

BBS515 Nesneye Yönelik Programlama

Ders 1

Zümra Kavafoğlu

Dersin Amacı

Nesneye Yönelik Programlama Kavramını Java Dili Üzerinden Öğrenmek

Nesneye yönelik programlama ile ilgili nesne, sınıf, kalıtım, çok biçimlilik, kapsülleme, arayüzler gibi temel kavramları tanıtmak

Bu kavramları kullanarak Java programlama diliyle program yazma becerisi kazanmak

Puanlama

- 3 programlama ödevi (% 20)
- 1 arasınay (%40)
- Final (%40)
- Derslere düzenli katılım beklenmektedir.

Ödevler

- Temel konularla ilgili toplam 3 programlama ödeviniz olacaktır.
- Programlama ancak program yazarak öğrenenebilir, bu sebeple ödevlerinizi mutlaka yapmaya çalışın.
- Ödev ile ilgili takıldığınız yerlerde e-posta ile veya ofis saatinde yardım isteyebilirsiniz.
- Ödevlerinizin kodu tamamen size ait olmalıdır. Aksi tespit edildiği takdirde o ödevden hiç not alamazsınız.

Sınavlar

- Arasınave ve final sınavı yazılı sınav şeklinde yapılacaktır. Sınavda karşılaşabileceğiniz soru çeşitleri:
 - program yazma
 - verilen programda hata bulma
 - verilen programın çıktısını yazma
- Yazılı sınavlarda başarılı olmak için bol bol kod pratiği yapmalısınız. Bunun için :
 - * her dersten sonra sunumda gösterilen kodları kendiniz de yazın
 - * ödevlerinizi mutlaka yapın.
 - * öncelikle ders sayfasına konulan alıştırmaları ve ek olarak kaynak kitaplarda yer alan alıştırmaları yapmaya çalışın.
- Derse düzenli çalışın, konuların birbiri üstüne geliştiğini, önceki konularınızda eksiklik olursa yeni konuları anlamakta sıkıntı yaşayacağınızı unutmayın.

İletişim

- e-posta adresi : zdemir@hacettepe.edu.tr
- Ofis saati: Çarşamba 16:00 - 18:00
- **Sınav günleri ofis saati yapılmamaktadır.**
- Ödevler veya dersle ilgili diğer konular için e-posta ile de soru sorabilirsiniz. Sorduğunuz soruların cevabı bir sonraki ofis saatinde e-posta ile cevaplanır.
- Sınavlar, ödevler ve derslerle ilgili tüm materyaller ve duyurular ders sayfasında yayınlanacaktır. Ders sayfasını sık sık kontrol etmeyi unutmayın.

<https://zumrakavafoglu.github.io/courses/bbs515.html>

Bu derste neler öğreneceğiz?

- Programlama Dilleri ve Seviyeleri
- Programlama Dillerinin Tarihçesi
- Tipik bir Java geliştirme Ortamı
- JRE ve JDK
- Örnek bir Java kodu
- IDE (Tümleşik Geliştirme Ortamı)

Programlama Dilleri ve Seviyeleri

- **Programlama dili**, bir makineye özellikle de bilgisayara bir takım talimatlar vermek için tasarlanmış yapay bir dildir.
- Bilgisayarlar sayısal elektronik yapıda olduklarından anladıkları tek şey ikili(binary) sayı düzenindeki 1 ve 0 rakamlarıdır. Bu bakımdan bir bilgisayarla aynı dili konuşabiliyor olsaydık ona yapmasını istediğimiz şeyleri belli bir sırada 1 ve 0'lardan oluşan bir dizi halinde iletebilmemiz gerekirdi. Farklı makinelerin farklı dizi yapılarını anlayabilmesi de cabası! *
- **Makine Dili(1. nesil programlama dili)**, işlemcinin verilen komutlar doğrultusunda çalıştırılmasını sağlayan ve işlemci mimarisine göre değişen en alt seviyedeki programlama dilidir. Bu dil sadece 0 ve 1 ikililerinin anlamlı kombinasyonlarından meydana gelmektedir. Bu nedenle, makine dilinin anlaşılması çok güçtür.

* Java , Numan Pekgöz, Pusula Yayıncılık



Python, JavaScript,...

Java, C#,...

C++, C,...

Assembly

Makine Dili



Programlama Dilleri ve Seviyeleri

- **Assembly(Çevirme Dili)**, belirli bir bilgisayar mimarisine özelleşmiş düşük seviye bir programlama dilidir. Makine dilinden daha anlaşılırdır ama yüksek seviye dillere göre öğrenmesi ve yazması çok zordur.

Assembly

```
title    Hello World Program

dosseg
.model small
.stack 100h

.data
hello_message db 'Hello World!',0dh,0ah,'$'

.code
main proc
    mov     ax,@data
    mov     ds,ax

    mov     ah,9
    mov     dx,offset hello_message
    int     21h

    mov     ax,4C00h
    int     21h
main endp
end main
```

C

```
#include<stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}
```



Python, JavaScript,...

Java, C#,...

C++, C,...

Assembly

Makine Dili



Programlama Dilleri ve Seviyeleri

- **Yüksek Seviye Diller**, gerçek dile(İngilizce) daha yakın komutlara sahip, bilindik matematik işlemlerini kapsayan, yazması ve öğrenmesi daha kolay programlama dilleridir.
- Günümüzde sıklıkla kullanılan C, C++, Java, Python gibi diller yüksek seviye programlama dilleridir.
- Bu dilleri makine diline çevirmek için derleyiciler kullanılır.



Python, JavaScript,...

Java, C#,...

C++, C,...

Assembly

Makine Dili



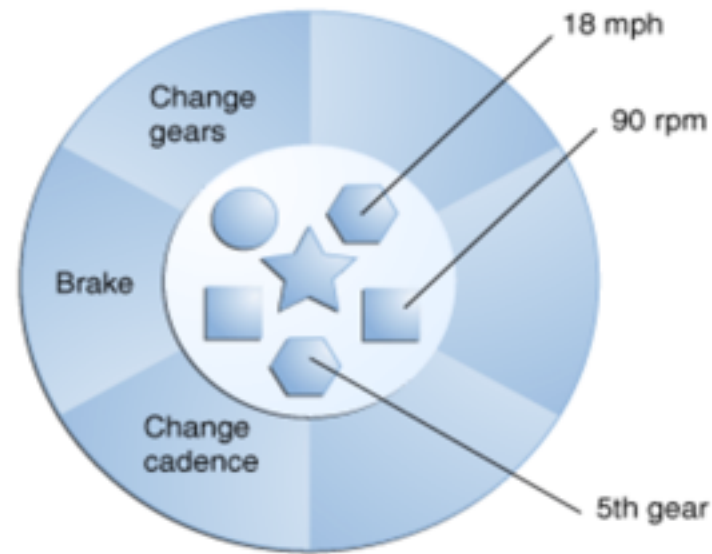
Yapısal ve Nesne Yönelimli Programlama Dili Farkları

Yapısal Programlama Dili	Nesne Yönelimli Programlama Dili
Program fonksiyon adı verilen küçük parçalara bölünmüştür.	Program nesne adı verilen küçük parçalara bölünmüştür.
Top-down yaklaşım	Bottom-up yaklaşım
Yeni veri ve fonksiyon eklemek zordur, diğer bir deyişle genişletilebilirliği kolay değildir.	Yeni veri ve fonksiyon eklemeyi kolaylaştıran bir yapıdadır.
Genellikle tüm veriler fonksiyonlar tarafından erişilebilirdir.	Veriler özel(private) veya genel(public) olarak tanımlanabilir ve böylece verinin erişilebilirliği kontrol altında tutulabilir.

Nesne nedir

- Yazılım nesneleri, gerçek hayattaki nesnelere benzer özellikler taşıyan yapılardır.
- Gerçek nesnelerin ortak iki özelliği vardır: bir duruma ve bir davranışa sahiptirler.
- Örneğin masa lambanızın iki farklı durumu vardır, “Açık” ve “Kapalı”, ve iki mümkün davranışı vardır “Aç” ve “Kapa”.
- Yazılım nesneleri de benzer şekilde durumlara ve davranışlara sahiptirler. Bir yazılım nesnesi durumunu değişkenler içinde tutarken, davranışını belirleyen metotlara sahiptir.
- Metotlar ait oldukları nesnenin durumu üzerinde işlem yapar ve nesneler arası iletişimde temel mekanizmayı oluşturur.
- Bir nesnenin iç durumunu saklamaya ve tüm iletişimleri metodlar aracılığıyla sağlamasına veri kapsüllemesi denir ve bu nesne yönelimli yazılımın temel prensiplerindendir.

Nesne nedir



A bicycle modeled as a software object.

- Örneğin bir bisikleti ele alalım. Bisikletin durumunu belirleyerek (hız, pedal ritmi ve vites) ve bu durumu değiştirecek metodlar tanımlayarak, dış dünyanın onu nasıl kullanabileceğinin kontrolünü elde etmiş oluruz. Örneğin bisiklet yalnızca 6 vitesse sahipse, vites değiştirmek için tanımlanmış bir metod 1'den az veya 6'dan büyük bir değeri kabul etmeyecektir.

Nesne nedir

- Nesnelerle kod yazmanın bir çok faydası vardır
 - **Modülerlik:** Bir nesnenin kodu, sistemdeki diğer nesnelerin kodundan bağımsız olarak yazılabilir ve değiştirilebilir. Bir kere oluşturulduğunda sistemin istenilen yerinde rahatça kullanılabilir.
 - **Bilgi saklama:** Bir nesneyle yalnızca metodları aracılığıyla ilişki kurularak, onun iç yapısı dış dünyadan gizli hale getirilebilir.
 - **Kodun yeniden kullanılması:** Eğer bir nesne halihazırda mevcutsa (başka bir yazılımcı tarafından yazılmış da olabilir), bu nesneyi kendi programınızda kullanabilirsiniz. Böylelikle uzman yazılımcıların yazdığı ve test ettiği karmaşık ve göreve-özel nesneleri kendi programınızda güvenle kullanabilirsiniz.
 - **Ekleme ve hata ayıklama kolaylığı:** Eğer herhangi bir nesne problem çıkarıyorsa onu uygulamanızdan çıkarabilir ve yerine yeni bir nesne ekleyebilirsiniz. Bu gerçek dünyadaki mekanik problemleri çözmeye benzerdir. Eğer bir civata bozulursa, bütün makineyi değil onu değiştirirsiniz.

C ve C++'ın tarihçesi

- **C**
 - Yapısal bir programlama dilidir.
 - 1972'de Unix işletim sistemi için geliştirilmeye başlanmıştır.
- **C++**
 - Nesne Yönelimli bir programlama dilidir.
 - 1979'da Danimarkalı Bilgisayar Bilimcisi Bjarne Stroustrup'un doktora tezi için "Sınıf içeren C" üzerine çalışmasıyla geliştirilmeye başlanmıştır.
 - 1985'de ilk ticari sürümü yayımlanmıştır.

Java'nın Tarihçesi

- **Java**

- Nesne Yönelimli bir programlama dilidir.
- 1991'de James Gosling, Mike Sheridan, ve Patrick Naughton tarafından ticari elektronik cihazlar için geliştirilmeye başlanmıştır.
- 1995'de ilk sürümü "Write Once, Run Everywhere"("Bir defa yaz, her yerde çalıştır") mottosuyla Sun Microsystems tarafından yayınlanmıştır.
- C++'dan farklı olarak makine diliyle iletişimi bir sanal makine aracılığıyla sağlar.
- Yazılımcılara tanıdık gelmesi için C/C++ benzeri bir syntax stiliyle tasarlanmıştır.

Neden Java?

Java

- Nesne yönelimli programlama kurallarının en net olduğu dillerdendir
- Bu özelliği ile büyük kurumsal projelerde kullanılmaktadır
- İşletim sistemi ve donanım bağımsızdır
- İnternet sayfaları üstünden çalıştırılabilir
- Gömülü sistemlerde kullanılmaktadır
- Cep telefonlarında kullanılmaktadır (Android)
- En önemlisi kuralları ve yapısı sayesinde karmaşık projelerde anlaşılır ve takip edilebilir bir yapısı bulunmaktadır

Tipik bir Java Geliştirme Ortamı

- **Java Programları aşağıdaki 5 aşamadan oluşur**

- **Düzenleme**

- ▶ Programcı bir editör kullanarak kaynak kodu yazar ve kodun olduğu dosyayı .java uzantısıyla diske kaydeder.

- **Derleme**

- ▶ javac(Java Derleyicisi) kullanılarak kaynak kodundan bytecode üretilir, bytecode .class dosyasında saklanır.

- **Yükleme**

- ▶ Sınıf yükleyici .class dosyasından bytecodeları hafızaya okur.

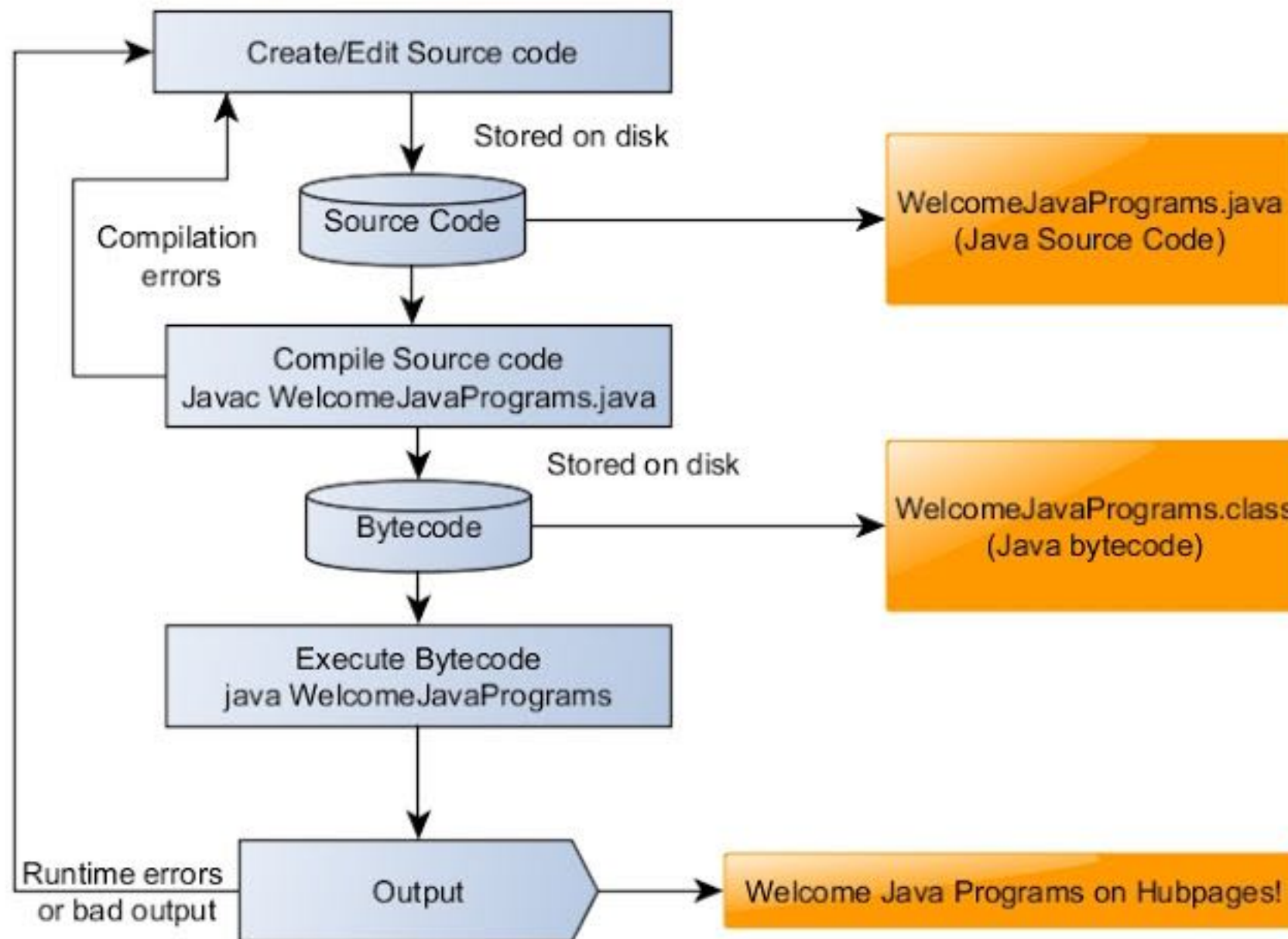
- **Doğrulama**

- ▶ Bytecode doğrulayıcı geçerli olduklarından ve güvenlik kısıtlamalarını ihlal etmediklerinden emin olmak için bytecodeları inceler.

- **Çalıştırma**

- ▶ Java Sanal Makinesi(Java Virtual Machine/JVM) bytecodeları makine diline çevirir.

Tipik bir Java geliştirme ortamı



JRE ve JDK

- **JRE (Java Runtime Environment)** Java ile yazılmış bir programın çalıştırılması için gerekli bileşenleri içeren bir yazılım paketidir (Bir Java Sanal Makinesi içerir)
- **JDK (Java Development Kit)** Java ile uygulama geliştirmek için gerekli araçları içeren bir yazılım paketidir. (javac derleyicisini içerir).
 - JRE, JDK'nın içinde yer alır.
 - Java ile uygulama geliştirmek için JDK kurulumu şarttır.
 - <http://www.oracle.com/technetwork/java/javase/downloads/index.html> adresinden indirip kurabilirsiniz.

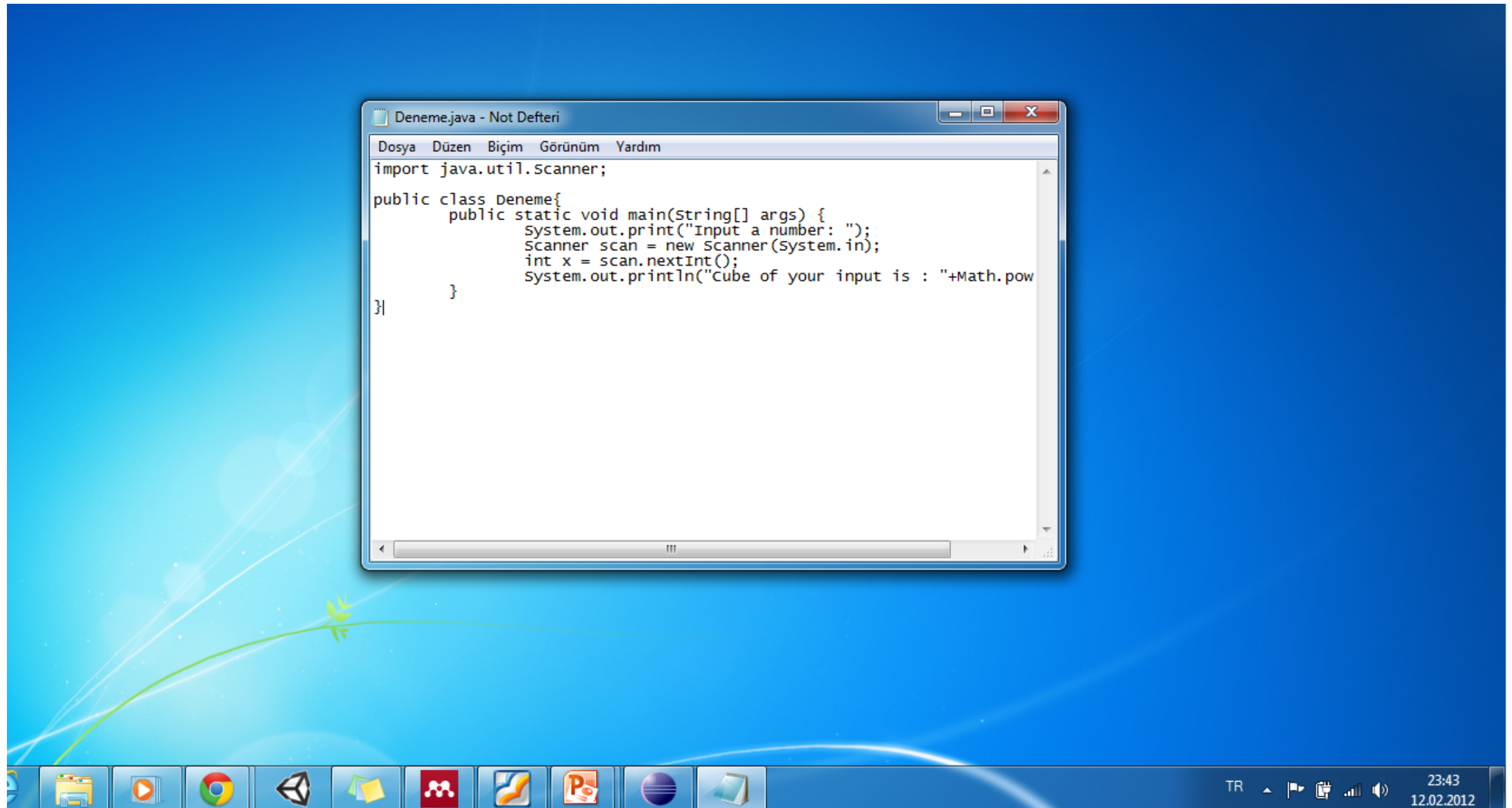
Java Versiyonları

Java

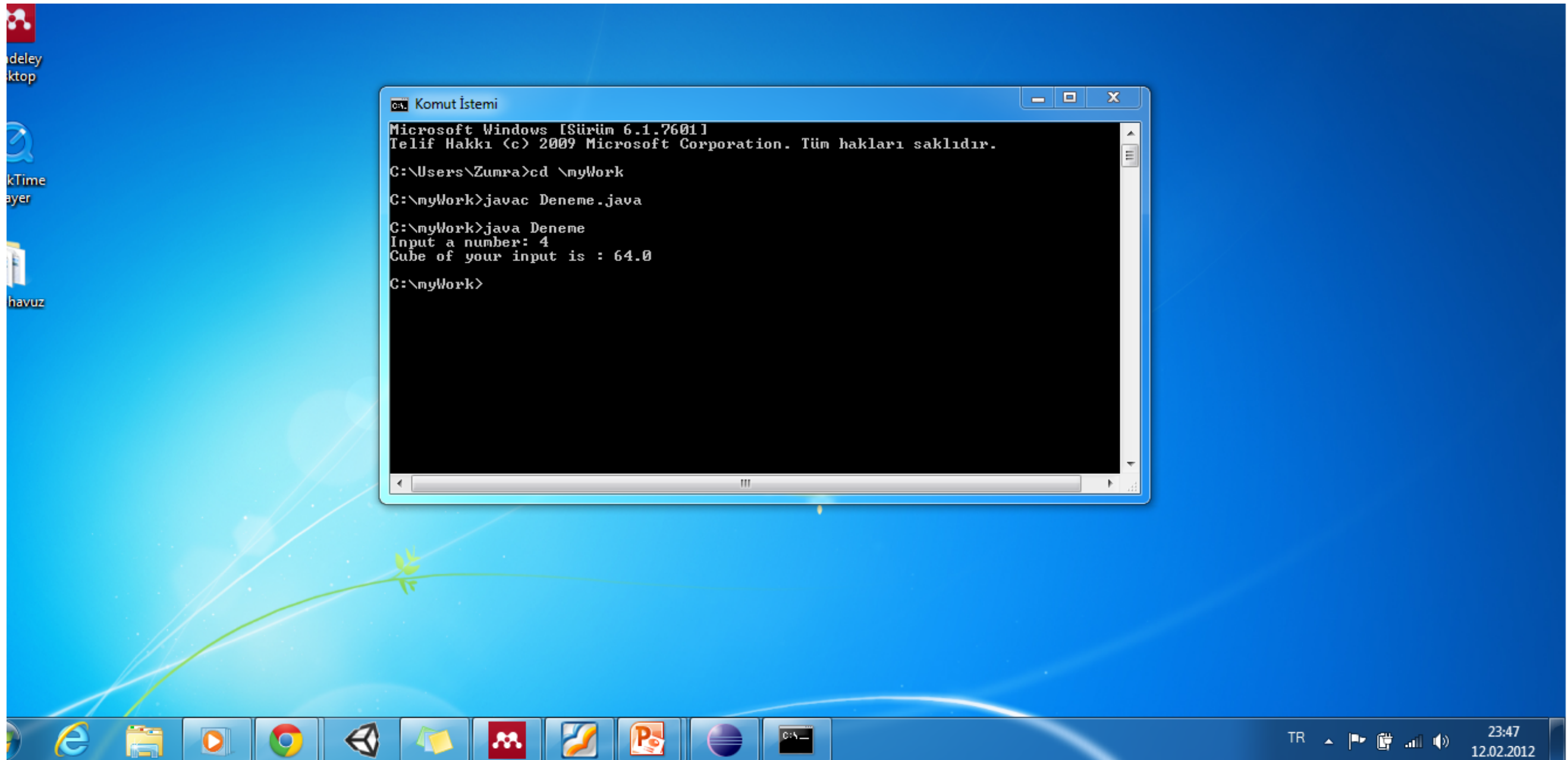
- Java SE (Standard Edition) : Temel sürüm. Kişisel projeler için kullanmaya uygun . Ders için bu sürümü indirip yüklemelisiniz.
- Java EE (Enterprise Edition) Kurumsal projeler için kullanılan sürüm. Kurumsal projeler için gerekli bir takım araçları barındırır.

Java kodu nasıl çalıştırılır?

Java kodunu çalıştırmak için bir metin editörü (örneğin notepad) ve komut penceresi yeterli.



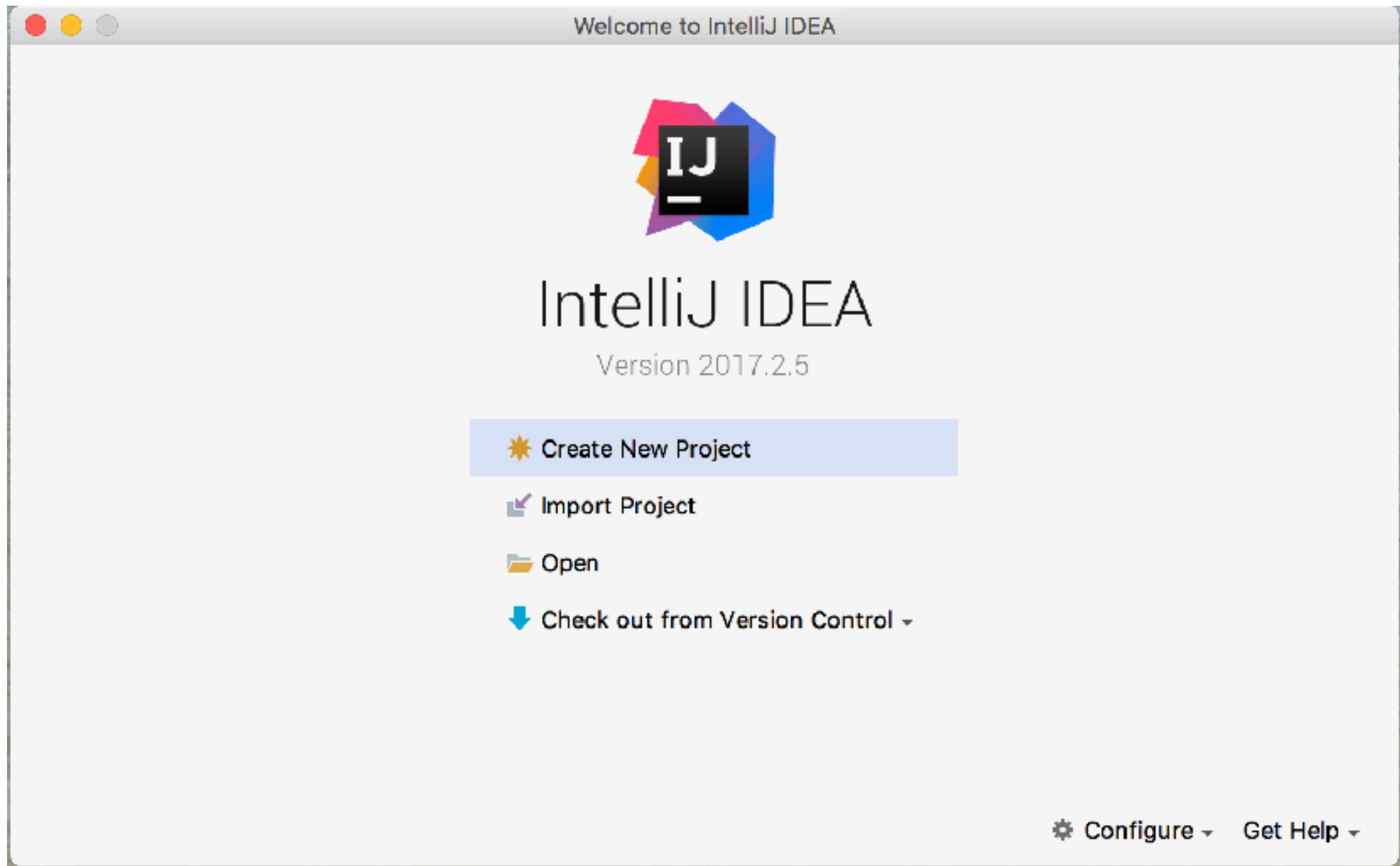
Java kodu nasıl çalıştırılır?



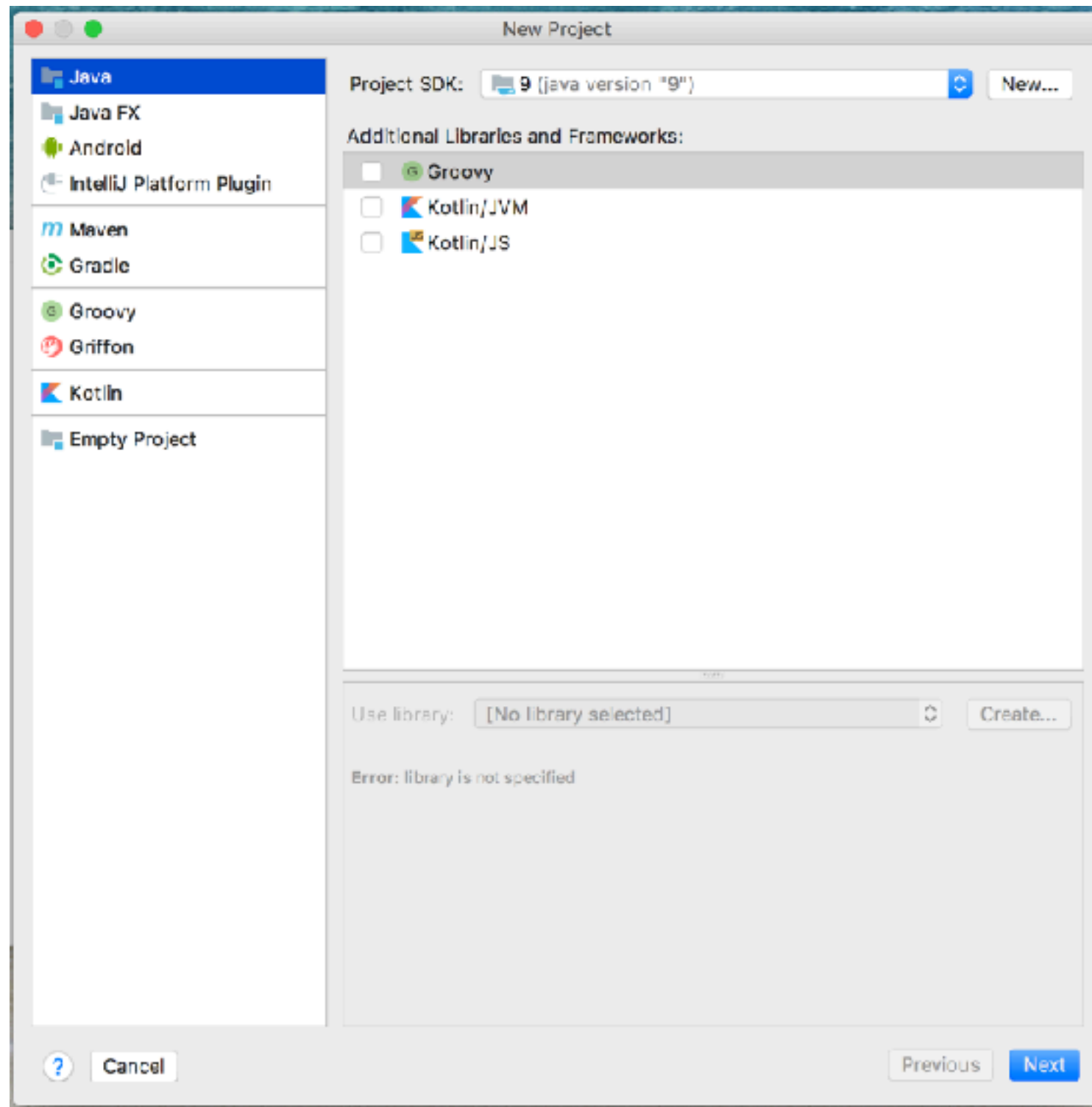
IDE(Integrated Development Environment / Tümleşik Geliştirme Ortamı)

- **Tümleşik geliştirme ortamı (IDE)** , bilgisayar programcılarının hızlı ve rahat bir şekilde yazılım geliştirebilmesini sağlayan araçlara sahip bir yazılım uygulamasıdır.
- Bir IDE genel olarak,
 - Programlama diline göre sözdizimi renklendirmesi yapabilen kod yazım editörü ve
 - Tümleşik bir derleyici, yorumlayıcı ve hata ayıklayıcıya sahiptir, böylece kod yazım sürecini hızlandırır.
- Belli başlı Java ideleri
 - IntelliJ
 - Eclipse
 - Netbeans
 - JCreator

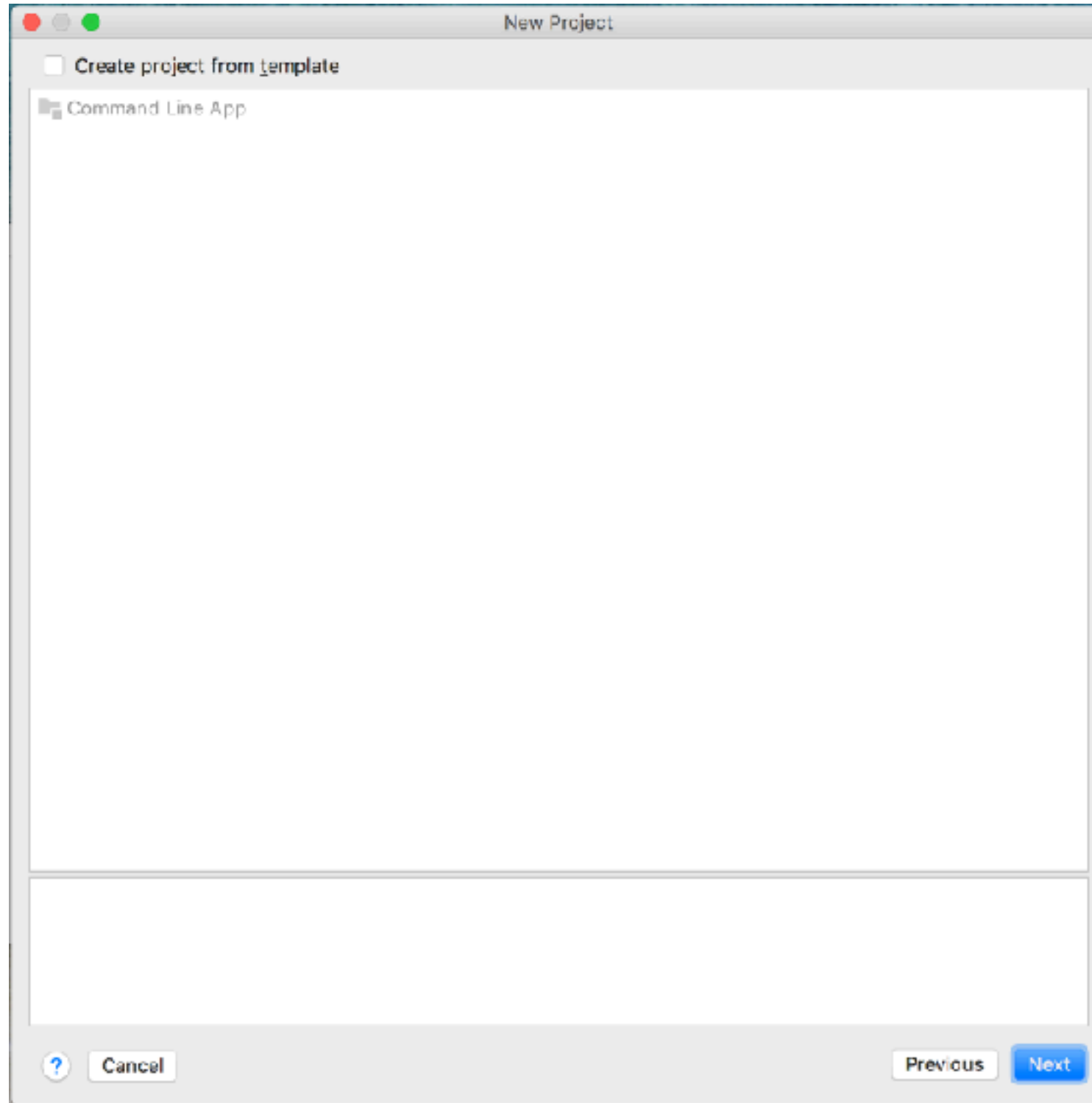
Java ile ilk program : IntelliJ ile proje oluřturmak



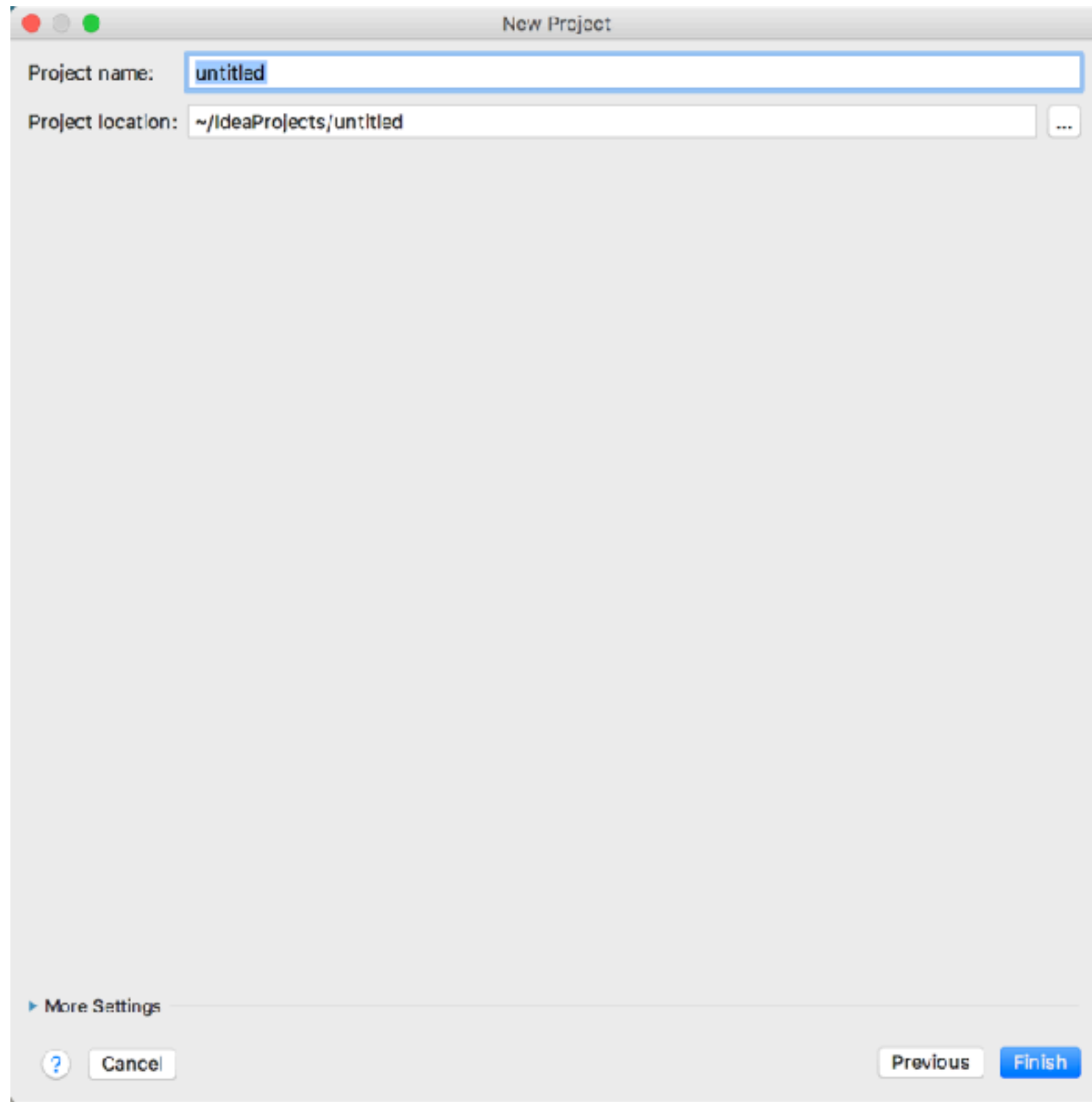
Java ile ilk program : IntelliJ ile proje oluřturmak



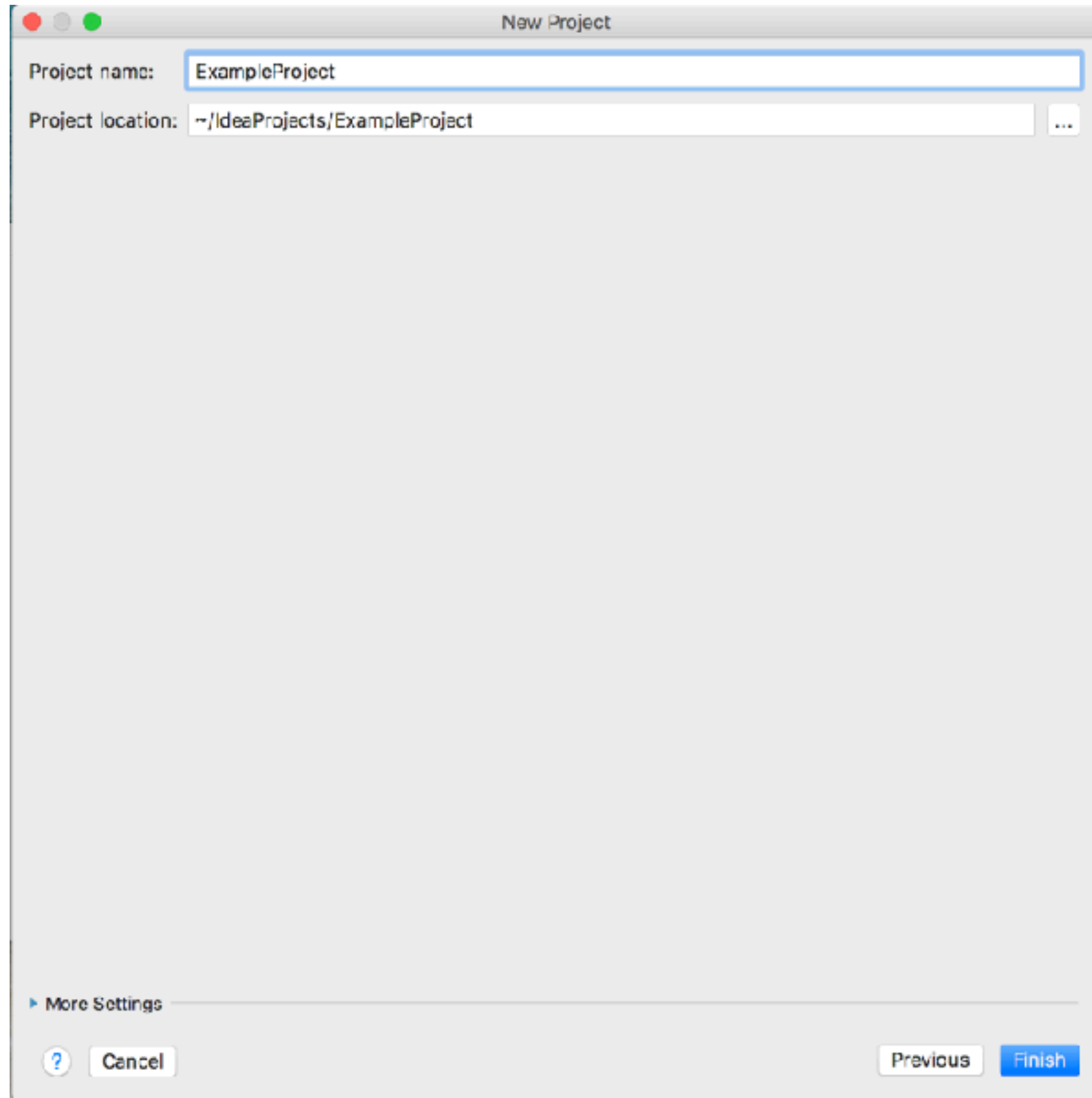
Java ile ilk program : IntelliJ ile proje oluřturmak



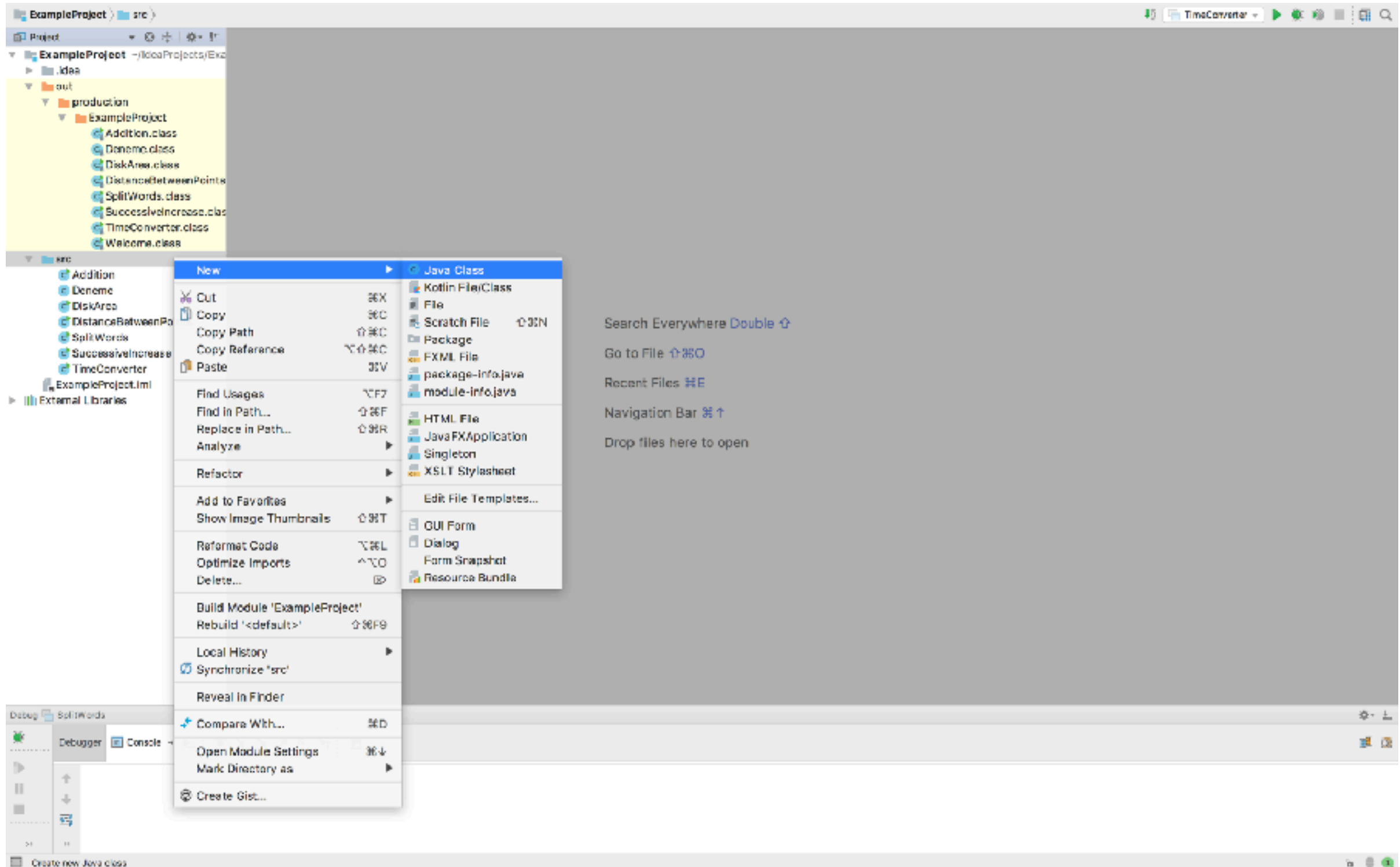
Java ile ilk program : IntelliJ ile proje oluşturmak



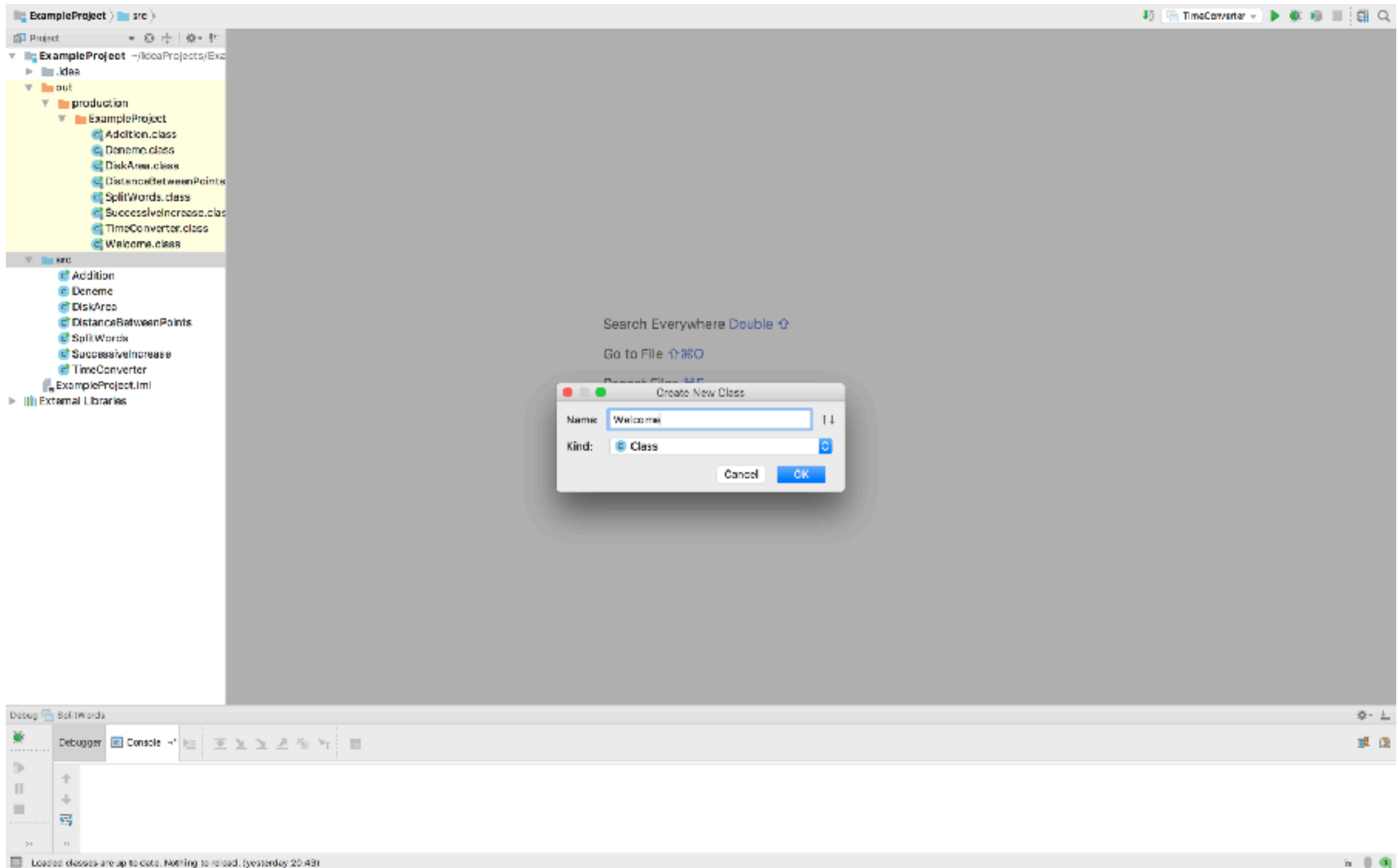
Java ile ilk program : IntelliJ ile proje oluřturmak



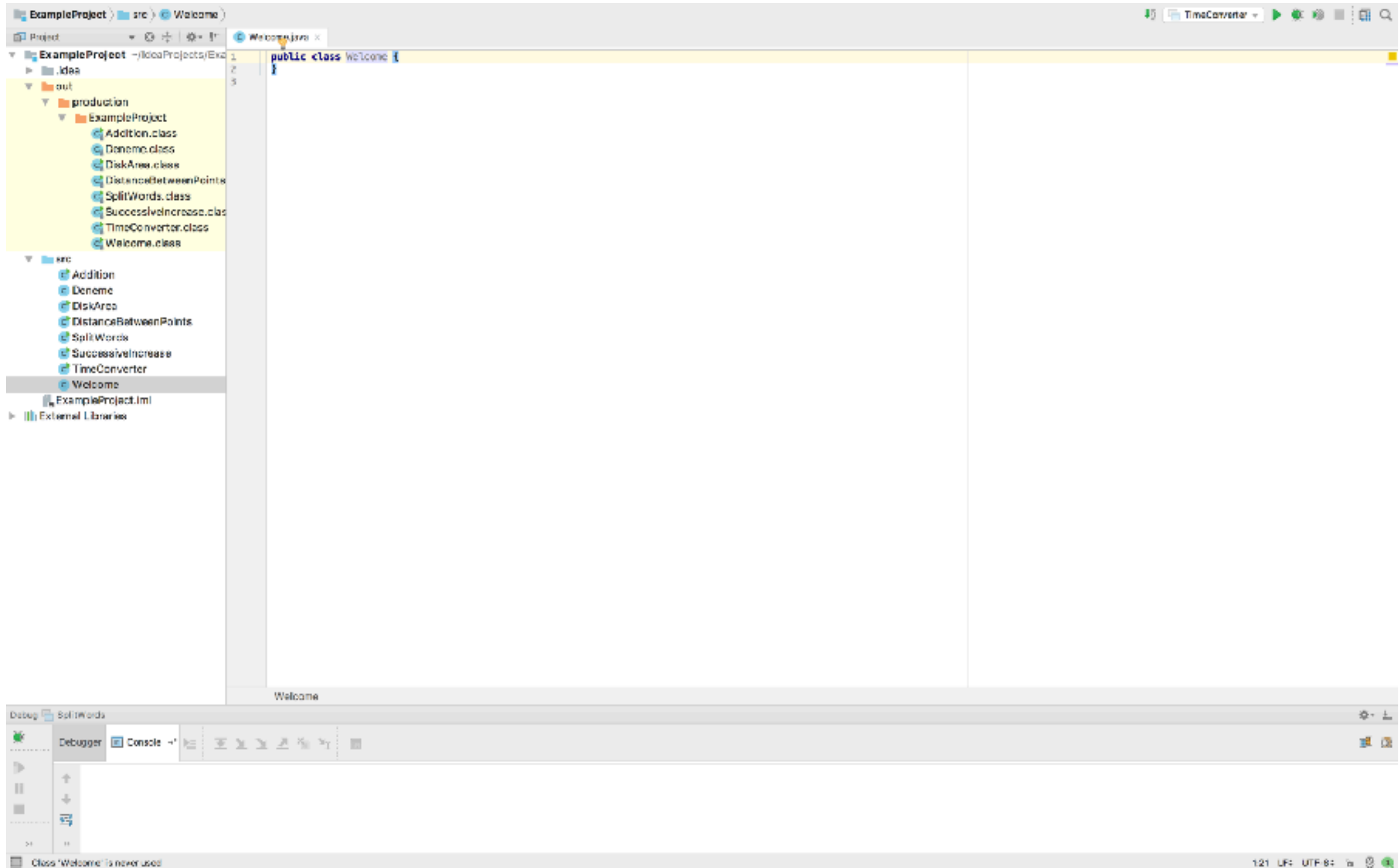
Java ile ilk program : Bir java dosyası oluşturmak



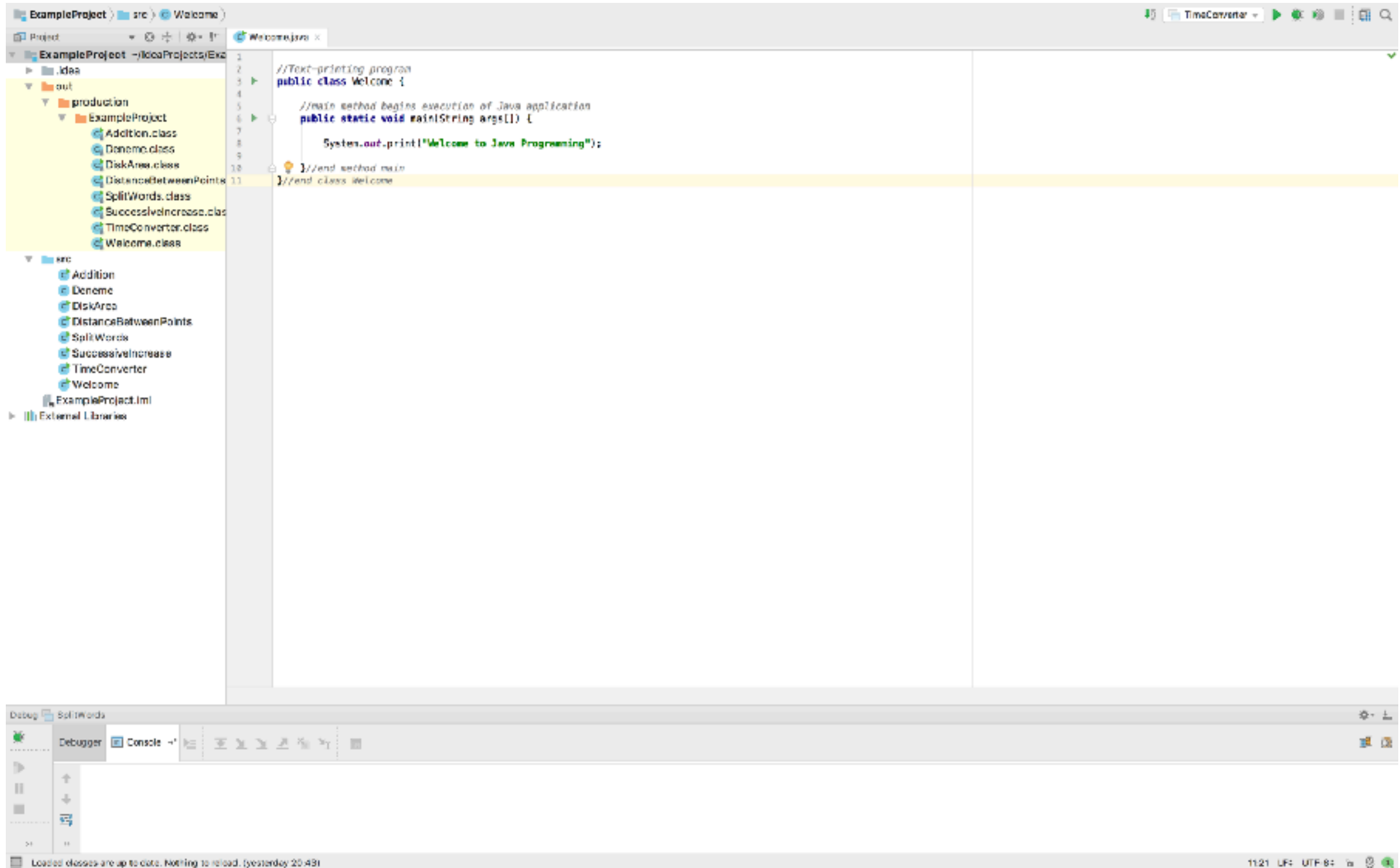
Java ile ilk program : Bir java dosyası oluşturmak



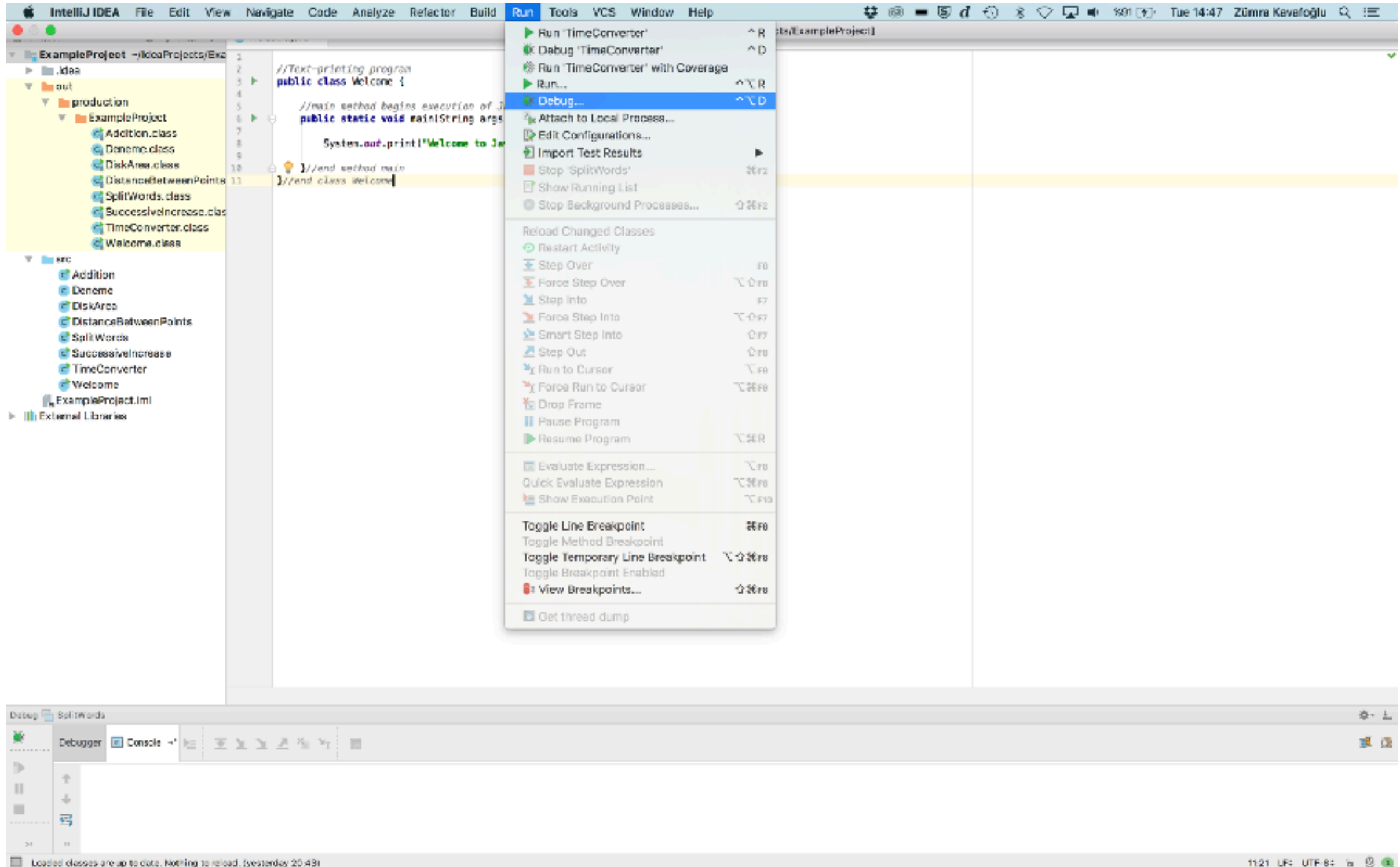
Java ile ilk program : Bir java dosyası oluşturmak



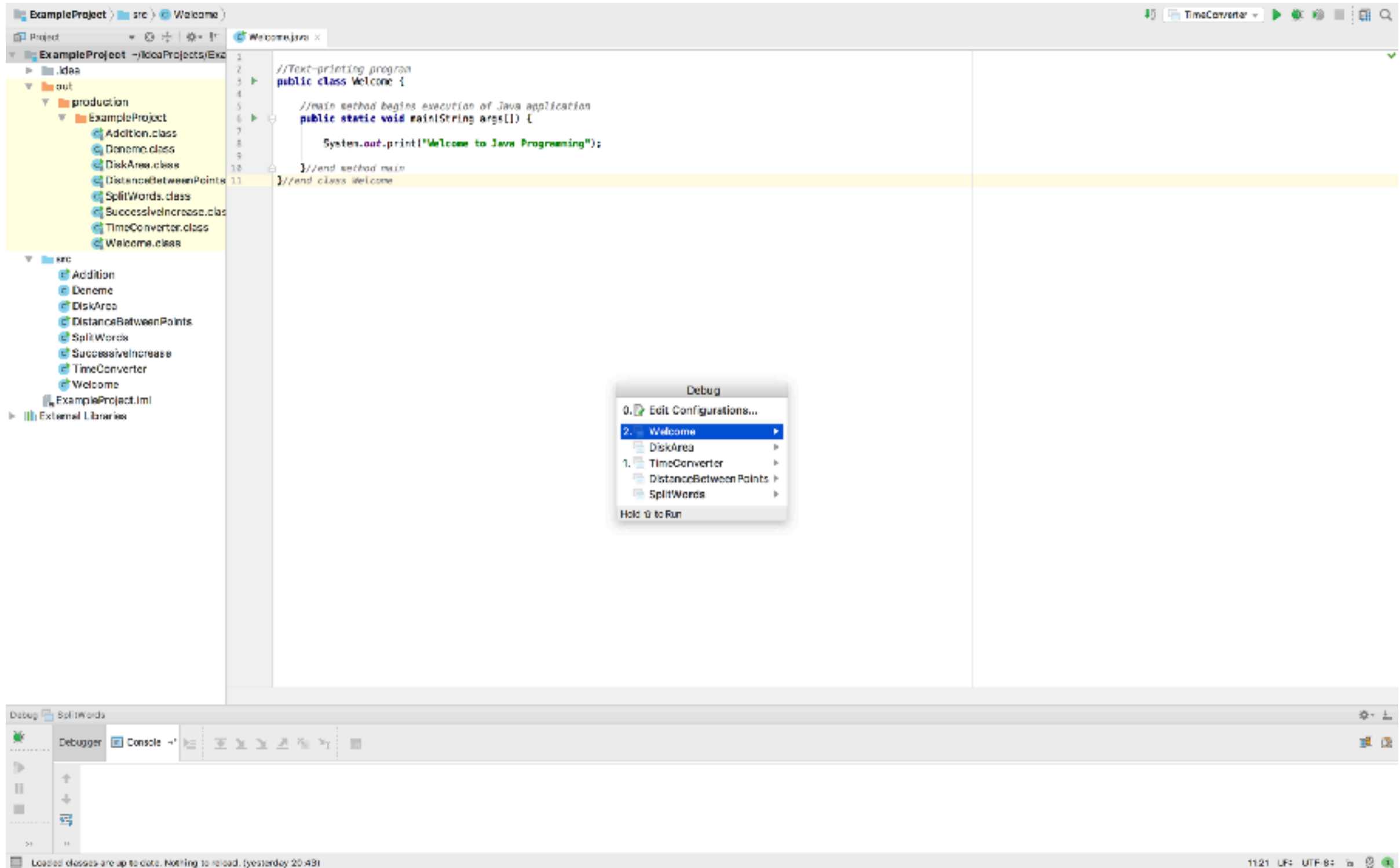
Java ile ilk program : Bir java dosyası oluşturmak



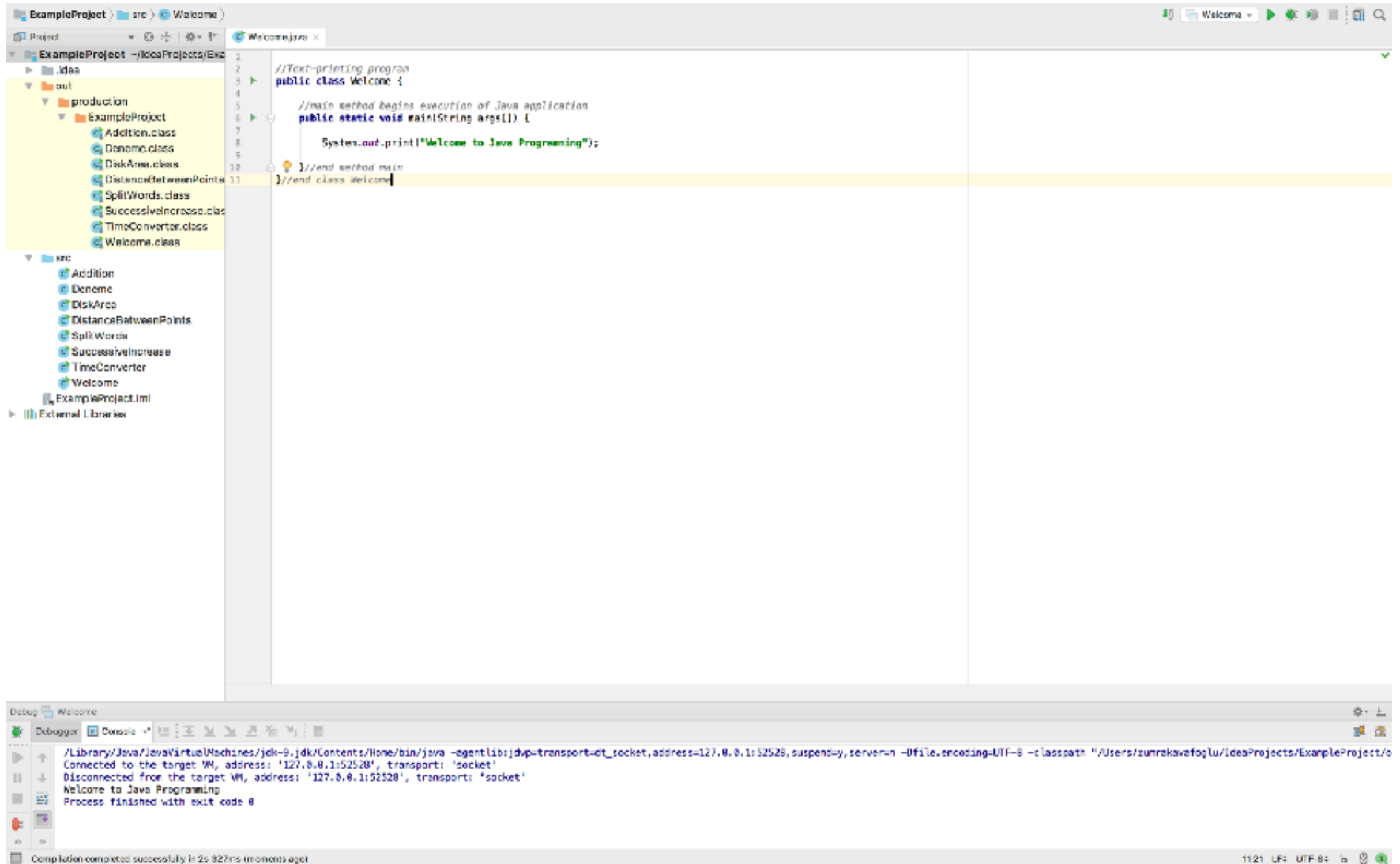
Java ile ilk program : Debug ile programı çalıştırmak



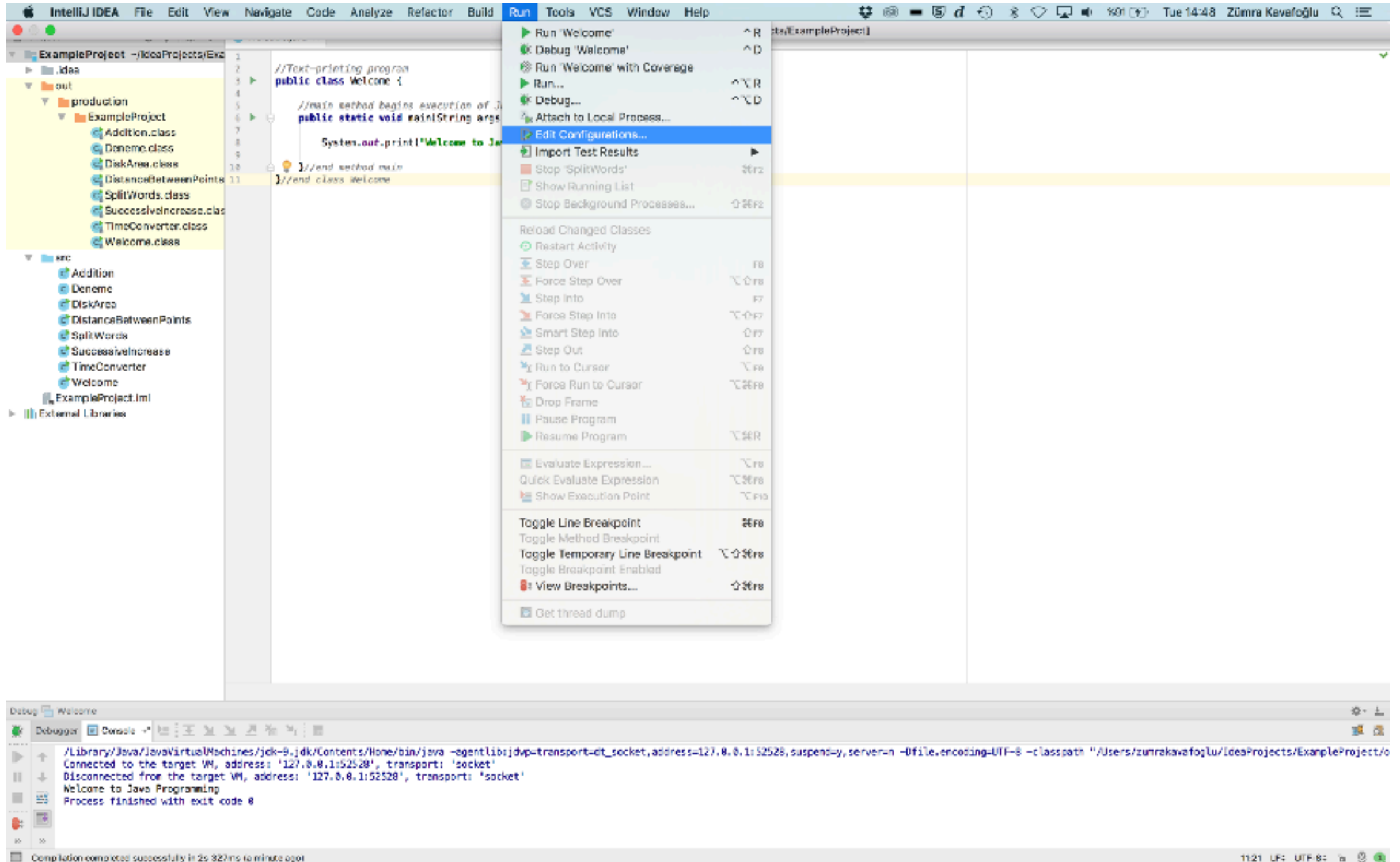
Java ile ilk program : Debug ile programı çalıştırmak



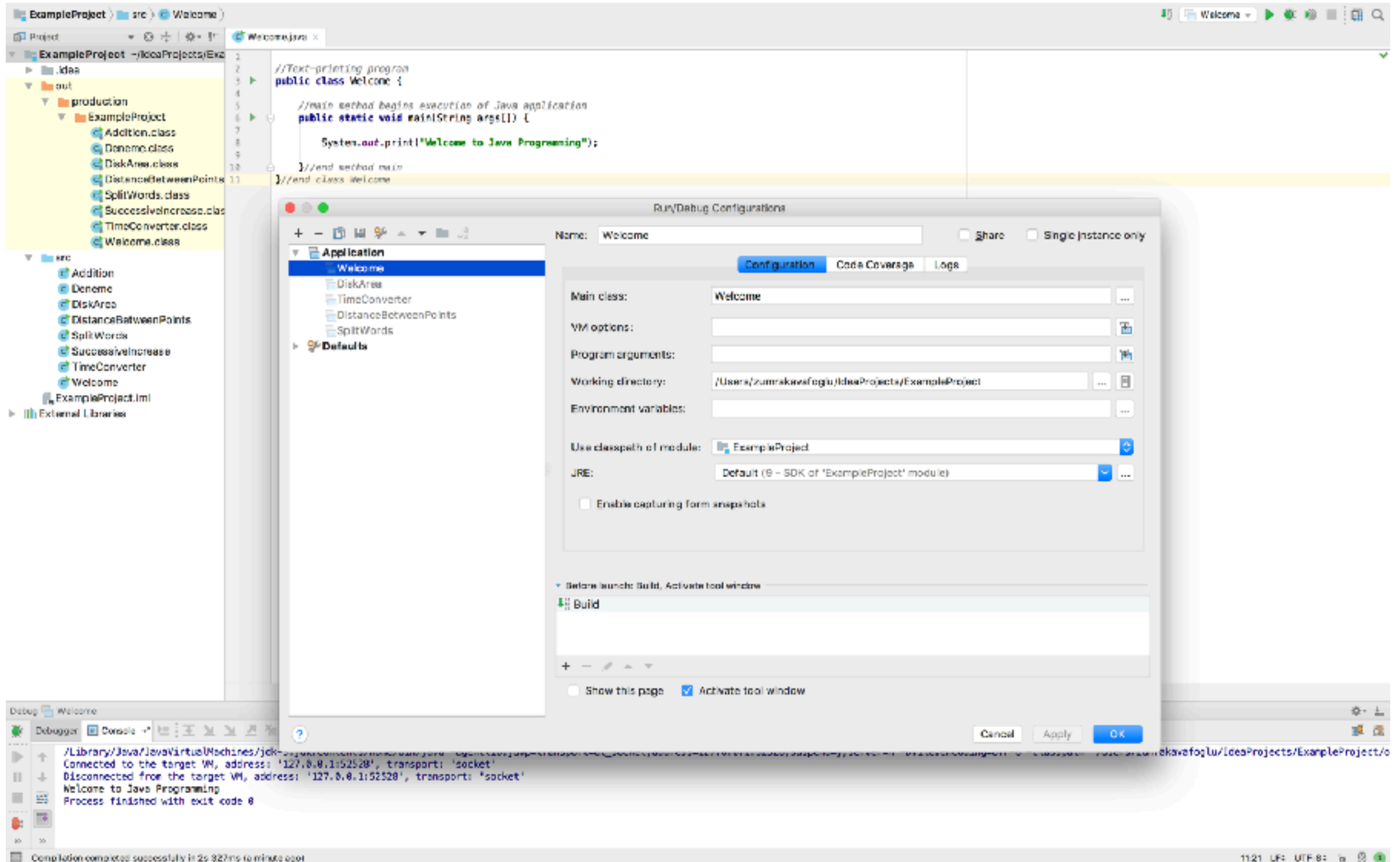
Java ile ilk program : Debug ile programı çalıştırmak



Java ile ilk program : Sınıf konfigürasyonlarını düzenlemek



Java ile ilk program : Sınıf konfigürasyonlarını düzenlemek



Java ile ilk program

Problem: Konsola bir mesaj yazdırmak

Neler öğreneceğiz:

- Yorum satırları
- Sınıf deklarasyonu
- İsimlendirme kuralları
- Anahtar sözcükler
- main metodu
- İfadeler
- Derleme Hatası

Java ile ilk program : Ekrana bir metin satırı yazdırmak

```
//Text-printing program
public class Welcome {

    public static void main(String args[]){
        //main method begins execution of Java application
        System.out.println("Welcome to Java Programming");
    }//end method main
}//end class Welcome 1
```

Çıktı

```
Welcome to Java Programming!
```


Java ile ilk program : Yorum Satırları

```
//Text-printing program
public class Welcome {

    public static void main(String args[]){
        //main method begins execution of Java application
        System.out.println("Welcome to Java Programming");
    }//end method main
}//end class Welcome 1
```

- Bir java programında yorumlar kod hakkında açıklamalar yazmak için kullanılır.
- Java derleyicisi yorumları yok sayar.
- Tek satırlık yorumlar:
//Bu bir yorumdur.
- Birden fazla satırlı yorumlar:
/* Bunlar yorumlardır
Bunlar yorumlardır */
- Her programın başına o programın ne yaptığını yorum olarak not etmeyi alışkanlık haline getirmelisiniz.
- Her sınıf deklarasyonunun başına adınız, tarihi ve sınıfın genel görevini açıklamaya çalışın.
-

Java ile ilk program : Sınıf ismi

```
//Text-printing program
public class Welcome {

    public static void main(String args[]){
        //main method begins execution of Java application
        System.out.println("Welcome to Java Programming");
    }//end method main
}//end class Welcome 1
```

İsimlendirme Kuralları

- İsmi yazılan program hakkında fikir vermesi gerekir.
- İlk karakter bir rakam olamaz.
- İsim sadece harflerden, rakamlardan, alt çizgi _ ve dolar \$ işaretlerinden oluşabilir.
- İsim genellikle büyük harfle başlar ve isimdeki her bir sözcük büyük harfle başlar. (ExampleProgram)
- İsimde boşluk olmaz.
- İsim javanın anahtar sözcüklerinden biri olamaz.
- Maksimum karakter sayısı sınırsızdır ancak genelde mümkün olduğunca az karakter kullanılmalıdır.
- İsimlerde Türkçe karakterler kullanılmaz.
- Java büyük harf - küçük harf duyarlıdır(case sensitive).

Java'nın anahtar sözcükleri

abstract	continue	for	new	switch
assert	default	goto	package	synchronize
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const*	float	native	super	while

Bazı yanlış isimler

2ab3 rakamla başlıyor

e*6 * işareti içeriyor

while java anahtar sözcüğü

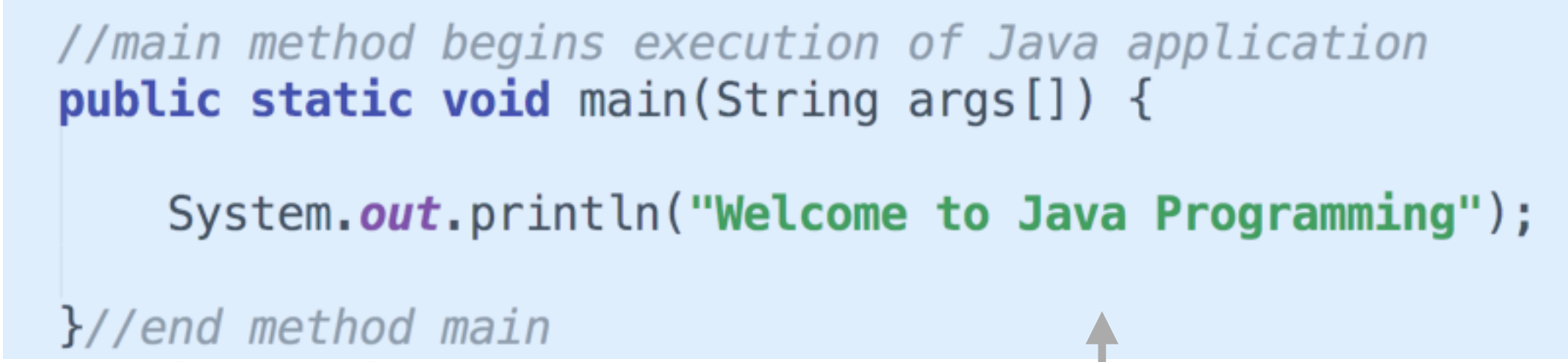
Java ile ilk program : Sınıf deklarasyonu

```
1
2  //Text-printing program
3  ▶ public class Welcome {
4
5      //main method begins execution of Java application
6  ▶  public static void main(String args[]) {
7
8      System.out.println("Welcome to Java Programming");
9
10     }//end method main
11 }//end class Welcome
12
```

- Java kodları(genel olarak) bir sınıfın içinde tanımlanmalıdır.
- Bu yüzden bütün programlar bir sınıf bloğu içerir.
- Her sınıfın ismini içeren bir deklarasyonu yapılmalıdır.
- Dosya adı sınıf adıyla aynı olmalıdır ve dosyanın uzantısı .java olmalıdır.
✓Örneğin bu sınıfı içeren dosyanın adı Welcome.java olmalıdır.

Java ile ilk program : Sınıf gövdesi ve parantezler

```
1
2 //Text-printing program
3 ▶ public class Welcome {
4
5     //main method begins execution of Java application
6 ▶ public static void main(String args[]) {
7
8     System.out.println("Welcome to Java Programming");
9
10    }//end method main
11 }//end class Welcome
12
```



Sınıf gövdesi

- Sınıf gövdesi tanımlanan sınıfın içerdiği kodlar bütününe verilen isimdir.
- Sınıf gövdesi süslü parantez { ile açılmalı ve süslü parantez } ile kapatılmalıdır.
- Sınıfları ileriki derslerde detaylı olarak öğreneceğiz, şimdilik yazacağınız programların içinde yer alacağı sınıflar tanımlamayı unutmamanız yeterli!

Java ile ilk program : main metodu

```
1
2  //Text-printing program
3  ▶ public class Welcome {
4
5      //main method begins execution of Java application
6  ▶  public static void main(String args[]) {
7
8      System.out.println("Welcome to Java Programming");
9
10     } //end method main
11 } //end class Welcome
12
```

Metot gövdesi

- Her java uygulaması içinde main metodu olan bir sınıf içermelidir.
- Uygulamalar main metoduyla çalışmaya başlar.
- main metodunun gövdesi { ile açılmalı ve } ile kapatılmalıdır.(Bu tüm metotlar için geçerlidir)
- Metotları ileriki derslerde detaylı olarak öğreneceğiz.
- Şimdilik her uygulamada yukarıdaki örnekte sarı ile taralı satırları kopyalayabilirsiniz.

Java ile ilk program : İfadeler

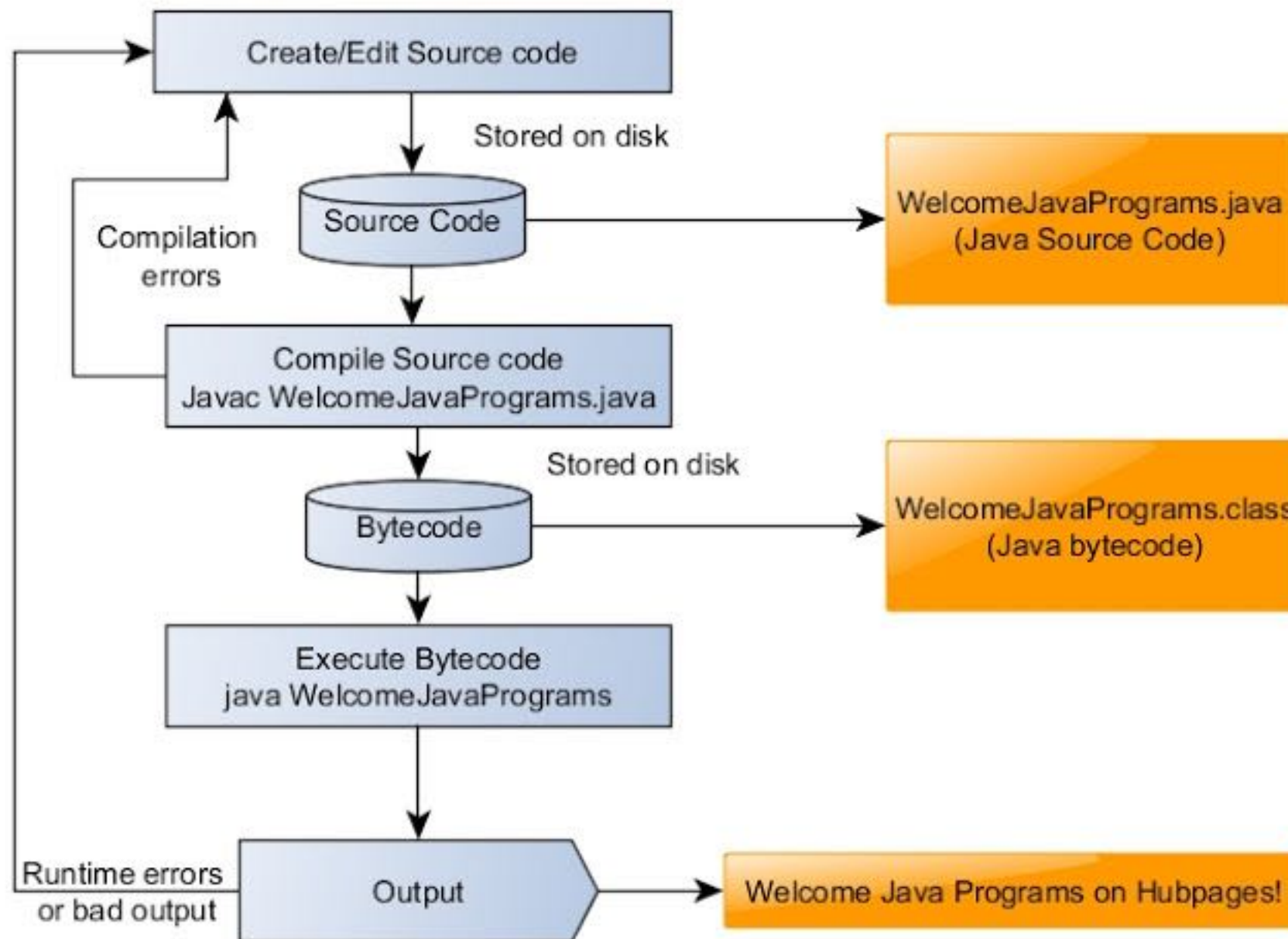
```
1
2  //Text-printing program
3 ▶ public class Welcome {
4
5      //main method begins execution of Java application
6 ▶      public static void main(String args[]) {
7
8          System.out.println("Welcome to Java Programming");
9
10         }//end method main
11     }//end class Welcome
12
```

- İfadeler(Statement) bilgisayara istenilen eylemi/işlemi yapması için verilen talimatlardır.
- İşaretli ifade ekrana (komut penceresine) çift tırnak içindeki metnin yazılması talimatını verir.
- Her ifade noktalı virgül ile bitmelidir.

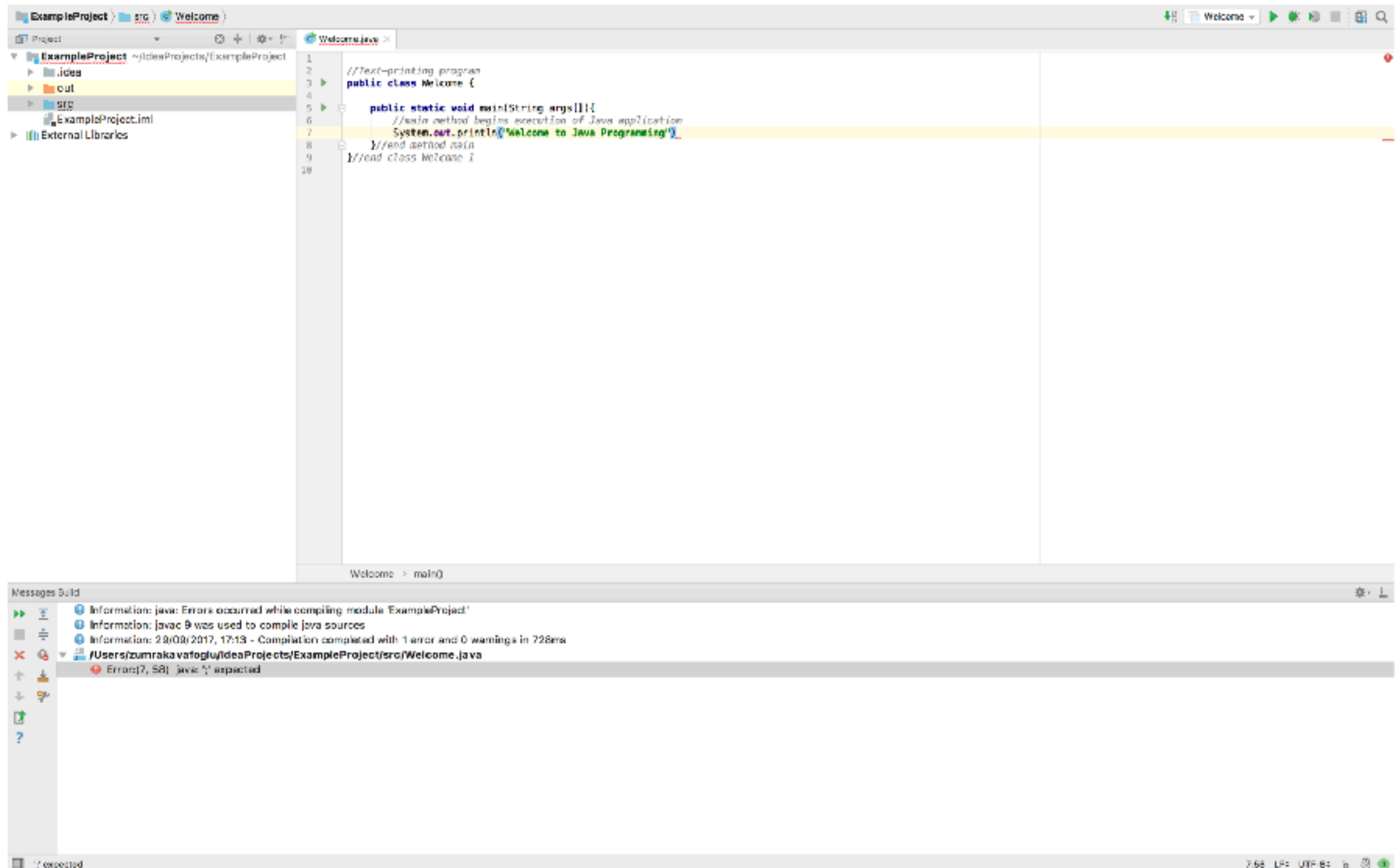
Derleme Hatası (Syntax Error / Compiler Error)

- Bir programlama dilinin syntax'ı o dilde doğru bir program yazmak için gerekli kuralları belirler.
- Derleyici Java'nın dil kurallarını(yani syntax'ini) ihlal eden bir kodla karşılaştığında derleme hatası ortaya çıkar.
- Bu durumda, derleyici bir .class dosyası üretmez, bunun yerine programcının yanlış kodu bulmasını ve düzeltmesine yardımcı olacak bir hata mesajı gösterir.
- Tüm derleme hataları düzeltilmeden program çalıştırılamaz.

Programı Derleme ve Çalıştırma



Derleme Hatası



Derleme Hatası

Program

```
1
2 //Text-printing program
3 public class Welcome {
4
5     public static void main(String args[]){
6         //main method begins execution of Java application
7         System.out.println("Welcome to Java Programming");
8     }//end method main
9 }//end class Welcome
10
```

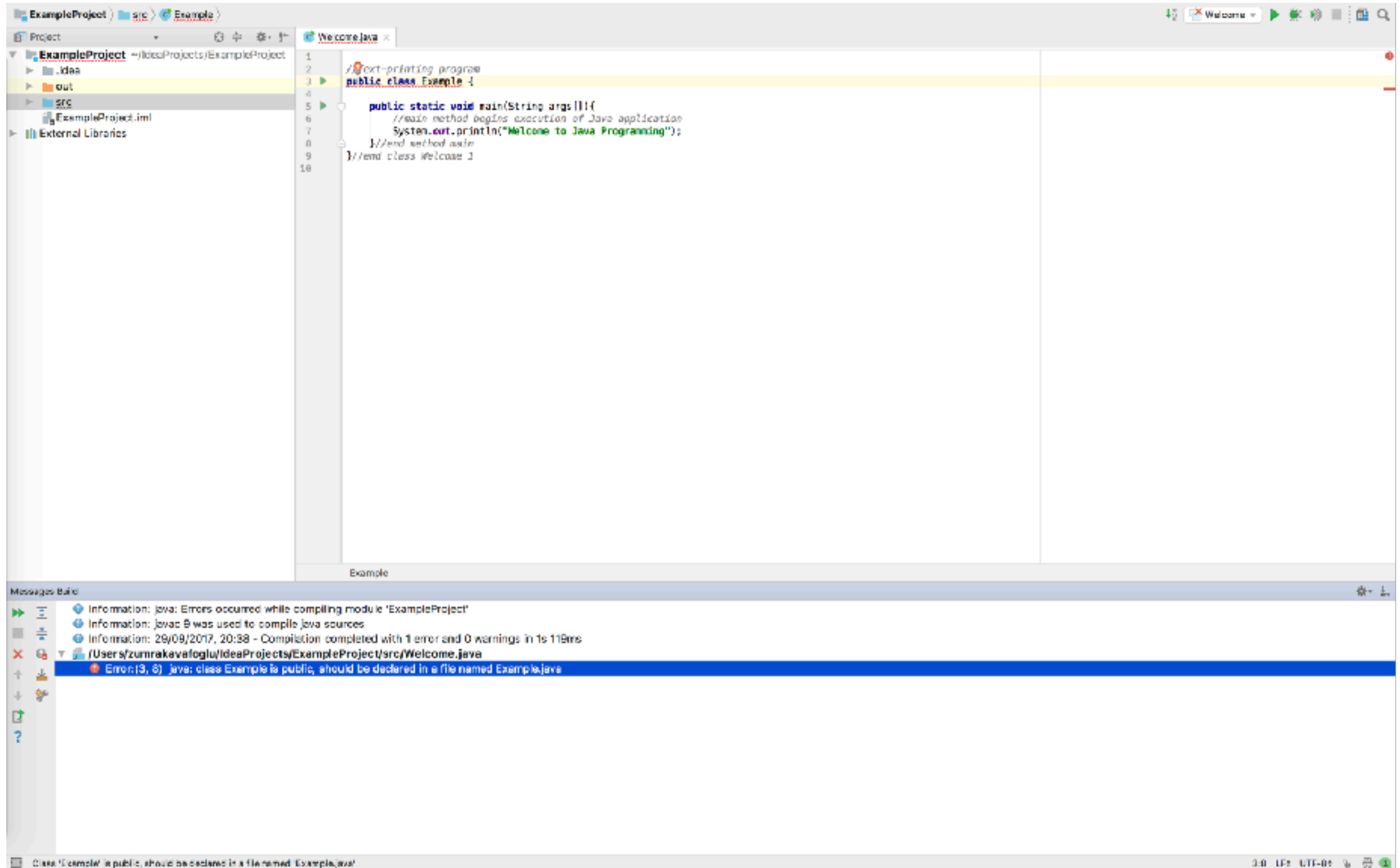
Hata Mesajı

- Information: java: Errors occurred while compiling module 'ExampleProject'
- Information: javac 9 was used to compile java sources
- Information: 29/09/2017, 17:13 - Compilation completed with 1 error and 0 warnings in 728ms

/Users/zumrakavafoglu/IdeaProjects/ExampleProject/src/Welcome.java

Error:(7, 58) java: ';' expected

Derleme Hatası



Derleme Hatası

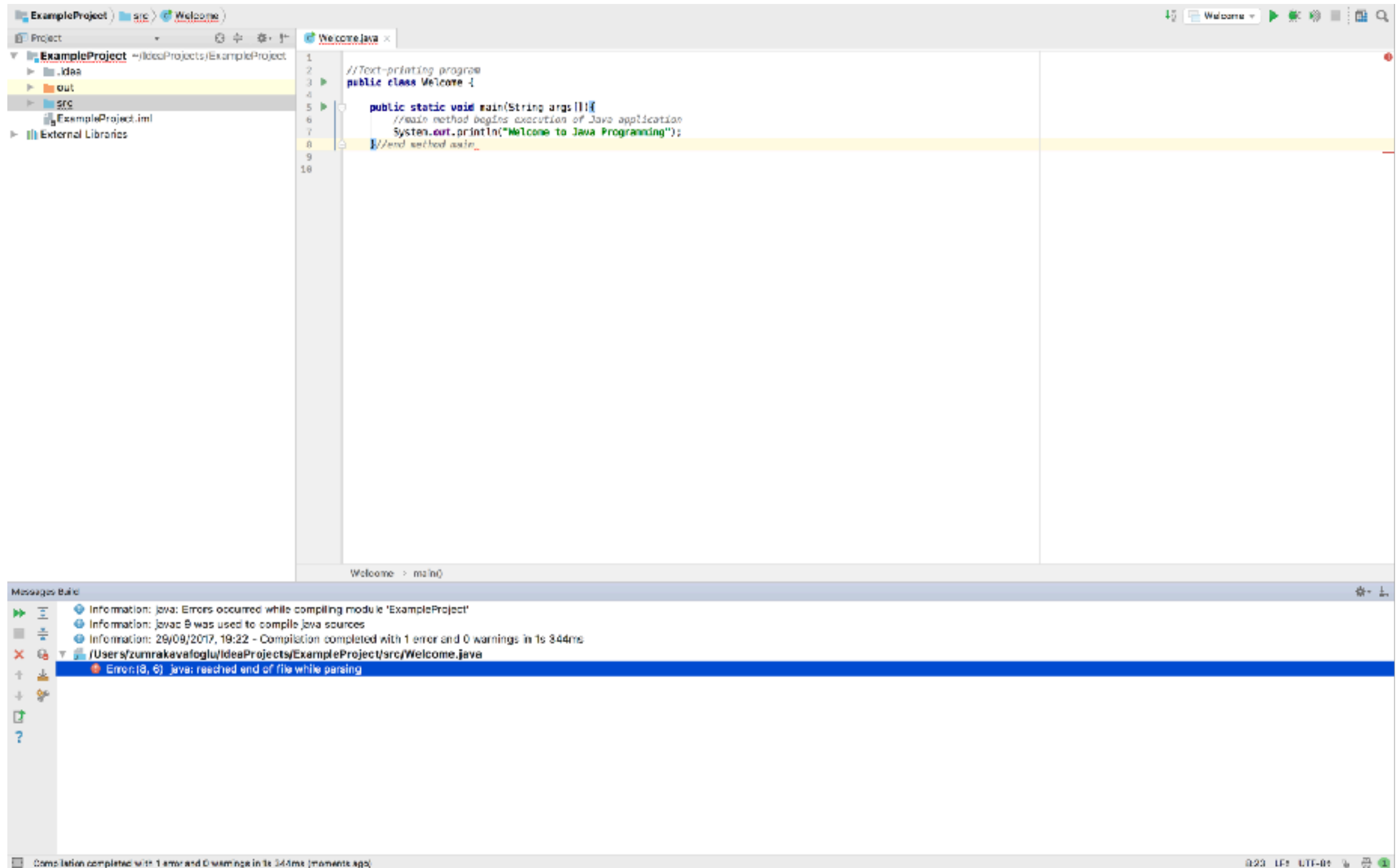
Program

```
Welcome.java x
1
2 //Text-printing program
3 public class Example {
4
5     public static void main(String args[]){
6         //main method begins execution of Java application
7         System.out.println("Welcome to Java Programming");
8     }//end method main
9 }//end class Welcome 1
10
```

Hata Mesajı

- Information: java: Errors occurred while compiling module 'ExampleProject'
- Information: javac 9 was used to compile java sources
- Information: 29/09/2017, 20:38 - Compilation completed with 1 error and 0 warnings in 1s 119ms
- ▼ /Users/zumrakavafoglu/IdeaProjects/ExampleProject/src/Welcome.java
 - Error:(3, 8) java: class Example is public, should be declared in a file named Example.java

Derleme Hatası



Derleme Hatası

- Hata mesajı her zaman programdaki hatayı tam olarak ifade etmeyebilir. Ancak bu tip hatalarla karşılaştıkça hangi mesajın hangi hatadan doğmuş olabileceğini öğrenmeye başlayacaksınız.
- Programdaki hata her zaman gösterilen satırda olmayabilir.

Program

```
1
2 //Text-printing program
3 public class Welcome {
4
5     public static void main(String args[]){
6         //main method begins execution of Java application
7         System.out.println("Welcome to Java Programming");
8     }//end method main
9
```

Hata Mesajı

- Information: java: Errors occurred while compiling module 'ExampleProject'
- Information: javac 9 was used to compile java sources
- Information: 29/09/2017, 19:22 - Compilation completed with 1 error and 0 warnings in 1s 344ms

▼ /Users/zumrakavafoglu/IdeaProjects/ExampleProject/src/Welcome.java

❗ Error:(8, 6) java: reached end of file while parsing

Bir sonraki derse kadar

- Kişisel bilgisayarlarınıza JDK ve IntelliJ kurunuz.
- Konsola “Merhaba Dünya” yazdıran ilk programınızı yazınız.