JLaby - The Java Laby

Marcel Schoen

ACME corp. 2005

Overview

Introduction

The JLaby is the Java version of an old OS/390 PL/I multiplayer game. That game was a fun project, born out of boredom when I still worked for a large company. Sometimes I didn't have any work to do at all, so I started creating games. One day, me and some friends of mine decided to start a persistent multiplayer game, written in PL/I, using a DB/2 database for persistency, and providing a "3D" view (though the graphics had to be "drawn" with text characters - see screenshots).

Years later - I had started an own company together with some friends - I decided that the request-oriented, stateless character of that old OS/390 game was actually a good fit for the web. So I collected the old sourcecode, reusing only the text "graphics", but rewriting everything else from scratch in a more modern manner (using Java servlets with JDBC).

Now that it works, there's only one catch: There is still no actual game in the whole thing. You can walk around in the 3D world and do lots of stuff, but... that's it. Anyway, maybe sometime some real game will arise from it. As for now, it's just fun doing it.

(continued on next page)

(continued from last page)

FAQ

1. **What is the JLaby?**
   The JLaby is a hobby project with no purpose other than having fun doing it. It implements a simple, rectangle-based 3D online world. It provides a persistent world and user state through a database. The cheat-avoiding design leaves all decisions to the server, the client never gets any information it doesn't need. For instance, the client never knows, which direction it is looking at in the game world. The server just tells the client what it sees. It's like in the real world. As long as you don't have a compass, you have no bloody idea about the direction your looking at (north, south..). The request-oriented design makes the implementation of an action game almost impossible, since the display of a client does not update until the user refreshes it. Therefore, other types of games should be created with JLaby [fill in fresh and original idea here].On the other hand, the design should scale very well and allows for any HTTP client to roam the JLaby world. That's why you can get into the JLaby with your webbrowser. And a MIDP client exists as well.
2. **What features are planned for the JLaby world?**

   - Handling of items, including use and combination (keys to lock and unlock doors, maps, teleporters, maybe weapons, spray to print stuff on the walls etc.), exchange with other users, stealing etc.
   - Bots (computer-controlled characters) that will interact with the players, maybe also with each other. Expect the first implementations to be very silly!
   - Allow some involvement of the users, for instance by providing means to upload a customized, unique text graphics file for the own character.
3. **Who the heck want's text-character graphics nowadays?**
   In this world of ultra-realistic sophisticated realtime 3D virtual realities, it may look a little bit outdated (only a little), but its simplicity is exactly the point. Well, ok, I just needed an excuse for my nostalgic feelings. Nevertheless, I really find it pretty cool myself! :-))
4. **Why do it if there is no game??**
   It's still better writing useless software than driving around and polluting the world, or killing innocent animals on a hunt! And, for the game question, maybe someone will have an idea about that sometime. But before you have, read the next point, please.
5. **Why not just create a fantasy / scifi RPG?**
   Yaawwwnn... Oh, really, c'mon... there is at least a dozen of such games on SourceForge alone. And it's a fact that of all people on earth, only a minority really digs stuff like that. The overwhelming success of games like "The Sims" (which is the first major blockbuster game known to have been bought by at least as many women as men) is proof enough. So, personally, I'd prefer something fresh, constructive (killing preferably not included), and most of all, humorous. Well, ok, I'm flexible about the killing part, as long as it fit's in in a more black humour (Monty Python?) style.
6. **What clients / platforms are supported?**

   - HTML (webbrowser)
   - MIDP (working skeleton code exists)
   - *J2SE* does not exist yet - do you wanna do it?
7. **Is it free?**
   It is indeed, as much as you can think, as it's released under the GNU Public License (GPL)

Package
com.jlaby.action

Contains classes related to player actions.

Package Specification

- http://www.poweredge.ch

There is more to this.

- Entry A
- Entry B

This is a new paragraph.

And another one.

com.jlaby.action
Class LabyAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
```

All Implemented Interfaces:
Serializable

Direct Known Subclasses:
WalkAction, LogoutAction, LoginLocalAction, LoginAction, ShowGuiAction, NoAction, TurnAction, CreateUserAction

public abstract class **LabyAction**
extends Object
implements Serializable

Default implementation of an action object.

Field Summary

`public static final`	`AUTHENTICATED_ACTION` Value: **2**
`public static final`	`PUBLIC_ACTION` Value: **1**

Constructor Summary

`public`	`LabyAction()`

Method Summary

`abstract String`	`getHandler()` Returns the name of the handler for the action.
`GameCharacter`	`getSource()` **Deprecated.** *The definition of this call depends on `suspend`, which is deprecated. Further, the results of this call were never well-defined.*
`abstract int`	`getType()` Must return the type of the action (`PUBLIC_ACTION` or `AUTHENTICATED_ACTION`).
`void`	`setSource(GameCharacter character)` Sets the originator of the action.

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

PUBLIC_ACTION

```
public static final int PUBLIC_ACTION
```

Constant value: **1**

AUTHENTICATED_ACTION

```
public static final int AUTHENTICATED_ACTION
```

Constant value: **2**

Constructors

LabyAction

```
public LabyAction()
```

Methods

getHandler

```
public abstract String getHandler()
```

Returns the name of the handler for the action.

getType

```
public abstract int getType()
```

Must return the type of the action (`PUBLIC_ACTION` or `AUTHENTICATED_ACTION`). This way the class defines itself to which type of action it belongs.

Returns:
the type of the action.

getSource

```
public final GameCharacter getSource()
```

Deprecated. *The definition of this call depends on* `suspend`*, which is deprecated. Further, the results of this call were never well-defined.*

Returns the originator (creator) of the action.

setSource

```
public final void setSource(GameCharacter character)
```

Sets the originator of the action.

Package
com.jlaby.action.actions

com.jlaby.action.actions
Class CreateUserAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
      |
      +-com.jlaby.action.actions.CreateUserAction
```

All Implemented Interfaces:
Serializable

public class **CreateUserAction**
extends LabyAction

Action for creating a new character (user)

Fields inherited from class `com.jlaby.action.LabyAction`
`AUTHENTICATED_ACTION`, `PUBLIC_ACTION`

Constructor Summary

`public`	`CreateUserAction(String name, String password1, String password2)`

Method Summary

`String`	`getCharacterName()`
`String`	`getHandler()` Returns the name of the handler for the action.
`String`	`getPassword1()`
`String`	`getPassword2()`
`int`	`getType()` Returns the type of the action (authenticated or public).

Methods inherited from class `com.jlaby.action.LabyAction`
`getHandler`, `getSource`, `getType`, `setSource`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

CreateUserAction

```
public CreateUserAction(String name,
                        String password1,
                        String password2)
```

Methods

getCharacterName

```
public String getCharacterName()
```

getPassword1

```
public String getPassword1()
```

getPassword2

```
public String getPassword2()
```

getType

```
public int getType()
```

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

```
public String getHandler()
```

Returns the name of the handler for the action.

com.jlaby.action.actions
Class LoginAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
     |
     +-com.jlaby.action.actions.LoginAction
```

All Implemented Interfaces:
Serializable

public class **LoginAction**
extends LabyAction

Action for character logging in

Fields inherited from class `com.jlaby.action.LabyAction`
`AUTHENTICATED_ACTION`, `PUBLIC_ACTION`

Constructor Summary

`public`	`LoginAction(String characterName, String password)`

Method Summary

`String`	`getCharacterName()`
`String`	`getHandler()` Returns the name of the handler for the action.
`String`	`getPassword()`
`int`	`getType()` Returns the type of the action (authenticated or public).

Methods inherited from class `com.jlaby.action.LabyAction`
`getHandler, getSource, getType, setSource`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

LoginAction

```
public LoginAction(String characterName,
                   String password)
```

Methods

getType

```
public int getType()
```

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

```
public String getHandler()
```

Returns the name of the handler for the action.

getCharacterName

```
public String getCharacterName()
```

getPassword

```
public String getPassword()
```

com.jlaby.action.actions
Class LoginLocalAction

```
java.lang.Object
   |
   +-com.jlaby.action.LabyAction
       |
       +-com.jlaby.action.actions.LoginLocalAction
```

All Implemented Interfaces:
Serializable

public class **LoginLocalAction**
extends LabyAction

Action for character logging in locally

Fields inherited from class com.jlaby.action.LabyAction
AUTHENTICATED_ACTION, PUBLIC_ACTION

Constructor Summary

public	LoginLocalAction(String characterName, String password)

Method Summary

String	getCharacterName()
String	getHandler() Returns the name of the handler for the action.
String	getPassword()
int	getType() Returns the type of the action (authenticated or public).

Methods inherited from class com.jlaby.action.LabyAction
getHandler, getSource, getType, setSource

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

LoginLocalAction

```
public LoginLocalAction(String characterName,
                        String password)
```

Methods

getType

```
public int getType()
```

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

```
public String getHandler()
```

Returns the name of the handler for the action.

getCharacterName

```
public String getCharacterName()
```

getPassword

```
public String getPassword()
```

com.jlaby.action.actions
Class LogoutAction

```
java.lang.Object
    |
    +-com.jlaby.action.LabyAction
        |
        +-com.jlaby.action.actions.LogoutAction
```

All Implemented Interfaces:
Serializable

public class **LogoutAction**
extends LabyAction

Action for character logging out of the Laby.

Fields inherited from class com.jlaby.action.LabyAction
AUTHENTICATED_ACTION, PUBLIC_ACTION

Constructor Summary

public	LogoutAction()

Method Summary

String	getHandler() Returns the name of the handler for the action.
int	getType() Returns the type of the action (authenticated or public).

Methods inherited from class com.jlaby.action.LabyAction
getHandler, getSource, getType, setSource

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

LogoutAction

public **LogoutAction**()

Methods

(continued from last page)

getType

```
public int getType()
```

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

```
public String getHandler()
```

Returns the name of the handler for the action.

com.jlaby.action.actions
Class NoAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
      |
      +-com.jlaby.action.actions.NoAction
```

All Implemented Interfaces:
Serializable

public class **NoAction**
extends LabyAction

Action for no action ("...nothing ever happens....")

Fields inherited from class com.jlaby.action.LabyAction
AUTHENTICATED_ACTION, PUBLIC_ACTION

Constructor Summary

public	NoAction()

Method Summary

String	getHandler() Returns the name of the handler for the action.
int	getType() Returns the type of the action (authenticated or public).

Methods inherited from class com.jlaby.action.LabyAction
getHandler, getSource, getType, setSource

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

NoAction

public **NoAction**()

Methods

(continued from last page)

getType

```
public int getType()
```

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

```
public String getHandler()
```

Returns the name of the handler for the action.

com.jlaby.action.actions
Class ShowGuiAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
      |
      +-com.jlaby.action.actions.ShowGuiAction
```

All Implemented Interfaces:
Serializable

public class **ShowGuiAction**
extends LabyAction

Action for showing a specific GUI (HTML page) to the user, such as a login etc.

Field Summary

public static final	CREATEUSER Value: **1**
public static final	LOGIN Value: **0**

Fields inherited from class com.jlaby.action.LabyAction
AUTHENTICATED_ACTION, PUBLIC_ACTION

Constructor Summary

public	ShowGuiAction(int gui) Constructor used by the client to create action objects.

Method Summary

int	getGui()
String	getHandler() Returns the name of the handler for the action.
int	getType() Returns the type of the action (authenticated or public).

Methods inherited from class com.jlaby.action.LabyAction
getHandler, getSource, getType, setSource

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

LOGIN

```
public static final int LOGIN
```

Constant value: **0**

CREATEUSER

```
public static final int CREATEUSER
```

Constant value: **1**

Constructors

ShowGuiAction

```
public ShowGuiAction(int gui)
```

Constructor used by the client to create action objects.

Methods

getGui

```
public final int getGui()
```

getType

```
public int getType()
```

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

```
public String getHandler()
```

Returns the name of the handler for the action.

com.jlaby.action.actions
Class TurnAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
      |
      +-com.jlaby.action.actions.TurnAction
```

All Implemented Interfaces:
Serializable

public class **TurnAction**
extends LabyAction

Action for character turning around

Field Summary

`public static final`	`AROUND` Value: **0**
`public static final`	`LEFT` Value: **2**
`public static final`	`RIGHT` Value: **1**

Fields inherited from class `com.jlaby.action.LabyAction`

`AUTHENTICATED_ACTION`, `PUBLIC_ACTION`

Constructor Summary

`public`	`TurnAction(int side)` Constructor used by the client to create action objects.

Method Summary

`String`	`getHandler()` Returns the name of the handler for the action.
`int`	`getSide()`
`int`	`getType()` Returns the type of the action (authenticated or public).

Methods inherited from class `com.jlaby.action.LabyAction`

`getHandler`, `getSource`, `getType`, `setSource`

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Fields

AROUND

`public static final int **AROUND**`

Constant value: **0**

RIGHT

`public static final int **RIGHT**`

Constant value: **1**

LEFT

`public static final int **LEFT**`

Constant value: **2**

Constructors

TurnAction

`public **TurnAction**(int side)`

Constructor used by the client to create action objects.

Methods

getSide

`public final int **getSide**()`

getType

`public int **getType**()`

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

`public String **getHandler**()`

(continued from last page)

Returns the name of the handler for the action.

com.jlaby.action.actions
Class WalkAction

```
java.lang.Object
  |
  +-com.jlaby.action.LabyAction
      |
      +-com.jlaby.action.actions.WalkAction
```

All Implemented Interfaces:
Serializable

public class **WalkAction**
extends LabyAction

Walking action (walk one step forward)

Fields inherited from class `com.jlaby.action.LabyAction`
`AUTHENTICATED_ACTION`, `PUBLIC_ACTION`

Constructor Summary

`public`	`WalkAction()`

Method Summary

`String`	`getHandler()` Returns the name of the handler for the action.
`int`	`getType()` Returns the type of the action (authenticated or public).

Methods inherited from class `com.jlaby.action.LabyAction`
`getHandler`, `getSource`, `getType`, `setSource`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

WalkAction

```
public WalkAction()
```

Methods

(continued from last page)

getType

`public int **getType**()`

Returns the type of the action (authenticated or public).

Returns:
The type of the action.

getHandler

`public String **getHandler**()`

Returns the name of the handler for the action.

Package com.jlaby.action.exception

com.jlaby.action.exception
Class UnsupportedActionException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.action.exception.UnsupportedActionException
```

All Implemented Interfaces:
Serializable

public class **UnsupportedActionException**
extends LabyException

Exception for cases where the client sent an Action object for which no appropriate handler exists (although this should not be possible at all as of now - maybe in future versions the whole action object stuff will become completely dynamic).

Constructor Summary

public	`UnsupportedActionException()` Constructs an unspecific exception.
public	`UnsupportedActionException(String msg)` Constructs an exception with a message text.
public	`UnsupportedActionException(Exception e)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

UnsupportedActionException

`public UnsupportedActionException()`

(continued from last page)

Constructs an unspecific exception.

UnsupportedActionException

`public` **`UnsupportedActionException`**`(String msg)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

UnsupportedActionException

`public` **`UnsupportedActionException`**`(Exception e)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

Package
com.jlaby.action.handlers

com.jlaby.action.handlers
Class ActionHandlerFactory

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.ActionHandlerFactory
```

public class **ActionHandlerFactory**
extends Object

This class creates laby action handler objects based on the given action object. Every `LabyAction` class defines its handler through its `getHandler()` method return value.

Constructor Summary

`public`	`ActionHandlerFactory()`

Method Summary

`static IActionHandler`	`getActionHandler(LabyAction actionObject)` Factory method which creates Singleton instances of handlers for certain types of actions.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ActionHandlerFactory

```
public ActionHandlerFactory()
```

Methods

getActionHandler

```
public static IActionHandler getActionHandler(LabyAction actionObject)
  throws UnsupportedActionException
```

Factory method which creates Singleton instances of handlers for certain types of actions.

Parameters:
`actionObject` - the laby action for which a handler is required.

Returns:
the handler for this action.

Throws:

(continued from last page)

`UnsupportedActionException` - if no appropriate handler for the action could be created. One possible case is that the class could not be found.

com.jlaby.action.handlers
Class CreateUserActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.CreateUserActionHandler
```

All Implemented Interfaces:
IActionHandler

public class **CreateUserActionHandler**
extends Object
implements IActionHandler

This class handles the action of creating a new user (character).

Constructor Summary

public	CreateUserActionHandler()

Method Summary

void	performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action) Creates a new user entry in the Laby user database.

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

performActionAsSoonAsPossibleBeforeItsTooLate

Constructors

CreateUserActionHandler

```
public CreateUserActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Creates a new user entry in the Laby user database.

Parameters:

(continued from last page)

`action` - A CreateUserAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`CreateUserAction`

com.jlaby.action.handlers
Interface IActionHandler

All Known Implementing Classes:

NoActionHandler, LoginLocalActionHandler, LoginActionHandler, TurnActionHandler, WalkActionHandler, LogoutActionHandler, CreateUserActionHandler, ShowGuiActionHandler

public interface **IActionHandler**
extends

Interface for handlers of actions. This interface is needed only on the server side. NOTE: Implementations of this interface must be completely thread-safe since only one (Singleton) instance will be used to handle all actions of each given type.

Method Summary

`void`	`performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)` Carries out some action(s) based on the action object.

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Carries out some action(s) based on the action object.

Parameters:

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException` - thrown if the action could not be handled.

com.jlaby.action.handlers
Class LoginActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.LoginActionHandler
```

All Implemented Interfaces:

IActionHandler

public class **LoginActionHandler**
extends Object
implements IActionHandler

This class handles a login action of a character.

Constructor Summary

`public`	`LoginActionHandler()`

Method Summary

`void`	`performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)` Executes the log in procedure for a user and sets the ID of the user in the HTTP session.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

LoginActionHandler

```
public LoginActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Executes the log in procedure for a user and sets the ID of the user in the HTTP session.

Parameters:

(continued from last page)

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`LoginAction`

com.jlaby.action.handlers
Class LoginLocalActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.LoginLocalActionHandler
```

All Implemented Interfaces:
IActionHandler

public class **LoginLocalActionHandler**
extends Object
implements IActionHandler

This class handles a local login action of a character.

Constructor Summary

public	LoginLocalActionHandler()

Method Summary

void	performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action) Executes the log in procedure for a user and sets the ID of the user in the HTTP session.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

LoginLocalActionHandler

public **LoginLocalActionHandler**()

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Executes the log in procedure for a user and sets the ID of the user in the HTTP session.

Parameters:

(continued from last page)

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`LoginAction`

com.jlaby.action.handlers
Class LogoutActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.LogoutActionHandler
```

All Implemented Interfaces:

IActionHandler

public class **LogoutActionHandler**
extends Object
implements IActionHandler

This class handles a logout action of a character.

Constructor Summary

public	LogoutActionHandler()

Method Summary

void	performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action) Carries out some action(s) based on the action object and the clients request.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

LogoutActionHandler

```
public LogoutActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Carries out some action(s) based on the action object and the clients request.

Parameters:

(continued from last page)

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`LogoutAction`

com.jlaby.action.handlers
Class NoActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.NoActionHandler
```

All Implemented Interfaces:
IActionHandler

public class **NoActionHandler**
extends Object
implements IActionHandler

This class handles "no" actions (created through requests to the Laby servlet without action parameters).

Constructor Summary

`public`	`NoActionHandler()`

Method Summary

`void`	`performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)` Empty method (for "no-op" actions used to just refresh the display, for instance).

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

NoActionHandler

```
public NoActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Empty method (for "no-op" actions used to just refresh the display, for instance).

Parameters:

(continued from last page)

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`NoAction`

com.jlaby.action.handlers
Class ShowGuiActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.ShowGuiActionHandler
```

All Implemented Interfaces:

IActionHandler

public class **ShowGuiActionHandler**
extends Object
implements IActionHandler

This class handles gui display actions. These are actually somewhat similar (logically) to a redirect; instead of continuing the internal logic, the servlet will be forced to send a "display gui xxx" answer to the client by throwing the appropriate exception.

Constructor Summary

`public`	`ShowGuiActionHandler()`

Method Summary

`void`	`performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)` Carries out some action(s) based on the action object and the clients request.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

ShowGuiActionHandler

```
public ShowGuiActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Carries out some action(s) based on the action object and the clients request.

(continued from last page)

Parameters:

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`ShowGuiAction`

com.jlaby.action.handlers
Class TurnActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.TurnActionHandler
```

All Implemented Interfaces:

IActionHandler

public class **TurnActionHandler**
extends Object
implements IActionHandler

This class handles a turning action of a character. It implements turning to the left or right (90 degree) or for 180 degree.

Constructor Summary

public	`TurnActionHandler()`

Method Summary

void	`performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)` Carries out some action(s) based on the action object and the clients request.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

TurnActionHandler

```
public TurnActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Carries out some action(s) based on the action object and the clients request.

Parameters:

(continued from last page)

`action` - A LabyAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`TurnAction`

com.jlaby.action.handlers
Class WalkActionHandler

```
java.lang.Object
  |
  +-com.jlaby.action.handlers.WalkActionHandler
```

All Implemented Interfaces:
IActionHandler

public class **WalkActionHandler**
extends Object
implements IActionHandler

This class handles walking actions of a character. It implements walking by moving the character 1 field forward (in its viewing direction).

Constructor Summary

public	`WalkActionHandler()`

Method Summary

void	`performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)` Checks if the game character can walk (i.e.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.action.handlers.IActionHandler`

`performActionAsSoonAsPossibleBeforeItsTooLate`

Constructors

WalkActionHandler

```
public WalkActionHandler()
```

Methods

performActionAsSoonAsPossibleBeforeItsTooLate

```
public void performActionAsSoonAsPossibleBeforeItsTooLate(LabyAction action)
  throws LabyException
```

Checks if the game character can walk (i.e. if there is nothing standing in its way) and if so, changes its coordinates accordingly.

(continued from last page)

Parameters:

`action` - A WalkAction object created based on the client HTTP request.

Throws:

`LabyException`

See Also:

`WalkAction`

Package
com.jlaby.admin.jdbc

com.jlaby.admin.jdbc
Class AbstractBatchJob

```
java.lang.Object
  |
  +-com.jlaby.admin.jdbc.AbstractBatchJob
```

All Implemented Interfaces:

ILabyConstants

Direct Known Subclasses:

ClearCharacterTable, BatchJob, ListCharacterTable

public abstract class **AbstractBatchJob**
extends Object
implements ILabyConstants

Base class used to create batch programs for database administration and maintenance.

Fields inherited from interface `com.jlaby.util.ILabyConstants`
`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Constructor Summary

`public`	`AbstractBatchJob()` Default constructor.

Method Summary

`abstract void`	`init(String[] args)` Initialisation method where the batch program might initialise any properties and values based on the commandline arguments.
`void`	`process(String[] args)` This method should be called in the main() method of the subclass; it kicks off the batch job process by first calling the "init()" method and then the "run()" method.
`abstract void`	`run()` The "worker" method which must contain the actual logic of the batch job.

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

AbstractBatchJob

```
public AbstractBatchJob()
```

Default constructor. It also reads the "configuration.properties" so that the batch program has access to the same configuration values as the Laby server has.

Methods

init

```
public abstract void init(String[] args)
  throws Exception
```

Initialisation method where the batch program might initialise any properties and values based on the commandline arguments.

Parameters:

`args` - the commandline arguments

Throws:

`java.lang.Exception`

run

```
public abstract void run()
  throws Exception
```

The "worker" method which must contain the actual logic of the batch job.

process

```
public final void process(String[] args)
```

This method should be called in the main() method of the subclass; it kicks off the batch job process by first calling the "init()" method and then the "run()" method.

Parameters:

`args` - the commandline arguments.

com.jlaby.admin.jdbc
Class BatchJob

```
java.lang.Object
  |
  +-com.jlaby.admin.jdbc.AbstractBatchJob
      |
      +-com.jlaby.admin.jdbc.BatchJob
```

All Implemented Interfaces:

ILabyConstants

public class **BatchJob**
extends AbstractBatchJob

Base class used to create batch programs for database administration and maintenance.

Fields inherited from interface `com.jlaby.util.ILabyConstants`
FALSE, PROP_CONFIG_FILE, PROP_DB_ORACLE_SEQNAME, PROP_DB_VENDOR, PROP_HTML_CREATE, PROP_HTML_INVENTORY, PROP_HTML_LOGIN, PROP_HTML_MAINPANEL, PROP_JDBC_DRIVER_CLASS, PROP_JDBC_DRIVER_URL, PROP_LOG_DESC, PROP_LOG_DIR, PROP_LOG_FILE, PROP_LOG_NAME, PROP_LOG_TRACE, PROP_SERVER_PORT, PROP_SQL_NEWUNIQUEID, PROP_SQL_QUERY_CREATEUSER, PROP_SQL_QUERY_GETVIEW, PROP_SQL_QUERY_LOADUSER, PROP_SQL_QUERY_SAVEUSER, PROP_SQL_UNIQUEID_MCKOI, PROP_SQL_UNIQUEID_ORACLE, PROP_URL_CREATE, PROP_URL_LABY, PROP_URL_LOGIN, TRUE

Constructor Summary

`public`	`BatchJob()`

Method Summary

`void`	`init(String[] args)` Initialisation method where the batch program might initialise any properties and values based on the commandline arguments.
`static void`	`main(String[] args)` Main entry method.
`void`	`run()` The "worker" method which must contain the actual logic of the batch job.

Methods inherited from class `com.jlaby.admin.jdbc.AbstractBatchJob`
`init, process, run`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Constructors

BatchJob

```
public BatchJob()
```

Methods

init

```
public void init(String[] args)
  throws Exception
```

Initialisation method where the batch program might initialise any properties and values based on the commandline arguments.

Parameters:

`args` - the commandline arguments

Throws:

`java.lang.Exception`

run

```
public void run()
  throws Exception
```

The "worker" method which must contain the actual logic of the batch job.

main

```
public static void main(String[] args)
```

Main entry method.

Parameters:

`args` - the commandline arguments.

com.jlaby.admin.jdbc
Class ClearCharacterTable

```
java.lang.Object
  |
  +-com.jlaby.admin.jdbc.AbstractBatchJob
      |
      +-com.jlaby.admin.jdbc.ClearCharacterTable
```

All Implemented Interfaces:
ILabyConstants

public class **ClearCharacterTable**
extends AbstractBatchJob

Administration utility class. This class clears the LABY_CHARACTER table.

Fields inherited from interface `com.jlaby.util.ILabyConstants`

`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Constructor Summary

`public`	`ClearCharacterTable()`

Method Summary

`void`	`init(String[] args)` Initialisation (not used here).
`static void`	`main(String[] args)` Main entry method.
`void`	`run()` Carries out the database operations.

Methods inherited from class `com.jlaby.admin.jdbc.AbstractBatchJob`

`init`, `process`, `run`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

ClearCharacterTable

```
public ClearCharacterTable()
```

Methods

main

```
public static void main(String[] args)
```

Main entry method.

Parameters:

`args` - the commandline arguments.

init

```
public void init(String[] args)
  throws Exception
```

Initialisation (not used here).

Parameters:

`args` - the commandline arguments

run

```
public void run()
  throws Exception
```

Carries out the database operations.

com.jlaby.admin.jdbc
Class ListCharacterTable

```
java.lang.Object
  |
  +-com.jlaby.admin.jdbc.AbstractBatchJob
      |
      +-com.jlaby.admin.jdbc.ListCharacterTable
```

All Implemented Interfaces:
ILabyConstants

public class **ListCharacterTable**
extends AbstractBatchJob

This class lists all the entries in the LABY_CHARACTER table.

Fields inherited from interface `com.jlaby.util.ILabyConstants`

`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Constructor Summary

`public`	`ListCharacterTable()`

Method Summary

`void`	`init(String[] args)` Initialisation (not used here).
`static void`	`main(String[] args)` Main entry method.
`void`	`run()` Carries out the database operations.

Methods inherited from class `com.jlaby.admin.jdbc.AbstractBatchJob`

`init`, `process`, `run`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

ListCharacterTable

```
public ListCharacterTable()
```

Methods

main

```
public static void main(String[] args)
```

Main entry method.

Parameters:

`args` - the commandline arguments.

init

```
public void init(String[] args)
  throws Exception
```

Initialisation (not used here).

Parameters:

`args` - the commandline arguments

run

```
public void run()
  throws Exception
```

Carries out the database operations.

Package

com.jlaby.bot

com.jlaby.bot
Class BotBrain

```
java.lang.Object
   |
   +-com.jlaby.bot.BotBrain
```

public class **BotBrain**
extends Object

Base class for bot KI implementations. In order to "deploy" bots in the central bot server, their logic must be implemented in a class derived from this one.

The second option is to create a bot in an own Java VM process which behaves just like a GUI client and may run on any remote client system. For such bots, the client classes should be used.

See Also:
Client

Constructor Summary

public	BotBrain()

Method Summary

LabyAction	think(GameCharacterInfo info) This method must implement the bot's logic.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

BotBrain

```
public BotBrain()
```

Methods

think

```
public LabyAction think(GameCharacterInfo info)
  throws LabyException
```

This method must implement the bot's logic. The default implementation of this method has the bot carrying out 90-degree turns every 3 seconds.

Parameters:
`info` - Information about the bot's environment. Based on this info, the bot must decide what action it wants to take.

(continued from last page)

Returns:

A LabyAction object. May be null if the bot does not want to take any action.

Throws:

`LabyException` - If the bot encountered an internal problem

Package com.jlaby.bot.server

com.jlaby.bot.server
Class BotThread

```
java.lang.Object
  |
  +-com.jlaby.bot.server.BotThread
```

All Implemented Interfaces:
Runnable

public class **BotThread**
extends Object
implements Runnable

This class implements the thread which is responsible for "driving" the bots (non-player-characters) in the laby game.

This class could have been derived from "java.lang.Thread" as well, but I prefer to stick to interfaces to retain more flexibility (hence the "start()" method).

Constructor Summary

`public`	`BotThread(IWorld world)` Constructs a BotThread.

Method Summary

`void`	`run()` Contains the logic which drives the bots.
`void`	`start()` Creates a new thread and runs this class in it.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `java.lang.Runnable`

`run`

Constructors

BotThread

`public BotThread(IWorld world)`

Constructs a BotThread.

Parameters:

`world` - Instance of the world in which

(continued from last page)

Methods

start

`public void **start**()`

Creates a new thread and runs this class in it.

run

`public void **run**()`

Contains the logic which drives the bots.

com.jlaby.bot.server
Class LabyBotServer

```
java.lang.Object
  |
  +-com.jlaby.bot.server.LabyBotServer
```

public class **LabyBotServer**
extends Object

Server for running multiple Laby bots within one Java VM instance.

Constructor Summary

public	LabyBotServer()

Method Summary

static void	main(String[] args) The main method used to start the server.

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

LabyBotServer

`public LabyBotServer()`

Methods

main

`public static void main(String[] args)`

The main method used to start the server. It takes the path of the configuration properties file as a single commandline parameter.

Parameters:

`args` - the commandline arguments.

Package

com.jlaby.character

com.jlaby.character
Class CharacterState

```
java.lang.Object
  |
  +-com.jlaby.character.CharacterState
```

public class **CharacterState**
extends Object

Helper class which handles state persistency of GameCharacter objects.

Constructor Summary

public	CharacterState()

Method Summary

static GameCharacter	load(String name) Load the GameCharacter from the database.
static void	save(GameCharacter character) Save the GameCharacter into the database.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

CharacterState

public **CharacterState**()

Methods

load

```
public static GameCharacter load(String name)
  throws UnknownUserException
```

Load the GameCharacter from the database.

Parameters:

name - The (unique) name of the game character.

Returns:

A GameCharacter object holding all information.

(continued from last page)

Throws:

`UnknownUserException` - If no entry with the given name could be found in the database.

save

```
public static void save(GameCharacter character)
  throws LabySQLException
```

Save the GameCharacter into the database.

Parameters:

`character` - The game character that should be saved.

Throws:

`LabySQLException` - thrown in case of database problems.

com.jlaby.character
Class GameCharacter

```
java.lang.Object
  |
  +-com.jlaby.character.GameCharacter
```

public class **GameCharacter**
extends Object

Object in the laby.

Field Summary

public static final	AWAKE Active user (logged in) Value: **1**
public static final	NON_PLAYER Value: **2**
public static final	PLAYER Type values Value: **1**
public static final	SLEEPING Inactive user (not logged in) Value: **0**

Constructor Summary

public	GameCharacter()

Method Summary

String	getAlias()
int	getDirection()
int	getID()
int	getLife()
String	getName()
int	getState()
int	getX()

int	getY()
int	getZ()
boolean	isStateful()
void	setAlias(String alias)
void	setDirection(int value)
void	setID(int id)
void	setLife(int value)
void	setName(String name)
void	setState(int value)
void	setStateful(boolean value)
void	setX(int value)
void	setY(int value)
void	setZ(int value)
String	toString()

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

PLAYER

`public static final int` **`PLAYER`**

Type values
Constant value: **1**

NON_PLAYER

`public static final int` **`NON_PLAYER`**

Constant value: **2**

(continued from last page)

SLEEPING

```
public static final int SLEEPING
```

Inactive user (not logged in)
Constant value: **0**

AWAKE

```
public static final int AWAKE
```

Active user (logged in)
Constant value: **1**

Constructors

GameCharacter

```
public GameCharacter()
```

Methods

toString

```
public String toString()
```

setID

```
public void setID(int id)
```

getID

```
public int getID()
```

setName

```
public void setName(String name)
```

getName

```
public String getName()
```

setAlias

```
public void setAlias(String alias)
```

(continued from last page)

getAlias

public String **getAlias**()

isStateful

public boolean **isStateful**()

setStateful

public void **setStateful**(boolean value)

setX

public void **setX**(int value)

getX

public int **getX**()

setY

public void **setY**(int value)

getY

public int **getY**()

setZ

public void **setZ**(int value)

getZ

public int **getZ**()

(continued from last page)

setDirection

public void **setDirection**(int value)

getDirection

public int **getDirection**()

setState

public void **setState**(int value)

getState

public int **getState**()

setLife

public void **setLife**(int value)

getLife

public int **getLife**()

Package
com.jlaby.client

com.jlaby.client
Class Client

```
java.lang.Object
  |
  +-com.jlaby.client.Client
```

Direct Known Subclasses:
TestClient

public abstract class **Client**
extends Object

Abstract base class for Java clients. It provides the functionality to connect and log in to a Laby server, send requests for Actions etc. All the HTTP communication is hidden and completely transparent to the developer.

Constructor Summary

public	`Client()`

Method Summary

static void	`centerOnScreen(java.awt.Component component)` This method takes a Component such as a Dialog or Window and calculates the coordinates appropriate to center it on the screen.
GameCharacterInfo	`getCharacterInfo()`
boolean	`isLoggedIn()` Determines if the client is currently logged in in a Laby world.
void	`login(String url, String username, String password)` Connects the client with a Laby HTTP server (servlet) and logs in with the given user account.
void	`logout()` Logs the current user out from the Laby world.
LabyAnswer	`performAction(LabyAction action)` Let the world handle any kind of action and return an appropriate answer.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Client

```
public Client()
```

(continued from last page)

Methods

login

```
public final void login(String url,
          String username,
          String password)
  throws LabyException
```

Connects the client with a Laby HTTP server (servlet) and logs in with the given user account. This method actually just sends a `LoginAction` object.

Parameters:

`url` - the URL of the Laby server

`username` - the name of the Laby user

`password` - the login password

Throws:

`LabyException` - thrown if connect or login failed (see causing exception for details)

getCharacterInfo

```
public final GameCharacterInfo getCharacterInfo()
```

performAction

```
public final LabyAnswer performAction(LabyAction action)
  throws LabyException
```

Let the world handle any kind of action and return an appropriate answer.

Parameters:

`action` - the action which the client wants to perform.

Returns:

the answer returned by the Laby world.

Throws:

`LabyException`

logout

```
public final void logout()
  throws LabyException
```

Logs the current user out from the Laby world.

isLoggedIn

```
public final boolean isLoggedIn()
```

Determines if the client is currently logged in in a Laby world.

(continued from last page)

centerOnScreen

```
public static void centerOnScreen(java.awt.Component component)
```

This method takes a Component such as a Dialog or Window and calculates the coordinates appropriate to center it on the screen. It then changes the coordinates of the given component to center it.

Parameters:

`component` - the component that should be centered on the screen

com.jlaby.client
Class GameCharacterInfo

```
java.lang.Object
  |
  +-com.jlaby.client.GameCharacterInfo
```

All Implemented Interfaces:
Serializable

public class **GameCharacterInfo**
extends Object
implements Serializable

Contains the information about a game character which should be available to this character. For example, its absolute position within the Laby should NOT be available to the character as long as no GPS-like device has been picked up.

Constructor Summary

public	GameCharacterInfo()

Method Summary

String	getAlias()
int	getDirection()
int	getLife()
String	getName()
Position	getPosition()
int	getState()
ViewObject[]	getViewObjects()
void	setAlias(String alias)
void	setDirection(int value)
void	setLife(int value)
void	setName(String name)
void	setPosition(Position position)
void	setState(int value)

void	setViewObjects(ViewObject[] viewObjects)

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GameCharacterInfo

`public` **`GameCharacterInfo`**`()`

Methods

setViewObjects

`public void` **`setViewObjects`**`(ViewObject[] viewObjects)`

getViewObjects

`public ViewObject[]` **`getViewObjects`**`()`

setName

`public void` **`setName`**`(String name)`

getName

`public String` **`getName`**`()`

setAlias

`public void` **`setAlias`**`(String alias)`

getAlias

`public String` **`getAlias`**`()`

(continued from last page)

setPosition

```
public void setPosition(Position position)
```

getPosition

```
public Position getPosition()
```

setDirection

```
public void setDirection(int value)
```

getDirection

```
public int getDirection()
```

setState

```
public void setState(int value)
```

getState

```
public int getState()
```

setLife

```
public void setLife(int value)
```

getLife

```
public int getLife()
```

com.jlaby.client
Class LabyAnswer

```
java.lang.Object
  |
  +-com.jlaby.client.LabyAnswer
```

All Implemented Interfaces:
Serializable

Direct Known Subclasses:
LoginAnswer, GameInfoAnswer

public abstract class **LabyAnswer**
extends Object
implements Serializable

Abstract base class for answers sent from the Laby server to Java clients. The server uses these classes to create an answer in a simple way, abstracted from the network transport mechanism, while the client uses it to deal with server answers in a more flexibel and extensible way.

Field Summary

public static final	ADMIN_RELATED Value: **300**
public static final	GAME_RELATED Value: **200**
public static final	LOGIN_RELATED Value: **100**

Constructor Summary

public	LabyAnswer(int type) Constructor used by the Laby server to create answer objects.

Method Summary

String	getClassName() Returns the name of the answer.
int	getType() Returns the type of the answer.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Fields

LOGIN_RELATED

`public static final int` **`LOGIN_RELATED`**

Constant value: **`100`**

GAME_RELATED

`public static final int` **`GAME_RELATED`**

Constant value: **`200`**

ADMIN_RELATED

`public static final int` **`ADMIN_RELATED`**

Constant value: **`300`**

Constructors

LabyAnswer

`public` **`LabyAnswer`**`(int type)`

Constructor used by the Laby server to create answer objects.

Parameters:
`type` - the type of answer (see constants).

Methods

getClassName

`public final String` **`getClassName`**`()`

Returns the name of the answer. Must be implemented specifically by each answer class. This method can be used by the client to determine the exact name (type) of the class for further casting.

getType

`public final int` **`getType`**`()`

Returns the type of the answer.

Returns:
the type of the answer (see constants).

Package

com.jlaby.client.answers

com.jlaby.client.answers
Class GameInfoAnswer

```
java.lang.Object
  |
  +-com.jlaby.client.LabyAnswer
    |
    +-com.jlaby.client.answers.GameInfoAnswer
```

All Implemented Interfaces:
Serializable

public class **GameInfoAnswer**
extends LabyAnswer

Answer for an authenticated and active client which informs informs him about the current state of the character etc.

Fields inherited from class `com.jlaby.client.LabyAnswer`
`ADMIN_RELATED`, `GAME_RELATED`, `LOGIN_RELATED`

Constructor Summary

`public`	`GameInfoAnswer(GameCharacterInfo characterInfo)` The detail type of the login related answer.

Method Summary

`GameCharacterInfo`	`getCharacterInfo()` Returns information about the game character.

Methods inherited from class `com.jlaby.client.LabyAnswer`
`getClassName`, `getType`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

GameInfoAnswer

`public GameInfoAnswer(GameCharacterInfo characterInfo)`

The detail type of the login related answer.

Parameters:
`detailType` - must be one of the constants.

Methods

(continued from last page)

getCharacterInfo

```
public GameCharacterInfo getCharacterInfo()
```

Returns information about the game character.

Returns:

a GameCharacterInfo object which contains filtered information (it contains all the information the GameCharacter is allowed to know).

com.jlaby.client.answers
Class LoginAnswer

```
java.lang.Object
  |
  +-com.jlaby.client.LabyAnswer
      |
      +-com.jlaby.client.answers.LoginAnswer
```

All Implemented Interfaces:
Serializable

public class **LoginAnswer**
extends LabyAnswer

Answer for a client that needs to carry out some login-related task or to be informed about something like that (login required, login failed, user unknown etc.)

Field Summary

`public static final`	`LOGIN_FAILED` Value: **200**
`public static final`	`LOGIN_NEWUSER_CREATION` Value: **300**
`public static final`	`LOGIN_REQUIRED` Value: **100**

Fields inherited from class `com.jlaby.client.LabyAnswer`

`ADMIN_RELATED`, `GAME_RELATED`, `LOGIN_RELATED`

Constructor Summary

`public`	`LoginAnswer(int detailType)` The detail type of the login related answer.

Method Summary

`int`	`getDetailType()` Returns the type of login-related answer.

Methods inherited from class `com.jlaby.client.LabyAnswer`

`getClassName, getType`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

LOGIN_REQUIRED

```
public static final int LOGIN_REQUIRED
```

Constant value: **100**

LOGIN_FAILED

```
public static final int LOGIN_FAILED
```

Constant value: **200**

LOGIN_NEWUSER_CREATION

```
public static final int LOGIN_NEWUSER_CREATION
```

Constant value: **300**

Constructors

LoginAnswer

```
public LoginAnswer(int detailType)
```

The detail type of the login related answer.

Parameters:
`detailType` - must be one of the constants.

Methods

getDetailType

```
public int getDetailType()
```

Returns the type of login-related answer.

Returns:
the type (must be one of the constants)

Package
com.jlaby.client.exception

com.jlaby.client.exception
Class RemoteException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.client.exception.RemoteException
```

All Implemented Interfaces:
Serializable

public class **RemoteException**
extends LabyException

This exception is thrown when a problem occurred during the creation or transmission of a HTTP POST request.

Constructor Summary

public	`RemoteException()` Constructs an unspecific exception.
public	`RemoteException(String txt)` Constructs an exception with a message text.
public	`RemoteException(Throwable causingException)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

RemoteException

`public RemoteException()`

Constructs an unspecific exception.

RemoteException

```
public RemoteException(String txt)
```

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

RemoteException

```
public RemoteException(Throwable causingException)
```

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.client.exception
Class UnsupportedAnswerException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.client.exception.UnsupportedAnswerException
```

All Implemented Interfaces:
Serializable

public class **UnsupportedAnswerException**
extends LabyException

Exception thrown when an answer sent from the server cannot be handled by the client.

Constructor Summary

`public`	`UnsupportedAnswerException()` Constructs an unspecific exception.
`public`	`UnsupportedAnswerException(String msg)` Constructs an exception with a message text.
`public`	`UnsupportedAnswerException(Exception e)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

UnsupportedAnswerException

```
public UnsupportedAnswerException()
```

Constructs an unspecific exception.

UnsupportedAnswerException

```
public UnsupportedAnswerException(String msg)
```

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

UnsupportedAnswerException

```
public UnsupportedAnswerException(Exception e)
```

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

Package com.jlaby.client.handler

com.jlaby.client.handler
Class BrowserClientHandler

```
java.lang.Object
  |
  +-com.jlaby.client.handler.BrowserClientHandler
```

All Implemented Interfaces:

ILabyConstants, IClientHandler

public class **BrowserClientHandler**
extends Object
implements IClientHandler, ILabyConstants

Handler for Webbrowser clients. It takes their simple GET-requests (with parameters) and creates `LabyAction` objects based on those parameters.

Mapping of LabyAction classes and GET parameters:

Fields inherited from interface `com.jlaby.util.ILabyConstants`

`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Constructor Summary

`public`	`BrowserClientHandler()` Constructs the Browser client handler.

Method Summary

`LabyAction`	`convertRequest(HttpServletRequest servletRequest)` Every handler must convert the request sent from the client into the appropriate Action object.
`void`	`sendAnswer(HttpServletResponse servletResponse, LabyAnswer answer)` This method has to create the appropriate HTML page for the answer given by the server and send it back to the Browser client.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.client.handler.IClientHandler`

`convertRequest`, `sendAnswer`

(continued from last page)

Constructors

BrowserClientHandler

```
public BrowserClientHandler()
```

Constructs the Browser client handler. This constructor reads the names of the HTML templates for the pages to be sent to the client from the configuration properties.

Methods

convertRequest

```
public LabyAction convertRequest(HttpServletRequest servletRequest)
  throws UnsupportedActionException
```

Every handler must convert the request sent from the client into the appropriate Action object. In case of a browser, the request is a simple string consisting of parameter/value pairs. In case of a Java client, the request will be a string with a base64 encoded serialized Action object.

Parameters:

`servletRequest` - the request sent from the client

Throws:

`UnsupportedActionException` - thrown if the request could not successfully be converted into a `LabyAction` object.

sendAnswer

```
public void sendAnswer(HttpServletResponse servletResponse,
          LabyAnswer answer)
```

This method has to create the appropriate HTML page for the answer given by the server and send it back to the Browser client.

Parameters:

`servletResponse` - the servlets response object

`answer` - the answer given by the server

com.jlaby.client.handler
Class ClientHandlerFactory

```
java.lang.Object
   |
   +-com.jlaby.client.handler.ClientHandlerFactory
```

public class **ClientHandlerFactory**
extends Object

This class creates instances of client handlers for client requests.

Field Summary

public static final	BROWSER_CLIENT Value: **0**
public static final	JAVA_CLIENT Value: **1**

Constructor Summary

public	ClientHandlerFactory()

Method Summary

static IClientHandler	getClientHandler(HttpServletRequest request) This method returns an appropriate handler for a given client request, based on the type of the client (which is determined from the requests headers).

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

BROWSER_CLIENT

`public static final int` **`BROWSER_CLIENT`**

Constant value: **0**

JAVA_CLIENT

`public static final int` **`JAVA_CLIENT`**

Constant value: **1**

(continued from last page)

Constructors

ClientHandlerFactory

`public **ClientHandlerFactory**()`

Methods

getClientHandler

```
public static IClientHandler getClientHandler(HttpServletRequest request)
```

This method returns an appropriate handler for a given client request, based on the type of the client (which is determined from the requests headers).

Supported HTTP header types so far:

`UserAgent:`
Type: `JavaLabyClient/1.0 [classname]`

where `classname` can be empty or, theoretically, the classname of an appropriate handler for the client. But as of now, only one type of Java client handler exists and this value will be ignored by the server.

If this HTTP header contains anything else than the value described above, the server will assume that the client is a web browser and use the `BrowserClientHandler`.

Parameters:

`request` - the HTTP servlet request sent from the client.

com.jlaby.client.handler
Interface IClientHandler

All Known Implementing Classes:

BrowserClientHandler, JavaClientHandler

---

public interface **IClientHandler**
extends

Interface for classes handling the response for a certain type of laby client. Note that an implementation of this interface must be thread-safe because only one instance of each handler type will be used within the servlet VM to serve all requests (from one type of client).

Method Summary

`LabyAction`	`convertRequest(HttpServletRequest servletRequest)` Every handler must convert the request sent from the client into the appropriate Action object.
`void`	`sendAnswer(HttpServletResponse servletResponse, LabyAnswer answer)` This method is called for answers of type ANSWER_STATE.

Methods

convertRequest

```
public LabyAction convertRequest(HttpServletRequest servletRequest)
  throws UnsupportedActionException
```

Every handler must convert the request sent from the client into the appropriate Action object. In case of a browser, the request is a simple string consisting of parameter/value pairs. In case of a Java client, the request will be a string with a base64 encoded serialized Action object.

Parameters:

`servletRequest` - the request sent from the client

---

sendAnswer

```
public void sendAnswer(HttpServletResponse servletResponse,
         LabyAnswer answer)
```

This method is called for answers of type ANSWER_STATE. It must return appropriate data for the client based on the type of the client. In case of an intelligent (Java) client, which can render its own 3D view, the data sent back should just be information about the objects. In case of a dumb (Browser) client, the answer must be a complete HTML page.

Parameters:

`servletResponse` - the servlets response object. The output stream of this response object must be used to send the answer back to the client.
`answer` - the answer information object

com.jlaby.client.handler
Class JavaClientHandler

```
java.lang.Object
   |
   +-com.jlaby.client.handler.JavaClientHandler
```

All Implemented Interfaces:
IClientHandler

public class **JavaClientHandler**
extends Object
implements IClientHandler

Handler for Java clients (J2SE or MIDP). This class uses serialization to retrieve Action objects from the client and send Answer objects back. All objects have to be Base64-encoded during the transmission.

Constructor Summary

public	`JavaClientHandler()`

Method Summary

`LabyAction`	`convertRequest(HttpServletRequest servletRequest)` Every handler must convert the request sent from the client into the appropriate Action object.
`void`	`sendAnswer(HttpServletResponse servletResponse, LabyAnswer answer)` This method is called for answers of type ANSWER_STATE.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.client.handler.IClientHandler`

`convertRequest, sendAnswer`

Constructors

JavaClientHandler

`public **JavaClientHandler**()`

Methods

(continued from last page)

convertRequest

```
public LabyAction convertRequest(HttpServletRequest servletRequest)
  throws UnsupportedActionException
```

Every handler must convert the request sent from the client into the appropriate Action object. In case of a browser, the request is a simple string consisting of parameter/value pairs. In case of a Java client, the request will be a string with a base64 encoded serialized Action object.

Parameters:

`servletRequest` - the request sent from the client

Throws:

`UnsupportedActionException` - thrown if the request could not successfully be converted into a `LabyAction` object.

sendAnswer

```
public void sendAnswer(HttpServletResponse servletResponse,
         LabyAnswer answer)
```

This method is called for answers of type ANSWER_STATE. It must return appropriate data for the client based on the type of the client. In case of an intelligent (Java) client, which can render its own 3D view, the data sent back should just be information about the objects. In case of a dumb (Browser) client, the answer must be a complete HTML page.

Parameters:

`servletResponse` - the servlets response object

`characterInfo` - The object containing information about the game character

Package
com.jlaby.client.http

com.jlaby.client.http
Class ConnectionFactory

```
java.lang.Object
   |
   +-com.jlaby.client.http.ConnectionFactory
```

public class **ConnectionFactory**
extends Object

Helper class for the creation of HTTP connections.

Constructor Summary

public	ConnectionFactory()

Method Summary

static java.net.HttpURLConnection	getHttpConnection(String urlstr)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

ConnectionFactory

```
public ConnectionFactory()
```

Methods

getHttpConnection

```
public static java.net.HttpURLConnection getHttpConnection(String urlstr)
  throws RemoteException
```

Package
com.jlaby.client.stub

com.jlaby.client.stub
Class HttpStubHelper

```
java.lang.Object
  |
  +-com.jlaby.client.stub.HttpStubHelper
```

public class **HttpStubHelper**
extends Object

Utility class used to send HTTP requests from a Java client to a Laby server (servlet). This class copes with issues such as serializing and Base64 encoding the Java objects to be sent and vice-versa for the objects retrieved in the servers' answer.

Constructor Summary

public	`HttpStubHelper(String url)` Constructs a HTTP stub helper for communication with a specific URL.
public	`HttpStubHelper(String url, int port)` Constructs a HTTP stub helper for communication with a specific URL and port number.

Method Summary

`LabyAnswer`	`sendAction(LabyAction action)` Serializes and Base64-encodes a given action object and sends it with a POST request to the server URL specified in the constructor.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

HttpStubHelper

```
public HttpStubHelper(String url)
```

Constructs a HTTP stub helper for communication with a specific URL.

Parameters:
`url` - the (HTTP) URL of the target server.

HttpStubHelper

```
public HttpStubHelper(String url,
                      int port)
```

Constructs a HTTP stub helper for communication with a specific URL and port number.

Parameters:
`url` - the (HTTP) URL of the target server.
`port` - the port number.

Methods

sendAction

```
public LabyAnswer sendAction(LabyAction action)
  throws RemoteException
```

Serializes and Base64-encodes a given action object and sends it with a POST request to the server URL specified in the constructor. This method also sets a particular HTTP header for the type of user agent and MIME type in order to allow to the server to distinguish this request from those sent by browser clients.

Parameters:

`action` - the action object that is to be sent.

Returns:

the answer returned from the server.

Throws:

`RemoteException` - thrown if there was any problem.

com.jlaby.client.stub
Class WorldStub

```
java.lang.Object
  |
  +-com.jlaby.client.stub.WorldStub
```

All Implemented Interfaces:

IWorld

public class **WorldStub**
extends Object
implements IWorld

This is the stub used by a client to communicate with a remote world on a server. The stub should NOT be created directly but by using the factory class `WorldFactory`.

Fields inherited from interface `com.jlaby.world.IWorld`
`EAST`, `NORTH`, `SOUTH`, `WEST`

Constructor Summary

`public`	`WorldStub(String url)` Constructs the stub object.
`public`	`WorldStub(String url, int port)` Constructs the stub object.

Method Summary

`LabyAnswer`	`handleAction(LabyAction action)` This method serializes the action and sends it to the "real" world server in a HTTP request (Base64 encoded).
`void`	`initialize()` Initializes the world access stub.

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.world.IWorld`
`handleAction, initialize`

Constructors

WorldStub

```
public WorldStub(String url)
```

(continued from last page)

Constructs the stub object. DO NOT CALL THIS CONSTRUCTOR DIRECTLY! Use the factory method of the class `WorldFactory` instead.

Parameters:

`url` - the HTTP URL of the target laby server (servlet).

See Also:

`WorldFactory`

WorldStub

```
public WorldStub(String url,
                 int port)
```

Constructs the stub object. DO NOT CALL THIS CONSTRUCTOR DIRECTLY! Use the factory method of the class `WorldFactory` instead.

Parameters:

`url` - the HTTP URL of the target laby server (servlet).

See Also:

`WorldFactory`

Methods

initialize

```
public void initialize()
  throws InitializationFailedException
```

Initializes the world access stub.

Throws:

`InitializationFailedException` - if something went wrong and the stub could not be created.

handleAction

```
public LabyAnswer handleAction(LabyAction action)
  throws LabyException
```

This method serializes the action and sends it to the "real" world server in a HTTP request (Base64 encoded).

Parameters:

`action` - the action that should be handled by the server.

Returns:

the answer sent from the server.

Throws:

`LabyException` - thrown if the action could not be handled.

Package com.jlaby.client.view.text

com.jlaby.client.view.text
Class MainPanel

```
java.lang.Object
  |
  +-com.jlaby.html.LabyHtmlPanel
      |
      +-com.jlaby.client.view.text.MainPanel
```

public class **MainPanel**
extends LabyHtmlPanel

This class handles the main HTML page of the laby. It loads the template and fills in the 3d view, the messages and inventory before displaying it.

Constructor Summary

public	MainPanel(String templateResource, HttpServletResponse servletResponse)

Method Summary

void	update(String[] view, String[] messages)

Methods inherited from class com.jlaby.html.LabyHtmlPanel

display, replace, replace, replaceURL

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

MainPanel

```
public MainPanel(String templateResource,
                 HttpServletResponse servletResponse)
```

Methods

update

```
public void update(String[] view,
          String[] messages)
```

com.jlaby.client.view.text
Class MultiSidedGraphics

```
java.lang.Object
  |
  +-com.jlaby.client.view.text.TextGraphics
      |
      +-com.jlaby.client.view.text.MultiSidedGraphics
```

public class **MultiSidedGraphics**
extends TextGraphics

Helper class which makes it easier to implement objects for the laby. It provides some methods to load graphics from resource files in the classpath etc.

Fields inherited from class `com.jlaby.client.view.text.TextGraphics`

`FACING_PLAYER`, `LOOKING_AWAY`, `LOOKING_LEFT`, `LOOKING_RIGHT`, `m_color`, `m_colorType`, `m_data`, `m_name`, `m_type`, `MULTI_COLOR`, `MULTI_SIDED`, `SINGLE_COLOR`, `SINGLE_SIDED`, `UNDEFINED`

Constructor Summary

`public`	`MultiSidedGraphics()`

Method Summary

`String[]`	`getGraphicsData(int position, int direction, int distance)` Default implementation for multi-sided objects.
`String[]`	`getMaskData(int position, int direction, int distance)` Default implementation for multi-sided objects.

Methods inherited from class `com.jlaby.client.view.text.TextGraphics`

`getColor, getColorCharacter, getColorType, getColorValue, getColorValue, getGraphics, getGraphicsData, getMask, getMaskData, getName, getType, init, initialize`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MultiSidedGraphics

`public MultiSidedGraphics()`

Methods

(continued from last page)

getGraphicsData

```
public String[] getGraphicsData(int position,
         int direction,
         int distance)
```

Default implementation for multi-sided objects.

getMaskData

```
public String[] getMaskData(int position,
         int direction,
         int distance)
```

Default implementation for multi-sided objects.

com.jlaby.client.view.text
Class SingleSidedGraphics

```
java.lang.Object
  |
  +-com.jlaby.client.view.text.TextGraphics
      |
      +-com.jlaby.client.view.text.SingleSidedGraphics
```

public class **SingleSidedGraphics**
extends TextGraphics

Helper class which makes it easier to implement objects for the laby. It provides some methods to load graphics from resource files in the classpath etc.

Fields inherited from class `com.jlaby.client.view.text.TextGraphics`

FACING_PLAYER, LOOKING_AWAY, LOOKING_LEFT, LOOKING_RIGHT, m_color, m_colorType, m_data, m_name, m_type, MULTI_COLOR, MULTI_SIDED, SINGLE_COLOR, SINGLE_SIDED, UNDEFINED

Constructor Summary

public	SingleSidedGraphics()

Method Summary

String[]	getGraphicsData(int position, int direction, int distance) Default implementation for single-sided objects.
String[]	getMaskData(int position, int direction, int distance) Default implementation for single-sided objects.

Methods inherited from class `com.jlaby.client.view.text.TextGraphics`

getColor, getColorCharacter, getColorType, getColorValue, getColorValue, getGraphics, getGraphicsData, getMask, getMaskData, getName, getType, init, initialize

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

SingleSidedGraphics

`public SingleSidedGraphics()`

Methods

(continued from last page)

getGraphicsData

```
public String[] getGraphicsData(int position,
         int direction,
         int distance)
```

Default implementation for single-sided objects.

getMaskData

```
public String[] getMaskData(int position,
         int direction,
         int distance)
```

Default implementation for single-sided objects.

com.jlaby.client.view.text
Class TextGraphics

```
java.lang.Object
  |
  +-com.jlaby.client.view.text.TextGraphics
```

Direct Known Subclasses:
SingleSidedGraphics, WallGraphics, MultiSidedGraphics

public abstract class **TextGraphics**
extends Object

Helper class which makes it easier to implement objects for the laby. It provides some methods to load graphics from resource files in the classpath etc.

Field Summary

public static final	FACING_PLAYER Value: **2**
public static final	LOOKING_AWAY Value: **1**
public static final	LOOKING_LEFT Value: **4**
public static final	LOOKING_RIGHT Value: **3**
protected	m_color
protected	m_colorType
protected	m_data
protected	m_name
protected	m_type
public static final	MULTI_COLOR Value: **2**
public static final	MULTI_SIDED Value: **2**
public static final	SINGLE_COLOR Flags for the color settings Value: **1**

public static final	SINGLE_SIDED Flags for the graphics type Value: **1**
public static final	UNDEFINED Constants for the direction flag of an object Value: **-1**

Constructor Summary

public	TextGraphics()

Method Summary

String	getColor()
static String	getColorCharacter(String color)
int	getColorType()
static String	getColorValue(char character)
static String	getColorValue(String character)
String[]	getGraphics(int position, int direction, int distance) This method is called by the view renderer class.
abstract String[]	getGraphicsData(int position, int direction, int distance) This method must return an array of Strings of same length containing the graphics of the object specified by position, direction and distance.
String[]	getMask(int position, int direction, int distance) This method is called by the view renderer class.
abstract String[]	getMaskData(int position, int direction, int distance) This method must return an array of Strings of same length containing the graphics of the object specified by position, direction and distance.
String	getName()
int	getType()
void	init(String name, int type, int colorType, String color, Properties data) Constructs the
static void	initialize()

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Fields

SINGLE_COLOR

```
public static final int SINGLE_COLOR
```

Flags for the color settings
Constant value: **1**

MULTI_COLOR

```
public static final int MULTI_COLOR
```

Constant value: **2**

SINGLE_SIDED

```
public static final int SINGLE_SIDED
```

Flags for the graphics type
Constant value: **1**

MULTI_SIDED

```
public static final int MULTI_SIDED
```

Constant value: **2**

UNDEFINED

```
public static final int UNDEFINED
```

Constants for the direction flag of an object
Constant value: **-1**

LOOKING_AWAY

```
public static final int LOOKING_AWAY
```

Constant value: **1**

FACING_PLAYER

```
public static final int FACING_PLAYER
```

Constant value: **2**

LOOKING_RIGHT

```
public static final int LOOKING_RIGHT
```

Constant value: **3**

LOOKING_LEFT

`public static final int LOOKING_LEFT`

Constant value: **4**

m_colorType

`protected int m_colorType`

m_color

`protected java.lang.String m_color`

m_type

`protected int m_type`

m_name

`protected java.lang.String m_name`

m_data

`protected java.util.Properties m_data`

Constructors

TextGraphics

`public TextGraphics()`

Methods

initialize

`public static void initialize()`

(continued from last page)

init

```
public void init(String name,
         int type,
         int colorType,
         String color,
         Properties data)
```

Constructs the

getColorCharacter

```
public static String getColorCharacter(String color)
```

getColorValue

```
public static String getColorValue(String character)
```

getColorValue

```
public static String getColorValue(char character)
```

getName

```
public String getName()
```

getColor

```
public String getColor()
```

getColorType

```
public int getColorType()
```

getType

```
public int getType()
```

getGraphics

```
public String[] getGraphics(int position,
         int direction,
         int distance)
```

This method is called by the view renderer class.

getMask

```
public String[] getMask(int position,
         int direction,
         int distance)
```

This method is called by the view renderer class.

getGraphicsData

```
public abstract String[] getGraphicsData(int position,
         int direction,
         int distance)
```

This method must return an array of Strings of same length containing the graphics of the object specified by position, direction and distance. NOTE: If the object is invisible from the current point of view (a writing on a wall, for example, is not visible from the other side of the wall) the method should return an array with 0 entries (String[0]).

Parameters:

`position` - The number of the field in the view area (see above).

`direction` - The direction which the object is looking into (see constants)

`distance` - The distance from the viewer to the object (0-5)

Returns:

An array of Strings of same length, containing the graphics of the object, or a String-array with length 0 if the object is invisible.

getMaskData

```
public abstract String[] getMaskData(int position,
         int direction,
         int distance)
```

This method must return an array of Strings of same length containing the graphics of the object specified by position, direction and distance. NOTE: If the object is invisible from the current point of view (a writing on a wall, for example, is not visible from the other side of the wall) the method should return an array with 0 entries (String[0]).

Parameters:

`position` - The number of the field in the view area (see above).

`direction` - The direction which the object is looking into (see constants)

`distance` - The distance from the viewer to the object (0-5)

Returns:

An array of Strings of same length, containing the graphics of the object, or a String-array with length 0 if the object is invisible.

com.jlaby.client.view.text
Class TextGraphicsFactory

```
java.lang.Object
   |
   +-com.jlaby.client.view.text.TextGraphicsFactory
```

public class **TextGraphicsFactory**
extends Object

Helper class which makes it easier to implement objects for the laby. It provides some methods to load graphics from resource files in the classpath etc.

Constructor Summary

public	TextGraphicsFactory()

Method Summary

static void	initialize(String indexProperties)
static TextGraphics	load(int objectType) Factory method which caches all graphics in a static Hashtable to ensure that they are loaded only once during the Servlets lifespan.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

TextGraphicsFactory

public **TextGraphicsFactory**()

Methods

initialize

public static void **initialize**(String indexProperties)

load

public static TextGraphics **load**(int objectType)

Factory method which caches all graphics in a static Hashtable to ensure that they are loaded only once during the Servlets lifespan.

com.jlaby.client.view.text
Class TextView

```
java.lang.Object
  |
  +-com.jlaby.client.view.text.TextView
```

public class **TextView**
extends Object

The class which renders the "3D" view. Usage of this class by other classes: 1. initialize once then, for every image update, 3. 1-n x addGraphics(...) 4. image = renderView();

Constructor Summary

public	TextView()

Method Summary

void	addGraphics(int line, int column, String[] graphics, String color)
void	addGraphics(int line, int column, String[] graphics, String[] mask)
static void	initialize()
String[]	renderView() Renders a string picture with tags

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

TextView

public **TextView**()

Methods

renderView

public String[] **renderView**()

Renders a string picture with tags

(continued from last page)

initialize

```
public static void initialize()
```

addGraphics

```
public void addGraphics(int line,
         int column,
         String[] graphics,
         String[] mask)
```

addGraphics

```
public void addGraphics(int line,
         int column,
         String[] graphics,
         String color)
```

com.jlaby.client.view.text
Class View3D

```
java.lang.Object
  |
  +-com.jlaby.client.view.text.View3D
```

public class **View3D**
extends Object

The class which renders the "3D" view.

Constructor Summary

`public`	`View3D()`

Method Summary

`static String[]`	`renderView(GameCharacterInfo characterInfo)` Renders a 3D string picture.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

View3D

`public View3D()`

Methods

renderView

`public static String[] renderView(GameCharacterInfo characterInfo)`

Renders a 3D string picture.

com.jlaby.client.view.text
Class WallGraphics

```
java.lang.Object
  |
  +-com.jlaby.client.view.text.TextGraphics
      |
      +-com.jlaby.client.view.text.WallGraphics
```

public class **WallGraphics**
extends TextGraphics

Helper class which makes it easier to implement objects for the laby. It provides some methods to load graphics from resource files in the classpath etc.

Fields inherited from class `com.jlaby.client.view.text.TextGraphics`

FACING_PLAYER, LOOKING_AWAY, LOOKING_LEFT, LOOKING_RIGHT, m_color, m_colorType, m_data, m_name, m_type, MULTI_COLOR, MULTI_SIDED, SINGLE_COLOR, SINGLE_SIDED, UNDEFINED

Constructor Summary

public	WallGraphics()

Method Summary

String[]	getGraphicsData(int position, int direction, int distance) Special implementation for wall graphics.
String[]	getMaskData(int position, int direction, int distance) Special implementation for wall graphics.

Methods inherited from class `com.jlaby.client.view.text.TextGraphics`

getColor, getColorCharacter, getColorType, getColorValue, getColorValue, getGraphics, getGraphicsData, getMask, getMaskData, getName, getType, init, initialize

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

WallGraphics

`public WallGraphics()`

Methods

(continued from last page)

getGraphicsData

```
public String[] getGraphicsData(int position,
         int direction,
         int distance)
```

Special implementation for wall graphics. The difference to normal multi-sided graphics is that the direction of the "object" is always given, but it depends on the position in the view rectangle.

getMaskData

```
public String[] getMaskData(int position,
         int direction,
         int distance)
```

Special implementation for wall graphics. The difference to normal multi-sided graphics is that the direction of the "object" is always given, but it depends on the position in the view rectangle.

Package
com.jlaby.config

com.jlaby.config
Class Configuration

```
java.lang.Object
  |
  +-com.jlaby.config.Configuration
```

public class **Configuration**
extends Object

Helper class for accessing configuration values set in a specific properties file in the classpath or from the local filesystem.

Constructor Summary

public	Configuration()

Method Summary

static int	getNumericProperty(String property)
static int	getNumericProperty(String property, int defaultValue)
static String	getProperty(String property)
static String	getProperty(String property, String defaultValue)
static void	load(String resource)

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Configuration

public **Configuration**()

Methods

load

public static void **load**(String resource)

getProperty

```
public static String getProperty(String property)
```

getProperty

```
public static String getProperty(String property,
         String defaultValue)
```

getNumericProperty

```
public static int getNumericProperty(String property)
```

getNumericProperty

```
public static int getNumericProperty(String property,
         int defaultValue)
```

Package com.jlaby.exception

com.jlaby.exception
Class ExceptionHandler

```
java.lang.Object
  |
  +-com.jlaby.exception.ExceptionHandler
```

public class **ExceptionHandler**
extends Object

Generic handler for exceptions, used to provide a consistent and centralized handling of all uncaught exceptions.

Constructor Summary

`public`	`ExceptionHandler()`

Method Summary

`static void`	`handleException(Throwable e)` This method takes an exception and carries out the following steps:

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ExceptionHandler

`public` **`ExceptionHandler`**`()`

Methods

handleException

`public static void` **`handleException`**`(Throwable e)`

This method takes an exception and carries out the following steps:

- create a HTML error page if an appropriate servlet output stream can be found.
- print the stack trace in `System.out`
- log the exception in the Laby log file.

Parameters:

`e` - the exception that should be handled.

com.jlaby.exception
Class InvalidPasswordException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.exception.InvalidPasswordException
```

All Implemented Interfaces:
Serializable

public class **InvalidPasswordException**
extends LabyException

This exception is thrown when the password presented by a user was incorrect

Constructor Summary

public	InvalidPasswordException() Constructs an unspecific exception.
public	InvalidPasswordException(String txt) Constructs an exception with a message text.
public	InvalidPasswordException(Throwable causingException) Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

InvalidPasswordException

```
public InvalidPasswordException()
```

Constructs an unspecific exception.

InvalidPasswordException

`public InvalidPasswordException(String txt)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

InvalidPasswordException

`public InvalidPasswordException(Throwable causingException)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.exception
Class LabyException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
```

All Implemented Interfaces:
Serializable

Direct Known Subclasses:
WorldException, LabySQLException, UserNotLoggedInException, InvalidPasswordException, UnknownUserException, RemoteException, UnsupportedAnswerException, UnsupportedActionException

public class **LabyException**
extends RuntimeException

The mother of all exceptions thrown in the Laby software realms. It is derived from java.lang.RuntimeException in order to make exception handling easier. This way, method signatures can stay lean and simple, and the exceptions can be handled at the appropriate level.

If you don't like this approach: Go f... yourself!

Constructor Summary

`public`	`LabyException()` Constructs an unspecific exception.
`public`	`LabyException(String txt)` Constructs an exception with a message text.
`public`	`LabyException(Throwable causingException)` Constructs an exception with a reference to another exception which caused the problem.

Method Summary

`Throwable`	`getCausingException()` Returns a reference to the wrapped exception which originally caused the problem.

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

(continued from last page)

Constructors

LabyException

`public **LabyException**()`

Constructs an unspecific exception.

LabyException

`public **LabyException**(String txt)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

LabyException

`public **LabyException**(Throwable causingException)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

Methods

getCausingException

`public Throwable **getCausingException**()`

Returns a reference to the wrapped exception which originally caused the problem.

Returns:

the wrapped exception object.

com.jlaby.exception
Class StackTraceCollector

```
java.lang.Object
  |
  +-java.io.OutputStream
      |
      +-com.jlaby.exception.StackTraceCollector
```

All Implemented Interfaces:
Flushable, Closeable

public class **StackTraceCollector**
extends OutputStream

This class allows to collect a stack trace output into a string array which can then be used, for example, to log the stack trace in a file.

Constructor Summary

public	`StackTraceCollector(Throwable exception)` Constructs a collector with a given exception.

Method Summary

`String[]`	`getStackTrace()` Returns the stack trace of the exception given in the constructor as an array of String objects.
`void`	`write(byte[] data)` Implementation of the abstract method from the superclass.
`void`	`write(byte[] data, int off, int len)` Implementation of the abstract method from the superclass.
`void`	`write(int val)` Method not implemented, but necessary because it is abstract in the superclass.

Methods inherited from class `java.io.OutputStream`

`close, flush, write, write, write`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `java.io.Closeable`

`close`

Methods inherited from interface `java.io.Flushable`

`flush`

Constructors

StackTraceCollector

```
public StackTraceCollector(Throwable exception)
```

Constructs a collector with a given exception.

Parameters:
`exception` - the exception whose stack trace should be collected.

Methods

getStackTrace

```
public String[] getStackTrace()
```

Returns the stack trace of the exception given in the constructor as an array of String objects.

Returns:
the String array with the stack trace.

write

```
public void write(int val)
```

Method not implemented, but necessary because it is abstract in the superclass.

write

```
public void write(byte[] data)
```

Implementation of the abstract method from the superclass.

write

```
public void write(byte[] data,
         int off,
         int len)
```

Implementation of the abstract method from the superclass.

com.jlaby.exception
Class UnknownUserException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.exception.UnknownUserException
```

All Implemented Interfaces:
Serializable

public class **UnknownUserException**
extends LabyException

This exception is thrown when the name presented by a user was unknown (could not be found in the db)

Constructor Summary

public	UnknownUserException() Constructs an unspecific exception.
public	UnknownUserException(String txt) Constructs an exception with a message text.
public	UnknownUserException(Throwable causingException) Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class com.jlaby.exception.LabyException

getCausingException

Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

UnknownUserException

public **UnknownUserException**()

Constructs an unspecific exception.

UnknownUserException

```
public UnknownUserException(String txt)
```

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

UnknownUserException

```
public UnknownUserException(Throwable causingException)
```

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.exception
Class UserNotLoggedInException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.exception.UserNotLoggedInException
```

All Implemented Interfaces:
Serializable

public class **UserNotLoggedInException**
extends LabyException

This exception is thrown when the user found in the session information is not logged in.

Constructor Summary

public	`UserNotLoggedInException()` Constructs an unspecific exception.
public	`UserNotLoggedInException(String txt)` Constructs an exception with a message text.
public	`UserNotLoggedInException(Throwable causingException)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

UserNotLoggedInException

`public` **`UserNotLoggedInException()`**

Constructs an unspecific exception.

UserNotLoggedInException

```
public UserNotLoggedInException(String txt)
```

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

UserNotLoggedInException

```
public UserNotLoggedInException(Throwable causingException)
```

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

Package
com.jlaby.html

com.jlaby.html
Class CreateUserPanel

```
java.lang.Object
  |
  +-com.jlaby.html.LabyHtmlPanel
      |
      +-com.jlaby.html.CreateUserPanel
```

All Implemented Interfaces:
ILabyConstants

public class **CreateUserPanel**
extends LabyHtmlPanel
implements ILabyConstants

This class handles the login HTML page of the laby.

Fields inherited from interface com.jlaby.util.ILabyConstants

FALSE, PROP_CONFIG_FILE, PROP_DB_ORACLE_SEQNAME, PROP_DB_VENDOR, PROP_HTML_CREATE, PROP_HTML_INVENTORY, PROP_HTML_LOGIN, PROP_HTML_MAINPANEL, PROP_JDBC_DRIVER_CLASS, PROP_JDBC_DRIVER_URL, PROP_LOG_DESC, PROP_LOG_DIR, PROP_LOG_FILE, PROP_LOG_NAME, PROP_LOG_TRACE, PROP_SERVER_PORT, PROP_SQL_NEWUNIQUEID, PROP_SQL_QUERY_CREATEUSER, PROP_SQL_QUERY_GETVIEW, PROP_SQL_QUERY_LOADUSER, PROP_SQL_QUERY_SAVEUSER, PROP_SQL_UNIQUEID_MCKOI, PROP_SQL_UNIQUEID_ORACLE, PROP_URL_CREATE, PROP_URL_LABY, PROP_URL_LOGIN, TRUE

Constructor Summary

public	CreateUserPanel(String templateResource, HttpServletResponse servletResponse)

Method Summary

void	update()

Methods inherited from class com.jlaby.html.LabyHtmlPanel

display, replace, replace, replaceURL

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

CreateUserPanel

```
public CreateUserPanel(String templateResource,
                       HttpServletResponse servletResponse)
```

Methods

update

```
public void update()
```

com.jlaby.html
Class LabyHtmlPanel

```
java.lang.Object
   |
   +-com.jlaby.html.LabyHtmlPanel
```

Direct Known Subclasses:
MainPanel, LoginPanel, CreateUserPanel

public abstract class **LabyHtmlPanel**
extends Object

This class allows to read a HTML template and have parts of it replaced dynamically.

Constructor Summary

public	LabyHtmlPanel(String templateResource, HttpServletResponse servletResponse)

Method Summary

void	display(HttpServletResponse servletResponse)
void	replace(String toReplace, String line)
void	replace(String toReplace, String[] lines)
void	replaceURL(String toReplace, String url)

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

LabyHtmlPanel

```
public LabyHtmlPanel(String templateResource,
                     HttpServletResponse servletResponse)
```

Methods

display

```
public final void display(HttpServletResponse servletResponse)
```

(continued from last page)

replace

```
public final void replace(String toReplace,
         String[] lines)
```

replace

```
public final void replace(String toReplace,
         String line)
```

replaceURL

```
public final void replaceURL(String toReplace,
         String url)
```

com.jlaby.html
Class LoginPanel

```
java.lang.Object
  |
  +-com.jlaby.html.LabyHtmlPanel
      |
      +-com.jlaby.html.LoginPanel
```

All Implemented Interfaces:
ILabyConstants

public class **LoginPanel**
extends LabyHtmlPanel
implements ILabyConstants

This class handles the login HTML page of the laby.

Fields inherited from interface `com.jlaby.util.ILabyConstants`

FALSE, PROP_CONFIG_FILE, PROP_DB_ORACLE_SEQNAME, PROP_DB_VENDOR, PROP_HTML_CREATE, PROP_HTML_INVENTORY, PROP_HTML_LOGIN, PROP_HTML_MAINPANEL, PROP_JDBC_DRIVER_CLASS, PROP_JDBC_DRIVER_URL, PROP_LOG_DESC, PROP_LOG_DIR, PROP_LOG_FILE, PROP_LOG_NAME, PROP_LOG_TRACE, PROP_SERVER_PORT, PROP_SQL_NEWUNIQUEID, PROP_SQL_QUERY_CREATEUSER, PROP_SQL_QUERY_GETVIEW, PROP_SQL_QUERY_LOADUSER, PROP_SQL_QUERY_SAVEUSER, PROP_SQL_UNIQUEID_MCKOI, PROP_SQL_UNIQUEID_ORACLE, PROP_URL_CREATE, PROP_URL_LABY, PROP_URL_LOGIN, TRUE

Constructor Summary

public	`LoginPanel(String templateResource, HttpServletResponse servletResponse)`

Method Summary

void	`update()`

Methods inherited from class `com.jlaby.html.LabyHtmlPanel`

`display, replace, replace, replaceURL`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

LoginPanel

```
public LoginPanel(String templateResource,
                  HttpServletResponse servletResponse)
```

Methods

update

```
public void update()
```

Package com.jlaby.inventory

com.jlaby.inventory
Class Inventory

```
java.lang.Object
   |
   +-com.jlaby.inventory.Inventory
```

public class **Inventory**
extends Object

This class handles the inventory.

Constructor Summary

public	Inventory()

Method Summary

static String[]	display(PrintWriter out)

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

Inventory

```
public Inventory()
```

Methods

display

```
public static String[] display(PrintWriter out)
```

Package
com.jlaby.jdbc

com.jlaby.jdbc
Interface IUniqueIdGenerator

All Known Implementing Classes:
IdGenerator, IdGenerator

public interface **IUniqueIdGenerator**
extends

Interface for database-specific implementations of mechanisms for creating unique IDs.

Method Summary

`int`	`createNewID()` Creates a new, unique integer ID value.
`void`	`initialize()` This method must initialize and / or configure the ID generator.

Methods

createNewID

```
public int createNewID()
```

Creates a new, unique integer ID value.

Returns:
the unique integer value

initialize

```
public void initialize()
  throws LabyException
```

This method must initialize and / or configure the ID generator. It may throw any kind of LabyException if necessary.

com.jlaby.jdbc
Class JdbcUtil

```
java.lang.Object
  |
  +-com.jlaby.jdbc.JdbcUtil
```

public class **JdbcUtil**
extends Object

Helper class for database access.

Constructor Summary

`public`	`JdbcUtil()`

Method Summary

`static GameCharacter`	`createCharacter(String userName)` This simplifying method creates a new user record in the database.
`static void`	`createConnection()` Create a physical connection to the database based on the configuration values in the main configuration file.
`static void`	`dispose()` Free all resources and dispose of the JDBC Utility class
`static java.sql.ResultSet`	`executeQuery(String query)` Executes any query.
`static GameCharacter[]`	`getCharacters(int leftX, int leftY, int rightX, int rightY)` This method is used to determine which characters are in the viewing area of the current character.
`static boolean`	`isConnected()` Determines if there is currently a connection to the database or not.
`static GameCharacter`	`loadCharacter(String name)` This simplifying method reads the record of a game character from the database and fills it into an existing GameCharacter object.
`static void`	`saveCharacter(GameCharacter character)` This simplifying method saves a GameCharacter into an existing record in the database.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

JdbcUtil

```
public JdbcUtil()
```

Methods

isConnected

```
public static boolean isConnected()
```

Determines if there is currently a connection to the database or not.

Returns:
true if the connection exists.

getCharacters

```
public static GameCharacter[] getCharacters(int leftX,
         int leftY,
         int rightX,
         int rightY)
```

This method is used to determine which characters are in the viewing area of the current character. It "scans" a certain area for other characters.

Parameters:
`leftX` - The "upper left" x-coordinate (column) of the rectangular area
`leftY` - The "upper left" y-coordinate (row) of the rectangular area
`rightX` - The "lower right" x-coordinate (column) of the rectangular area
`rightY` - The "lower right" y-coordinate (row) of the rectangular area

Returns:
An array of characters found in that area. The array has a length of 0, if no characters had been found.

Throws:
`LabyException` - Thrown in case of DB problems

createCharacter

```
public static GameCharacter createCharacter(String userName)
  throws LabyException
```

This simplifying method creates a new user record in the database.

Parameters:
`userID` - The new users unique name ID

Returns:
The character object that represents the character.

Throws:
`LabyException` - Thrown in case of DB problems

(continued from last page)

loadCharacter

```
public static GameCharacter loadCharacter(String name)
  throws LabyException
```

This simplifying method reads the record of a game character from the database and fills it into an existing GameCharacter object.

Please note: Since a game character can both be a human player or a non-player-character (a bot), this method has intentionally not been named "getUser()".

Parameters:

`name` - The (unique) name of the game character.

Returns:

The character object that represents the character.

Throws:

`LabyException` - Thrown in case of DB problems

saveCharacter

```
public static void saveCharacter(GameCharacter character)
  throws LabyException
```

This simplifying method saves a GameCharacter into an existing record in the database.

Parameters:

`character` - The character object that should be saved

Throws:

`LabyException` - Thrown in case of DB problems

createConnection

```
public static void createConnection()
```

Create a physical connection to the database based on the configuration values in the main configuration file.

executeQuery

```
public static java.sql.ResultSet executeQuery(String query)
  throws Exception
```

Executes any query. Used mainly for admin purposes.

Parameters:

`query` - the SQL query string to be executed.

Throws:

`Exception` - thrown if there was a database problem

dispose

```
public static void dispose()
  throws Exception
```

Free all resources and dispose of the JDBC Utility class

Package com.jlaby.jdbc.exception

com.jlaby.jdbc.exception
Class LabySQLException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.jdbc.exception.LabySQLException
```

All Implemented Interfaces:
Serializable

Direct Known Subclasses:
NoRecordFoundException, NoRecordUpdatedException

public class **LabySQLException**
extends LabyException

Exception for db related problems.

Constructor Summary

public	LabySQLException() Constructs an unspecific exception.
public	LabySQLException(String msg) Constructs an exception with a message text.
public	LabySQLException(Exception e) Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class com.jlaby.exception.LabyException
getCausingException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

(continued from last page)

LabySQLException

public **LabySQLException**()

Constructs an unspecific exception.

LabySQLException

public **LabySQLException**(String msg)

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

LabySQLException

public **LabySQLException**(Exception e)

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.jdbc.exception
Class NoRecordFoundException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.jdbc.exception.LabySQLException
                      |
                      +-com.jlaby.jdbc.exception.NoRecordFoundException
```

All Implemented Interfaces:
Serializable

public class **NoRecordFoundException**
extends LabySQLException

Thrown if a select statement returned zero records.

Constructor Summary

`public`	`NoRecordFoundException()` Constructs an unspecific exception.
`public`	`NoRecordFoundException(String msg)` Constructs an exception with a message text.
`public`	`NoRecordFoundException(Exception e)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

NoRecordFoundException

`public` **`NoRecordFoundException`**`()`

(continued from last page)

Constructs an unspecific exception.

NoRecordFoundException

`public NoRecordFoundException(String msg)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

NoRecordFoundException

`public NoRecordFoundException(Exception e)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.jdbc.exception
Class NoRecordUpdatedException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.jdbc.exception.LabySQLException
                      |
                      +-com.jlaby.jdbc.exception.NoRecordUpdatedException
```

All Implemented Interfaces:
Serializable

public class **NoRecordUpdatedException**
extends LabySQLException

Thrown if an update statement returned zero records.

Constructor Summary

public	NoRecordUpdatedException() Constructs an unspecific exception.
public	NoRecordUpdatedException(String msg) Constructs an exception with a message text.
public	NoRecordUpdatedException(Exception e) Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

NoRecordUpdatedException

`public NoRecordUpdatedException()`

(continued from last page)

Constructs an unspecific exception.

NoRecordUpdatedException

`public` **`NoRecordUpdatedException`**`(String msg)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

NoRecordUpdatedException

`public` **`NoRecordUpdatedException`**`(Exception e)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

Package com.jlaby.jdbc.mckoi

com.jlaby.jdbc.mckoi
Class IdGenerator

```
java.lang.Object
  |
  +-com.jlaby.jdbc.mckoi.IdGenerator
```

All Implemented Interfaces:
ILabyConstants, IUniqueIdGenerator

---

public class **IdGenerator**
extends Object
implements IUniqueIdGenerator, ILabyConstants

McKoi-specific implementation of a unique ID generator.

Fields inherited from interface `com.jlaby.util.ILabyConstants`

FALSE, PROP_CONFIG_FILE, PROP_DB_ORACLE_SEQNAME, PROP_DB_VENDOR, PROP_HTML_CREATE, PROP_HTML_INVENTORY, PROP_HTML_LOGIN, PROP_HTML_MAINPANEL, PROP_JDBC_DRIVER_CLASS, PROP_JDBC_DRIVER_URL, PROP_LOG_DESC, PROP_LOG_DIR, PROP_LOG_FILE, PROP_LOG_NAME, PROP_LOG_TRACE, PROP_SERVER_PORT, PROP_SQL_NEWUNIQUEID, PROP_SQL_QUERY_CREATEUSER, PROP_SQL_QUERY_GETVIEW, PROP_SQL_QUERY_LOADUSER, PROP_SQL_QUERY_SAVEUSER, PROP_SQL_UNIQUEID_MCKOI, PROP_SQL_UNIQUEID_ORACLE, PROP_URL_CREATE, PROP_URL_LABY, PROP_URL_LOGIN, TRUE

Constructor Summary

public	`IdGenerator(java.sql.Connection connection)` Constructs a generator for unique IDs which uses an ORACLE Sequence.

Method Summary

int	`createNewID()` Creates a new, unique integer ID value.
void	`initialize()` This method initializes the generator by reading the name of the Sequence object and the SQL query from the configuration.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.jdbc.IUniqueIdGenerator`

`createNewID, initialize`

Constructors

(continued from last page)

IdGenerator

```
public IdGenerator(java.sql.Connection connection)
```

Constructs a generator for unique IDs which uses an ORACLE Sequence.

Parameters:

`connection` - the JDBC database connection.

Methods

initialize

```
public void initialize()
  throws LabyException
```

This method initializes the generator by reading the name of the Sequence object and the SQL query from the configuration.

createNewID

```
public int createNewID()
  throws LabySQLException
```

Creates a new, unique integer ID value.

Returns:

the unique integer value

Package com.jlaby.jdbc.oracle

com.jlaby.jdbc.oracle
Class IdGenerator

```
java.lang.Object
  |
  +-com.jlaby.jdbc.oracle.IdGenerator
```

All Implemented Interfaces:
ILabyConstants, IUniqueIdGenerator

public class **IdGenerator**
extends Object
implements IUniqueIdGenerator, ILabyConstants

Oracle-specific implementation of a unique ID generator, working with Oracle sequence objects.

Fields inherited from interface `com.jlaby.util.ILabyConstants`

`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Constructor Summary

`public`	`IdGenerator(java.sql.Connection connection)` Constructs a generator for unique IDs which uses an ORACLE Sequence.

Method Summary

`int`	`createNewID()` Creates a new, unique integer ID value.
`void`	`initialize()` This method initializes the generator by reading the name of the Sequence object and the SQL query from the configuration.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.jdbc.IUniqueIdGenerator`

`createNewID, initialize`

Constructors

(continued from last page)

IdGenerator

```
public IdGenerator(java.sql.Connection connection)
```

Constructs a generator for unique IDs which uses an ORACLE Sequence.

Parameters:
`connection` - the JDBC database connection.

Methods

initialize

```
public void initialize()
  throws LabyException
```

This method initializes the generator by reading the name of the Sequence object and the SQL query from the configuration.

createNewID

```
public int createNewID()
  throws LabySQLException
```

Creates a new, unique integer ID value.

Returns:
the unique integer value

Package
com.jlaby.log

com.jlaby.log
Class Log

```
java.lang.Object
  |
  +-com.jlaby.log.Log
```

public class **Log**
extends Object

Constructor Summary

public	`Log()`

Method Summary

static void	`error(Object objref, String methodName, String text)` Log an error message with a free text.
static void	`error(String className, String methodName, String text)` Log an error message with a free text.
static void	`info(Object objref, String methodName, String text)` Log an info message with a free text.
static void	`info(String className, String methodName, String text)` Log an info message with a free text.
static void	`initialize()`
static boolean	`isLoggingActive()` Returns true when logging is active
static boolean	`isTraceActive()` Returns true when traceing is active
static void	`severe(Object objref, String methodName, String text)` Log a severe error message with a free text.
static void	`severe(String className, String methodName, String text)` Log a severe error message with a free text.
static void	`trace(Object objref, String methodName, String text)` Log a trace message with a free text.
static void	`trace(String className, String methodName, String text)` Log a trace error message with a free text.
static void	`traceEntry(Object objref, String methodName)` Log a trace message with a free text and parameters.
static void	`traceEntry(Object objref, String methodName, Object[] params)` Log a trace message with a free text and parameters.

`static void`	`traceEntry(String className, String methodName)` Log a trace message with a free text and parameters.
`static void`	`traceEntry(String className, String methodName, Object[] params)` Log a trace message with a free text and parameters.
`static void`	`traceExit(Object objref, String methodName, long rc)` Log a trace message with a free text and parameters.
`static void`	`traceExit(String className, String methodName, long rc)` Log a trace message with a free text and parameters.
`static void`	`warning(Object objref, String methodName, String text)` Log a warning message with a free text.
`static void`	`warning(String className, String methodName, String text)` Log a warning message with a free text.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Log

```
public Log()
```

Methods

initialize

```
public static void initialize()
```

info

```
public final static void info(Object objref,
        String methodName,
        String text)
```

Log an info message with a free text.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`text` - The text of the info message.

(continued from last page)

info

```
public final static void info(String className,
         String methodName,
         String text)
```

Log an info message with a free text.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`text` - The text of the info message.

warning

```
public final static void warning(Object objref,
         String methodName,
         String text)
```

Log a warning message with a free text.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`text` - The text of the warning message.

warning

```
public final static void warning(String className,
         String methodName,
         String text)
```

Log a warning message with a free text.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`text` - The text of the warning message.

error

```
public final static void error(Object objref,
         String methodName,
         String text)
```

Log an error message with a free text.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`text` - The text of the error message.

error

```
public final static void error(String className,
         String methodName,
         String text)
```

Log an error message with a free text.

(continued from last page)

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`text` - The text of the error message.

severe

```
public final static void severe(Object objref,
         String methodName,
         String text)
```

Log a severe error message with a free text.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`text` - The text of the severe message.

severe

```
public final static void severe(String className,
         String methodName,
         String text)
```

Log a severe error message with a free text.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`text` - The text of the severe message.

trace

```
public final static void trace(Object objref,
         String methodName,
         String text)
```

Log a trace message with a free text.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`text` - The text of the trace message.

trace

```
public final static void trace(String className,
         String methodName,
         String text)
```

Log a trace error message with a free text.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`text` - The text of the trace message.

(continued from last page)

traceEntry

```
public final static void traceEntry(String className,
         String methodName,
         Object[] params)
```

Log a trace message with a free text and parameters. This method shall be called when a method has been entered.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`params` - An array of elements to be inserted into the message

traceEntry

```
public final static void traceEntry(Object objref,
         String methodName,
         Object[] params)
```

Log a trace message with a free text and parameters. This method shall be called when a method has been entered.

Parameters:

`objRef` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`params` - An array of elements to be inserted into the message

traceEntry

```
public static void traceEntry(String className,
         String methodName)
```

Log a trace message with a free text and parameters. This method shall be called when a method has been entered.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.

traceEntry

```
public static void traceEntry(Object objref,
         String methodName)
```

Log a trace message with a free text and parameters. This method shall be called when a method has been entered.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.

traceExit

```
public final static void traceExit(String className,
         String methodName,
         long rc)
```

Log a trace message with a free text and parameters. This method shall be called before a method is exited.

Parameters:

`className` - The name of the class that created the message.
`methodName` - The method where the message originated.
`rc` - The return code of the method

traceExit

```
public final static void traceExit(Object objref,
         String methodName,
         long rc)
```

Log a trace message with a free text and parameters. This method shall be called before a method is exited.

Parameters:

`objref` - The reference to the object that created the message.
`methodName` - The method where the message originated.
`rc` - The return code of the method

isLoggingActive

```
public final static boolean isLoggingActive()
```

Returns true when logging is active

Returns:

true if logging is active

isTraceActive

```
public final static boolean isTraceActive()
```

Returns true when traceing is active

Returns:

true if traceing is active

com.jlaby.log
Class LogManager

```
java.lang.Object
  |
  +-com.jlaby.log.LogManager
```

All Implemented Interfaces:
ILabyConstants

public class **LogManager**
extends Object
implements ILabyConstants

The LogManager ... log.dir = directory for log files log.name = logical name of log (internal) log.desc = description of log (internal) log.file = name of log file log.trace = name of trace file

Fields inherited from interface `com.jlaby.util.ILabyConstants`
`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Method Summary

`MessageLogger`	`getErrorLogger()` Returns the error logger
`MessageLogger`	`getMessageLogger()` Returns the message logger
`TraceLogger`	`getTraceLogger()` Returns the trace logger
`static LogManager`	`instance()` Returns a reference to an existing LogManager object.

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods

instance

```
public static LogManager instance()
```

(continued from last page)

Returns a reference to an existing LogManager object. If no LogManager instance was created to this point, a new one is created with the default constructor and its reference is returned. If any other class created an instance of the LogManager class before, the first created instance will be returned.

Returns:
LogManager Reference to a LogManager object

getMessageLogger

`public MessageLogger getMessageLogger()`

Returns the message logger

Returns:
the message logger

getErrorLogger

`public MessageLogger getErrorLogger()`

Returns the error logger

Returns:
the error logger

getTraceLogger

`public TraceLogger getTraceLogger()`

Returns the trace logger

Returns:
the trace logger

com.jlaby.log
Class SingleLineFormatter

```
java.lang.Object
  |
  +-Formatter
     |
     +-com.jlaby.log.SingleLineFormatter
```

Direct Known Subclasses:
SingleLineTraceFormatter

public class **SingleLineFormatter**
extends Formatter

SingleLineFormatter is used to output a single log-entry containing the following format:

- date/time
- severity (as string: INFO/WARNING/ERROR/FATAL)
- text which describes the severity

the tab (\t) is used as field delimiter

Field Summary

protected static	LOG_FIELD_END
protected static	LOG_FIELD_START

Constructor Summary

public	SingleLineFormatter() Creates new SingleLineFormatter

Method Summary

String	extractClassName(String className) This method extracts the classname from the passed object.
String	format(ILogRecord record) Overloaded method of Formatter, formats the log record
String	getFormattedText(ILogRecord record) This method returns the text of the logrecord.
static String	getTypeText(long messType) This method returns a string represenatation of the appropriate message type

Methods inherited from class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

(continued from last page)

Fields

LOG_FIELD_START

`protected static java.lang.String **LOG_FIELD_START**`

LOG_FIELD_END

`protected static java.lang.String **LOG_FIELD_END**`

Constructors

SingleLineFormatter

`public **SingleLineFormatter**()`

Creates new SingleLineFormatter

Methods

format

`public String **format**(ILogRecord record)`

Overloaded method of Formatter, formats the log record

Parameters:
`record` - the ILogRecord object containing the log informations

Returns:
Formatted line

---

getFormattedText

`protected String **getFormattedText**(ILogRecord record)`

This method returns the text of the logrecord. By default the `com.ibm.logging.ILogRecord.getText` method is invoked. Subclasses can provide an alternative implementation for example

Parameters:
`record` - the ILogRecord object containing the log information

Returns:
the text string of the log record

---

getTypeText

`protected static String **getTypeText**(long messType)`

This method returns a string represenatation of the appropriate message type

Parameters:
`record` - the IRecord Object containing the log-type as a long

(continued from last page)

Returns:

Formatted line

extractClassName

```
protected String extractClassName(String className)
```

This method extracts the classname from the passed object. Only the classname is returned, the leading package qualifier is omited. E.g. com.acme.util.AClass is substituted to AClass

Parameters:

`aString` - a string from which the classname should be extracted

Returns:

A string representing the classname without the leading package qualifier

com.jlaby.log
Class SingleLineTraceFormatter

```
java.lang.Object
  |
  +-Formatter
     |
     +-com.jlaby.log.SingleLineFormatter
        |
        +-com.jlaby.log.SingleLineTraceFormatter
```

public class **SingleLineTraceFormatter**
extends SingleLineFormatter

SingleLineTraceFormatter is used to output a single trace log-entry containing the following format:

- date/time
- trace type (as string: enter,exit,trace)
- text which describes the severity

the tab (\t) is used as field delimiter

Fields inherited from class `com.jlaby.log.SingleLineFormatter`

`LOG_FIELD_END`, `LOG_FIELD_START`

Constructor Summary

`public`	`SingleLineTraceFormatter()` Creates new SingleLineTraceFormatter

Method Summary

`String`	`getFormattedText(ILogRecord record)` This method returns a formatted text, but it doesn't support logrecords which have more than 10 arguments to print out.

Methods inherited from class `com.jlaby.log.SingleLineFormatter`

`extractClassName`, `format`, `getFormattedText`, `getTypeText`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

SingleLineTraceFormatter

`public` **`SingleLineTraceFormatter`**`()`

Creates new SingleLineTraceFormatter

(continued from last page)

Methods

getFormattedText

```
public String getFormattedText(ILogRecord record)
```

This method returns a formatted text, but it doesn't support logrecords which have more than 10 arguments to print out.

Package

com.jlaby.server

com.jlaby.server
Class WorldImpl

```
java.lang.Object
  |
  +-com.jlaby.server.WorldImpl
```

All Implemented Interfaces:

IWorld

public class **WorldImpl**
extends Object
implements IWorld

This class loads and holds the world data.

Fields inherited from interface `com.jlaby.world.IWorld`
`EAST`, `NORTH`, `SOUTH`, `WEST`

Constructor Summary

`public`	`WorldImpl()` Constructs the

Method Summary

`LabyAnswer`	`handleAction(LabyAction action)` This method deals with an action created by a "client" (local or remote).
`void`	`initialize()`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `com.jlaby.world.IWorld`
`handleAction, initialize`

Constructors

WorldImpl

`public WorldImpl()`

Constructs the

Methods

(continued from last page)

initialize

```
public void initialize()
  throws InitializationFailedException
```

handleAction

```
public LabyAnswer handleAction(LabyAction action)
  throws LabyException
```

This method deals with an action created by a "client" (local or remote). It creates the appropriate handler for the action, has it handled and returns the answer for the client.

Parameters:

`action` - the action created /sent by a client.

Returns:

the answer for the client.

Throws:

`LabyException` - thrown if the action could not be handled.

Package com.jlaby.server.jetty

com.jlaby.server.jetty
Class LabyHttpListener

```
java.lang.Object
   |
   +-com.jlaby.server.jetty.LabyHttpListener
```

public class **LabyHttpListener**
extends Object

Constructor Summary

public	LabyHttpListener(InetAddrPort addr)
public	LabyHttpListener()

Method Summary

void	customizeRequest(HttpConnection connection, HttpRequest request)
void	destroy()
String	getDefaultScheme()
String	getHost()
HttpServer	getHttpServer()
int	getPort()
java.net.ServerSocket	getServerSocket()
boolean	isDestroyed()
boolean	isLowOnResources()
boolean	isOutOfResources()
boolean	isStarted()
void	persistConnection(HttpConnection connection)
void	setHost(String host)
void	setHttpServer(HttpServer server)

void	setPort(int port)
void	start()
void	stop()

Methods inherited from class `java.lang.Object`
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

LabyHttpListener

```
public LabyHttpListener(InetAddrPort addr)
```

LabyHttpListener

```
public LabyHttpListener()
```

Methods

setHttpServer

```
public void setHttpServer(HttpServer server)
```

setHost

```
public void setHost(String host)
  throws java.net.UnknownHostException
```

setPort

```
public void setPort(int port)
```

persistConnection

```
public void persistConnection(HttpConnection connection)
```

(continued from last page)

isOutOfResources

```
public boolean isOutOfResources()
```

isLowOnResources

```
public boolean isLowOnResources()
```

getServerSocket

```
public java.net.ServerSocket getServerSocket()
```

getPort

```
public int getPort()
```

getHttpServer

```
public HttpServer getHttpServer()
```

getHost

```
public String getHost()
```

getDefaultScheme

```
public String getDefaultScheme()
```

customizeRequest

```
public void customizeRequest(HttpConnection connection,
         HttpRequest request)
```

start

```
public void start()
  throws Exception
```

(continued from last page)

stop

```
public void stop()
  throws InterruptedException
```

destroy

```
public void destroy()
```

isDestroyed

```
public boolean isDestroyed()
```

isStarted

```
public boolean isStarted()
```

com.jlaby.server.jetty
Class LabyServer

```
java.lang.Object
  |
  +-com.jlaby.server.jetty.LabyServer
```

All Implemented Interfaces:
ILabyConstants, Runnable

public class **LabyServer**
extends Object
implements Runnable, ILabyConstants

This Jetty-based server can run, depending on the configuration, both the servlet(s) required to deal with requests from clients as well a an in-process instance of a Bot thread. This is especially useful for local testing in combination with the local pure Java database "McKoi".

Nested Class Summary

class	`LabyServer.ShutdownHook` LabyServer.ShutdownHook

Fields inherited from interface `com.jlaby.util.ILabyConstants`

`FALSE`, `PROP_CONFIG_FILE`, `PROP_DB_ORACLE_SEQNAME`, `PROP_DB_VENDOR`, `PROP_HTML_CREATE`, `PROP_HTML_INVENTORY`, `PROP_HTML_LOGIN`, `PROP_HTML_MAINPANEL`, `PROP_JDBC_DRIVER_CLASS`, `PROP_JDBC_DRIVER_URL`, `PROP_LOG_DESC`, `PROP_LOG_DIR`, `PROP_LOG_FILE`, `PROP_LOG_NAME`, `PROP_LOG_TRACE`, `PROP_SERVER_PORT`, `PROP_SQL_NEWUNIQUEID`, `PROP_SQL_QUERY_CREATEUSER`, `PROP_SQL_QUERY_GETVIEW`, `PROP_SQL_QUERY_LOADUSER`, `PROP_SQL_QUERY_SAVEUSER`, `PROP_SQL_UNIQUEID_MCKOI`, `PROP_SQL_UNIQUEID_ORACLE`, `PROP_URL_CREATE`, `PROP_URL_LABY`, `PROP_URL_LOGIN`, `TRUE`

Constructor Summary

public	`LabyServer()` Constructs a server.

Method Summary

static void	`beginShutdown()` Ensure a VM exit after a given maximum time.
static void	`destroy()` Shuts down the Jetty server by stopping and removing all listeners first and stopping and destroying the server subsequently.
static void	`main(String[] arg)` The main method which creates and invokes the server.
void	`run()` This Thread ensures that when a shutdown has been initialized, the Java VM will be terminated after max.

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Methods inherited from interface `java.lang.Runnable`

```
run
```

Constructors

LabyServer

```
public LabyServer()
```

Constructs a server.

Methods

main

```
public static void main(String[] arg)
```

The main method which creates and invokes the server.

beginShutdown

```
public static void beginShutdown()
```

Ensure a VM exit after a given maximum time.

destroy

```
public static void destroy()
```

Shuts down the Jetty server by stopping and removing all listeners first and stopping and destroying the server subsequently. This method implements a somewhat "heavy" exception handling, the reason being the fact that the shutdown must be as robust and safe as possible.

run

```
public void run()
```

This Thread ensures that when a shutdown has been initialized, the Java VM will be terminated after max. 15 seconds, under all circumstances, no matter what happens in any other thread. PLEASE NOTE: If the program is in self-testing mode, it will NOT exit (as other tests may be made subsequently in the same VM instance).

com.jlaby.server.jetty
Class LabyServer.ShutdownHook

```
java.lang.Object
   |
   +-java.lang.Thread
      |
      +-com.jlaby.server.jetty.LabyServer.ShutdownHook
```

All Implemented Interfaces:
Runnable

public class **LabyServer.ShutdownHook**
extends Thread

This class ensures that the server is destroyed cleanly should the JVM be killed with a signal (CTRL-C).

Fields inherited from class java.lang.Thread

MAX_PRIORITY, MIN_PRIORITY, NORM_PRIORITY

Constructor Summary

public	LabyServer.ShutdownHook()

Method Summary

void	run()

Methods inherited from class java.lang.Thread

activeCount, checkAccess, countStackFrames, currentThread, destroy, dumpStack, enumerate, getAllStackTraces, getContextClassLoader, getDefaultUncaughtExceptionHandler, getId, getName, getPriority, getStackTrace, getState, getThreadGroup, getUncaughtExceptionHandler, holdsLock, interrupt, interrupted, isAlive, isDaemon, isInterrupted, join, join, join, resume, run, setContextClassLoader, setDaemon, setDefaultUncaughtExceptionHandler, setName, setPriority, setUncaughtExceptionHandler, sleep, sleep, start, stop, stop, suspend, toString, yield

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.lang.Runnable

run

Constructors

(continued from last page)

LabyServer.ShutdownHook

public **LabyServer.ShutdownHook**()

Methods

run

public void **run**()

com.jlaby.server.jetty
Class LogSinkAdapter

```
java.lang.Object
  |
  +-com.jlaby.server.jetty.LogSinkAdapter
```

public class **LogSinkAdapter**
extends Object

This facade class catches log calls from the Jetty server in order to log them through TARSECs' standard logger (like log4j, for instance). This ensures that the log messages will all have the same format.

Constructor Summary

public	LogSinkAdapter()

Method Summary

void	destroy()
String	getOptions()
boolean	isDestroyed()
boolean	isStarted()
void	log(String formattedLog)
void	log(String tag, Object msg, Frame frame, long time)
void	setOptions(String options)
void	start()
void	stop()

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

LogSinkAdapter

public **LogSinkAdapter**()

(continued from last page)

Methods

getOptions

```
public String getOptions()
```

setOptions

```
public void setOptions(String options)
```

log

```
public void log(String tag,
         Object msg,
         Frame frame,
         long time)
```

log

```
public void log(String formattedLog)
```

start

```
public void start()
```

stop

```
public void stop()
```

destroy

```
public void destroy()
```

isDestroyed

```
public boolean isDestroyed()
```

isStarted

```
public boolean isStarted()
```

(continued from last page)

Package com.jlaby.server.servlet

com.jlaby.server.servlet
Class LabyServlet

```
java.lang.Object
  |
  +-HttpServlet
      |
      +-com.jlaby.server.servlet.LabyServlet
```

All Implemented Interfaces:
ILabyConstants

public class **LabyServlet**
extends HttpServlet
implements ILabyConstants

The main servlet of the Laby.

Fields inherited from interface `com.jlaby.util.ILabyConstants`

FALSE, PROP_CONFIG_FILE, PROP_DB_ORACLE_SEQNAME, PROP_DB_VENDOR, PROP_HTML_CREATE, PROP_HTML_INVENTORY, PROP_HTML_LOGIN, PROP_HTML_MAINPANEL, PROP_JDBC_DRIVER_CLASS, PROP_JDBC_DRIVER_URL, PROP_LOG_DESC, PROP_LOG_DIR, PROP_LOG_FILE, PROP_LOG_NAME, PROP_LOG_TRACE, PROP_SERVER_PORT, PROP_SQL_NEWUNIQUEID, PROP_SQL_QUERY_CREATEUSER, PROP_SQL_QUERY_GETVIEW, PROP_SQL_QUERY_LOADUSER, PROP_SQL_QUERY_SAVEUSER, PROP_SQL_UNIQUEID_MCKOI, PROP_SQL_UNIQUEID_ORACLE, PROP_URL_CREATE, PROP_URL_LABY, PROP_URL_LOGIN, TRUE

Constructor Summary

`public`	`LabyServlet()`

Method Summary

`void`	`destroy()`
`String`	`getServletInfo()` This methode returns information about the servlet.
`void`	`init(ServletConfig p_config)` The init method of the HTTPServlet class.
`static void`	`initialize()` This initialization is done the first time the servlet is called.
`static boolean`	`isInitialized()`
`void`	`service(HttpServletRequest httpRequest, HttpServletResponse httpResponse)` This methode handles the http request.

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructors

LabyServlet

```
public LabyServlet()
```

Methods

service

```
public void service(HttpServletRequest httpRequest,
         HttpServletResponse httpResponse)
  throws ServletException,
         IOException
```

This methode handles the http request.

Parameters:

`req` - The http request.

`res` - The http response.

Returns:

none.

Throws:

`ServletException` - Exception thrown by servlet method.

`IOException` - Exception thrown by IO.

initialize

```
public static void initialize()
  throws Exception
```

This initialization is done the first time the servlet is called. Its main purpose is to trigger the initialization of all static variables.

isInitialized

```
public static boolean isInitialized()
```

init

```
public void init(ServletConfig p_config)
  throws ServletException
```

The init method of the HTTPServlet class.

(continued from last page)

destroy

```
public void destroy()
```

getServletInfo

```
public String getServletInfo()
```

This methode returns information about the servlet.

Returns:
information about the servlet.

Package
com.jlaby.test

com.jlaby.test
Class TestClient

```
java.lang.Object
  |
  +-com.jlaby.client.Client
      |
      +-com.jlaby.test.TestClient
```

All Implemented Interfaces:
java.awt.event.ActionListener

public class **TestClient**
extends Client
implements java.awt.event.ActionListener

Object in the laby.

Constructor Summary

public	`TestClient()`

Method Summary

void	`actionPerformed(java.awt.event.ActionEvent evt)`
void	`display()` Shows the help window centered on screen.
static void	`main(String[] args)`

Methods inherited from class `com.jlaby.client.Client`

`centerOnScreen, getCharacterInfo, isLoggedIn, login, logout, performAction`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `java.awt.event.ActionListener`

`actionPerformed`

Constructors

TestClient

`public TestClient()`

Methods

main

```
public static void main(String[] args)
```

display

```
public void display()
```

Shows the help window centered on screen.

actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent evt)
```

Package
com.jlaby.util

com.jlaby.util
Class Base64

```
java.lang.Object
  |
  +-com.jlaby.util.Base64
```

public class **Base64**
extends Object

This class is used for Base64-encoding (as specified in RFC1521). PLEASE NOTE: There is no 76 character limit for the resulting string. The primary purpose of this class is to encode username and passwords in HTTP/Proxy authentication requests and to be as small, lightweight and fast as possible.

Constructor Summary

`public`	`Base64()` Default constructor

Method Summary

`static byte[]`	`decode(String text)` Decode Base64 encoded data which is contained in a String.
`static String`	`encode(byte[] source)` Encode a byte array Base64 compliant.
`static String`	`encode(String source)` Encode a textstring Base64 compliant.

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Base64

`public Base64()`

Default constructor

Methods

decode

`public static byte[] decode(String text)`

Decode Base64 encoded data which is contained in a String.

Parameters:
`text` - The String containing the Base64 encoded data.

(continued from last page)

Returns:
A byte-array with the decoded data.

encode

```
public static String encode(byte[] source)
```

Encode a byte array Base64 compliant.

Parameters:
`source` - A byte-array containing the data to be encoded

Returns:
A String object holding a Base64-compliant String

encode

```
public static String encode(String source)
```

Encode a textstring Base64 compliant.

Parameters:
`source` - A String object containing the text to be encoded

Returns:
A String object holding a Base64-compliant String

com.jlaby.util
Class CurrentThread

```
java.lang.Object
   |
   +-com.jlaby.util.CurrentThread
```

public class **CurrentThread**
extends Object

Maps a name (for log entries) to the reference of threads. This is necessary in order to be able to find out which log entries in a log file belong to the same request thread (if several requests were handled at the same time).

Constructor Summary

public	`CurrentThread()`

Method Summary

static void	`add()` Adds the current thread to the internal list with its current name.
static void	`add(String threadName)` Adds the current thread to the internal list with a given name.
static String	`getBareName()` Returns the name of the current Thread in the internal list.
static String	`getName()` Returns the name of the current Thread in the internal list.
static void	`remove()` Removes the current thread from the internal list.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

CurrentThread

public **CurrentThread**()

Methods

remove

public static void **remove**()

(continued from last page)

Removes the current thread from the internal list.

add

`public static void add()`

Adds the current thread to the internal list with its current name.

add

`public static void add(String threadName)`

Adds the current thread to the internal list with a given name.

Parameters:

`threadName` - The name for the thread.

getName

`public static String getName()`

Returns the name of the current Thread in the internal list.

Returns:

The name of the current thread with surrounding brackets and blanks (ready to be used in other Strings).

getBareName

`public static String getBareName()`

Returns the name of the current Thread in the internal list.

Returns:

The bare name of the current thread.

com.jlaby.util
Class FileUtil

```
java.lang.Object
  |
  +-com.jlaby.util.FileUtil
```

public class **FileUtil**
extends Object

Generic file- and path-handling utility class

Constructor Summary

`public`	`FileUtil()`

Method Summary

`static String`	`endPathWithSeparator(String path)` This little method ensures that a path string ends with a file separator character by appending one if necessary.
`static void`	`makeDirsForFile(String filename)` Create all subdirectories for a given file path.
`static byte[]`	`readBinaryFile(String filename)` Reads a binary file and returns the content of the file as a byte array.
`static String[]`	`readTextFile(String filename)` Reads a textfile and returns the content of the file as a String array.
`static String[]`	`readTextFileStream(Reader reader)` Reads a textfile and returns the content of the file as a String array.
`static void`	`writeTextFile(String filename, String[] content)` Writes a textfile and creates the content of the file from a String array.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

FileUtil

`public FileUtil()`

Methods

(continued from last page)

endPathWithSeparator

```
public static String endPathWithSeparator(String path)
```

This little method ensures that a path string ends with a file separator character by appending one if necessary. It's just two lines of code, but its used very often in a lot of classes throughout the TARSEC package, that's why it has been put into this own method here.

Parameters:

`path` - The path that should get a file separator char.

readBinaryFile

```
public static byte[] readBinaryFile(String filename)
  throws IOException
```

Reads a binary file and returns the content of the file as a byte array. Be careful to not use this method on very large files.

Parameters:

`filename` - The absolute path of the binary file.

readTextFile

```
public static String[] readTextFile(String filename)
  throws IOException
```

Reads a textfile and returns the content of the file as a String array. Be careful to not use this method on very large files.

Parameters:

`filename` - The absolute path of the textfile.

readTextFileStream

```
public static String[] readTextFileStream(Reader reader)
  throws IOException
```

Reads a textfile and returns the content of the file as a String array. Be careful to not use this method on very large files.

Parameters:

`filename` - The absolute path of the textfile.

writeTextFile

```
public static void writeTextFile(String filename,
         String[] content)
  throws IOException
```

Writes a textfile and creates the content of the file from a String array.

Parameters:

`filename` - The absolute path of the textfile.

`content` - The contents of the textfile.

makeDirsForFile

```
public static void makeDirsForFile(String filename)
  throws IOException
```

(continued from last page)

Create all subdirectories for a given file path. This is usefull when you want to create a new file and want to make sure that the whole path exists.

Parameters:

`filepath` - Absolute path of the file, including the file itself.

com.jlaby.util
Interface ILabyConstants

All Known Implementing Classes:

LabyServer, LabyServlet, IdGenerator, IdGenerator, LogManager, BrowserClientHandler, AbstractBatchJob, LoginPanel, CreateUserPanel

public interface **ILabyConstants**
extends

This interface contains constants used throughout the whole rest of the sources in order to avoid hard-coded strings (when referencing a property name, for instance). This makes later changes easier and the code more robust.

Field Summary

`public static final`	`FALSE` A String value representing a boolean value FALSE Value: **`false`**
`public static final`	`PROP_CONFIG_FILE` The property specifying the path of the configuration properties file. Value: **`config.file`**
`public static final`	`PROP_DB_ORACLE_SEQNAME` The property specifying the name of the ORACLE sequence object used to create unique id's. Value: **`oracle.uniqueid.sequence.name`**
`public static final`	`PROP_DB_VENDOR` The property specifying the name of the vendor of the database used. Value: **`database.vendor`**
`public static final`	`PROP_HTML_CREATE` The property specifying the path of the HTML template for the "Create new user" page. Value: **`create.user.html.template`**
`public static final`	`PROP_HTML_INVENTORY` The property specifying the path of the HTML template for the inventory page. Value: **`inventory.html.template`**
`public static final`	`PROP_HTML_LOGIN` The property specifying the path of the HTML template for the "Log in" page. Value: **`login.user.html.template`**
`public static final`	`PROP_HTML_MAINPANEL` The property specifying the path of the HTML template for the laby game page. Value: **`mainpanel.html.template`**
`public static final`	`PROP_JDBC_DRIVER_CLASS` The property specifying the JDBC driver class. Value: **`jdbc.driver.class`**

`public static final`	`PROP_JDBC_DRIVER_URL` The property specifying the JDBC driver URL. Value: **`jdbc.driver.url`**
`public static final`	`PROP_LOG_DESC` The property specifying the description for the Logger instance. Value: **`log.desc`**
`public static final`	`PROP_LOG_DIR` The property specifying the directory for the JLog log files. Value: **`log.dir`**
`public static final`	`PROP_LOG_FILE` The property specifying the name of the standard log messages file. Value: **`log.file`**
`public static final`	`PROP_LOG_NAME` The property specifying the JLog name of the Logger instance. Value: **`log.name`**
`public static final`	`PROP_LOG_TRACE` The property specifying the name of the trace log file. Value: **`log.trace`**
`public static final`	`PROP_SERVER_PORT` The property specifying the port for the server listener Value: **`server.port`**
`public static final`	`PROP_SQL_NEWUNIQUEID` The property specifying the SQL statement for creating a new unique id. Value: **`sql.query.newuniqueid`**
`public static final`	`PROP_SQL_QUERY_CREATEUSER` The property specifying the SQL statement for creating data for a new user. Value: **`sql.query.createuserdata`**
`public static final`	`PROP_SQL_QUERY_GETVIEW` The property specifying the SQL statement for retrieving data about a user's view. Value: **`sql.query.getviewusers`**
`public static final`	`PROP_SQL_QUERY_LOADUSER` The property specifying the SQL statement for loading a user's state. Value: **`sql.query.loaduserdata`**
`public static final`	`PROP_SQL_QUERY_SAVEUSER` The property specifying the SQL statement for saving a user's state. Value: **`sql.query.saveuserdata`**
`public static final`	`PROP_SQL_UNIQUEID_MCKOI` The property specifying the SQL statement for loading a user's state. Value: **`sql.query.uniqueid.mckoi`**
`public static final`	`PROP_SQL_UNIQUEID_ORACLE` The property specifying the SQL statement for loading a user's state. Value: **`sql.query.uniqueid.oracle`**

`public static final`	`PROP_URL_CREATE` The property specifying the URL used to get to the "Create user" page. Value: **`url.gui.createuser`**
`public static final`	`PROP_URL_LABY` The property specifying the URL used to get to the laby game. Value: **`url.laby`**
`public static final`	`PROP_URL_LOGIN` The property specifying the URL used to get to the "Log in" page. Value: **`url.gui.login`**
`public static final`	`TRUE` A String value representing a boolean value TRUE Value: **`true`**

Fields

TRUE

```
public static final java.lang.String TRUE
```

A String value representing a boolean value TRUE
Constant value: **`true`**

FALSE

```
public static final java.lang.String FALSE
```

A String value representing a boolean value FALSE
Constant value: **`false`**

PROP_SERVER_PORT

```
public static final java.lang.String PROP_SERVER_PORT
```

The property specifying the port for the server listener
Constant value: **`server.port`**

PROP_CONFIG_FILE

```
public static final java.lang.String PROP_CONFIG_FILE
```

The property specifying the path of the configuration properties file.
Constant value: **`config.file`**

PROP_LOG_DIR

```
public static final java.lang.String PROP_LOG_DIR
```

The property specifying the directory for the JLog log files.
Constant value: **`log.dir`**

PROP_LOG_TRACE

```
public static final java.lang.String PROP_LOG_TRACE
```

(continued from last page)

The property specifying the name of the trace log file.
Constant value: **`log.trace`**

PROP_LOG_FILE

`public static final java.lang.String` **`PROP_LOG_FILE`**

The property specifying the name of the standard log messages file.
Constant value: **`log.file`**

PROP_LOG_NAME

`public static final java.lang.String` **`PROP_LOG_NAME`**

The property specifying the JLog name of the Logger instance.
Constant value: **`log.name`**

PROP_LOG_DESC

`public static final java.lang.String` **`PROP_LOG_DESC`**

The property specifying the description for the Logger instance.
Constant value: **`log.desc`**

PROP_SQL_UNIQUEID_MCKOI

`public static final java.lang.String` **`PROP_SQL_UNIQUEID_MCKOI`**

The property specifying the SQL statement for loading a user's state.
Constant value: **`sql.query.uniqueid.mckoi`**

PROP_SQL_UNIQUEID_ORACLE

`public static final java.lang.String` **`PROP_SQL_UNIQUEID_ORACLE`**

The property specifying the SQL statement for loading a user's state.
Constant value: **`sql.query.uniqueid.oracle`**

PROP_SQL_QUERY_LOADUSER

`public static final java.lang.String` **`PROP_SQL_QUERY_LOADUSER`**

The property specifying the SQL statement for loading a user's state.
Constant value: **`sql.query.loaduserdata`**

PROP_SQL_QUERY_SAVEUSER

`public static final java.lang.String` **`PROP_SQL_QUERY_SAVEUSER`**

The property specifying the SQL statement for saving a user's state.
Constant value: **`sql.query.saveuserdata`**

PROP_SQL_QUERY_CREATEUSER

`public static final java.lang.String` **`PROP_SQL_QUERY_CREATEUSER`**

The property specifying the SQL statement for creating data for a new user.
Constant value: **`sql.query.createuserdata`**

PROP_SQL_QUERY_GETVIEW

`public static final java.lang.String` **`PROP_SQL_QUERY_GETVIEW`**

The property specifying the SQL statement for retrieving data about a user's view.
Constant value: **`sql.query.getviewusers`**

PROP_SQL_NEWUNIQUEID

`public static final java.lang.String` **`PROP_SQL_NEWUNIQUEID`**

The property specifying the SQL statement for creating a new unique id.
Constant value: **`sql.query.newuniqueid`**

PROP_DB_VENDOR

`public static final java.lang.String` **`PROP_DB_VENDOR`**

The property specifying the name of the vendor of the database used.
Constant value: **`database.vendor`**

PROP_DB_ORACLE_SEQNAME

`public static final java.lang.String` **`PROP_DB_ORACLE_SEQNAME`**

The property specifying the name of the ORACLE sequence object used to create unique id's.
Constant value: **`oracle.uniqueid.sequence.name`**

PROP_JDBC_DRIVER_CLASS

`public static final java.lang.String` **`PROP_JDBC_DRIVER_CLASS`**

The property specifying the JDBC driver class.
Constant value: **`jdbc.driver.class`**

PROP_JDBC_DRIVER_URL

`public static final java.lang.String` **`PROP_JDBC_DRIVER_URL`**

The property specifying the JDBC driver URL.
Constant value: **`jdbc.driver.url`**

PROP_URL_LOGIN

`public static final java.lang.String` **`PROP_URL_LOGIN`**

The property specifying the URL used to get to the "Log in" page.
Constant value: **`url.gui.login`**

PROP_URL_CREATE

`public static final java.lang.String` **`PROP_URL_CREATE`**

The property specifying the URL used to get to the "Create user" page.
Constant value: **`url.gui.createuser`**

(continued from last page)

PROP_URL_LABY

```
public static final java.lang.String PROP_URL_LABY
```

The property specifying the URL used to get to the laby game.
Constant value: **`url.laby`**

PROP_HTML_INVENTORY

```
public static final java.lang.String PROP_HTML_INVENTORY
```

The property specifying the path of the HTML template for the inventory page.
Constant value: **`inventory.html.template`**

PROP_HTML_MAINPANEL

```
public static final java.lang.String PROP_HTML_MAINPANEL
```

The property specifying the path of the HTML template for the laby game page.
Constant value: **`mainpanel.html.template`**

PROP_HTML_LOGIN

```
public static final java.lang.String PROP_HTML_LOGIN
```

The property specifying the path of the HTML template for the "Log in" page.
Constant value: **`login.user.html.template`**

PROP_HTML_CREATE

```
public static final java.lang.String PROP_HTML_CREATE
```

The property specifying the path of the HTML template for the "Create new user" page.
Constant value: **`create.user.html.template`**

com.jlaby.util
Class LabyVersion

```
java.lang.Object
  |
  +-com.jlaby.util.LabyVersion
```

public class **LabyVersion**
extends Object

Reads the version information from "version.properties" (which is in the root of the classpath) and provides statis methods to access this information.

Constructor Summary

`public`	`LabyVersion()`

Method Summary

`static String`	`getBuildCVSTag()` Returns the CVS tag with which all the project files had been tagged for the build.
`static String`	`getBuildID()` Returns the build ID number.
`static String`	`getReleaseID()` Returns the release ID (which is the public version number of the product or library).
`static String`	`getVersion()` Returns the release ID (which is the public version number of the product or library).

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

LabyVersion

`public LabyVersion()`

Methods

getBuildCVSTag

`public static String getBuildCVSTag()`

Returns the CVS tag with which all the project files had been tagged for the build.

(continued from last page)

Returns:
The CVS tag string.

getBuildID

```
public static String getBuildID()
```

Returns the build ID number.

Returns:
The build ID.

getReleaseID

```
public static String getReleaseID()
```

Returns the release ID (which is the public version number of the product or library).

Returns:
The release ID.

getVersion

```
public static String getVersion()
```

Returns the release ID (which is the public version number of the product or library).

Returns:
The release ID.

com.jlaby.util
Class PropertyUtil

```
java.lang.Object
   |
   +-com.jlaby.util.PropertyUtil
```

public class **PropertyUtil**
extends Object

Utility class for property handling

Constructor Summary

public	PropertyUtil()

Method Summary

static Properties	load(String resource) Load Properties either from a file in the filesystem or in the classpath.

Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructors

PropertyUtil

public **PropertyUtil**()

Methods

load

public static Properties **load**(String resource)

Load Properties either from a file in the filesystem or in the classpath.

com.jlaby.util
Class StringUtil

```
java.lang.Object
  |
  +-com.jlaby.util.StringUtil
```

public class **StringUtil**
extends Object

Generic String handling toolbox.

Constructor Summary

public	`StringUtil()`

Method Summary

static int	`getASCII(String txt)` Get the ASCII-code of a single character.
static String	`getCharacter(int ascii)` Returns the character for a certain ASCII-code.
static String[]	`patch(String[] text, String old, String patch)` Run through all entries of a string array and patch every textstring by replacing one or several occurrences of a text in it with another text.
static String	`patch(String text, String old, String patch)` Patch a textstring by replacing one or several occurrences of a text in it with another text.
static String[]	`sortAscending(String[] source)` Sort a String array ascending.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

StringUtil

```
public StringUtil()
```

Methods

sortAscending

```
public static String[] sortAscending(String[] source)
```

(continued from last page)

Sort a String array ascending.

Parameters:

`source` - The String array which will be sorted.

Returns:

The new array, sorted in ascending order.

patch

```
public static String[] patch(String[] text,
         String old,
         String patch)
```

Run through all entries of a string array and patch every textstring by replacing one or several occurrences of a text in it with another text.

Parameters:

`text` - The text array that should be patched.
`old` - The text that should be replaced.
`patch` - The replacement text.

Returns:

The patched text array.

patch

```
public static String patch(String text,
         String old,
         String patch)
```

Patch a textstring by replacing one or several occurrences of a text in it with another text.

Parameters:

`text` - The text that should be patched.
`old` - The text that should be replaced.
`patch` - The replacement text.

Returns:

The patched text.

getASCII

```
public static int getASCII(String txt)
```

Get the ASCII-code of a single character.

Parameters:

`txt` - A string containing a single character.

Returns:

The ASCII-value of the character.

getCharacter

```
public static String getCharacter(int ascii)
```

Returns the character for a certain ASCII-code.

(continued from last page)

Parameters:

`ascii` - The ascii code (value from 0 - 255).

Returns:

A String containing a single character.

com.jlaby.util
Class Util

```
java.lang.Object
  |
  +-com.jlaby.util.Util
```

public class **Util**
extends Object

Utility class

Constructor Summary

public	`Util()`

Method Summary

`static void`	`disposeCurrentThreadInfo()`
`static String`	`extractFile(String resource)` Extracts a file which must be somewhere in the CLASSPATH to the system specific temp directory and returns a reference to the extracted file.
`static Object`	`getObject(byte[] data)`
`static byte[]`	`getObjectData(Object object)`
`static PrintWriter`	`getOut()`
`static HttpSession`	`getSession()`
`static java.awt.Image`	`loadImage(String picname, java.awt.Frame parent, Object client)` Load an image either from a local file or a URL.
`static java.awt.Image`	`loadImage(String picname, Object client)` Load an image either from a local file or a URL.
`static java.awt.Image`	`loadImageFromFile(String filename, java.awt.Frame parent)` Load an image from a GIF or JPEG file, using the MediaTracker to make sure the image is loaded completely before the thread continues.
`static void`	`println(String txt)` Print to servlets output stream
`static void`	`setOut(PrintWriter out)`
`static void`	`setSession(HttpSession session)`

Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

Constructors

Util

```
public Util()
```

Methods

setOut

```
public static void setOut(PrintWriter out)
```

setSession

```
public static void setSession(HttpSession session)
```

disposeCurrentThreadInfo

```
public static void disposeCurrentThreadInfo()
```

getObjectData

```
public static byte[] getObjectData(Object object)
  throws IOException
```

getObject

```
public static Object getObject(byte[] data)
  throws IOException,
         ClassNotFoundException
```

getSession

```
public static HttpSession getSession()
```

getOut

```
public static PrintWriter getOut()
```

(continued from last page)

println

```
public static void println(String txt)
```

Print to servlets output stream

Parameters:
`text` - The text to print

loadImage

```
public static java.awt.Image loadImage(String picname,
         Object client)
```

Load an image either from a local file or a URL. It may also be in an archive file within the classpath; it that case, it's beeing loaded by the classloader as a resource.

Parameters:
`picname` - The name of the imagefile, either absolute or relative.

loadImage

```
public static java.awt.Image loadImage(String picname,
         java.awt.Frame parent,
         Object client)
```

Load an image either from a local file or a URL. It may also be in an archive file within the classpath; in that case, it's beeing loaded by the classloader as a resource.

Parameters:
`picname` - The name of the imagefile, either absolute or relative.
`parent` - The parent Frame (which is used for the MediaTracker). This value may be null.

loadImageFromFile

```
public static java.awt.Image loadImageFromFile(String filename,
         java.awt.Frame parent)
```

Load an image from a GIF or JPEG file, using the MediaTracker to make sure the image is loaded completely before the thread continues.

Parameters:
`filename` - The absolute path of the image file.

extractFile

```
public static String extractFile(String resource)
```

Extracts a file which must be somewhere in the CLASSPATH to the system specific temp directory and returns a reference to the extracted file. Use slashes as package delimiters. For example, if you want to extract a GIF image file which is in the archive inside the package COM.tarsec.icons and is named Acme.gif, use this parameter:

```
/COM/tarsec/icons/Acme.gif
```

Parameters:
`resource` - The name of the resource to unpack.

Returns:

(continued from last page)

The absolute path of the extracted file.extract file: " + re

Package
com.jlaby.view

com.jlaby.view
Interface IRepresentationTypes

All Known Implementing Classes:
ViewObject

public interface **IRepresentationTypes**
extends

Constants for types of objects (or, to be more specific, graphical representations of objects, be it characters, items or whatever).

Field Summary

public static final	CHARACTER Value: **2**
public static final	COLUMN Value: **4**
public static final	FACING_PLAYER Value: **2**
public static final	LOOKING_AWAY Value: **1**
public static final	LOOKING_LEFT Value: **4**
public static final	LOOKING_RIGHT Value: **3**
public static final	TREE Value: **3**
public static final	UNDEFINED Value: **-1**
public static final	WALL Value: **1**

Fields

WALL

```
public static final int WALL
```

(continued from last page)

Constant value: **1**

CHARACTER

```
public static final int CHARACTER
```

Constant value: **2**

TREE

```
public static final int TREE
```

Constant value: **3**

COLUMN

```
public static final int COLUMN
```

Constant value: **4**

UNDEFINED

```
public static final int UNDEFINED
```

Constant value: **-1**

LOOKING_AWAY

```
public static final int LOOKING_AWAY
```

Constant value: **1**

FACING_PLAYER

```
public static final int FACING_PLAYER
```

Constant value: **2**

LOOKING_RIGHT

```
public static final int LOOKING_RIGHT
```

Constant value: **3**

LOOKING_LEFT

```
public static final int LOOKING_LEFT
```

Constant value: **4**

com.jlaby.view
Class ViewObject

```
java.lang.Object
  |
  +-com.jlaby.view.ViewObject
```

All Implemented Interfaces:

IRepresentationTypes, Serializable

public class **ViewObject**
extends Object
implements Serializable, IRepresentationTypes

Object in the laby.

Fields inherited from interface `com.jlaby.view.IRepresentationTypes`
`CHARACTER`, `COLUMN`, `FACING_PLAYER`, `LOOKING_AWAY`, `LOOKING_LEFT`, `LOOKING_RIGHT`, `TREE`, `UNDEFINED`, `WALL`

Constructor Summary

`public`	`ViewObject()` Constructs the

Method Summary

`int`	`getDirection()`
`int`	`getObjectType()`
`Position`	`getPosition()`
`void`	`setDirection(int direction)`
`void`	`setObjectType(int objectType)`
`void`	`setPosition(Position position)`

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

ViewObject

```
public ViewObject()
```

Constructs the

Methods

setObjectType

```
public void setObjectType(int objectType)
```

getObjectType

```
public int getObjectType()
```

setDirection

```
public void setDirection(int direction)
```

setPosition

```
public void setPosition(Position position)
```

getDirection

```
public int getDirection()
```

getPosition

```
public Position getPosition()
```

Package
com.jlaby.world

com.jlaby.world
Interface IWorld

All Known Implementing Classes:
WorldImpl, WorldStub

public interface **IWorld**
extends

This class loads and holds the world data.

Field Summary

`public static final`	`EAST` Value: **3**
`public static final`	`NORTH` Value: **1**
`public static final`	`SOUTH` Value: **2**
`public static final`	`WEST` Value: **4**

Method Summary

`LabyAnswer`	`handleAction(LabyAction action)` This method has to deal with an action created by a "client" (local or remote).
`void`	`initialize()` Initializes the world (sounds nice, doesn't it?).

Fields

NORTH

```
public static final int NORTH
```

Constant value: **1**

SOUTH

```
public static final int SOUTH
```

Constant value: **2**

(continued from last page)

EAST

```
public static final int EAST
```

Constant value: **3**

WEST

```
public static final int WEST
```

Constant value: **4**

Methods

initialize

```
public void initialize()
  throws InitializationFailedException
```

Initializes the world (sounds nice, doesn't it?). In other words, it does in a few milliseconds what took god six days... so to speak...

Throws:

`InitializationFailedException` - if something went wrong and the world could not be created.

handleAction

```
public LabyAnswer handleAction(LabyAction action)
  throws LabyException
```

This method has to deal with an action created by a "client" (local or remote). It has to have the action handled by the appropriate handler and then create an answer to be returned to the client.

Parameters:

`action` - the action created /sent by a client.

Returns:

the answer for the client.

Throws:

`LabyException` - thrown if the action could not be handled.

com.jlaby.world
Class Position

```
java.lang.Object
  |
  +-com.jlaby.world.Position
```

All Implemented Interfaces:

Serializable

public class **Position**
extends Object
implements Serializable

This class is a holder for information about the position of any object within the Laby world.

Constructor Summary

public	`Position()` Default constructor.
public	`Position(int x, int y, int z)` Constructs a position object.

Method Summary

int	`getX()` Returns the x-coordinate.
int	`getY()` Returns the y-coordinate.
int	`getZ()` Returns the z-coordinate.
void	`setX(int value)` Sets the x-coordinate.
void	`setY(int value)` Sets the y-coordinate.
void	`setZ(int value)` Sets the z-coordinate.

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

Position

```
public Position()
```

Default constructor.

Position

```
public Position(int x,
                int y,
                int z)
```

Constructs a position object.

Parameters:

`x` - the x-coordinate
`y` - the y-coordinate
`z` - the z-coordinate

Methods

setX

```
public void setX(int value)
```

Sets the x-coordinate.

Parameters:

`value` - the value for the x-coordinate.

setY

```
public void setY(int value)
```

Sets the y-coordinate.

Parameters:

`value` - the value for the y-coordinate.

setZ

```
public void setZ(int value)
```

Sets the z-coordinate.

Parameters:

`value` - the value for the z-coordinate.

getX

```
public int getX()
```

Returns the x-coordinate.

Returns:

the value of the x-coordinate.

(continued from last page)

getY

```
public int getY()
```

Returns the y-coordinate.

Returns:
the value of the y-coordinate.

getZ

```
public int getZ()
```

Returns the z-coordinate.

Returns:
the value of the z-coordinate.

com.jlaby.world
Class WorldFactory

```
java.lang.Object
  |
  +-com.jlaby.world.WorldFactory
```

public class **WorldFactory**
extends Object

This class creates a singleton world object instance for a local world or a stub for a remote world.

Constructor Summary

`public`	`WorldFactory()`

Method Summary

`static IWorld`	`createLocalWorld()` Creates a local world instance.
`static IWorld`	`createRemoteWorld(String url)` Creates a stub for a HTTP connection to a remote world.
`static IWorld`	`createRemoteWorld(String url, int port)` Creates a stub for a HTTP connection to a remote world.
`static IWorld`	`getWorld()` Returns the reference to the Singleton instance of the world object (or the world stub in case of a remote client).

Methods inherited from class `java.lang.Object`
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

WorldFactory

```
public WorldFactory()
```

Methods

getWorld

```
public static IWorld getWorld()
  throws NotCreatedException
```

Returns the reference to the Singleton instance of the world object (or the world stub in case of a remote client).

(continued from last page)

Returns:
the singleton world reference.

Throws:
`thrown` - if the world had not been created yet.

createLocalWorld

```
public static IWorld createLocalWorld()
  throws WorldException
```

Creates a local world instance. This method is typically only used once, by the Laby server / servlet during the initialization process. Clients will use the `createRemoteWorld()` method to communicate with such a world.

Returns:
a reference to the singleton world object instance.

Throws:
`thrown` - if the world could not be created.

createRemoteWorld

```
public static IWorld createRemoteWorld(String url)
  throws WorldException
```

Creates a stub for a HTTP connection to a remote world.

Parameters:
`url` - the URL of the Laby world servlet.

Returns:
a reference to the singleton world object instance (actually just a stub).

Throws:
`thrown` - if the world could not be created.

createRemoteWorld

```
public static IWorld createRemoteWorld(String url,
         int port)
  throws WorldException
```

Creates a stub for a HTTP connection to a remote world.

Parameters:
`url` - the URL of the Laby world servlet.
`port` - the port number of the servlet.

Returns:
a reference to the singleton world object instance (actually just a stub).

Throws:
`thrown` - if the world could not be created.

Package
com.jlaby.world.exception

com.jlaby.world.exception
Class InitializationFailedException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.world.exception.WorldException
                      |
                      +-com.jlaby.world.exception.InitializationFailedException
```

All Implemented Interfaces:
Serializable

public class **InitializationFailedException**
extends WorldException

This exception is thrown when a problem occurred during the initialization of a newly created Laby world object (remote or local).

Constructor Summary

`public`	`InitializationFailedException()` Constructs an unspecific exception.
`public`	`InitializationFailedException(String txt)` Constructs an exception with a message text.
`public`	`InitializationFailedException(Throwable causingException)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

InitializationFailedException

`public` **`InitializationFailedException()`**

(continued from last page)

Constructs an unspecific exception.

InitializationFailedException

`public` **`InitializationFailedException`**`(String txt)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

InitializationFailedException

`public` **`InitializationFailedException`**`(Throwable causingException)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.world.exception
Class NotCreatedException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.world.exception.WorldException
                      |
                      +-com.jlaby.world.exception.NotCreatedException
```

All Implemented Interfaces:
Serializable

public class **NotCreatedException**
extends WorldException

This exception is thrown when an attempt was made to access the static, singleton instance of the world object before it had actually been created.

Constructor Summary

`public`	`NotCreatedException()` Constructs an unspecific exception.
`public`	`NotCreatedException(String txt)` Constructs an exception with a message text.
`public`	`NotCreatedException(Throwable causingException)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

NotCreatedException

`public **NotCreatedException**()`

Constructs an unspecific exception.

NotCreatedException

`public **NotCreatedException**(String txt)`

Constructs an exception with a message text.

Parameters:

`txt` - the exception message text.

NotCreatedException

`public **NotCreatedException**(Throwable causingException)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:

`causingException` - the problem-causing exception.

com.jlaby.world.exception
Class WorldException

```
java.lang.Object
  |
  +-java.lang.Throwable
      |
      +-java.lang.Exception
          |
          +-java.lang.RuntimeException
              |
              +-com.jlaby.exception.LabyException
                  |
                  +-com.jlaby.world.exception.WorldException
```

All Implemented Interfaces:
Serializable

Direct Known Subclasses:
NotCreatedException, InitializationFailedException

public class **WorldException**
extends LabyException

Generic world-related exception.

Constructor Summary

public	`WorldException()` Constructs an unspecific exception.
public	`WorldException(String txt)` Constructs an exception with a message text.
public	`WorldException(Throwable causingException)` Constructs an exception with a reference to another exception which caused the problem.

Methods inherited from class `com.jlaby.exception.LabyException`

`getCausingException`

Methods inherited from class `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods inherited from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

(continued from last page)

WorldException

`public **WorldException**()`

Constructs an unspecific exception.

WorldException

`public **WorldException**(String txt)`

Constructs an exception with a message text.

Parameters:
`txt` - the exception message text.

WorldException

`public **WorldException**(Throwable causingException)`

Constructs an exception with a reference to another exception which caused the problem. The causing exception will be wrapped into this exception and can be retrieved later using the "getCausingException()" method.

Parameters:
`causingException` - the problem-causing exception.

Index