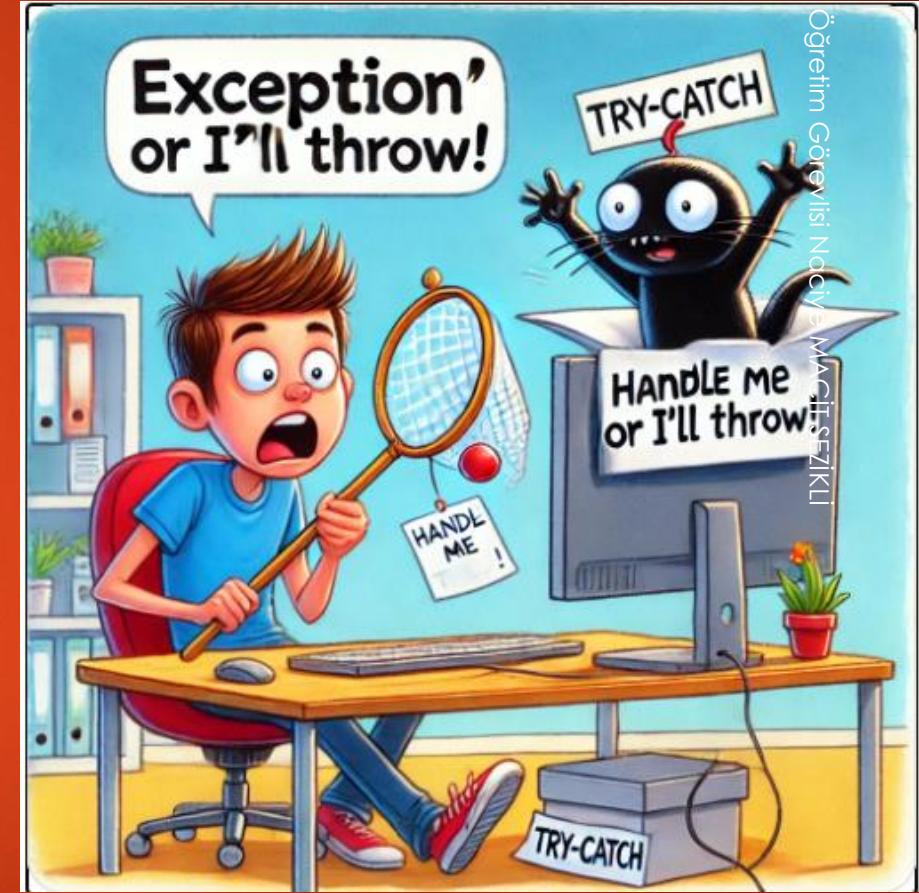


# exceptions

İSTİSNALAR



Öğretim Görevlisi Naciye MAGİSSEZİKİLİ

# Exceptions(İSTİSNALAR) Nedir?

- ▶ Exception, Java'da hataları temsil eden bir sınıfıtır.
- ▶ İçerisinde hata kodları, bilgi mesajları ve ilgili metodlar bulunur
- ▶ Programın çalışma zamanında (runtime) karşılaşabileceği hataları ele alır.
- ▶ Bu hatalarla başa çıkmak için kullanılan bir mekanizmadır
- ▶ Exceptions, kodunuzun beklenmedik durumlarla daha güvenli ve düzenli bir şekilde başa çıkmasını sağlar.

# Exeptions(İSTİSNALAR) Nedir?

- ▶ Bir exception, programın normal akışını bozan olağan dışı bir durumdur. Java'da bu durumlar Exception sınıfından türeyen nesnelerle temsil edilir.

## Örneğin:

- Bir dizinin sınırını aşmaya çalışmak (ArrayIndexOutOfBoundsException)
- Bir sayıyı sıfıra bölmek (ArithmeticException)
- Geçersiz bir dosya yolunu açmaya çalışmak (FileNotFoundException)

# Exceptions(İSTİSNALAR) Nedir?

## ► Kod Çalışma Süreci:

- Bir sınıfta tanımlanan bir metot çağrıldığında kod çalışmaya başlar.
- Kod satır satır okunurken bir hata oluşursa, bir exception durumu meydana gelir.

## ► Öngörülebilir ve Öngörülemeyen Hatalar:

- Geliştiriciler, bilinen hata durumlarını önceden öngörerek kodlarını bu hataları ele alacak şekilde düzenlemelidir.
- Ancak, kullanıcılar programı kullanırken her zaman beklenmeyen durumlar oluşabilir.
- Bir exception meydana geldiğinde, kodun yürütülmesi kesintiye uğrar ve hata bilgileri genellikle log dosyalarına kaydedilir.

# Exceptions(İSTİSNALAR) Nedir?



# Exeptions(İSTİSNALAR) Nedir?

## ► **Exception Handling (Hata Yönetimi):**

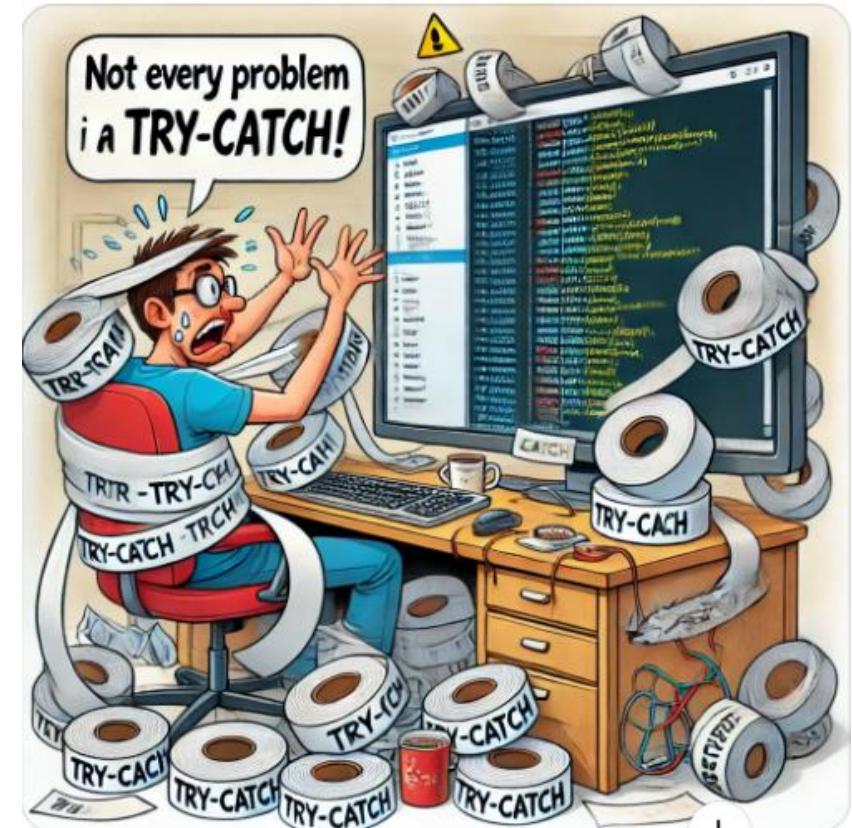
- Hata oluştuğunda, sebebinin tespit etmek ve çözüm üretmek gereklidir.
- Hata o an çözülemiyorsa bile, sistemin çökmesini önlemek ve çalışmaya devam etmesini sağlamak için alternatif çözümler geliştirilmelidir.(kesinti olmadan)

## ► **Exception Yönetiminin Amacı:**

- Sistemin güvenilirliğini artırmak.
- Kullanıcı deneyimini iyileştirmek.
- Programın beklenmedik durumlara karşı dayanıklı olmasını sağlamak.

# Exceptions(İSTİSNALAR) Nedir?

- ▶ Kodun beklenildiği gibi çalışmaması durumunda, kullanıcı deneyiminin aksamaması için işlemleri bir try-catch bloğunda ele almak yaygın bir yaklaşımdır.
- ▶ Bu yöntem, hataların kontrol altına alınmasını ve programın çökmesini önlemeyi amaçlar.
- ▶ Ancak, tüm kodları rastgele try-catch bloklarına sarmak doğru bir yöntem değildir.
- ▶ Gereksiz try-catch kullanımı, hataların maskelenmesine ve gerçek sorunların gizlenmesine neden olabilir.
- ▶ try-catch blokları, yalnızca belirli ve anlamlı hata senaryoları için dikkatli bir şekilde kullanılmalıdır.



# Java Exception Türleri



Throwable

Exception

Error

Runtime  
Exception  
Other  
Exception

## Checked Exception (Derleme Zamanı Hataları)

- Derleme (compile) zamanında kontrol edilen istisnalardır.
- Kodunuzda bu tür istisnalar için ya bir try-catch bloğu kullanmalı ya da metot imzasına throws anahtar kelimesi eklemelisiniz.

Örnekler:  
•IOException  
•SQLException  
•FileNotFoundException

## Unchecked Exceptions (Runtime Hataları)

Derleme zamanında kontrol edilmezler; çalışma zamanı sırasında oluşurlar.

Genellikle programlama hatalarından kaynaklanır.

Örnekler:  
•ArithmaticException (0'a bölme)  
•NullPointerException (null bir referansa erişim)  
•ArrayIndexOutOfBoundsException (dizi sınırlarını aşmak)

## Errors

Exception sınıfından değil, Error sınıfından türetilirler.  
Sistem seviyesindeki ciddi sorunları temsil eder ve genellikle program tarafından ele alınmaz.

Örnekler:  
•StackOverflowError  
•OutOfMemoryError

Exeptions1.java

Exceptions1.java X

ileriNesneTabanliProgramlama\_Guz\_24\_25 > src > hafta7 > Exceptions1 > calculateModulus(int, int) : void

```
1 package hafta7;
2
3 public class Exceptions1 {
4     public static void main(String[] args) {
5         calculateModulus(10, 0);
6     }
7     public static void calculateModulus(int x, int y) {
8         System.out.println(x%y);
9     }
10}
11<
```

Problems @ Javadoc Declaration Console Terminal



<terminated> Exceptions1 [Java Application] C:\Users\nmacit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (28 Kas 2024 11:53:45 – 11:53:45) [pid: 25224]

Exception in thread "main" java.lang.ArithmetiException: / by zero  
at ileriNesneTabanliProgramlama\_Guz\_24\_25/hafta7.Exceptions1.calculateModulus  
(Exceptions1.java:8)  
at ileriNesneTabanliProgramlama\_Guz\_24\_25/hafta7.Exceptions1.main(Exceptions1.java:5)

Exeptions1.java

Exceptions1.java X

ileriNesneTabanlıProgramlama\_Guz\_24\_25 > src > hafta7 > Exceptions1 > calculateModulus(int, int) : void

```
1 package hafta7;  
2  
3 public class Exceptions1 {  
4     public static void main(String[] args) {  
5         calculateModulus(10, 0);  
6     }  
7     public static void calculateModulus(int x, int y) {  
8         if(y==0) {  
9             System.out.println("0 a bölüm hatası");  
10        }else {  
11            System.out.println(x%y);  
12        }  
13    }  
14 }
```

Problems @ Javadoc Declaration Console Terminal

<terminated> Exceptions1 [Java Application] C:\Users\nmacit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-

0 a bölüm hatası

Öğretim Görevlisi Naciye MACIT SEZİKLİ

```
Exeptions1.java *Exceptions1.java X
ileriNesneTabanlıProgramlama_Guz_24_25 src hafta7 Exceptions1 calculateModulus(int, int) : void
1 package hafta7;
2
3 public class Exceptions1 {
4     public static void main(String[] args) {
5         calculateModulus(10, 0);
6     }
7     public static void calculateModulus(int x, int y) {
8         try {
9             System.out.println(x%y);
10        } catch (ArithmaticException e) {
11            System.out.println("0 a bölümme hatası ");
12        }
13    }
14 }
```

Problems @ Javadoc Declaration Console Terminal

<terminated> Exceptions1 [Java Application] C:\Users\nmacit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\java

0 a bölümme hatası

```
1 package hafta7;
2 public class Exceptions1 {
3     public static void main(String[] args) {
4         calculateModulus(10, 0);
5     }
6     public static void calculateModulus(int x, int y) {
7         try {
8             System.out.println(x%y);
9             System.out.println("try içi görev tamamlandı");
10        } catch (ArithmetricException e) {
11            System.out.println("catch bölümü çalıştı");
12            System.out.println("0 a bölümme hatası ");
13        }
14        System.out.println("kodlar buradan çalışmaya devam eder mi?");
```

Problems @ Javadoc Declaration Console Terminal

<terminated> Exceptions1 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (28 Kas 2024)

catch bölümü çalıştı

0 a bölümme hatası

kodlar buradan çalışmaya devam eder mi?

Exceptions1.java

Exceptions1.java ×

ileriNesneTabanlıProgramlama\_Guz\_24\_25 > src > hafta7 > Exceptions1 > main(String[]) : void

```
1 package hafta7;
2 public class Exceptions1 {
3     public static void main(String[] args) {
4         calculateModulus(10, 3);
5     }
6     public static void calculateModulus(int x, int y) {
7         try {
8             System.out.println(x%y);
9             System.out.println("try içi görev tamamlandı");
10        } catch (ArithmaticException e) {
11            System.out.println("catch bölümü çalıştı");
12            System.out.println("0 a bölümme hatası ");
13        }
14        System.out.println("kodlar buradan çalışmaya devam eder mi?");
```

Problems @ Javadoc Declaration Console × Terminal

<terminated> Exceptions1 [Java Application] C:\Users\nmacit\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (28 Kas 2024 11:58:

1

try içi görev tamamlandı

kodlar buradan çalışmaya devam eder mi?

```
Exeptions1.java *Exceptions1.java X
ileriNesneTabanlıProgramlama_Guz_24_25 src hafta7 Exceptions1 calculateModulus(int, int) : void
1 package hafta7;
2 public class Exceptions1 {
3     public static void main(String[] args) {
4         calculateModulus(10, 0);
5     }
6     public static void calculateModulus(int x, int y) {
7         try {
8             if(y==0) {
9                 throw new ArithmeticException();
10            //bazen biz custom işlerimizde throw fırlatırız.
11        }
12        System.out.println(x%y);
13    } catch (ArithmeticException e) {
14        System.out.println("catch bölümü çalıştı");
15    }
16 }
17 }
```

Problems @ Javadoc Declaration Console Terminal  
<terminated> Exceptions1 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (28 I

catch bölümü çalıştı  
Öğretim Görevlisi Naciye MACİT SEZİKLİ

Exeptions1.java Exeptions1.java Exception2.java

ileriNesneTabanliProgramlama\_Guz\_24\_25 src hafta7 Exception2 convertPhoneNumberToInt(String) : void

```
1 package hafta7;
2 import java.util.Scanner;
3 public class Exception2 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Lütfen bir telefon numarası giriniz (sadece rakam): ");
7         String phoneNumber = scanner.nextLine();
8         convertPhoneNumberToInt(phoneNumber);
9     }
10    public static void convertPhoneNumberToInt(String phoneNumber) {
11        // Telefon numarasını tam sayıya çevir
12        int intPhoneNumber = Integer.valueOf(phoneNumber);
13        System.out.println("Dönüştürülen Telefon Numarası: " + intPhoneNumber);
14    }
15 }
```

Problems @ Javadoc Declaration Console Terminal



<terminated> Exception2 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (28 Kas 2024 12:25:08 – 12:25:15) [pid: 23536]

Lütfen bir telefon numarası giriniz (sadece rakam): 525y4545f

Exception in thread "main" java.lang.NumberFormatException: For input string: "525y4545f"

```
at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
at java.base/java.lang.Integer.parseInt(Integer.java:588)
at java.base/java.lang.Integer.valueOf(Integer.java:935)
at ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exception2.convertPhoneNumberToInt
```

# NumberFormatException

Exeptions1.java Exceptions1.java \*Exception2.java

```
1 package hafta7;
2 import java.util.Scanner;
3 public class Exception2 {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Lütfen bir telefon numarası giriniz (sadece rakam): ");
7         String phoneNumber = scanner.nextLine();
8         convertPhoneNumberToInt(phoneNumber);
9     }
10    public static void convertPhoneNumberToInt(String phoneNumber) {
11        try {
12            // Telefon numarasını tam sayıya çevir
13            int intPhoneNumber = Integer.valueOf(phoneNumber);
14            System.out.println("Dönüştürülen Telefon Numarası: " + intPhoneNumber);
15        } catch (NumberFormatException e) {
16            // Hatalı giriş için kullanıcıya bilgi ver
17            System.out.println("Lütfen yalnızca rakam içeren bir telefon numarası giriniz!");
18            System.out.println("Hata Detayı: " + e.getMessage());
19        }
20    }
}
```

Problems @ Javadoc Declaration Console X Terminal  
<terminated> Exception2 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\j  
Lütfen bir telefon numarası giriniz (sadece rakam): 0553582654  
Dönüştürülen Telefon Numarası: 553582654

Problems @ Javadoc Declaration Console X Terminal  
<terminated> Exception2 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw  
Lütfen bir telefon numarası giriniz (sadece rakam): 0558478558fd  
Lütfen yalnızca rakam içeren bir telefon numarası giriniz!  
Hata Detayı: For input string: "0558478558fd"

# StringIndexOutOfBoundsException

The screenshot shows an Eclipse IDE interface with the following details:

- Project Explorer:** Shows files like Exeptions1.java, Exceptions1.java, Exception2.java, \*Exceptions3.java, and ileriNesneTabanlıProgramlama\_Guz\_24\_25.
- Code Editor:** Displays the code for \*Exceptions3.java. The code reads a user's name and prints the character at index 10. The code is as follows:

```
3 import java.util.Scanner;
4
5 public class Exceptions3 {
6     public static void main(String[] args) {
7         // Kullanıcıdan bir kullanıcı adı al
8         Scanner scanner = new Scanner(System.in);
9         System.out.print("Lütfen kullanıcı adınızı giriniz: ");
10        String username = scanner.nextLine();
11        // Kullanıcı adından belirli bir karakteri seçmeye çalış
12        getCharacter(username, 10);
13    }
14    public static void getCharacter(String username, int idx) {
15        char selectedChar = username.charAt(idx);
16        System.out.println("Kullanıcı adının " + (idx + 1) + ". karakteri: " + selectedChar);
17    }
}
```

- Console Output:** Shows the terminal window with the following output:

```
Lütfen kullanıcı adınızı giriniz: NaciyeMa
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: Index 10 out of bounds for
length 8
        at java.base/jdk.internal.util.Preconditions$1.apply(Preconditions.java:55)
        at java.base/jdk.internal.util.Preconditions$1.apply(Preconditions.java:52)
        at java.base/jdk.internal.util.Preconditions$4.apply(Preconditions.java:213)
        at java.base/jdk.internal.util.Preconditions$4.apply(Preconditions.java:213)
```
- Status Bar:** Shows the application is terminated and provides system information.

Exceptions3.java

```
    String username = scanner.nextLine(),
11     // Kullanıcı adından belirli bir karakteri seçmeye çalış
12     getCharacter(username, 10);
13 }
14 public static void getCharacter(String username, int idx) {
15     try {
16         // Belirtilen indeksteki karakteri al
17         char selectedChar = username.charAt(idx);
18         System.out.println("Kullanıcı adının " + (idx + 1) + ". karakteri: " + selectedChar);
19     } catch (StringIndexOutOfBoundsException e) {
20         // Hatalı indeks durumu için kullanıcıya bilgi ver
21         System.err.println("Girilen kullanıcı adı yeterince uzun değil! Lütfen en az "
22             + (idx + 1) + " karakter içeren bir kullanıcı adı giriniz.");
23         System.err.println("Hata Detayı: " + e.getMessage());
24     }
25     System.out.println("Uygulama çalışmaya devam ediyor.");
26 }
27 }
28 }
```

Problems Declaration Console Terminal

<terminated> Exceptions3 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (29 Kas 2024 20:07:18 – 20:07:23) [pid: 29844]

Lütfen kullanıcı adınızı giriniz: naciye

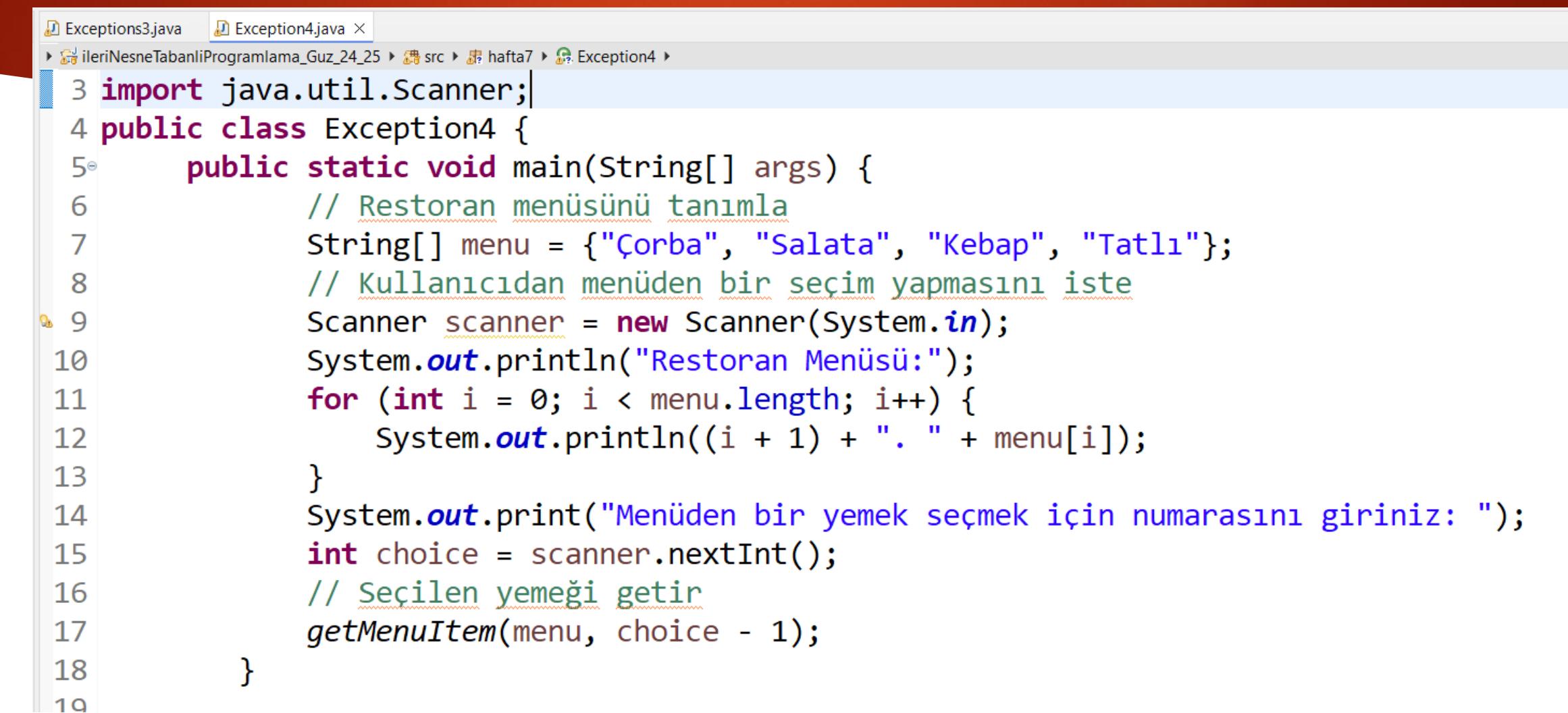
Girilen kullanıcı adı yeterince uzun değil! Lütfen en az 11 karakter içeren bir kullanıcı adı giriniz.

Hata Detayı: Index 10 out of bounds for length 6

Öğretim Görevisi Naciye MACİT SEZİK

Uygulama çalışmaya devam ediyor.

# ArrayIndexOutOfBoundsException



The screenshot shows a Java code editor with the file `Exception4.java` selected. The code demonstrates how to handle an `ArrayIndexOutOfBoundsException` by catching it and printing a custom error message. The code uses a scanner to read user input and prints a menu of items.

```
3 import java.util.Scanner;
4 public class Exception4 {
5     public static void main(String[] args) {
6         // Restoran menüsünü tanımla
7         String[] menu = {"Çorba", "Salata", "Kebap", "Tatlı"};
8         // Kullanıcıdan menüden bir seçim yapmasını iste
9         Scanner scanner = new Scanner(System.in);
10        System.out.println("Restoran Menüsü:");
11        for (int i = 0; i < menu.length; i++) {
12            System.out.println((i + 1) + ". " + menu[i]);
13        }
14        System.out.print("Menüden bir yemek seçmek için numarasını giriniz: ");
15        int choice = scanner.nextInt();
16        // Seçilen yemeği getir
17        getMenuItem(menu, choice - 1);
18    }
19}
```

# ArrayIndexOutOfBoundsException

```
16         getMenuItem(menu, choice - 1);
17     }
18     public static void getMenuItem(String[] menu, int idx) {
19         String selectedItem = menu[idx];
20         System.out.println("Seçtiğiniz yemek: " + selectedItem);
21     }
22 }
```

The screenshot shows the Eclipse IDE interface with the following details:

- Java Editor:** Displays the Java code for `Exception4.java`.
- Console Tab:** Shows the execution output:
  - Restoran Menüsü:
  - 1. Çorba
  - 2. Salata
  - 3. Kebap
  - 4. Tatlı
- Console Output:**

```
Menüden bir yemek seçmek için numarasını giriniz: 8
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 7 out of bounds for length 4
        at ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exception4.getMenuItem(Exception4.java:19)
        at ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exception4.main(Exception4.java:16)
```

# ArrayIndexOutOfBoundsException

```
19
20     public static void getMenuItem(String[] menu, int idx) {
21         try {
22             // Seçilen menü öğesini al
23             String selectedItem = menu[idx];
24             System.out.println("Seçtiğiniz yemek: " + selectedItem);
25         } catch (ArrayIndexOutOfBoundsException e) {
26             // Hatalı seçim durumu için kullanıcıya bilgi ver
27             System.out.println("Geçersiz bir seçim yaptınız! Lütfen 1 ile "
28                               + menu.length + " arasında bir sayı giriniz.");
29         }
30         System.out.println("Restoran uygulaması çalışmaya devam ediyor.");
31     }
32 }
33 }
```

Problems @ Javadoc Declaration Console X Terminal

<terminated> Exception4 [Java Application] C:\Users\nmacit\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_23.0.0.v20240919-1706\jre\bin\javaw.exe (29 Kas 2024 20:54:53 – 20:54:57) [pid: 12496]

Restoran Menüsü:

1. Çorba
2. Salata
3. Kebap
4. Tatlı

Menüden bir yemek seçmek için numarasını giriniz: 8

Geçersiz bir seçim yaptınız! Lütfen 1 ile 4 arasında bir sayı giriniz.

Restoran uygulaması çalışmaya devam ediyor.

```
*Exceptions5.java ×
ileriNesneTabanlıProgramlama_Guz_24_25 > src > hafta7 > Exceptions5 > calculateAverage(int[], int, int) : void

1 package hafta7;
2 public class Exceptions5 {
3     public static void main(String[] args) {
4         int[] grades = {85, 90, 78};
5         int divisor = 0;
6         int index = 5;
7         // Notları kontrol et ve ortalamayı hesapla
8         calculateAverage(grades, index, divisor);
9     }
10    public static void calculateAverage(int[] grades, int idx, int divisor) {
11        try {
12            // Bölme işlemi ve diziden eleman alma
13            int selectedGrade = grades[idx];
14            int average = selectedGrade / divisor;
15            System.out.println("Seçilen not: " + selectedGrade);
16            System.out.println("Ortalama: " + average);
17
18        } catch (ArithmaticException e) {
19            // Bölme hatası durumunda
20            System.out.println("Bölme işlemi hatalı: Sıfıra bölme yapmayınız! "
21                + e.getMessage());
22        } catch (ArrayIndexOutOfBoundsException e) {
23            // Geçersiz dizin hatası durumunda
24            System.out.println("Hatalı indeks kullanımı: Bu indeks dizide mevcut
değil! " + e.getMessage());
25        }
26    }
27 }
```

\*Exceptions5.java ×

```
ileriNesneTabanlıProgramlama_Guz_24_25 › src › hafta7 › Exceptions5 › calculateAverage(int[], int, int) : void
3   public static void main(String[] args) {
4       int[] grades = {85, 90, 78};
5       int divisor = 0;
6       int index = 5;
7       // Notları kontrol et ve ortalamayı hesapla
8       calculateAverage(grades, index, divisor);
9   }
10  public static void calculateAverage(int[] grades, int idx, int
divisor) {
11      try {
12          // Bölme işlemi ve diziden eleman alma
13          int selectedGrade = grades[idx];
14          int average = selectedGrade / divisor;
15          System.out.println("Seçilen not: " + selectedGrade);
16          System.out.println("Ortalama: " + average);
17
18      } catch (Exception e) {
19          // Bölme hatası durumunda
20          System.out.println("Bölme işlemi hatalı: Sıfıra bölme
yapmayınız! "
21              + e.getMessage());
22      } catch (ArrayIndexOutOfBoundsException e) {
23          // Geçersiz dizin hatası durumunda
24          System.out.println("Hatalı indeks kullanımı: Bu indeks
dizide mevcut değil! " + e.getMessage());
25      }
26  }
```

# NullPointerException

The screenshot shows the Eclipse IDE interface with the following details:

- Project Structure:** The project is named "ileriNesneTabanliProgramlama\_Guz\_24\_25". Inside it, there are two Java files: "Exceptions5.java" and "Exceptions6.java".
- Code Editor:** The code for "Exceptions6.java" is displayed. It contains a main method that demonstrates the conversion of student names to uppercase using the `toUpperCase()` method of the `String` class. The code includes comments in Turkish explaining the purpose of each variable and line.
- Output Console:** The bottom part of the interface shows the terminal output. It starts with the application's output: "Büyük harfli adı: ALİ VELİ". This is followed by an exception stack trace:

```
Büyük harfli adı: ALİ VELİ
Exception in thread "main" Büyük harfli adı:
java.lang.NullPointerException: Cannot invoke "String.toUpperCase()" because
"studentName" is null
    at
ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exceptions6.convertToUpperCase
(Exceptions6.java:20)
Ogrencim Görevi: MACIT SEZIKLI
    at ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exceptions6.main
```

Exceptions5.java Exceptions6.java

ileriNesneTabanlıProgramlama\_Guz\_24\_25 > src > hafta7 > Exceptions6 > convertToUpperCase(String) : void

```
3 public class Exceptions6 {  
4     public static void main(String[] args) {  
5         // Geçerli bir öğrenci adı  
6         String studentName1 = "Ali Veli";  
7         convertToUpperCase(studentName1);  
8         // Boş bir metin  
9         String studentName2 = "";  
10        convertToUpperCase(studentName2);  
11        // null değerli bir öğrenci adı  
12        String studentName3 = null;  
13        convertToUpperCase(studentName3);  
14    }  
15    public static void convertToUpperCase(String studentName) {  
16        try {  
17            // Öğrenci adını büyük harfe çevir  
18            String upperCaseName = studentName.toUpperCase();  
19            System.out.println("Büyük harfli adı: " + upperCaseName);  
20        } catch (NullPointerException e) {  
21            // null değeri kontrol et ve kullanıcıya bilgi ver  
22            System.err.println("null bir öğrenci adı ile işlem yapılamaz."  
23                                         + "| Lütfen geçerli bir ad giriniz.");  
24        }  
25    }  
26}
```

Büyük harfli adı: ALİ VELİ  
Büyük harfli adı:  
null bir öğrenci adı ile işlem yapılamaz. Lütfen geçerli bir ad giriniz.

# IllegalArgumentException

The screenshot shows an Eclipse IDE interface with the following details:

- Project Structure:** The project is named "ileriNesneTabanliProgramlama\_Guz\_24\_25". The package is "hafta7", and the class is "Exceptions7". The method being tested is "checkAccountBalance(double)".
- Code Content:**

```
1 package hafta7;
2
3 public class Exceptions7 {
4     public static void main(String[] args) {
5         // Gecerli bir bakiye
6         checkAccountBalance(1500);
7         // Gecersiz bir bakiye (negatif değer)
8         checkAccountBalance(-500);
9     }
10    //Exception olustugunda bunu ilan eder ve geri cekilirsiniz, buna da throw exception
11    // try-catch kullanmazsiniz.Burda programin durmasi kesinlikle isteniyor.
12    public static void checkAccountBalance(double balance) {
13        // Hesap bakiyesini kontrol et
14        if (balance < 0) {
15            // Gecersiz bir bakiye icin hata fırlat
16            throw new IllegalArgumentException("Hesap bakiyesi negatif olamaz: "
17                + balance);
18        } else {
19            System.out.println("Hesap bakiyesi: " + balance + " TL");
20        }
21    }
22 }
```
- Console Output:**

```
Hesap bakiyesi: 1500.0 TL
Exception in thread "main" java.lang.IllegalArgumentException: Hesap bakiyesi negatif
olamaz: -500.0
        at ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exceptions7.checkAccountBalance
(Exceptions7.java:17)
        at ileriNesneTabanliProgramlama_Guz_24_25/hafta7.Exceptions7.main
```

# Java Exception Türleri



Throwable

Exception

Error

Runtime  
Exception  
Other  
Exception

## Checked Exception (Derleme Zamanı Hataları)

- Derleme (compile) zamanında kontrol edilen istisnalardır.
- Kodunuzda bu tür istisnalar için ya bir try-catch bloğu kullanmalı ya da metot imzasına throws anahtar kelimesi eklemelisiniz.

Örnekler:  
•IOException  
•SQLException  
•FileNotFoundException

## Unchecked Exceptions (Runtime Hataları)

Derleme zamanında kontrol edilmezler; çalışma zamanı sırasında oluşurlar.

Genellikle programlama hatalarından kaynaklanır.

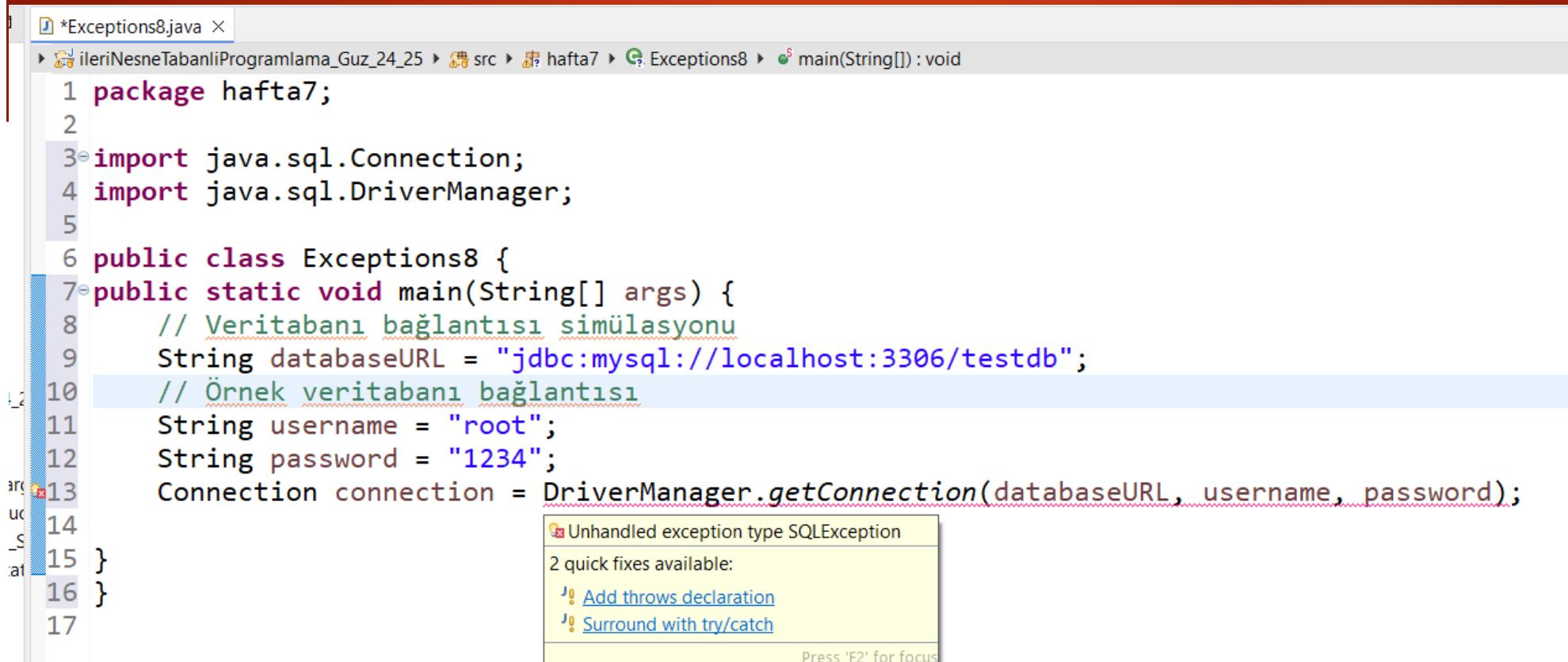
Örnekler:  
•ArithmaticException (0'a bölme)  
•NullPointerException (null bir referansa erişim)  
•ArrayIndexOutOfBoundsException (dizi sınırlarını aşmak)

## Errors

Exception sınıfından değil, Error sınıfından türetilirler.  
Sistem seviyesindeki ciddi sorunları temsil eder ve genellikle program tarafından ele alınmaz.

Örnekler:  
•StackOverflowError  
•OutOfMemoryError

# Compile Time Exception SQLException

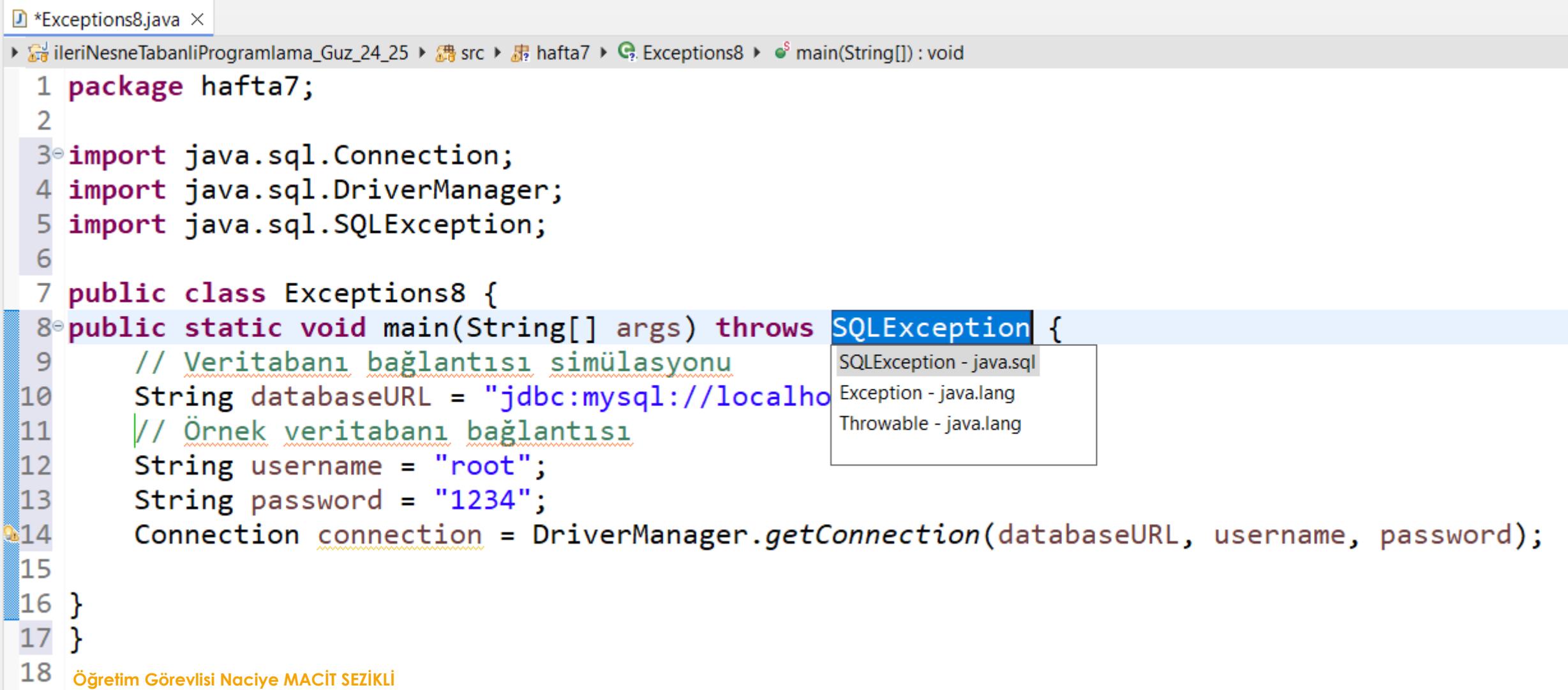


The screenshot shows a Java code editor with the file `*Exceptions8.java` open. The code attempts to establish a database connection using the `DriverManager.getConnection` method. A tooltip is displayed at the cursor position, indicating an `Unhandled exception type SQLException`. It provides two quick fix options: `Add throws declaration` and `Surround with try/catch`.

```
1 package hafta7;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5
6 public class Exceptions8 {
7     public static void main(String[] args) {
8         // Veritabanı bağlantısı simülasyonu
9         String databaseURL = "jdbc:mysql://localhost:3306/testdb";
10        // Örnek veritabanı bağlantısı
11        String username = "root";
12        String password = "1234";
13        Connection connection = DriverManager.getConnection(databaseURL, username, password);
14
15    }
16 }
17
```

# Compile Time Exception SQLException

```
*Exceptions8.java ×
ileriNesneTabanlıProgramlama_Guz_24_25 › src › hafta7 › Exceptions8 › main(String[]) : void
1 package hafta7;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.SQLException;
6
7 public class Exceptions8 {
8     public static void main(String[] args) throws SQLException {
9         // Veritabanı bağlantısı simülasyonu
10        String databaseURL = "jdbc:mysql://localhost";
11        // Örnek veritabanı bağlantısı
12        String username = "root";
13        String password = "1234";
14        Connection connection = DriverManager.getConnection(databaseURL, username, password);
15
16    }
17 }
18 Öğretim Görevlisi Naciye MACİT SEZİKLİ
```



A screenshot of an IDE showing Java code. The code is a simple program that attempts to connect to a MySQL database using JDBC. It imports the necessary classes from the java.sql package and defines a main method that throws an SQLException. A tooltip is displayed over the word 'SQLException' in the throws clause, listing three definitions: 'SQLException - java.sql', 'Exception - java.lang', and 'Throwable - java.lang'. The IDE interface includes a navigation bar at the top and a code editor with syntax highlighting.

# Compile Time Exception FileNotFoundException

The screenshot shows a Java code editor with the following code:

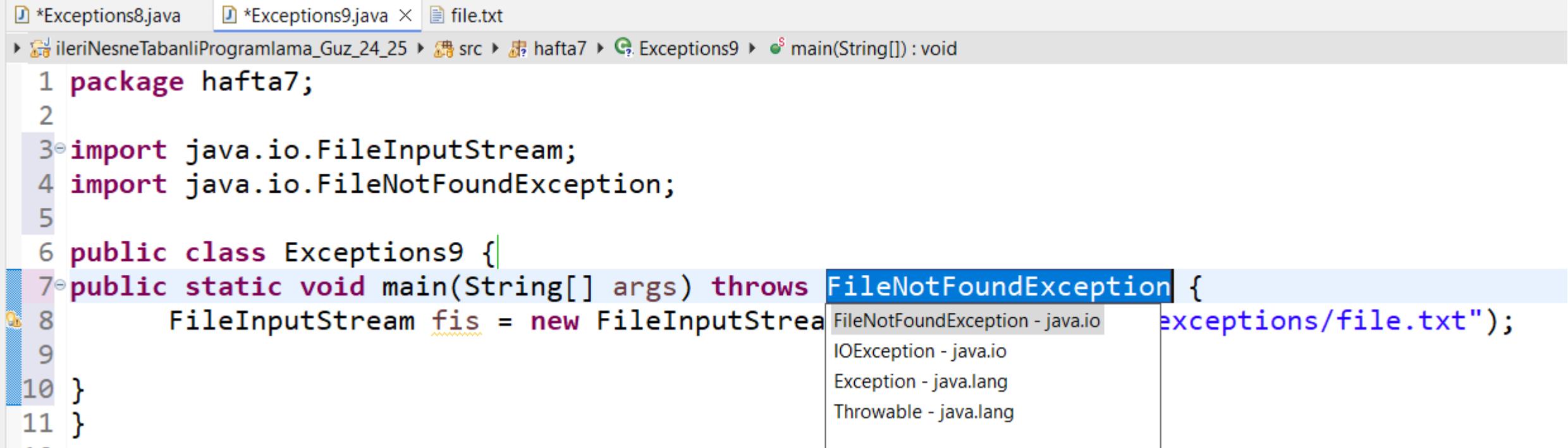
```
1 package hafta7;
2
3 import java.io.FileInputStream;
4
5 public class Exceptions9 {
6     public static void main(String[] args) {
7         FileInputStream fis = new FileInputStream("src/main/java/d27exceptions/file.txt");
8     }
9 }
10 }
11 }
```

A tooltip is displayed over the line of code that creates the `FileInputStream`. The tooltip contains the following information:

- Unhandled exception type FileNotFoundException
- 2 quick fixes available:
  - Add throws declaration
  - Surround with try/catch

At the bottom right of the tooltip, it says "Press 'F2' for focus".

# Compile Time Exception FileNotFoundException



The screenshot shows a Java code editor with the following code:

```
1 package hafta7;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5
6 public class Exceptions9 {
7     public static void main(String[] args) throws FileNotFoundException {
8         FileInputStream fis = new FileInputStream("exceptions/file.txt");
9     }
10 }
11 }
```

A tooltip is displayed over the word `FileNotFoundException` in the `throws` clause, listing the following options:

- FileNotFoundException - java.io
- IOException - java.io
- Exception - java.lang
- Throwable - java.lang

# Compile Time Exception IOException

The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "ileriNesneTabanlıProgramlama\_Guz\_24\_25" located at "src\hafta7\Exceptions9". The current file is "Exceptions9.java".
- Code Content:**

```
1 package hafta7;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5
6 public class Exceptions9 {
7     public static void main(String[] args) throws FileNotFoundException {
8         FileInputStream fis = new FileInputStream("src/main/java/d27exceptions/file.txt");
9         int k=0;
10        //okunan değerleri bunun içinde saklayacağız
11        while((k=fis.read())!=-1){//compile time hatası read önce hata verdi
12            System.out.println(k);
13        }
14    }
15 }
16 }
```
- Tooltips and Error Indicators:** A tooltip is displayed over the line "while((k=fis.read())!=-1)". It says "Unhandled exception type IOException" and provides two quick fixes:
  - Add throws declaration
  - Surround with try/catch
- Status Bar:** At the bottom left, it says "Öğretim Görevlisi Naciye MACİT SEZİKLİ".

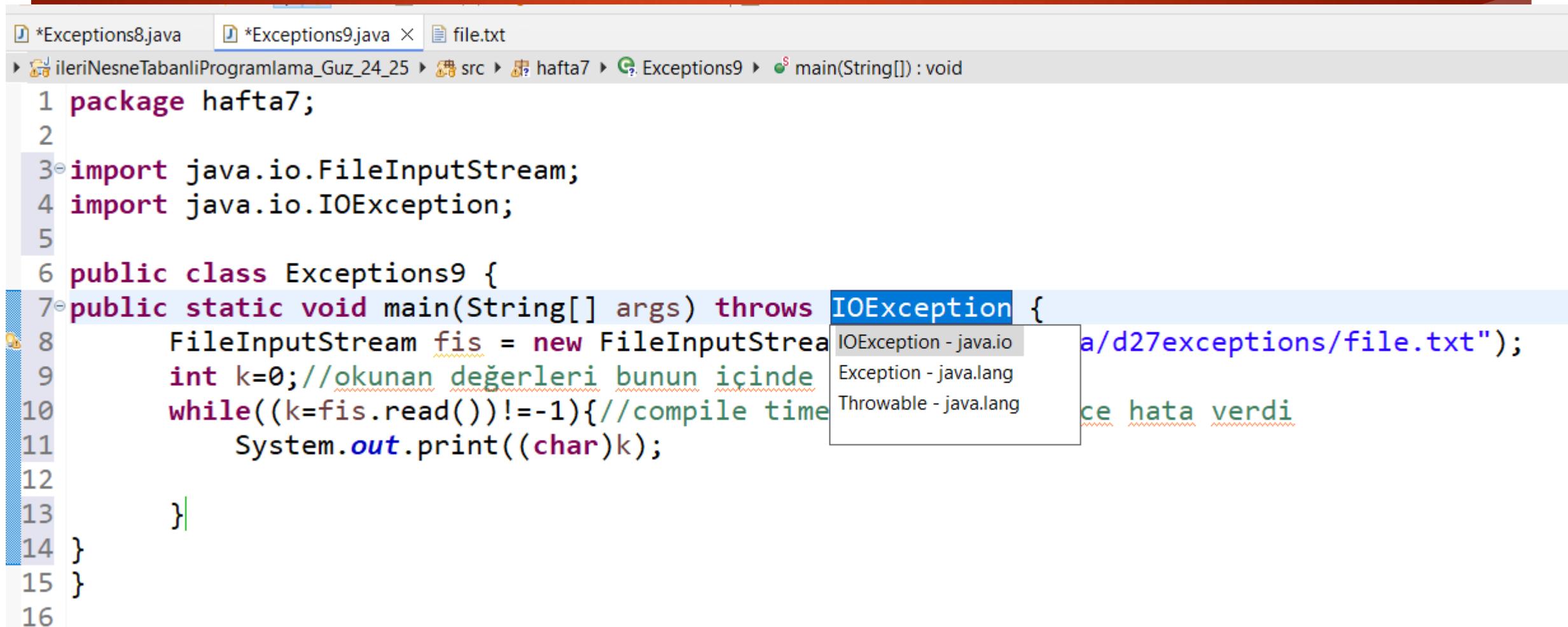
# Compile Time Exception IOException

The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "ileriNesneTabanlıProgramlama\_Guz\_24\_25" located at "src\hafta7\Exceptions9". The current file is "Exceptions9.java".
- Code Content:**

```
1 package hafta7;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5
6 public class Exceptions9 {
7     public static void main(String[] args) throws FileNotFoundException {
8         FileInputStream fis = new FileInputStream("src/main/java/d27exceptions/file.txt");
9         int k=0;
10        //okunan değerleri bunun içinde saklayacağız
11        while((k=fis.read())!=-1){//compile time hatası read önce hata verdi
12            System.out.println(k);
13        }
14    }
15 }
16 }
```
- Tooltips and Error Indicators:** A tooltip is displayed over the line "while((k=fis.read())!=-1)". It says "Unhandled exception type IOException" and provides two quick fixes:
  - Add throws declaration
  - Surround with try/catch
- Status Bar:** The status bar at the bottom right says "Press 'F2' for focus".

# Compile Time Exception IOException



The screenshot shows a Java code editor with the following code:

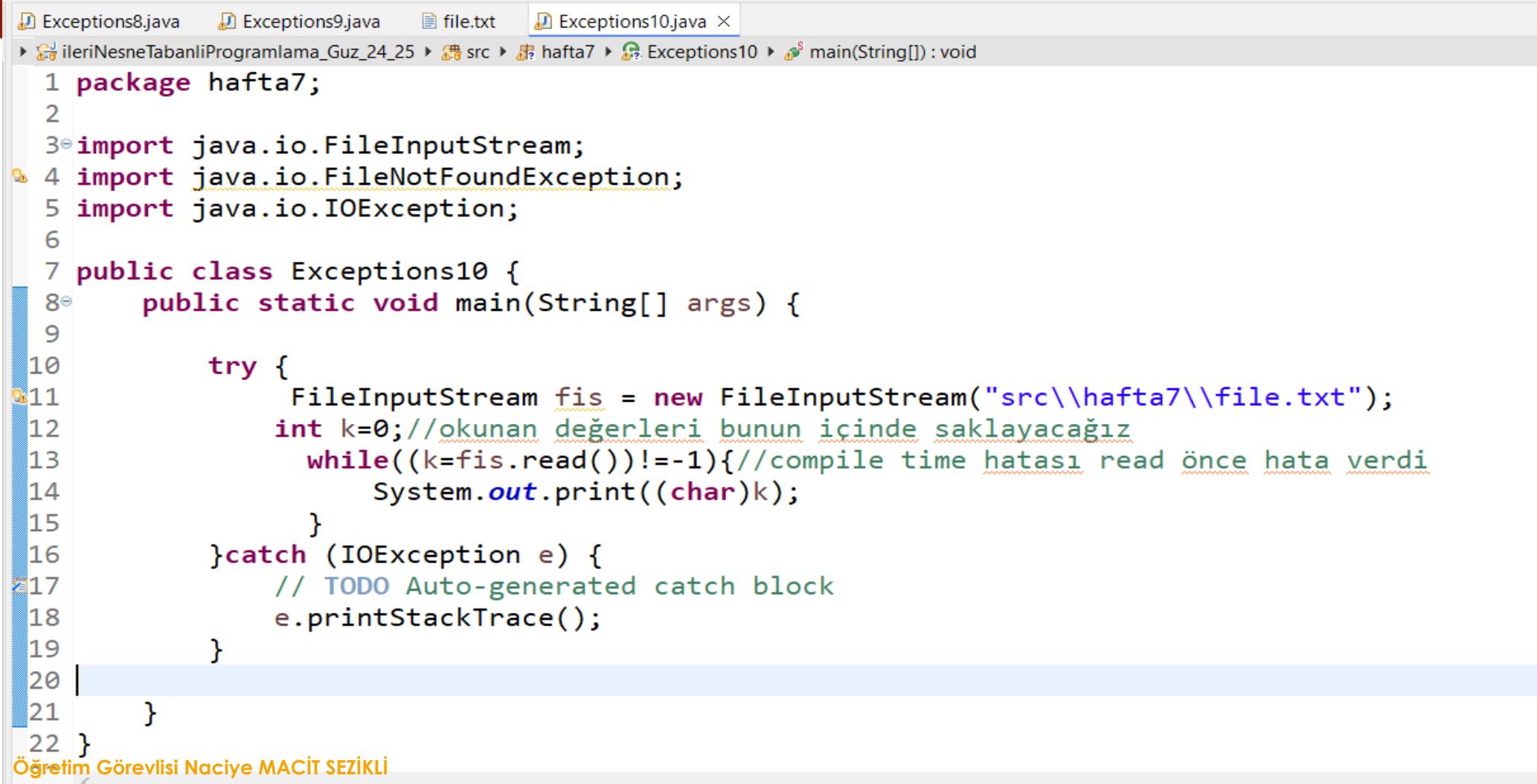
```
1 package hafta7;
2
3 import java.io.FileInputStream;
4 import java.io.IOException;
5
6 public class Exceptions9 {
7     public static void main(String[] args) throws IOException {
8         FileInputStream fis = new FileInputStream("a/d27exceptions/file.txt");
9         int k=0; //okunan değerleri bunun içinde
10        while((k=fis.read())!=-1){ //compile time
11            System.out.print((char)k);
12        }
13    }
14 }
15 }
```

A tooltip is displayed over the word `IOException` in line 7, listing three options:

- IOException - java.io
- Exception - java.lang
- Throwable - java.lang

The text "ce hata verdi" is visible at the bottom right of the tooltip.

# Compile Time Exception TRY CATCH



The screenshot shows a Java code editor with the following details:

- Project Structure:** ileriNesneTabanlıProgramlama\_Guz\_24\_25 > src > hafta7 > Exceptions10 > Exceptions10.java
- Code Content:**

```
1 package hafta7;
2
3 import java.io.FileInputStream;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6
7 public class Exceptions10 {
8     public static void main(String[] args) {
9
10        try {
11            FileInputStream fis = new FileInputStream("src\\hafta7\\file.txt");
12            int k=0;//okunan değerleri bunun içinde saklayacağız
13            while((k=fis.read())!=-1){//compile time hatası read önce hata verdi
14                System.out.print((char)k);
15            }
16        }catch (IOException e) {
17            // TODO Auto-generated catch block
18            e.printStackTrace();
19        }
20    }
21 }
22 }
```
- Annotations:** The code contains several annotations in Turkish:
  - //okunan değerleri bunun içinde saklayacağız
  - //compile time hatası read önce hata verdi
- Bottom Bar:** The bottom bar displays the text "Öğretim Görevlisi Naciye MACİT SEZİKLİ".

# Finally blogu

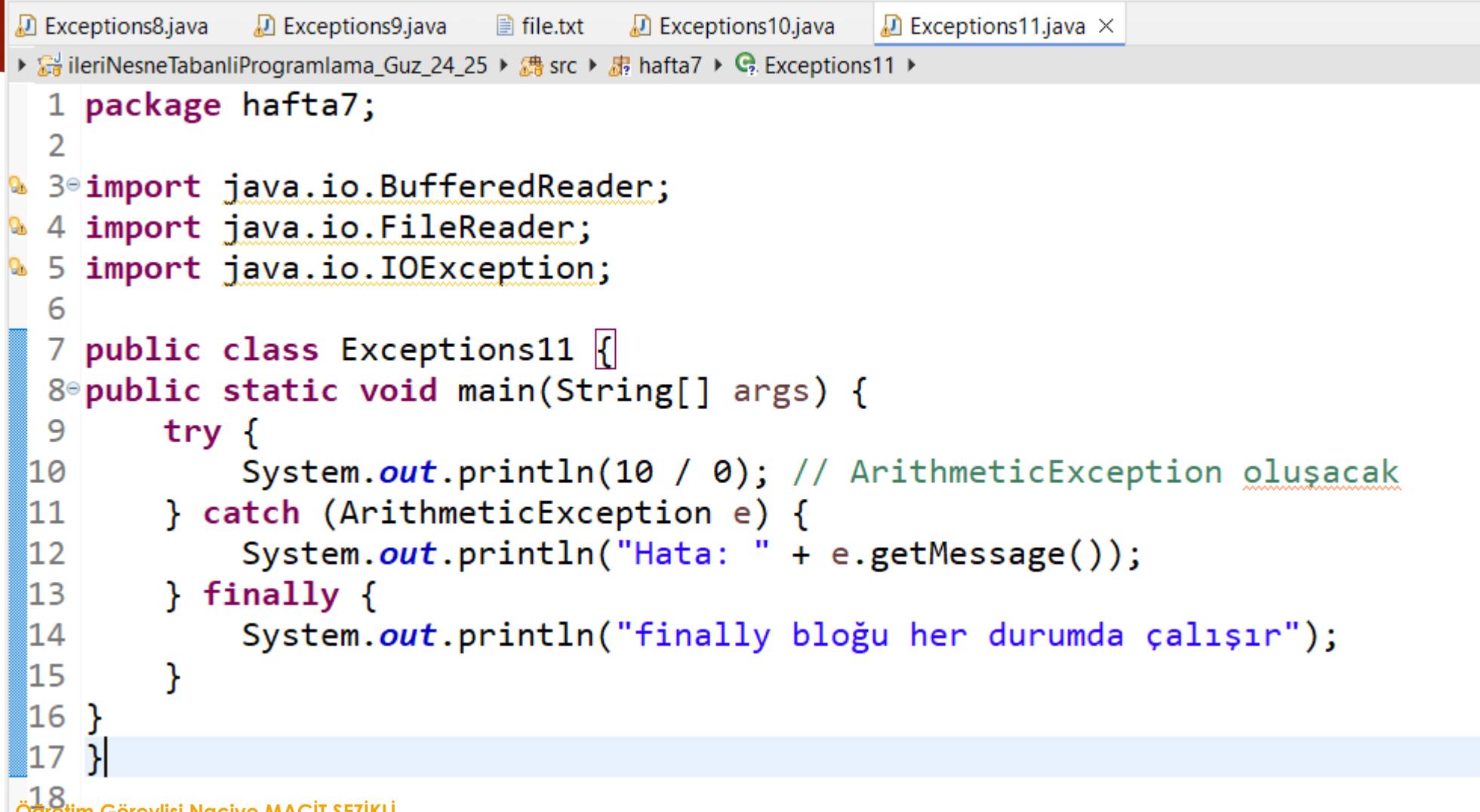
## ► finally blogu,

- Java'da **try-catch** blokları ile birlikte kullanılan ve her durumda çalışacak olan bir kısımdır.
- finally blogunun temel amacı, hata olsa da olmasa da her durumda çalıştırılması gereken kodları yerleştirmektir.
- Genellikle temizleme işlemleri (örneğin, açık kaynakların kapatılması veya dosya bağlantılarının kapatılması) için kullanılır.

# Finally blogu

- ▶ Kaynakların serbest bırakılması : (veritabani bağlantısının kesilmesi, close() methodu ile), Özellikle eski Java versiyonlarında (Java 7 öncesi), veritabanı bağlantıları gibi kaynaklar manuel olarak kapatılmalıdır. Bu tür kaynaklar, uzun süre açık bırakılırsa sistemde gereksiz yere yer kaplar ve potansiyel performans sorunlarına neden olabilir. Java 7 sonrası otomatiklesti ( try-with-resources ifadesi ile)
- ▶ Dosyaların kapatılması : (az önce dosya bağlantısı yapmıştık. fis.close(); ile kapatılabilirdi) Dosyaları okuma veya yazma işlemi bittikten sonra kapatmak gereklidir. Aksi halde, dosya kolları gereksiz yere sistemde yer kaplar.
- ▶ Her durumda çalıştırılması gereken temizleme işlemleri : (Bazı işlemler, başarılı olsun ya da olmasın, belirli bir sürecin sonunda her zaman yapılmalıdır. Örneğin, bir log dosyasına her işlemin başlangıç ve bitiş zamanını kaydetmek gibi. BufferedWriter ile)

# Finally bloğu



```
1 package hafta7;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6
7 public class Exceptions11 {
8     public static void main(String[] args) {
9         try {
10             System.out.println(10 / 0); // ArithmeticException olacak
11         } catch (ArithmaticException e) {
12             System.out.println("Hata: " + e.getMessage());
13         } finally {
14             System.out.println("finally bloğu her durumda çalışır");
15         }
16     }
17 }
```