Java 2 SE Reflection

abelski

Introduction

What is Reflection?

- Reflection is the ability to examine the objects and their classes during runtime. This amazing ability allows us modifying objects behavior - during runtime - as well.

What is it Good For?

- Using the reflection capabilities we can develop Java applications with special capabilities otherwise wouldn't have been possible to develop.

  ### Automated Extensibility Capabilities

  Using the reflection classes we can develop Java applications that during runtime they will automatically extend themselves with new classes either generated automatically by our code or received from third parties.

  ### Development Environments

  Using the reflection classes we can develop development environments with various reflective capabilities (e.g. automatic code completion, adding new components and visually control them as in VB, advanced debugging capabilities, advanced testing tools etc.).

Drawbacks

- Using reflection has few drawbacks that should be taken into consideration when exploring the possibilities to use reflection in our code.

  ### Performance Price

  Using reflection capabilities has a performance price.

  ### Security Issues

  Using the reflection capabilities requires runtime permission which might be not available by the security manager where the code is executed.

  ### Internal Exposure

  Reflection allows performing actions that otherwise wouldn't have been possible (e.g. accessing private fields and methods). Using reflection in our code and having it allowed by the security code is a long run risk.

Drawbacks

- Using reflection has few drawbacks that should be taken into consideration when exploring the possibilities to use reflection in our code.

  ### Performance Price

  Using reflection capabilities has a performance price.

  ### Security Issues

  Using the reflection capabilities requires runtime permission which might be not available by the security manager where the code is executed.

  ### Internal Exposure

  Reflection allows performing actions that otherwise wouldn't have been possible (e.g. accessing private fields and methods). Using reflection in our code and having it allowed by the security code is a long run risk.

Java 2 SE Reflection

abelski

Introduction

What is Reflection?

- Reflection is the ability to examine the objects and their classes during runtime. This amazing ability allows us modifying objects behavior - during runtime - as well.

Try to run the ReflectionDemo code you can find on this course's server. This code is a good demonstration for the reflection capabilities to inspect during runtime a given class's members. Make sure that when you run this code sample you provide it (via the command line) with a full qualified name of a class you want to inspect.

What is it Good For?

- Using the reflection capabilities we can develop Java applications with special capabilities otherwise wouldn't have been possible to develop.

  Automated Extensibility Capabilities

  Using the reflection classes we can develop Java applications that during runtime they will automatically extend themselves with new classes either generated automatically by our code or received from third parties.

  Development Environments

  Using the reflection classes we can develop development environments with various reflective capabilities (e.g. automatic code completion, adding new components and visually control them as in VB, advanced debugging capabilities, advanced testing tools etc.).

Drawbacks

- Using reflection has few drawbacks that should be taken into consideration when exploring the possibilities to use reflection in our code.

  Performance Price

  Using reflection capabilities has a performance price.

  Security Issues

  Using the reflection capabilities requires runtime permission which might be not available by the security manager where the code is executed.

  Internal Exposure

  Reflection allows performing actions that otherwise wouldn't have been possible (e.g. accessing private fields and methods). Using reflection in our code and having it allowed by the security code is a long run risk.

Drawbacks

- Using reflection has few drawbacks that should be taken into consideration when exploring the possibilities to use reflection in our code.

  Performance Price

  Using reflection capabilities has a performance price.

  Security Issues

  Using the reflection capabilities requires runtime permission which might be not available by the security manager where the code is executed.

  Internal Exposure

  Reflection allows performing actions that otherwise wouldn't have been possible (e.g. accessing private fields and methods). Using reflection in our code and having it allowed by the security code is a long run risk.