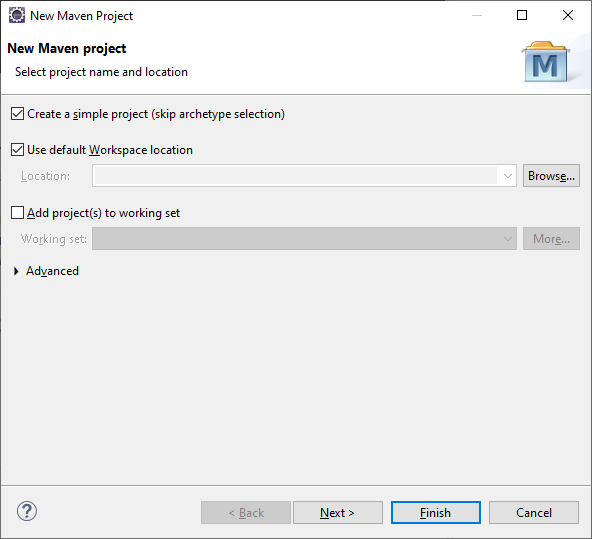
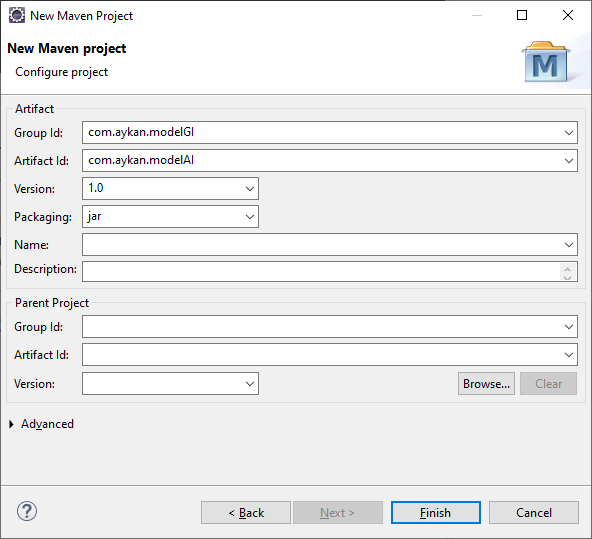
<https://www.youtube.com/watch?v=5LH15ggLaAs&list=PL4ET09KoRZtT3ROOSDcDVVXzSQpOz1Cdb&index=2>





<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.ramazan.model</groupId>

<artifactId>com.ramazan.model</artifactId>

<version>1.0</version>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

## Notlar

Class File Editor - Source not found - Eclipse

Source Not Found JRE Classes Eclipse

https://www.youtube.com/watch?v=UCXMDIxe9OY

source attachment configuration

https://feeling.sourceforge.io/update/

<https://medium.com/@metinalniacik/equals-ve-hashcode-metotlar%C4%B1n%C4%B1n-%C3%B6nemi-nedir-800c99af1519>

Equals ve HashCode Metotlarının Önemi Nedir?

Öncelikle **equals** metodunu inceleyelim;

İlgili metot, iki nesnenin içeriğinin aynı olup olmadığını kontrol eder.

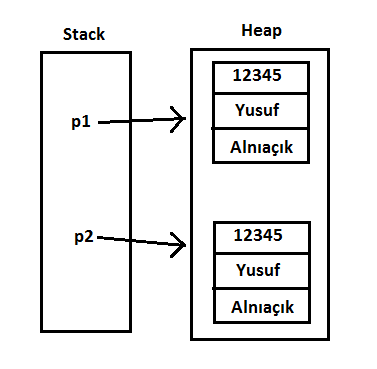
(==) operatörü adreslerin aynı olup olmadığını kontrol eder.

**ÖrneğinPerson p1 = new Person(12345L, “Yusuf”, “Alnaıçık”);Person p2 = new Person(12345L, “Yusuf”, “Alnıaçık”);**

**System.out.println(p1 == p2);**

Sonuç **false** olarak dönecektir. Çünkü nesneler farklı adreslere sahiptirler.

Image for post



Aynı şekil üzerinde equals metodunu da anlatalım.

Eğer equals metodu tanımlı değilse sonuç **false** olacaktır.

Person class’ına equals metodunu tanımlayalım;

Tüm alanların içeriklerinin aynı olup olmadığı kontrol eder. Bu durumda **true** sonucu dönecektir.

Gelelim collection’larda çok fazla önem taşıyan **hashcode** metoduna, Aşağıdaki code çalıştırıldığında üç kayıt görünecektir. Çünkü hashcode tanımlanmamıştır.

**Person p1 = new Person(12345L, “Yusuf”, “Alnıaçık”);Person p2 = new Person(12345L, “Yusuf”, “Alnıaçık”);Person p3 = new Person(13579L, “Dilek”, “Alnıaçık”);**

**Map<Person, String> liste = new HashMap<>();liste.put(p1, “Yusuf”);liste.put(p2, “Yusuf”);liste.put(p3, “Dilek”);**

**System.out.println(liste.size());**

p1 ve p2 nin tüm değerleri aynı olmasına rağmen Person class’ında hashcode metodu olmadığından sonuç üç olacaktır.

Örneklerimize hashcode ve equals metotlarını tanımlayarak devam edelim;

**Person p1 = new Person(12345L, “Yusuf”, “Alnıaçık”);Person p2 = new Person(12121L, “Yusuf”, “Alnıaçık”);Person p3 = new Person(13579L, “Dilek”, “Alnıaçık”);**

**Map<Person, String> liste = new HashMap<>();liste.put(p1, “Yusuf”);liste.put(p2, “Yusuf”);liste.put(p3, “Dilek”);**

**System.out.println(liste.size());**

HashCode metodu identityNumber’a bakıyor. Eğer identityNumber unique (eşsiz) ise bunu farklı bir kayıt sayarak listeye ekliyor. Sonuç olarak üç değeri döner.

**Person p1 = new Person(12345L, “Yusuf”, “Alnıaçık”);Person p2 = new Person(12345L, “Metin”, “Alnıaçık”);Person p3 = new Person(12345L, “Dilek”, “Alnıaçık”);**

**Map<Person, String> liste = new HashMap<>();liste.put(p1, “Yusuf”);liste.put(p2, “Metin”);liste.put(p3, “Dilek”);**

**System.out.println(liste.size());**

Yukarıdaki örnekte ise biraz değişiklik yaptık ve kodun çalışma şeklinde biraz farklılıklar oluştu.

Öncelikle, p1 kaydı listeye ekleniyor. Daha sonra p2 kaydı ile p1 kaydının identityNumber değerleri aynı olduğundan dolayı kaydın tamamen aynı olup olmadığını anlamak için equals metoduna kullanılarak p1’in ve p2'nin tüm alanlarının içerikleri aynı olup olmadığı kontrol ediliyor ve aynı olmadığı tespit edilerek, bu kayıt listeye ekleniyor.

Son olarak p3'ün kontrolü yapılıyor. p3'ün identityNumber değeri p1 ve p2 ile aynı olduğundan p1 ve p3 için equals metodu çalışır ve tüm alanların içerikleri aynı olmadığından dolayı sonuc **false** döner. Daha sonra p2 ve p3 için equals metodu çalışır ve durum bu kayıtlar içinde aynıdır. Sonuç yine **false** döner. Yani bir kaydın identityNumber değeri kaç kayıt ile aynı olursa, ilgili kayıtların tümü ile equals metodu çalışacaktır.

**Person p1 = new Person(12345L, “Yusuf”, “Alnıaçık”);Person p2 = new Person(12345L, “Dilek”, “Alnıaçık”);Person p3 = new Person(12345L, “Dilek”, “Alnıaçık”);**

**Map<Person, String> liste = new HashMap<>();liste.put(p1, “Yusuf”);liste.put(p2, “Dilek”);liste.put(p3, “Dilek”);**

**System.out.println(liste.size());**

Son olarak yukarıdaki örneği inceleyelim. İlk olarak p1 nesnesi listeye eklenecektir. p2 nesnesinin identityNumber değeri p1 ile aynı olduğundan equals metodu çalışacaktır. Tüm değerler kontrol edildiğinde farklılık görülmektedir. Bundan dolayı sonuç **false** dönecektir ve kayıt listeye eklenecektir. p3'ün identityNumber değeri p1 ile aynı olduğundan equals metoduna gidilir. Nesnelerin alanları karşılaştırıldığında, farklılıklar bulunduğundan dolayı değer **false** dönecektir. p2'nin ve p3'ün identityNumber değerleri aynı olduğundan equals metoduna gidilir. Tüm alanların değerleri aynı olduğundan, sonuç **true**dönecektir. Bu durumda, her iki kaydın aynı olduğu anlamına gelir. Bu durumda ise listeye kayıt ekleme işlemi gerçekleşmez. Toplam kayıt sayısı iki olarak görülmektedir.

/\*\*

Category ile Product Arasında bire çok ilişki varsa

Yani 1 category’nin 🡪 birden çok product var ise

Category Sınıfın’da bir tane Liste tanımlarız product için

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

public List<Product> Products { get; set; }

}

public class Product

{

public int Id { get; set; }

public string Name { get; set; }

public string ImageUrl { get; set; }

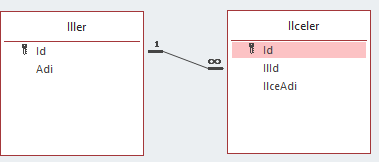
public decimal Price { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; }

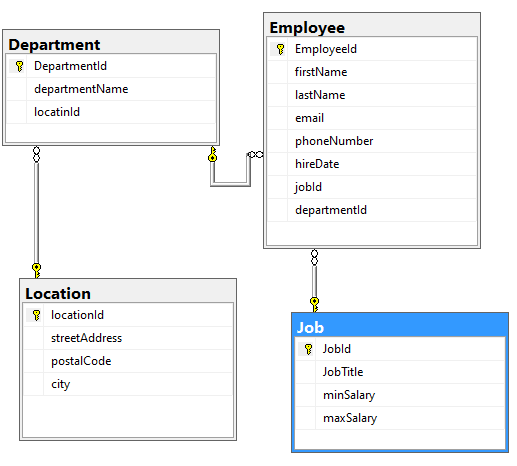
}

\*/



Bir ilin 🡪 çok ilçesi var

Bire (Class ismi ) 🡪 çok ilişki (List<T>)





Hata 1= Could not calculate build plan: Plugin org.apache.maven.plugins:maven-resources-plugin:2.6 or one of its dependencies could not be resolved: Failed to read artifact descriptor for org.apache.maven.plugins:maven-resources-plugin:jar:2.6

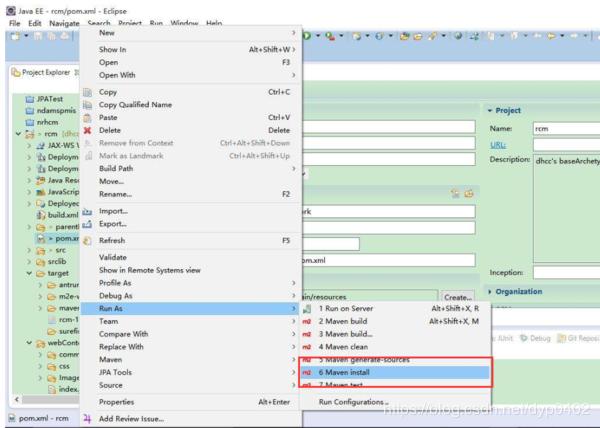
Plugin org.apache.maven.plugins:maven-resources-plugin:2.6 or one of its dependencies could not be resolved: Failed to read artifact descriptor for org.apache.maven.plugins:maven-resources-plugin:jar:2.6

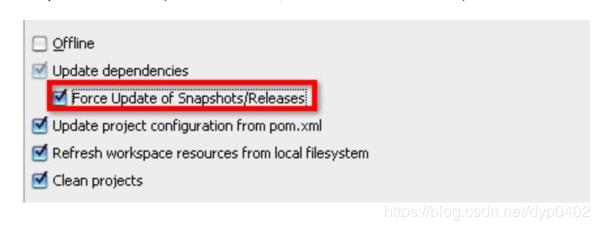
Çözüm 1

This error occurs because maven-resource-plugin:2.6 is the corresponding dependency file in the maven local repository. The solution is as follows:

1. In the eclipse, find the pom.xml file under the maven project, and add the contents of the red box below.



2. Open the dos command console, enter the directory of the maven project (including the level of the pom.xml file), and then select maven install and press Enter to find that maven will automatically download the missing files.  


3. After the download is complete, right-click the maven project in eclipse - Maven-Update Project Check the Force Update of Snapshots/Releases option and click the OK button. The error has disappeared.  


## Git Proje Atma

<https://www.youtube.com/watch?v=-tq5V07UXXc>

# Github'da Repository oluşturma, Proje gönderme ve Commit etme

…or create a new repository on the command line

echo "# javaSpring" >> README.md

git init

git add README.md

git commit -m "first commit"

git remote add origin https://github.com/ramazanorhanor/javaSpring.git

git push -u origin master

…or push an existing repository from the command line

git remote add origin https://github.com/ramazanorhanor/javaSpring.git

git push -u origin master

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model

$ git init

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

buraya geçtik

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

$ git add .

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

$ git commit -m "javaSpring projesi git commit -m komutu"

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

$ git remote add origin https://github.com/ramazanorhanor/javaSpring.git

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

$ git push -u origin master

Projeyi gönderdik

Şimdi ise bir değişiklik yaptığımızda projeyi tekrardan nasıl göndeririz onun komutlarını yazıyoruz

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master) $ git add .

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

$ git commit -m "word dosyası ekledik"

1. Aşama

ramaz@DESKTOP-JQVK3QL MINGW64 ~/eclipse\_workspace2020EE/com.ramazan.model (master)

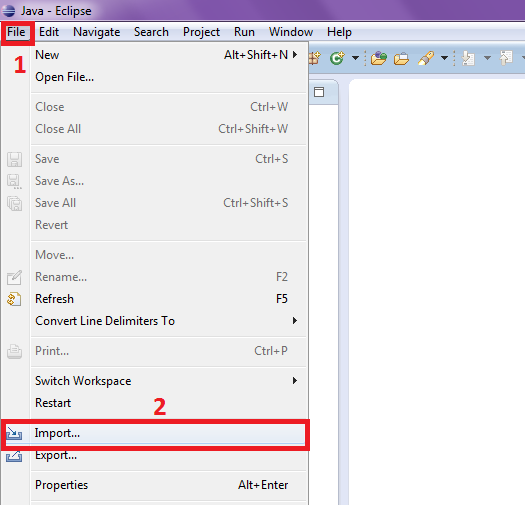
$ git push -u origin master

Sorun1 = java-eclipse-ile-olusturulmus-projeyi-baska-bilgisayarda-acmak/

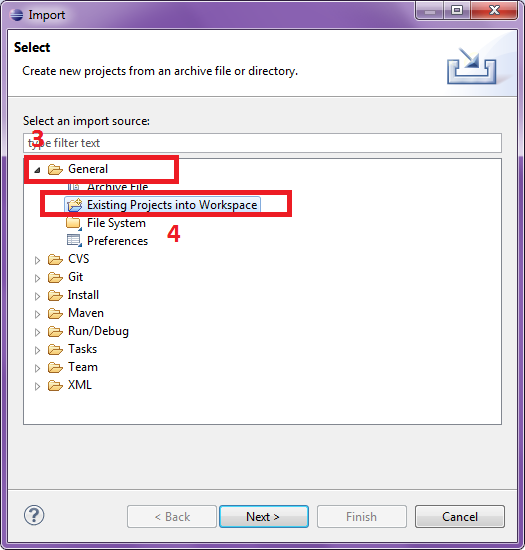
Çözüm1

Eclipse ile Java geliştirirken çalıştığınız projeyi sol taraftaki projeler alanından sildiğinizde veya daha önceden çalıştığınız bir projeyi başka bir bilgisayardaki Eclipse üzerinde çalıştırmak istediğinizde proje dosyalarını eclipse çalışma ortamına (workspace) tanıtmanız gerekiyor. Projeyi tanıtmak için aşağıdaki yolu izliyoruz.

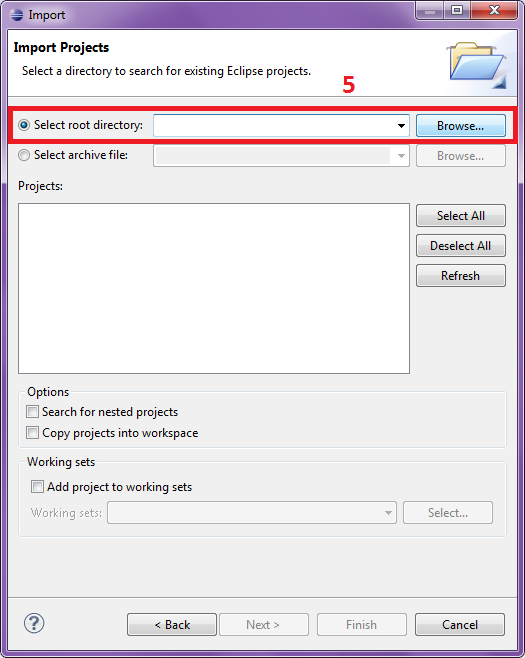
1. Öncelikle Eclipse’te sol üstteki **File** menüsünden **Import**‘ta tıklıyoruz.

[](https://www.akifarslan.com.tr/wp-content/uploads/2014/03/eclipse1.png)

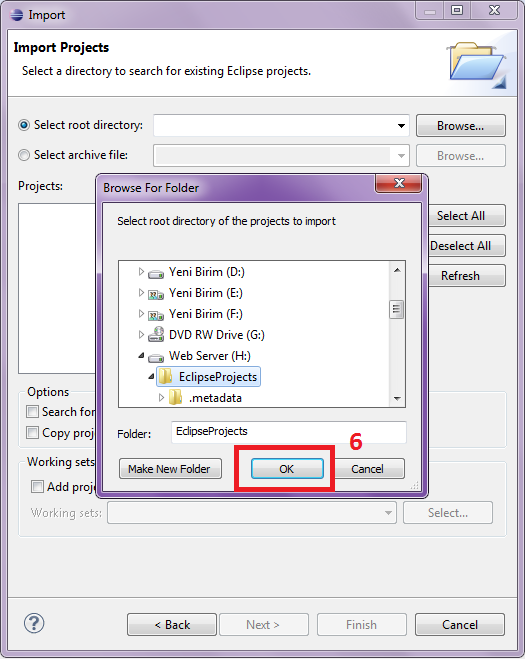
2. Açılan pencerede **General** tıklayarak açılan seçeneklerden **Existing Projects into Workspace** seçeneğini seçerek **Next** diyoruz.

[](https://www.akifarslan.com.tr/wp-content/uploads/2014/03/eclipse2.png)

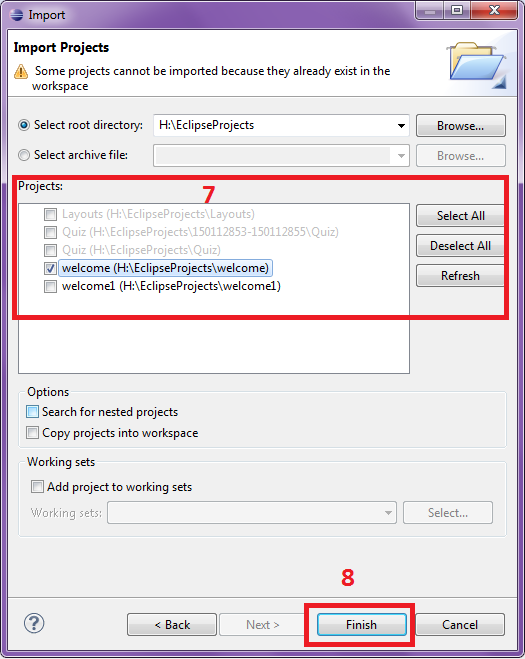
3. Aşağıdaki resimde de görüldüğü gibi **Select root directory** kısmındaki **Browse** butonuna tıklıyoruz.

[](https://www.akifarslan.com.tr/wp-content/uploads/2014/03/eclipse3.png)

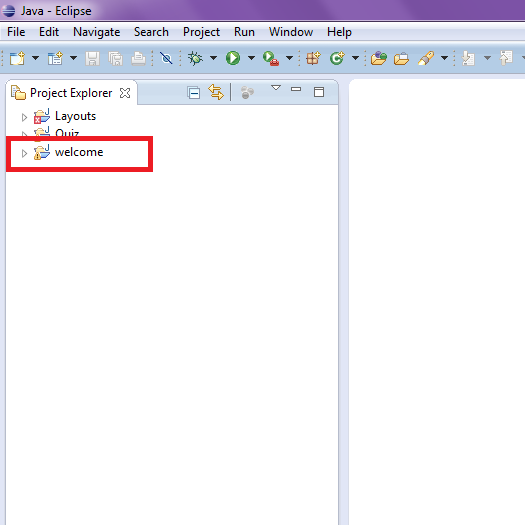
4. Açılan pencerede proje ana klasörünü seçeceğiz. Eğer projeyi diğer projelerinizin bulunduğu klasöre attıysanız bu klasörü seçmeniz yeterli. Ancak açmak istediğiniz proje başka bir klasörde ise bu projenin ana klasörünü seçmemiz gerekiyor. Benim tüm projelerim EclipseProjects klasörü içinde bulunuyor. Yeni açmak istediğim projeyide bu klasöre kopyaladığım için bu klasörü seçtim. Kendimize göre seçimi yaptıktan sonra OK diyoruz.

[](https://www.akifarslan.com.tr/wp-content/uploads/2014/03/eclipse4.png)

5. Yeni açılan pencerede projeler listelenecektir. Burada açmak istediğimiz projeyi seçiyoruz. Benim bilgisayarımda welcome ve welcome1 projeleri tanımlanmamış olduğu için bu iki proje koyu gözüküyor. Diğer projeler eclipsete tanımlı olduğu için soluk görünmektedir. Eklemek istediğimiz projeyi seçerek **Finish** diyoruz.

[](https://www.akifarslan.com.tr/wp-content/uploads/2014/03/eclipse5.png)

6. Aşağıdaki resimde de görüldüğü gibi eclipsete tanımlı olmayan welcome projesini workspace tanımlamış olduk. Artık projemiz üzerinde çalışabiliriz.

[](https://www.akifarslan.com.tr/wp-content/uploads/2014/03/eclipse6.png)

## Oku ve araştır

Oku1 @Transient nedir nasıl kullanılır

Cevap1