Nolix

Nolix Exceptions

2019-02-01

Table of contents

1 Introduction

1.1 What Nolix Exceptions are

Nolix Exceptions are custom Exceptions of the Nolix library.

1.2 Why to use Nolix Exceptions

When Exceptions are thrown **directly** and Nolix Exceptions are used, then you have the following advantages.

- There exist **suitable** types of Nolix Exceptions for the most situations.
- Nolix Exceptions provide **consistent** error messages.
- Nolix Exceptions have different constructors for taking exactly all available information to create error messages that are as **informative** as possible.

1.3 Where Nolix Exceptions are

Nolix Exceptions are defined in the Nolix library. To use Nolix Exceptions, import the Nolix library into your project.

2 How to import a Nolix Exception class

```
import ch.nolix.core.invalidArgumentException.NegativeArgumentException;

…

var invalidArgumentExceptionClass = InvalidArgumentException.cass;

…
```

3 How to throw a meaningful Exception

```
public void setAmount(int amount) {

  if (amount < 0) {
    throw new NegativeArgumentException(amount);
  }

 …
}
```

If the given amount is e.g. -25, the error message of the NegativeArgumentException will be:

“The given argument ‘-25’ is negative.”

4 How to include the argument’s name in the error message

```
public void setAmount(int amount) {

  if (amount < 0) {
    throw new NegativeArgumentException(”amount”, amount);
  }

  …
}
```

If the given amount is e.g. -25, the error message of the NegativeArgumentException will be:

“The given amount ‘-25’ is negative.”

5 Where to find common argument names as constants

```
import ch.nolix.core.constants.VariableNameCatalogue;

…

public void setAmount(int amount) {

  if (amount < 0) {
    throw
    new NegativeArgumentException(VariableNameCatalogue.NAME, amount);
  }

  …
}

…
```

The VariableNameCatalogue provides constants of common argument names and can be found in the package ‘ch.nolix.core.constants’. These constants are just strings.

If the given amount is e.g. -25, the error message of the NegativeArgumentException will be:

“The given amount ‘-25’ is negative.”

6 Types of Nolix Exceptions

Exception	Purpose
BiggerArgumentException	Purposed to be thrown when a given argument is undesirably **bigger than** a given limit.
EmptyArgumentException	Purposed to be thrown when a given argument is undesirably **empty**.
EqualArgumentException	Purposed to be thrown when a given argument **equals** undesirably a given value.
FalseArgumentException	Purposed to be thrown when a given argument is undesirably **false**.
InRangeArgumentException	Purposed to be thrown when a given argument is undesirably **in a given range**.
InvalidArgumentException	Purposed to be thrown when a given argument is **not valid** of any reason.
NegativeArgumentException	Purposed to be thrown when a given argument is undesirably **negative**.
NonBiggerArgumentException	Purposed to be thrown when a given argument is undesirably not **bigger** than a given limit.
NonEmptyArgumentException	Purposed to be thrown when a given argument is undesirably **not empty**.
NonNegativeArgumentException	Purposed to be thrown when a given argument is undesirably **not negative**.
NonPositiveArgumentException	Purposed to be thrown when a given argument is undesirably **not positive**.
NonRepresentingArgumentException	Purposed to be thrown when a given argument does undesirably **not represent** an object of a certain type.
NonSmallerArgumentException	Purposed to be thrown when a given argument is **not smaller** than a given limit.

Exception	**Purpose**
NonNullArgumentException	Purposed to be thrown when a given argument is undesirably **not null**.
NullArgumentException	Purposed to be thrown when a given argument is undesirably **null**.
OutOfRangeArgumentException	Purposed to be thrown when a given argument is undesirably **not in a given range**.
PositiveArgumentException	Purposed to be thrown when a given argument is undesirably **positive**.
SmallerArgumentException	Purposed to be thrown when a given argument is undesirably **smaller** than a given limit.
TrueArgumentException	Purposed to be thrown when a given argument is undesirably **true**.
UnequalArgumentException	Purposed to be thrown when a given argument does undesirably **not equal** a given value.
ZeroArgumentException	Intended to be thrown when a number is 0.