Chapter 16: Exception Handling

TAO Yida

taoyd@sustech.edu.cn

Exception

- An ***exception*** is an indication of a problem that occurs during a program's execution. It would disrupt the normal flow of instructions.

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter an integer numerator: ");
    int numerator = scanner.nextInt();
    System.out.print("Enter an integer denominator: ");
    int denominator = scanner.nextInt();
    int result = quotient(numerator, denominator);
    System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
    scanner.close();
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

Three executions of the program

```
Enter an integer numerator: 3
Enter an integer denominator: 2
Result: 3 / 2 = 1
```

A normal execution, where the result is calculated correctly.

```
Enter an integer numerator: 3
Enter an integer denominator: 0
Exception in thread "main" java.lang.ArithmeticException: / by zero
at ExceptionExample.quotient(ExceptionExample.java:15)
at ExceptionExample.main(ExceptionExample.java:10)
```

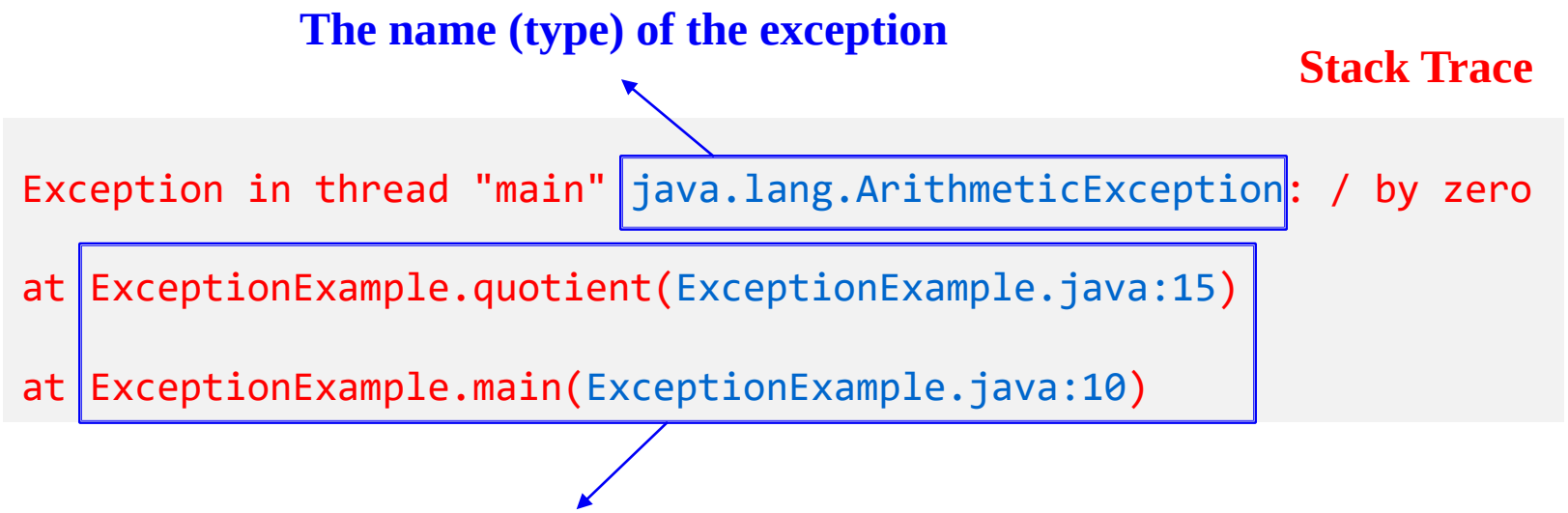
An execution where the "/ by zero" exception is thrown and the program terminates

Three executions of the program

```
Enter an integer numerator: 3
Enter an integer denominator: a
Exception in thread "main" java.util.InputMismatchException
at java.util.Scanner.throwFor(Unknown Source)
at java.util.Scanner.next(Unknown Source)
at java.util.Scanner.nextInt(Unknown Source)
at java.util.Scanner.nextInt(Unknown Source)
at ExceptionExample.main(ExceptionExample.java:9)
```

An execution where the "InputMismatch" exception is thrown and the program terminates

Exception (Cont.)



Stack trace contains the path of execution that led to the exception!!!

Exception handling

- An exception would disrupt program execution flows (for example, causing crashes).
- **Exception handling** is a nice feature of the Java language that can help you write **robust** and **fault-tolerant** programs.
- With exception handling, a program can **continue executing (rather than terminating)** after dealing with a problem. It is very useful in **mission-critical** or **business-critical** computing.

try-catch statement syntax

```
try {
    // code that might throw an exception
} catch( ExceptionType1 e1 ) {
    // code that handles type1 exception
} catch( ExceptionType2 e2 ) {
    // code that handles type2 exception
} catch( ExceptionType3 e3 ) {
    // code that handles type3 exception
} ...
```

Exception parameter
e1 is a local variable in the catch block

At least one catch block or a finally block must **immediately** follow the try block ("immediately" means no content in between)

Handling the two exceptions

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    boolean continueLoop = true;
    do {
        try {
            System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}
```

Enclose the code that might throw an exception in a try block

Each catch block (exception handler) handles a certain type of exception.

The type is specified in the exception parameter.

Handling the two exceptions

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    boolean continueLoop = true;
    do {
        try {
            System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}
```

Loop until all inputs are valid

Let's examine the control flow of a typical case


```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
    boolean continueLoop = true;
    do {
        try {
            System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
            System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
        3   System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer numerator:
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
        3   System.out.print("Enter an integer numerator: ");
        4   int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer numerator: 3
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
        3   System.out.print("Enter an integer numerator: ");
        4   int numerator = scanner.nextInt();
        5   System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer denominator:
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
        3   System.out.print("Enter an integer numerator: ");
        4   int numerator = scanner.nextInt();
        5   System.out.print("Enter an integer denominator: ");
        6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer denominator: a
```

Exception occurs!

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
        3   System.out.print("Enter an integer numerator: ");
        4   int numerator = scanner.nextInt();
        5   System.out.print("Enter an integer denominator: ");
        6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

Skip the remaining statements in the try block (termination model)

The first catch block whose exception type matches the thrown one gets executed

```
Exception: java.util.InputMismatchException
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
3       System.out.print("Enter an integer numerator: ");
4       int numerator = scanner.nextInt();
5       System.out.print("Enter an integer denominator: ");
6       int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
7       System.err.printf("Exception: %s\n", inputMismatchException);
8       scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

What happens if we delete this line?

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
        3   System.out.print("Enter an integer numerator: ");
        4   int numerator = scanner.nextInt();
        5   System.out.print("Enter an integer denominator: ");
        6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

Exit the try-catch statement and continue with the loop condition test

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3   System.out.print("Enter an integer numerator: ");
    4   int numerator = scanner.nextInt();
    5   System.out.print("Enter an integer denominator: ");
    6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
    7   System.err.printf("Exception: %s\n", inputMismatchException);
    8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer numerator:
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3   System.out.print("Enter an integer numerator: ");
11  4   int numerator = scanner.nextInt();
    5   System.out.print("Enter an integer denominator: ");
    6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
    7   System.err.printf("Exception: %s\n", inputMismatchException);
    8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer numerator: 3
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3   System.out.print("Enter an integer numerator: ");
11  4   int numerator = scanner.nextInt();
12  5   System.out.print("Enter an integer denominator: ");
    6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer denominator:
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3   System.out.print("Enter an integer numerator: ");
11  4   int numerator = scanner.nextInt();
12  5   System.out.print("Enter an integer denominator: ");
13  6   int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
Enter an integer denominator: 0
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3       System.out.print("Enter an integer numerator: ");
11  4       int numerator = scanner.nextInt();
12  5       System.out.print("Enter an integer denominator: ");
13  6       int denominator = scanner.nextInt();
14          int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3       System.out.print("Enter an integer numerator: ");
11  4       int numerator = scanner.nextInt();
12  5       System.out.print("Enter an integer denominator: ");
13  6       int denominator = scanner.nextInt();
14          int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

Exception occurs !

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3       System.out.print("Enter an integer numerator: ");
11  4       int numerator = scanner.nextInt();
12  5       System.out.print("Enter an integer denominator: ");
13  6       int denominator = scanner.nextInt();
14          int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
16          System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

Skip the remaining statements in the try block

The first catch block whose exception type matches the thrown one gets executed

```
Exception: java.lang.ArithmeticException: / by zero
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
10  3       System.out.print("Enter an integer numerator: ");
11  4       int numerator = scanner.nextInt();
12  5       System.out.print("Enter an integer denominator: ");
13  6       int denominator = scanner.nextInt();
14          int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
16          System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

Exit the try-catch statement and continue with the loop condition test

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
    11  4   int numerator = scanner.nextInt();
    12  5   System.out.print("Enter an integer denominator: ");
    13  6   int denominator = scanner.nextInt();
    14      int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

```
Enter an integer numerator:
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
    12  5   System.out.print("Enter an integer denominator: ");
    13  6   int denominator = scanner.nextInt();
    14      int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

```
Enter an integer numerator: 3
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
20  12  5   System.out.print("Enter an integer denominator: ");
    13  6   int denominator = scanner.nextInt();
    14      int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

```
Enter an integer denominator:
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
20  12  5   System.out.print("Enter an integer denominator: ");
21  13  6   int denominator = scanner.nextInt();
    14      int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

```
Enter an integer denominator: 2
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
20  12  5   System.out.print("Enter an integer denominator: ");
21  13  6   int denominator = scanner.nextInt();
22  14      int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
15  return numerator / denominator;
}
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    boolean continueLoop = true;
    do {
        try {
            System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
20  12  5   System.out.print("Enter an integer denominator: ");
21  13  6   int denominator = scanner.nextInt();
22  14      int result = quotient(numerator, denominator);
24          System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
        7   System.err.printf("Exception: %s\n", inputMismatchException);
        8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17
}

public static int quotient(int numerator, int denominator) {
23 15   return numerator / denominator;
}
```

```
Result: 3 / 2 = 1
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18 10 3     System.out.print("Enter an integer numerator: ");
19 11 4     int numerator = scanner.nextInt();
20 12 5     System.out.print("Enter an integer denominator: ");
21 13 6     int denominator = scanner.nextInt();
22 14       int result = quotient(numerator, denominator);
24          System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
25          scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            7 System.err.printf("Exception: %s\n", inputMismatchException);
            8 scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9  17
}

public static int quotient(int numerator, int denominator) {
23 15   return numerator / denominator;
}
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    boolean continueLoop = true;
    do {
        try {
            System.out.print("Enter an integer numerator: ");
            int numerator = scanner.nextInt();
            System.out.print("Enter an integer denominator: ");
            int denominator = scanner.nextInt();
            int result = quotient(numerator, denominator);
            System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
            scanner.close();
            continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            System.err.printf("Exception: %s\n", inputMismatchException);
            scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
            System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop);
}

public static int quotient(int numerator, int denominator) {
    return numerator / denominator;
}
```

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
20  12  5   System.out.print("Enter an integer denominator: ");
21  13  6   int denominator = scanner.nextInt();
22  14      int result = quotient(numerator, denominator);
24          System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
25          scanner.close();
26          continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            7   System.err.printf("Exception: %s\n", inputMismatchException);
            8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17   27
}

public static int quotient(int numerator, int denominator) {
23  15  return numerator / denominator;
}
```

No exception occurs, continue with the loop condition test

```
public static void main(String[] args) {
1   Scanner scanner = new Scanner(System.in);
2   boolean continueLoop = true;
    do {
        try {
18  10  3   System.out.print("Enter an integer numerator: ");
19  11  4   int numerator = scanner.nextInt();
20  12  5   System.out.print("Enter an integer denominator: ");
21  13  6   int denominator = scanner.nextInt();
22  14      int result = quotient(numerator, denominator);
24          System.out.printf("Result: %d / %d = %d\n", numerator, denominator, result);
25          scanner.close();
26          continueLoop = false;
        } catch(InputMismatchException inputMismatchException) {
            7   System.err.printf("Exception: %s\n", inputMismatchException);
            8   scanner.nextLine(); // discard input so user can try again
        } catch(ArithmeticException arithmeticException) {
    16      System.err.printf("Exception: %s\n", arithmeticException);
        }
    } while(continueLoop); 9   17   27
}

public static int quotient(int numerator, int denominator) {
23  15  return numerator / denominator;
}
```

Exit loop and main method terminates normally