Nolix

Nolix Exceptions

2023-12-03

Table of contents

Table of contents 2

1 Introduction 3

1.1 What Nolix Exceptions are 3

1.2 Why to use Nolix Exceptions 3

1.3 Where Nolix Exceptions are 3

2 Basics 4

2.1 Import Nolix Exceptions 4

2.2 Throw a meaningful Nolix Exception 4

2.3 Include the argument’s name in the error message 5

2.4 Use constants for common argument names 6

3 Define a custom InvalidArgumentException 7

4 Types of InvalidArgumentExceptions 8

4.1 For general cases 8

4.2 For numbers 9

4.3 For structure 10

4.4 For attributes 11

1 Introduction

1.1 What Nolix Exceptions are

Nolix Exceptions are Exceptions for invalid **arguments**.

1.2 Why to use Nolix Exceptions

- There exist **suitable** types of Nolix Exceptions for the most situations.
- Nolix Exceptions provide **consistent** error messages.
- Nolix Exceptions have different creation methods for saving all available information. So, the Nolix Exceptions provide error messages that are as **informative** as possible.

1.3 Where Nolix Exceptions are

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

2 Basics

2.1 Import Nolix Exceptions

```
import ch.nolix.core.errorcontrol.invalidargumentexception.*;
```

All Nolix Exceptions are in the ch.nolix.core.errorcontrol.invalidargumentexception package.

2.2 Throw a meaningful Nolix Exception

```
public void setDeliveryAmount(int deliveryAmount) {

  if (deliveryAmount < 0) {
    throw NegativeArgumentException.forArgument(deliveryAmount);
  }
  …
}
```

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

"The given argument '-25' is negative."

The forArgument static method of NegativeArgumentException creates a new NegativeArgumentException for the given argument.

2.3 Include the argument's name in the error message

```
public void setDeliveryAmount(int deliveryAmount) {

  if (deliveryAmount < 0) {
    throw
    NegativeArgumentException.forArgumentNameAndArgument(
      "delivery amount",
      deliveryAmount
    );
  }
  …
}
```

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

"The given delivery amount '-25' is negative."

The forArgumentNameAndArgument static method of NegativeArgumentException creates a new NegativeArgumentException for the given argument name and argument.

2.4 Use constants for common argument names

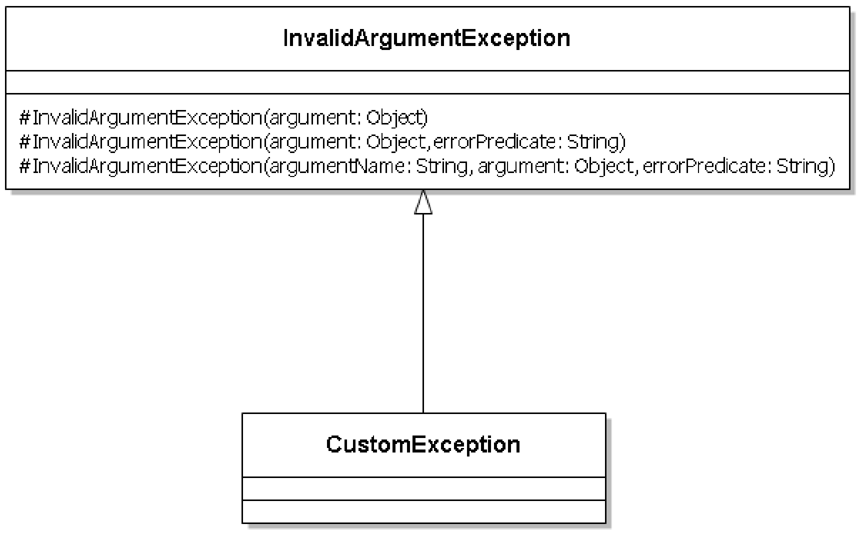
```
import ch.nolix.core.programatom.name.LowerCaseCatalogue;
…
public void setAmount(int amount) {

  if (amount < 0) {
    throw
    NegativeArgumentException.forArgumentNameAndArgument(
      LowerCaseCatalogue.AMOUNT,
      amount
    );
  }
  …
}
```

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

"The given amount '-25' is negative."

The LowerCaseCatalogue provides constants for common argument names. These constants are Strings. The LowerCaseCatalogue is in the ch.nolix.core.programatom.name package.

3 Define a custom InvalidArgumentException


InvalidArgumentException
#InvalidArgumentException(argument: Object)
#InvalidArgumentException(argument: Object,errorPredicate: String)
#InvalidArgumentException(argumentName: String, argument: Object, errorPredicate: String)
CustomException

```
public class CustomException extends InvalidArgumentException {
  …
}
```

A custom InvalidArgumentException can be defined by inheriting from it. The constructors of the custom InvalidArgumentException must call one of the super constructors.

4 Types of InvalidArgumentExceptions

4.1 For general cases

Exception	Suppose
ArgumentDoesNotSupportMethodException	Supposed to be thrown when on a given argument is **tried to call an unsupported method**.
ArgumentIsNullException	Supposed to be thrown when a given argument is undesirably **null**.
ClosedArgumentException	Supposed to be thrown when an argument is undesirably **closed**.
EqualArgumentException	Supposed to be thrown when a given argument undesirably **equals** a given value.
InvalidArgumentException	Supposed to be thrown when a given argument is **not valid**.
UnequalArgumentException	Supposed to be thrown when a given argument does undesirably **not equal** a given value.
UnrepresentingArgumentException	Supposed to be thrown when a given argument **does not represent** an object of a wanted type.

4.2 For numbers

Exception	**Suppose**
ArgumentIsInRangeException	Supposed to be thrown when a given argument is **in an unwanted range**.
ArgumentIsOutOfRangeException	Supposed to be thrown when a given argument is **not in a wanted range**.
BiggerArgumentException	Supposed to be thrown when a given argument is undesirably **bigger** than a given maximum.
NegativeArgumentException	Supposed to be thrown when a given argument is undesirably **negative**.
NonNegativeArgumentException	Supposed to be thrown when a given argument is undesirably **not negative**.
NonPositiveArgumentException	Supposed to be thrown when a given argument is undesirably **not positive**.
PositiveArgumentException	Supposed to be thrown when a given argument is undesirably **positive**.
SmallerArgumentException	Supposed to be thrown when a given argument is undesirably **smaller** than a given minimum.

4.3 For structure

Exception	Suppose
ArgumentBelongsToParentException	Supposed to be thrown when a given argument **belongs** undesirably to a parent.
ArgumentContainsElementException	Supposed to be thrown when a given argument **contains** undesirably a given element.
ArgumentDoesNotBelongToParentException	Supposed to be thrown when a given argument does undesirable **not belong** to a parent.
ArgumentDoesNotContainElementException	Supposed to be thrown when a given argument does undesirably **not contain** a given element.
EmptyArgumentException	Supposed to be thrown when a given argument is undesirably **empty**.
NonEmptyArgumentException	Supposed to be thrown when a given argument is undesirably **not empty**.

4.4 For attributes

Exception	**Suppose**
ArgumentDoesNotHaveAttributeException	Supposed to be thrown when a given argument does undesirably **not have a specific attribute**.
ArgumentHasAttributeException	Supposed to be thrown when a given argument has undesirably **a specific attribute**.