

Исключения в JAVA

```
public class App {  
    public static void main(String[] args) throws Throwable {}  
}
```

```
public class App {  
    public static void main(String[] args) throws String {}  
}
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
        } catch (Throwable t) {}  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
        } catch (String s) {}  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        // Error - потомок Throwable  
        throw new Error();  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        throw new String("Hello!");  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        throw null;  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        Error ref = new Error(); // создаем экземпляр  
        throw ref;                // "бросаем" его  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        f(null);  
    }  
    public static void f(NullPointerException e) {  
        try {  
            throw e;  
        } catch (NullPointerException npe) {  
            f(npe);  
        }  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        double d = sqr(10.0);  
        System.out.println(d);  
    }  
    public static double sqr(double arg) {  
        // ...  
    }  
}
```

```
public class App {  
    public static double sqr(double arg) {  
        while (true);  
    }  
}
```

```
public class App {  
    public static double sqr(double arg) {  
        throw new RuntimeException();  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        double d = sqr(10.0);  
        System.out.println(d);  
    }  
    public static double sqr(double arg) {  
        throw new RuntimeException();  
    }  
}
```

```
public static int area(int width, int height) {  
    if (width < 0 || height < 0) {  
        // Что здесь должно быть?  
    }  
    return width * height;  
}
```

```
public class App {
    public static void main(String[] args) {
        System.err.println("#1.in");
        f(); // создаем фрейм, помещаем в стек, передаем в него управление
        System.err.println("#1.out"); // вернулись
    } // выходим из текущего фрейма, кончились инструкции

    public static void f() {
        System.err.println(".    #2.in");
        g(); // создаем фрейм, помещаем в стек, передаем в него управление
        System.err.println(".    #2.out"); //вернулись
    } // выходим из текущего фрейма, кончились инструкции

    public static void g() {
        System.err.println(".    .    #3.in");
        h(); // создаем фрейм, помещаем в стек, передаем в него управление
        System.err.println(".    .    #3.out"); // вернулись
    } // выходим из текущего фрейма, кончились инструкции

    public static void h() {
        System.err.println(".    .    .    #4.in");
        if (true) {
            System.err.println(".    .    .    #4.RETURN");
            return; // выходим из текущего фрейма по 'return'
        }
        System.err.println(".    .    .    #4.out"); // ПРОПУСКАЕМ
    }
}
```



```
>> #1.in
>> .    #2.in
>> .    .    #3.in
>> .    .    .    #4.in
>> .    .    .    #4.RETURN
>> .    .    #3.out
>> .    #2.out
>> #1.out
```

```
public class App {  
    public static void main(String[] args) {  
        System.err.println("#1.in");  
        f(); // создаем фрейм, помещаем в стек, передаем в него управление  
        System.err.println("#1.out"); // ПРОПУСТИЛИ!  
    }  
  
    public static void f() {  
        System.err.println(".    #2.in");  
        g(); // создаем фрейм, помещаем в стек, передаем в него управление  
        System.err.println(".    #2.out"); // ПРОПУСТИЛИ!  
    }  
  
    public static void g() {  
        System.err.println(".    .    #3.in");  
        h(); // создаем фрейм, помещаем в стек, передаем в него управление  
        System.err.println(".    .    #3.out"); // ПРОПУСТИЛИ!  
    }  
  
    public static void h() {  
        System.err.println(".    .    .    #4.in");  
        if (true) {  
            System.err.println(".    .    .    #4.THROW");  
            throw new Error(); // выходим со всей пачки фреймов ("раскрытие стека") по 'throw'  
        }  
        System.err.println(".    .    .    #4.out"); // ПРОПУСТИЛИ!  
    }  
}
```

```
>> #1.in
```

```
>> .    #2.in
```

```
>> .    .    #3.in
```

```
>> .    .    .    #4.in
```

```
>> .    .    .    #4.THROW
```

```
>> RUNTIME ERROR: Exception in thread "main" java.lang.Error
```

```
>> #1.in
>> .    #2.in
>> .    .    #3.in
>> .    .    .    #4.in
>> .    .    .    #4.THROW
>> .    #2.CATCH
>> .    #2.out
>> #1.out
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new RuntimeException();}  
            System.err.print(" 1");  
        } catch (Exception e) { // catch по Exception  
ПЕРЕХВАТЫВАЕТ RuntimeException  
            System.err.print(" 2");  
        }  
        System.err.println(" 3");  
    }  
}?? ЧТО ВЫВЕДЕМ
```

```
public class App {  
    public static void main(String[] args) throws Exception { // пока игнорируйте 'throws'  
        try {  
            System.err.print(" 0");  
            if (true) {throw new Exception();}  
            System.err.print(" 1");  
        } catch (RuntimeException e) {  
            System.err.print(" 2");  
        }  
        System.err.print(" 3");  
    }  
}
```

```
>> 0
```

```
>> RUNTIME EXCEPTION: Exception in thread "main" java.lang.Exception
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new Error();}  
            System.err.print(" 1");  
        } catch (Exception e) {  
            System.err.print(" 2");  
        }  
        System.err.print(" 3");  
    }  
}
```

```
>> 0
```

```
>> RUNTIME EXCEPTION: Exception in thread "main" java.lang.Error
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new RuntimeException();}  
            System.err.print(" 1");  
        } catch (RuntimeException e) {           // перехватили RuntimeException  
            System.err.print(" 2");  
            if (true) {throw new Error();} // но бросили Error  
        }  
        System.err.println(" 3");           // пропускаем - уже летит Error  
    }  
}
```

```
>> 0 2
```

```
>> RUNTIME EXCEPTION: Exception in thread "main" java.lang.Error
```



```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new RuntimeException();}  
            System.err.print(" 1");  
        } catch (RuntimeException e) { // перехватили RuntimeException  
            System.err.print(" 2");  
            if (true) {throw e;}      // и бросили ВТОРОЙ раз ЕГО ЖЕ  
        }  
        System.err.println(" 3");    // пропускаем - опять летит RuntimeException  
    }  
}
```

>> 0 2

>> RUNTIME EXCEPTION: Exception in thread "main" java.lang.RuntimeException

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new RuntimeException();}  
            System.err.print(" 1");  
        } catch (RuntimeException e) {          // перехватили RuntimeException  
            System.err.print(" 2");  
            if (true) {throw new Error();} // и бросили новый Error  
        } catch (Error e) { // хотя есть catch по Error "ниже", но мы в него не попадаем  
            System.err.print(" 3");  
        }  
        System.err.println(" 4");  
    }  
}
```

>> 0 2

>> RUNTIME EXCEPTION: Exception in thread "main" java.lang.Error

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new RuntimeException();}  
            System.err.print(" 1");  
        } catch (RuntimeException e) { // перехватили RuntimeException  
            System.err.print(" 2.1");  
            try {  
                System.err.print(" 2.2");  
                if (true) {throw new Error();} // и бросили новый Error  
                System.err.print(" 2.3");  
            } catch (Throwable t) { // перехватили Error  
                System.err.print(" 2.4");  
            }  
            System.err.print(" 2.5");  
        } catch (Error e) { // хотя есть catch по Error "ниже", но мы в него не попадаем  
            System.err.print(" 3");  
        }  
        System.err.println(" 4");  
    }  
}
```

```
>> 0 2.1 2.2 2.4 2.5 4
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            } catch (Exception e) {  
            } catch (RuntimeException e) {  
            }  
        }  
    }  
}
```

```
>> COMPILATION ERROR: Exception 'java.lang.RuntimeException' has already been caught
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
        } catch (Error e) {  
        } catch (RuntimeException e) {  
        }  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            throw new Exception();  
        } catch (RuntimeException e) {  
            System.err.println("catch RuntimeException");  
        } catch (Exception e) {  
            System.err.println("catch Exception");  
        } catch (Throwable e) {  
            System.err.println("catch Throwable");  
        }  
        System.err.println("next statement");  
    }  
}
```

```
>> catch Exception  
>> next statement
```

```
public class App {
    public static void main(String[] args) {
        try {
            Throwable t = new Exception(); // ссылка типа Throwable указывает на
            объект типа Exception
            throw t;
        } catch (RuntimeException e) {
            System.err.println("catch RuntimeException");
        } catch (Exception e) {
            System.err.println("catch Exception");
        } catch (Throwable e) {
            System.err.println("catch Throwable");
        }
        System.err.println("next statement");
    }
}
```

```
>> catch Exception
>> next statement
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.println("try");  
        } finally {  
            System.err.println("finally");  
        }  
    }  
}
```

```
>> try  
    >> finally
```



```
public class App {  
    public static void main(String[] args) {  
        try {  
            throw new RuntimeException();  
        } finally {  
            System.err.println("finally");  
        }  
    }  
}
```

>> finally

>> Exception in thread "main" java.lang.RuntimeException

```
public class App {  
    public static void main(String[] args) {  
        try {  
            return;  
        } finally {  
            System.err.println("finally");  
        }  
    }  
}
```

```
>> finally
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.exit(42);  
        } finally {  
            System.err.println("finally");  
        }  
    }  
}
```

```
>> Process finished with exit code 42
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            Runtime.getRuntime().halt(42);  
        } finally {  
            System.err.println("finally");  
        }  
    }  
}
```

>> Process finished with exit code 42

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.println("try");  
            if (true) {throw new RuntimeException();}  
        } finally {  
            System.err.println("finally");  
        }  
        System.err.println("more");  
    }  
}
```

```
>> try
```

```
>> finally
```

```
>> Exception in thread "main" java.lang.RuntimeException
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.println("try");  
            if (true) {return;}  
        } finally {  
            System.err.println("finally");  
        }  
        System.err.println("more");  
    }  
}
```

```
>> try
```

```
>> finally
```

```
public class App {  
    public static void main(String[] args) {  
        System.err.println(f());  
    }  
    public static int f() {  
        try {  
            return 0;  
        } finally {  
            return 1;  
        }  
    }  
}
```

```
>> 1
```

```
public class App {  
    public static void main(String[] args) {  
        System.err.println(f());  
    }  
    public static int f() {  
        try {  
            throw new RuntimeException();  
        } finally {  
            return 1;  
        }  
    }  
}
```

```
>> 1
```



```
public class App {  
    public static void main(String[] args) {  
        System.err.println(f());  
    }  
    public static int f() {  
        try {  
            return 0;  
        } finally {  
            throw new RuntimeException();  
        }  
    }  
}
```

```
>> Exception in thread "main" java.lang.RuntimeException
```

```
public class App {  
    public static void main(String[] args) {  
        System.err.println(f());  
    }  
    public static int f() {  
        try {  
            throw new Error();  
        } finally {  
            throw new RuntimeException();  
        }  
    }  
}
```

```
>> Exception in thread "main" java.lang.RuntimeException
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            // nothing  
            System.err.print(" 1");  
        } catch(Error e) {  
            System.err.print(" 2");  
        } finally {  
            System.err.print(" 3");  
        }  
        System.err.print(" 4");  
    }  
}
```

```
>> 0 1 3 4
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new Error();}  
            System.err.print(" 1");  
        } catch(Error e) {  
            System.err.print(" 2");  
        } finally {  
            System.err.print(" 3");  
        }  
        System.err.print(" 4");  
    }  
}
```

```
>> 0 2 3 4
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            if (true) {throw new RuntimeException();}  
            System.err.print(" 1");  
        } catch(Error e) {  
            System.err.print(" 2");  
        } finally {  
            System.err.print(" 3");  
        }  
        System.err.print(" 4");  
    }  
}
```

```
>> 0 3
```

```
>> RUNTIME ERROR: Exception in thread "main" java.lang.RuntimeException
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            try {  
                try {  
                    ...  
                } catch (Exception e) {  
                } finally {}  
            } catch (Exception e) {  
            } finally {}  
        } catch (Exception e) {  
        } finally {}  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            try {  
                ...  
            } catch (Exception e) {  
                ...  
            } finally {  
                ...  
            }  
        } catch (Exception e) {  
            try {  
                ...  
            } catch (Exception e) {  
                ...  
            } finally {  
                ...  
            }  
        } finally {  
            try {  
                ...  
            } catch (Exception e) {  
                ...  
            } finally {  
                ...  
            }  
        }  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            try {  
                System.err.print(" 1");  
                if (true) {throw new RuntimeException();}  
                System.err.print(" 2");  
            } catch (RuntimeException e) {  
                System.err.print(" 3"); // ЗАХОДИМ - есть исключение  
            } finally {  
                System.err.print(" 4"); // заходим всегда  
            }  
            System.err.print(" 5");      // заходим - выполнение УЖЕ в норме  
        } catch (Exception e) {  
            System.err.print(" 6");      // не заходим - нет исключения, УЖЕ перехвачено  
        } finally {  
            System.err.print(" 7");      // заходим всегда  
        }  
        System.err.print(" 8");          // заходим - выполнение УЖЕ в норме  
    }  
}
```

```
>> 0 1 3 4 5 7 8
```



```
public class App {  
    public static void main(String[] args) {  
        try {  
            System.err.print(" 0");  
            try {  
                System.err.print(" 1");  
                if (true) {throw new Exception();}  
                System.err.print(" 2");  
            } catch (RuntimeException e) {  
                System.err.print(" 3"); // НЕ заходим - есть исключение, но НЕПОДХОДЯЩЕГО ТИПА  
            } finally {  
                System.err.print(" 4"); // заходим всегда  
            }  
            System.err.print(" 5");      // не заходим - выполнение НЕ в норме  
        } catch (Exception e) {  
            System.err.print(" 6");      // ЗАХОДИМ - есть подходящее исключение  
        } finally {  
            System.err.print(" 7");      // заходим всегда  
        }  
        System.err.print(" 8");          // заходим - выполнение УЖЕ в норме  
    }  
}
```

```
>> 0 1 4 6 7 8
```

```
public class App {  
    public static void main(String[] args) {  
        f(); // тут ошибка компиляции  
    }  
  
    public static void f() throws Exception {  
    }  
}
```

```
public class App {  
    public static void main(String[] args) {  
        f();  
    }  
    public static void f() throws RuntimeException {  
    }  
}
```

```
public class App {  
    // пугаем ПРЕДКОМ исключений  
    public static void main(String[] args) throws IOException {  
        if (System.currentTimeMillis() % 2 == 0) {  
            throw new EOFException();  
        } else {  
            throw new FileNotFoundException();  
        }  
    }  
}
```

```
public class App {  
    public static void main(String[] args) throws IOException, InterruptedException  
{  
        f0();  
        f1();  
        f2();  
    }  
    public static void f0() throws EOFException {...}  
    public static void f1() throws FileNotFoundException {...}  
    public static void f2() throws InterruptedException {...}  
}
```

```
public class App {  
    // пугаем Exception  
    public static void main(String[] args) throws Exception {  
        Throwable t = new Exception(); // и лететь будет Exception  
        throw t; // но тут ошибка компиляции  
    }  
}
```

```
>> COMPILATION ERROR: unhandled exception: java.lang.Throwable
```

```
public class App {  
    public static void main(String[] args) throws Exception {  
        Object ref = "Hello!"; // ref указывает на строку  
        char c = ref.charAt(0); // но тут ошибка компиляции  
    }  
}
```

```
>> COMPILATION ERROR: Cannot resolve method 'charAt(int)'
```

```
public class App {  
    // пытаем Exception  
    public static void main(String[] args) throws Exception {  
        try {  
            Throwable t = new Exception(); // и лететь будет Exception  
            throw t; // но тут ошибка компиляции  
        } catch (Exception e) {  
            System.out.println("Перехвачено!");  
        }  
    }  
}
```

```
>> COMPILATION ERROR: unhandled exception: java.lang.Throwable
```

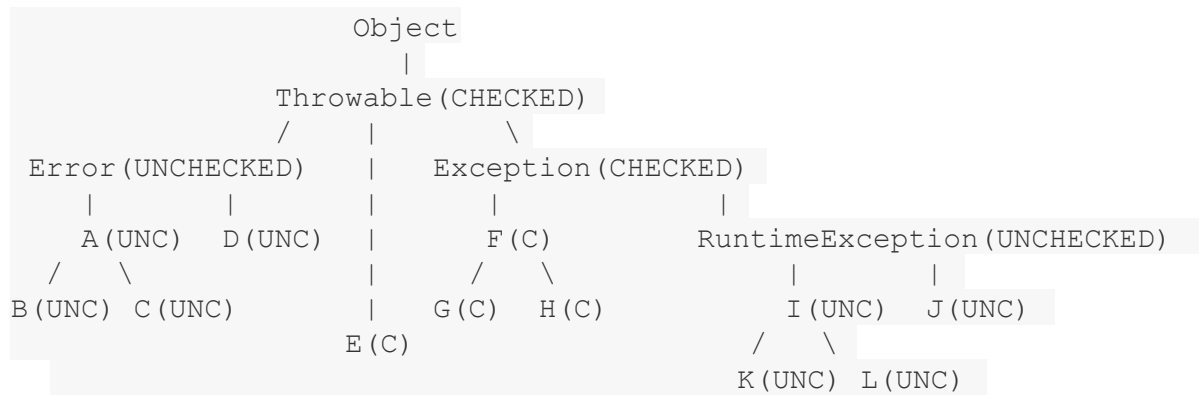
```
public class Parent {  
    // предок пугает IOException и InterruptedException  
    public void f() throws IOException, InterruptedException {}  
}
```

```
class Child extends Parent {  
    // а потомок пугает только потомком IOException  
    @Override  
    public void f() throws FileNotFoundException {}  
}
```

```
public class Parent {  
    public void f() throws IOException, InterruptedException {}  
}
```

```
class ChildB extends Parent {  
    @Override  
    public void f() throws Exception {}  
}
```

```
>> COMPILATION ERROR: overridden method does not throw 'java.lang.Exception'
```

```
public class TemporaryFileExample {
    public static void main(String[] args) {
        File tempFile = null;
        try {
            tempFile = File.createTempFile("tempfile", ".txt");
            try (BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile))) {
                writer.write("This is a temporary file.");
                writer.newLine();
                writer.write("It will be deleted after use.");
            }

            System.out.println("Temporary file written: " + tempFile.getAbsolutePath());
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (tempFile != null && tempFile.exists()) {
                tempFile.delete();
                System.out.println("Temporary file deleted: " + tempFile.getAbsolutePath());
            }
        }
    }
}
```