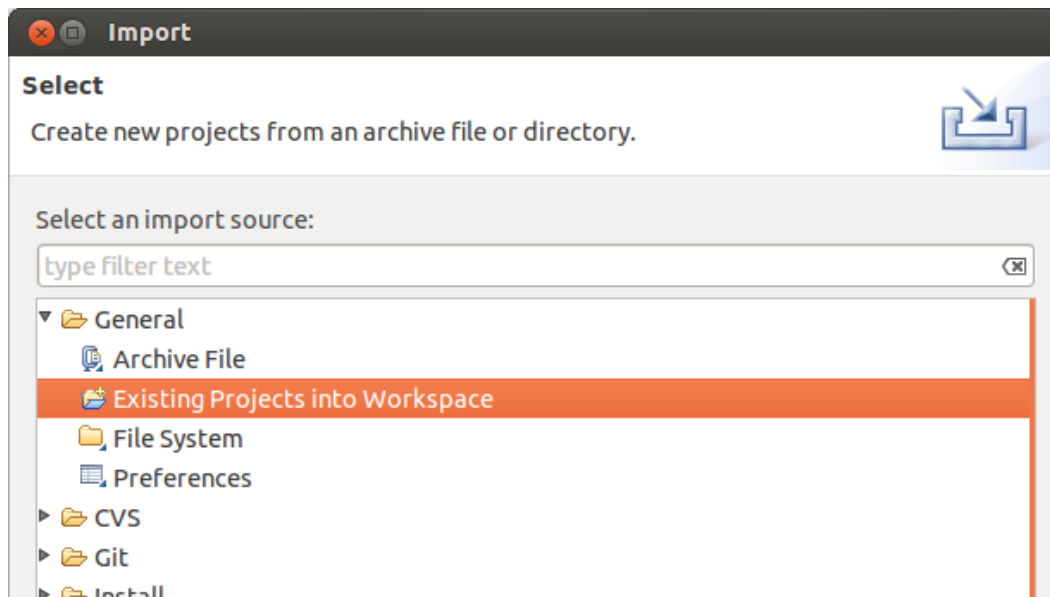
Eclipse'de projelerinizi dışa aktarmak için **File → Export → General → Archive File** yolunu izleyin ve sonunda dışa aktarmak istediğiniz projeyi seçin. Bu seçenek size projenizi .zip şeklinde aktarır.





9.2.Projeleri İçe Aktarmak

Projeyi içe aktarmak için File → Import → Existing Projects into Workspace yolunu izleyin. Projenizi bir arşiv dosyasından örneğin dışa aktardığınız bir .zip dosyasından içe aktarabilirsiniz.

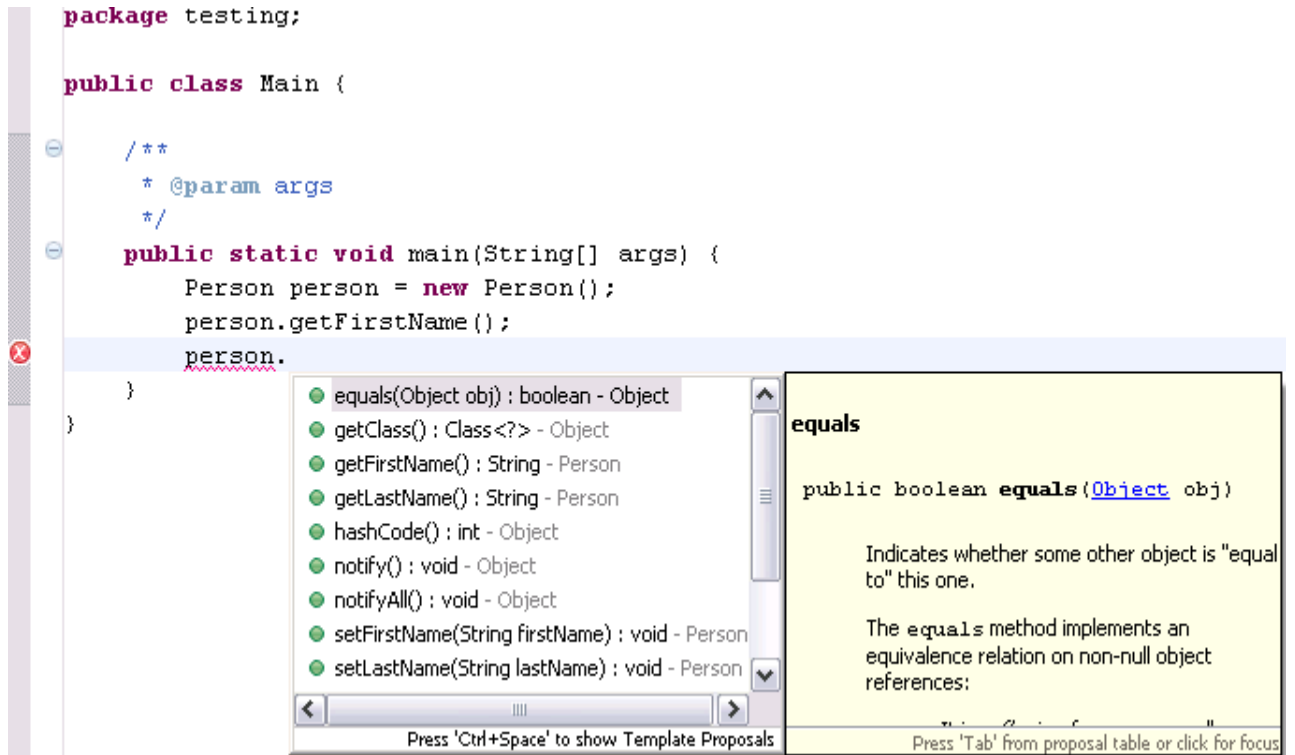


10. İçerik Yardımı ve Hızlı Düzeltme

10.1. İçerik yardımı

İçerik yardımı; geliştiriciye, içerik duyarlı kod tamamlama sağlayan bir Eclipse özelliğidir.

Ctrl+Space butonu seçilerek çalıştırılabilir.

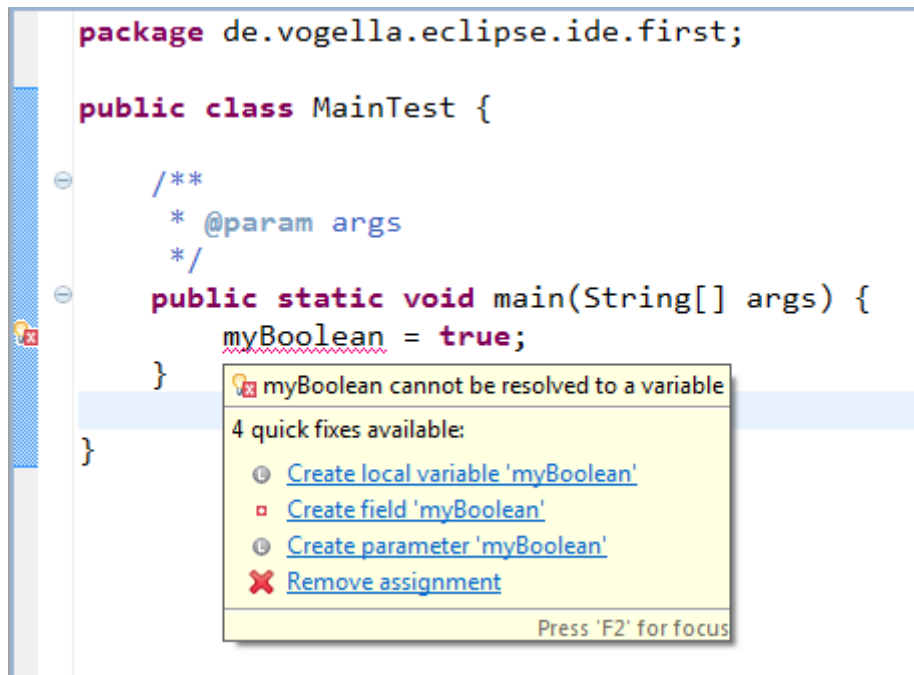


Örneğin, Java editörü içindeki kaynak dosyasına `sys.o` yazıp ardından Ctrl+Space butonlarını seçin. Bu eylem `sys.o` 'nun yerine `System.out.println("")` komutunu koyacaktır.

Bir nesneye referans yaptığınızda, örneğin `Person` tipinde tanımlanmış `person` nesnesi olsun ve bu nesnenin metotları görüntülenmek istensin; `person.` yazın ve ardından Ctrl+Space butonlarını seçin.

10.2. Hızlı Düzeltme

Eclipse bir sorun algıladığında editörün içindeki sorunlu metnin altını çizerek altı çizili metni seçin ve sorunun çözümü hakkındaki önerileri görmek için Ctrl+1 butonlarını kullanın. Bu özelliğe *Quick Fix* adı verilir. Örneğin, `myBoolean = true` yazın; Eğer `myBoolean` henüz tanımlanmamışsa, Eclipse bunu hata olarak belirtir. Değişkeni seçin ve Ctrl+1 butonlarını seçin, Eclipse yeni bir alan(field) veya yerel bir değişken oluşturmayı önerecektir.

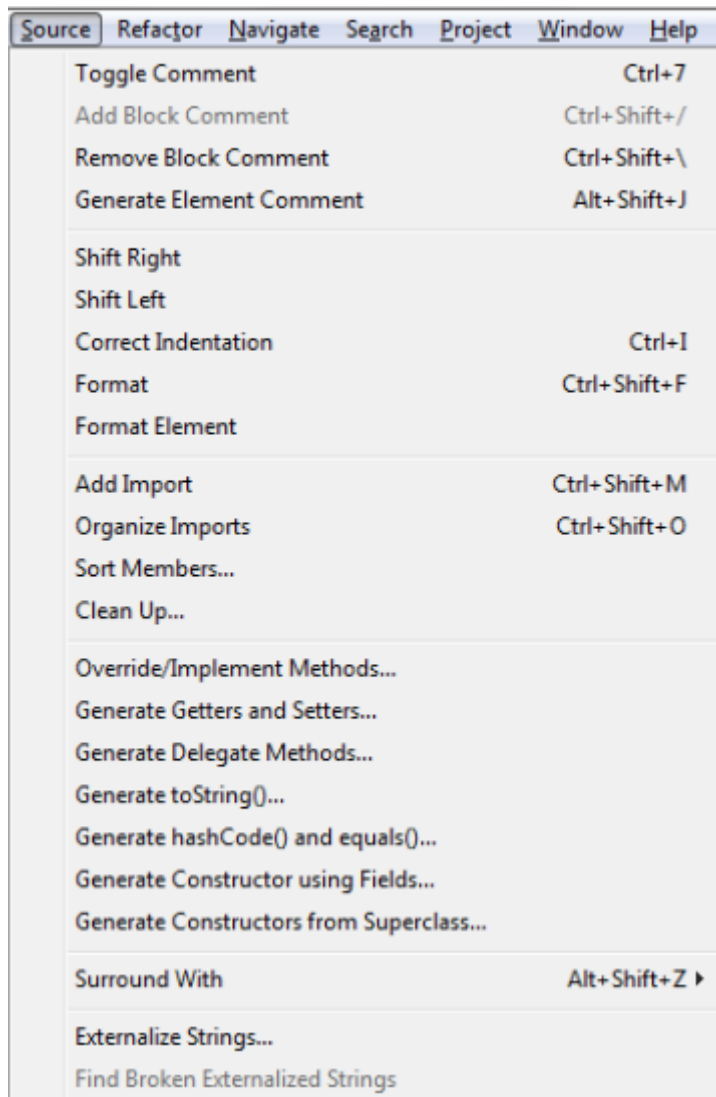


Quick Fix çok güçlü bir özelliktir. Yeni yerel değişkenler ve alanlar oluşturmanızın yanı sıra yeni metodlar ve sınıflar oluşturmanıza da olanak sağlar. Alternatif olarak, istisnalarının etrafına try-catch ifadeleri koyabilir. Ayrıca bir değişken bir ifade atayabilir ve çok daha fazlasını da yapabilir.

11.Kod Üretme

Eclipse kod üretmek için bir takım olanakları içinde bulundurur. Bu olanaklar, geliştirme sırasında önemli ölçüde zaman tasarrufu sağlayabilir.

Bu seçenekleri Source menüsünde bulabilirsiniz.

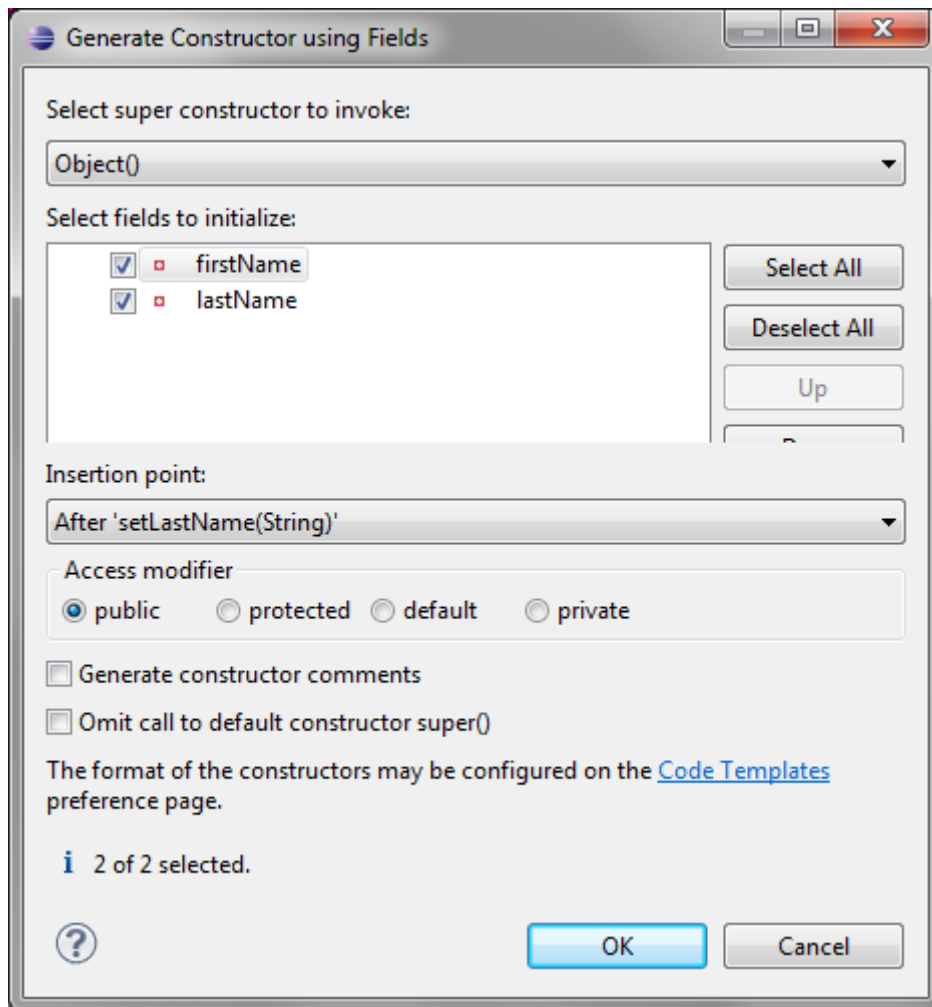


Kaynak üretimini test etmek için, `de.vogella.eclipse.ide.first` projeniz içerisinde aşağıdaki sınıfı oluşturun.

```
package de.vogella.eclipse.ide.first;

public class Person {
    private String firstName;
    private String lastName;
}
```

`Source` → `Generate Constructor using Fields` seçin, her iki alanı da işaretleyin ve OK butonunu seçin.



Source → Generate Getter and Setter kısmını seçin, tekrar her iki alanı da işaretleyip ardından OK butonunu seçin.

Source → Generate toString() kısmını seçin, tekrar her iki alanı da işaretleyin ve OK butonunu seçin.

Aşağıdaki sınıfı oluşturunuz.

```
package de.vogella.eclipse.ide.first;

public class Person {
    private String firstName;
    private String lastName;

    public Person(String firstName, String lastName) {
        super();
        this.firstName = firstName;
        this.lastName = lastName;
    }
}
```

```
public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}

@Override
public String toString() {
    return "Person [firstName=" + firstName + ", lastName=" + lastName
        + "]";
}
}
```

12. Alıştırma: kod üretimi ve içerik yardımı

12.1. Giriş

Bu alıştırma, kod üretiminin kullanımı ve İçerik Yardımı özelliğinin kullanımı hakkında uygulama yapacaksınız.

12.2 Proje Oluşturma

`com.vogella.ide.todo` isminde bir proje oluşturun

12.3. Sınıf Oluşturma

`com.vogella.ide.todo` isminde bir paket ve aşağıdaki sınıfı oluşturun.

```
package com.vogella.ide.todo;

import java.util.Date;

public class Todo {

    private long id;
```

```
private String summary = "";
private String description = "";
private boolean done = false;
private Date dueDate;

}
```

Bütün alanları kullanarak bir yapılandırıcı üretmek için [Source → Generate Constructor using Fields](#) kısmını seçin. Bütün alanlar için getter ve setter fonksiyonlarını oluşturmak için; [Source → Generate Getter and Setter](#) kısmını kullanın.

Oluşturulan sınıf, aşağıdaki gibi görünmelidir.

```
package com.vogella.ide.todo;

import java.util.Date;

public class Todo {

    private long id;
    private String summary = "";
    private String description = "";
    private boolean done = false;
    private Date dueDate;

    public Todo(long id, String summary, String description, boolean done,
        Date dueDate) {
        this.id = id;
        this.summary = summary;
        this.description = description;
        this.done = done;
        this.dueDate = dueDate;
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getSummary() {
        return summary;
    }

    public void setSummary(String summary) {
```



```

    this.summary = summary;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public boolean isDone() {
    return done;
}

public void setDone(boolean done) {
    this.done = done;
}

public Date getDueDate() {
    return dueDate;
}

public void setDueDate(Date dueDate) {
    this.dueDate = dueDate;
}
}

```

id ve *summary* alanlarına dayalı `Todo` sınıfından `toString()` metodunu üretmek için Eclipse'i kullanın. Bu işlem `Source → Generate toString()` adlı Eclipse menüsü kullanılarak yapılabilir.

13.Refactoring

13.1. Refactoring

Refactoring, davranışını değiştirmeden kodu yeniden yapılandırma işlemidir. Örneğin, Java sınıfını veya metodu yeniden adlandırmak refactoring aktivitesidir.

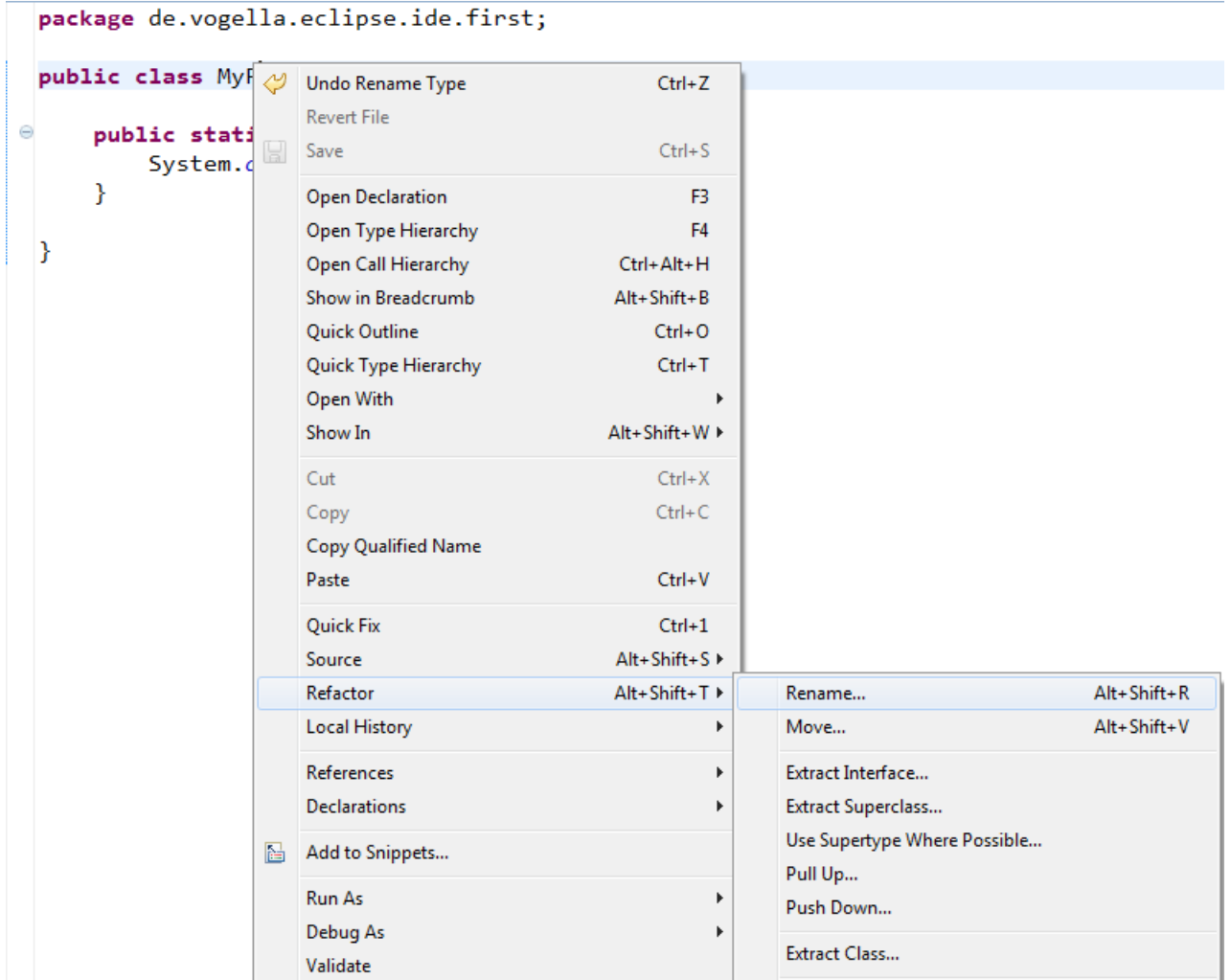
13.2.Eclipse'te Refactoring

Eclipse, yeniden adlandırma(renaming) ve taşıma(moving) gibi bir takım refactoring aktivitelerini destekler.

Rename refactoring'in kullanımına örnek olarak, sınıfınızın üstünde sağ tuşu seçerek (editörde veya Package Explorer' da) ve `Refactor → Rename` kısmını seçerek

sınıfınızı yeniden adlandırabilirsiniz. Eclipse, çalışma ortamınızdaki (workspace) sınıflar ve metotlar için bütün çağrılarını yeniden adlandırılmasını sağlayacaktır.

Aşağıdaki ekran görüntüsü bir sınıf için Rename refactoring'in nasıl çağırıldığını göstermektedir. Burada imleç sınıfın üzerinde konumlandırılmış ve kullanıcı içerik menüsünü görüntülemek için sağ tuş seçilmiştir.



En önemli refactoring işlemleri aşağıdaki tabloda listelenmiştir.

Tablo 1. Refactoring

Refactoring	Tanımlama
Rename	Bir değişkeni veya sınıfı yeniden adlandırır.
Extract Method	Editörde seçilen koda göre metot oluşturur.

Eclipse çok daha fazla refactoring işlemine sahiptir. Mevcut seçenekler Java editöründeki seçime bağlıdır. Çoğu durumda refactoring işlemini adlandırarak ,gerçekleştirilen eylemden fikir almalısınız.

14.Alıştırma: Refactoring

14.1. Hazırlık

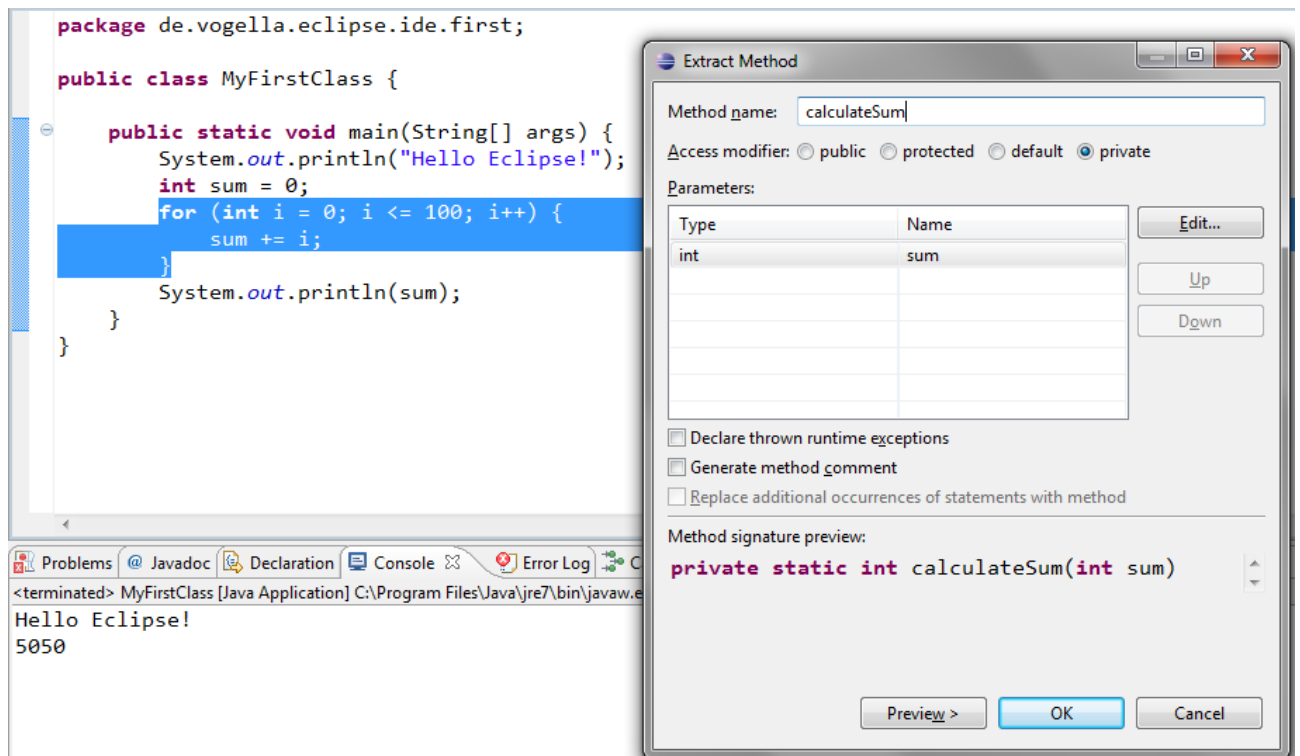
Sonraki örnekler için MyFirstClass sınıfını aşağıdaki koda göre değiştirin.

```
package de.vogella.eclipse.ide.first;

public class MyFirstClass {
    public static void main(String[] args) {
        System.out.println("Hello Eclipse!");
        int sum = 0;
        for (int i = 1; i <= 100; i++) {
            sum += i;
        }
        System.out.println(sum);
    }
}
```

14.2. Metot çıkartma

Kullanışlı bir refactoring işlemi de; kodu işaretleyip, ondan bir metot oluşturmaktır. Bu işlemi alıştırmada kullanmak için, for döngüsünü barındıran kodu işaretleyip sağ tuşu seçerek **Refactoring → Extract Method** kısmını seçin. Bu metoda **calculateSum** ismini verin.



Refactoring işleminden sonra sınıf aşağıda bulunan kod gibi görünmelidir.

```
package de.vogella.eclipse.ide.first;

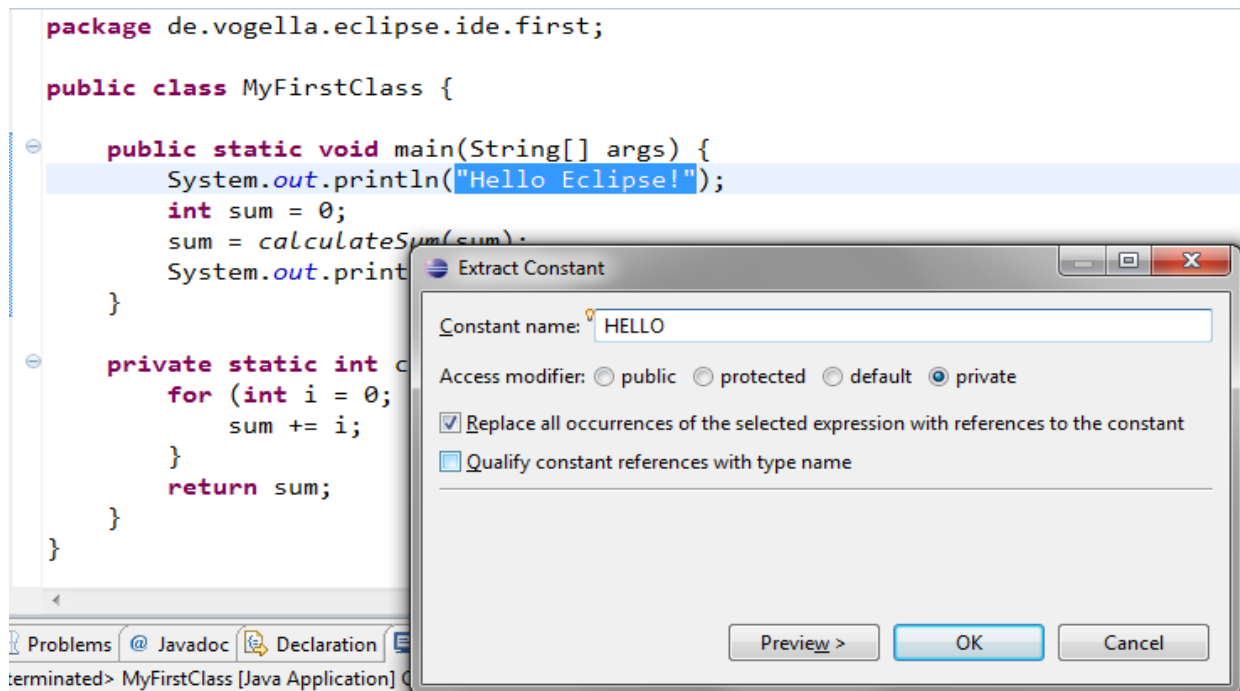
public class MyFirstClass {

    public static void main(String[] args) {
        System.out.println("Hello Eclipse!");
        int sum = 0;
        sum = calculateSum(sum);
        System.out.println(sum);
    }

    private static int calculateSum(int sum) {
        for (int i = 1; i <= 100; i++) {
            sum += i;
        }
        return sum;
    }
}
```

14.3. Sabit(Constant) Çıkartma

Ayrıca stringleri çıkartabilir ve stringlere dayalı yeni sabitler(constants) oluşturabilirsiniz. Bu örnek için kaynak kodunuzdaki Hello Eclipse! String'ini seçin ve *Refactor* → *Extract Constant* kısmını seçin. Yeni sabitinize "HELLO" ismini verin.



```
package de.vogella.eclipse.ide.first;
public class MyFirstClass {

    private static final String HELLO = "Hello Eclipse!";

    public static void main(String[] args) {
        System.out.println(HELLO);
        int sum = 0;
        sum = calculateSum(sum);
        System.out.println(sum);
    }

    private static int calculateSum(int sum) {
        for (int i = 1; i <= 100; i++) {
            sum += i;
        }
        return sum;
    }
}
```

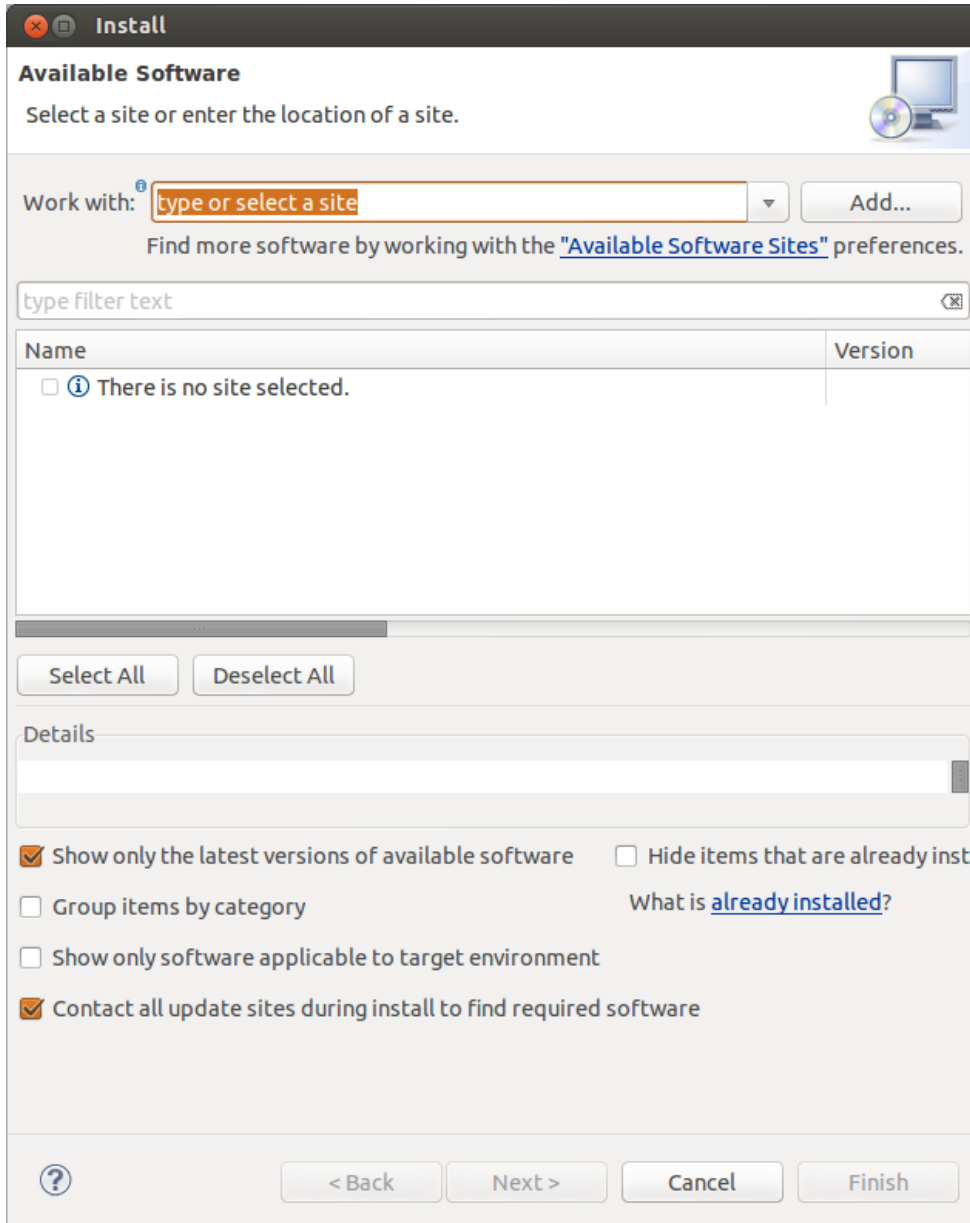
15. Güncellemeler ve Eklentilerin Yüklmesi

15.1 Eclipse Güncelleme Yöneticisi(Eclipse Update Manager)

Eclipse, yazılım bileşenlerini yüklemeniz ve güncellemeniz için Update Manager isimli bir yazılım bileşeni içerir.Kurulabilir yazılım bileşenleri `features` olarak adlandırılır ve eklentilerden(`plug-ins`) oluşur.

Bu özellikler güncelleme siteleri ve yazılım siteleri içinde yer alır. Bir güncelleme sitesi, yüklenebilir yazılım bileşenleri ve ek yapılandırma dosyaları içeren, web sunucusu üzerindeki bir dizin olabilir.

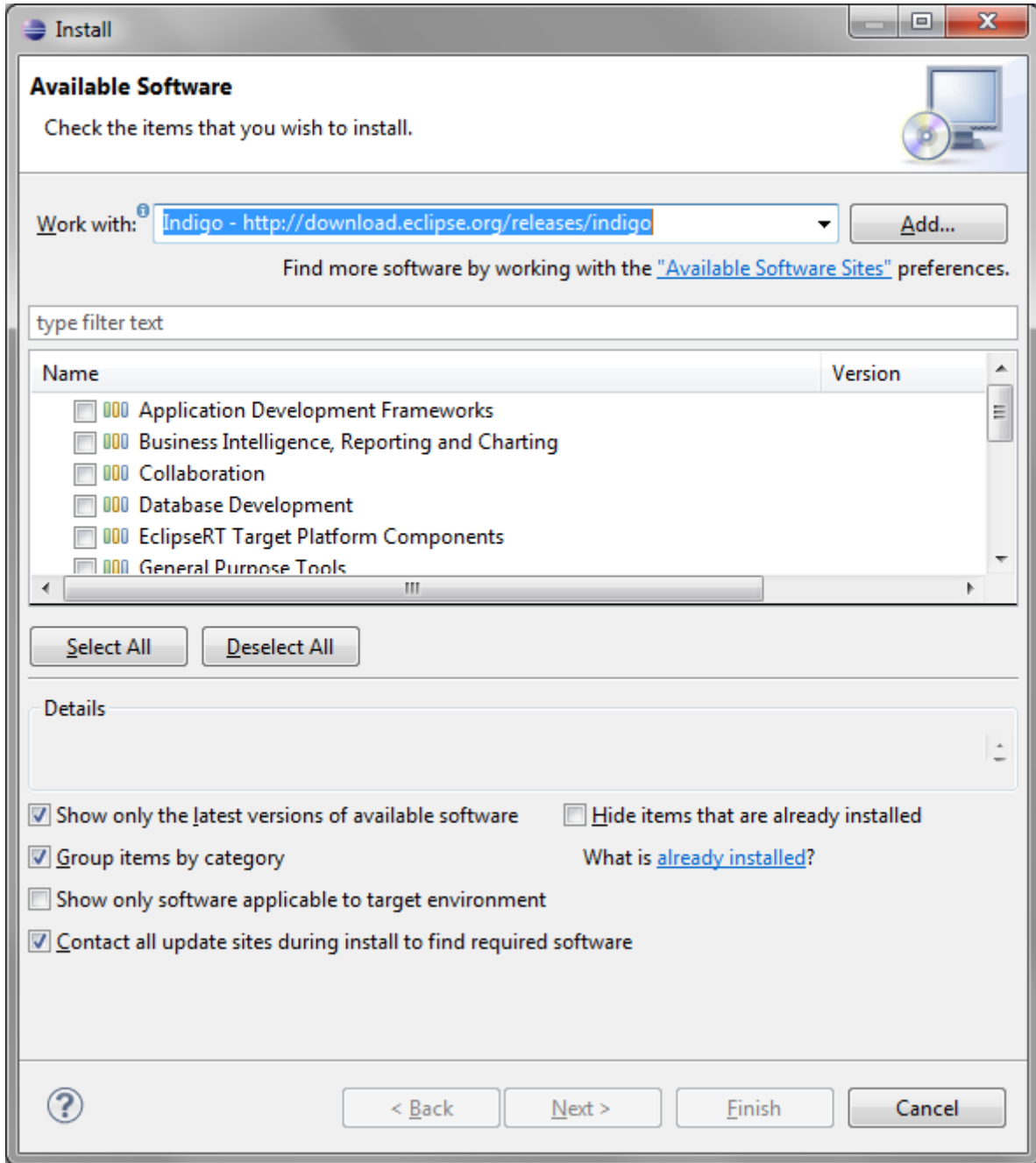
Bu yapılandırma dosyaları, yazılım bileşenleri hakkında bilgi sağlar. Eğer Eclipse, güncelleme sitesinde hangi yazılım bileşenin mevcut olduğunu kontrol ederse, sitenin içeriğini bilmek için sadece bu yapılandırma dosyalarının indirilmesi ve ayrıştırılması yeterlidir.



Eclipse kurulumunu g¼ncellemek iin, **Help → Check for Updates** seeneęini sein. Sistem, ¼nceden y¼klenmiř yazılım bileřenleri iin g¼ncelleřtirmeleri arar. Eęer g¼ncellenmiř bileřen bulursa, sizden g¼ncellemeyi onaylamanızı ister.

Yeni bir iřlevsellik y¼klemek iin, **Help → Install New Software** seeneęini sein.

Yeni bir yazılım bileřeni y¼klemek istiyorsanız, Work With listesinden bir URL seebilirsiniz.



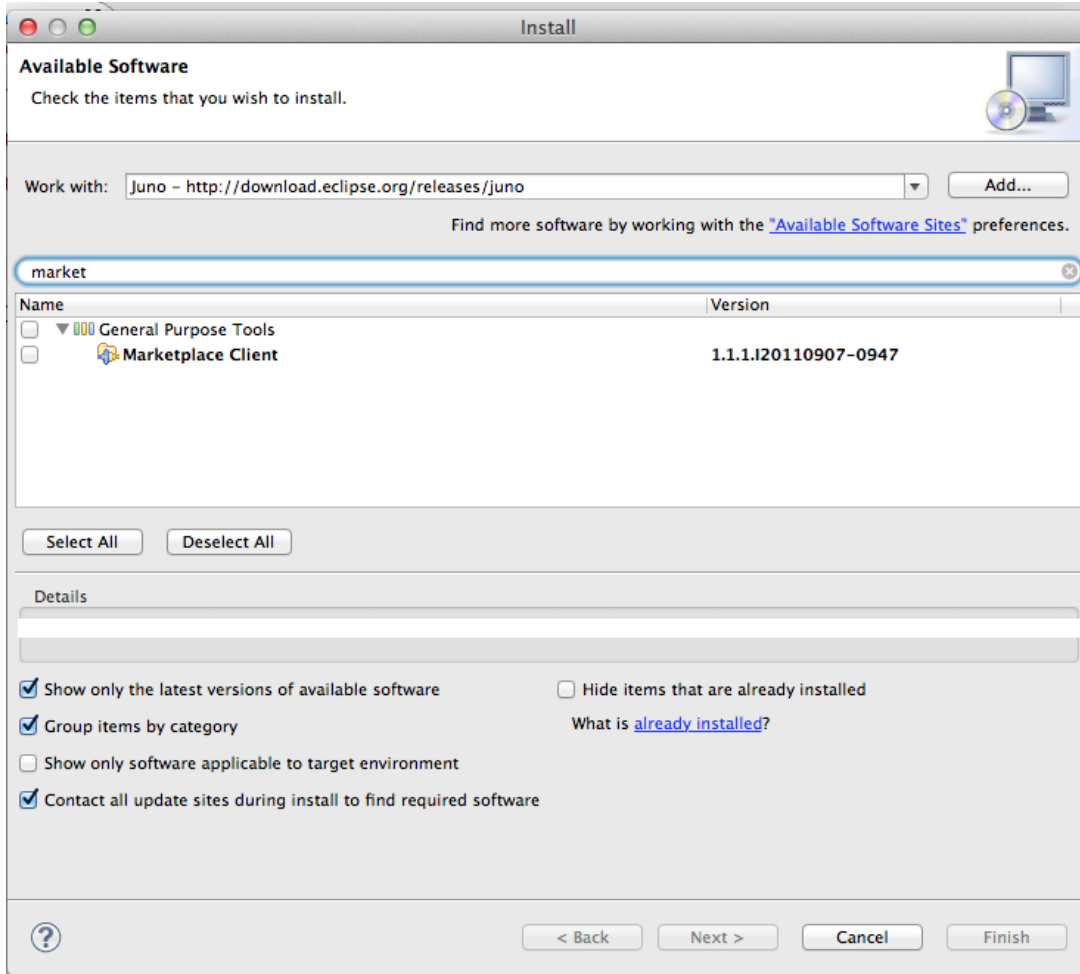
Bileşenleri yüklemek için Group items by category seçeneğinin işaretini kaldırmalısınız; çünkü halihazırda bütün plug-in'ler kategorize edilmemiştir. Eğer kategorize edilmemişlerse, gruplama devre dışı bırakılmadan gösterilmezler

15.2. Eclipse Marketplace

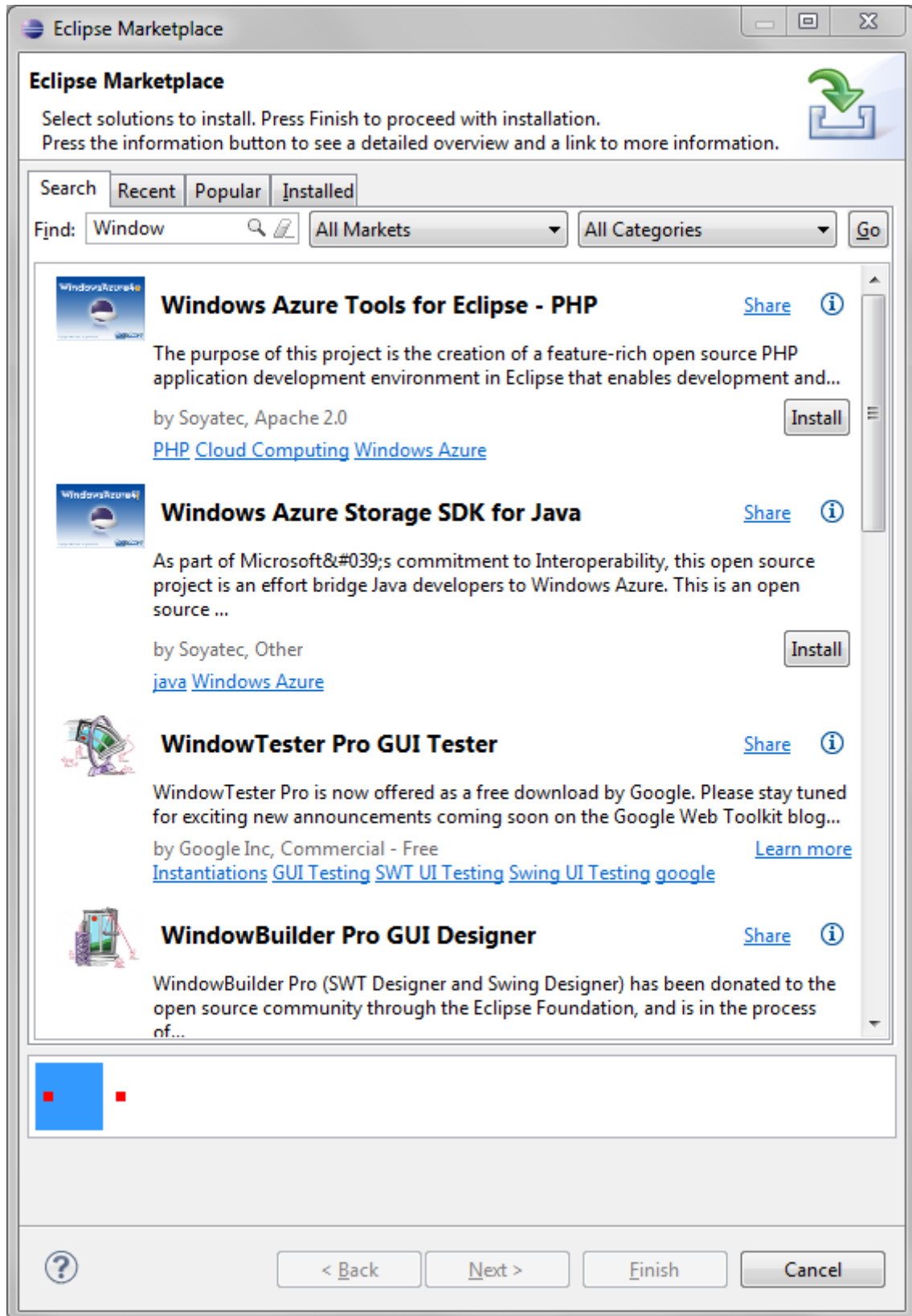
Eclipse aynı zamanda, yazılım bileşenlerinin Eclipse Marketplace istemcisinden yüklenmesini de içerir. Bu istemcinin faydası ise bileşenleri arama, popüler uzantıları keşfetme, açıklamaları ve derecelendirmeleri görebilmeyi sağlamasıdır.

Update Manager ile karşılaştırılırsa, yüklenebilen yazılım bileşenini içeren sitenin URL'sini bilmenize gerek yoktur.

Her Eclipse; Marketplace'i içinde barındırmaz. Kullanmadan önce sizin Marketplace istemcisinin yazılım bileşenini Eclipse'e yüklemeniz gerekmektedir. Aşağıdaki ekran görüntüsü, resmi bir Eclipse güncelleme sitesinden Marketplace'in nasıl yüklendiğini göstermektedir.



- Eclipse Marketplace'i açmak için **Help → Eclipse Marketplace** seçeneğın seçebilirsiniz.



Bileşenleri aramak için Find özelliğini kullanabilirsiniz. Install butonunu seçmek yükleme işlemini başlatır.

15.3 Bileşenleri Kaldırma

Eğer [Help → About](#) seçeneğini seçtikten sonra [Installation Details](#) butonunu seçerek, bileşenleri Eclipse IDE'nizden kaldırabilirsiniz.

15.4. Eclipse'i Yeniden Başlatma

Yeni bir yazılım bileşenini güncelleme veya yükleme işleminden sonra, değişikliklerin gerçekleştiğinden emin olmak için Eclipse'i yeniden başlatmalısınız.