Static inner class
Various combinations of inner class

Java Exception
Introduction
Stack
Stack Trace
Hierarchy
Try Catch
Exception Detail
Catch Multiple Exception
Finally
Checked And Unchecked
Throw
Throws
Custom Exception

Java String
String Introduction
Immutable String
StringBuilder
StringBuffer

Bir musibat seni o'ldirmadimi?
Demak shu musibat seni kuchliroq qiladi.
Unknown

Throws

Throws - tashlaydi, otadi deb tarjima qilinadi.

Throws

`throws` bu sodir bo'lishi mumkin bo'lgan exception ni ushlash javobgarlikidan qochib uni methodni chaqiruvchisiga yuklab qo'yishdir.

Yani metodda sodir bo'lishi mumkin bo'lgan exception ni `try-catch` ga olib o'tirmasdan shu metodni `throws` kelit so'zi bilan belgilaymiz. Natijada metodni chaqiruvchisi, metodda sodir bo'ladigan exception ni ushlashga majbur.

Dehqoncha aytsam `throw` kalit so'zi metodda `try-catch` ni ishlatmaslik qulayligini yaratad va sodir bo'lishi mumkin bo'lgan exception ni ushlashni metodni chaqiruvchisi bo'yniga yulab qo'yadi. Metoddni chaqiruvchi voy jallot bo'tda exception chiqar ekan deb uni ushlashga majbur bo'ladi.

`throws` kalit so'zi faqat `checked` exception lar bilan ishlatiladi. `Unchecked` exception lar bilan ishlatishga mano yo'q.

`throws` kalit so'zini **method** va **constructor** larga ishlatsak bo'ladi. Class larga ishlata olmaymiz.

`throws` kalit so'zi bilan faqat `Throwable` classidan nasil olgan class larni ishlatsak bo'ladi.

Namuna 1 oddiy holatdagi namuna

```
public class Main {
    public static void main(String[] args) {
        show();
    }

    public static void show() {
        System.out.println("Start showing");

        try {

            Thread.sleep(3000);
```

```
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        System.out.println("The End");
    }
}
```

Natija

```
Start showing
The End
```

Namunada

- `show()` metodi chaqirildi
- Metodda **'Start showing'** so'zi konsolga chiqarildi va dastur **3 sekund** davomida uxladi.
- Dastur uyg'ongandan keyin **'The End'** so'zi konsolga chiqdi.
- `Thread.sleep(3000)` - bu kod dasturni 3 sekund davomida uxlatadi.
- Ammo `sleep()` metodi `unchecked` bo'lgan `InterruptedException` sodir qilishi mumkin.
- Shu sababdan Compiler bu qatorni `try-catch` ga olishga majburlaydi.
- Agar `try-catch` yozmasak CompileTimError boladi.
-

`show` metodida `try-catch` ishlatmasak shu metoddi `throws` kelit so'zi bilan belgilashimiz kerak.

Namuna 2 `throws` kalit so'zi bilan.

```
public class Main {
    public static void main(String[] args) {
        try {
            show();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public static void show() throws InterruptedException {
        System.out.println("Start showing");
        Thread.sleep(3000);
        System.out.println("The End");
    }
}
```

Natija

```
Start showing
The End
```

Namunada

- Natijaga qarang. Natija birxil. Natija bo'yicha hech narsa o'zgarmadi.
- `show()` - metodida `InterruptedException` sodir bo'lishi mumkin edi.
- `show()` - metodida exception ni `try-catch` ga olmadi va `InterruptedException` ni ushlash javobgarligini `main` metodga yuklab qo'ydi.
- Yani `show` metodini `try-catch` ga olish javobgarligidan qochib uni `main` metodga yuklab qo'ydi.
- `show` metod nomi yonida `throws InterruptedException` deb yozdik. Bu degani shu metod `InterruptedException` sodir qilishi mumkin degani. Bu metodni chaquruvchilar `try-catch` ga olishlari kerak degani.
- `main` - metod `show` metodini chaqirganda qarasa u `InterruptedException` sodir qilishi mumkin ekan. Chunku yozib qo'yibdi *throws InterruptedException* deb.
- `main` metod `show` ni chaqirganda uni `try-catch` ichida yozishi kerak. Bo'lmasa CompilerTimeError bo'ladi.

Tepadagi namunada `main` metodni o'zi ham `show` da sodir bo'lishi mumkin bo'lgan `InterruptedException` ni ushlash javobgarligidan qochishi mumkin. Uni quyidagi namunada ko'ramiz.

Namuna 3

```java
public class Main {
    public static void main(String[] args) throws InterruptedException {
        show();
    }

    public static void show() throws InterruptedException {
        System.out.println("Start showing");
        Thread.sleep(3000);
        System.out.println("The End");
    }
}
```

Natija

```
Start showing
The End
```

Namunada

- `show` metodi sodir bo'lishi mumkin bo'lgan exception ni ushlamasdan uni `throws` qilgan edi.
- `main` metodni `show` metodini chaqiravotganida `try-catch` ichida chaqirishi kerak edi.

- `main` metodni `show` metodini chaqirayotganida `try-catch` ichida chaqirishi kerak edi.
- Ammo `main` metodi ham sodir bo'lishi mumkin bo'lgan `InterruptedException` ni ushlash javobgarligidan qochdi va `throws InterruptedException` deb `main` metodga yozildi.
- Shuning bilan `main` metod ham javobgarlikdan choqdi.
- Endi `InterruptedException` ni ushlashni `JVM` o'z bo'yniga oladi. Sababi dasturni ishga tushurish uchun `JVM` main metodni chaqiradi. Yani `main` metodni chaqirgan JVM boladi.

2022-01-03 1140 3 3 0

← Oldingisi Keyingisi →

Izoh qoldirish

Jo'natish

Izohlarni ko'rsatish