Flow Charts of SmartHomeClient

Table of Contents

[Introduction 2](#_Toc379902378)

[Class Structures 2](#_Toc379902379)

[package SmartHomeClient -> class SmartHomeClient 2](#_Toc379902380)

[Flow Chart of SmartHomeClient 3](#_Toc379902381)

[Legends of colour code 3](#_Toc379902382)

[Method: public static void main (String[] args) 3](#_Toc379902383)

[Method: private static void checkArguments(String[] args) 3](#_Toc379902384)

[Method: private static void handleUserInput(String UserInput) 4](#_Toc379902385)

[Method: private static String[] getCommand (String s) 5](#_Toc379902386)

[Method: private static void handleConnectCommand (String[] commands) 6](#_Toc379902387)

[Method: private static void handleSocketTransactionsCommands (String[] commands) 7](#_Toc379902388)

[Method: private static void handleExitStatus() 8](#_Toc379902389)

[Method: private static void printUsage() 8](#_Toc379902390)

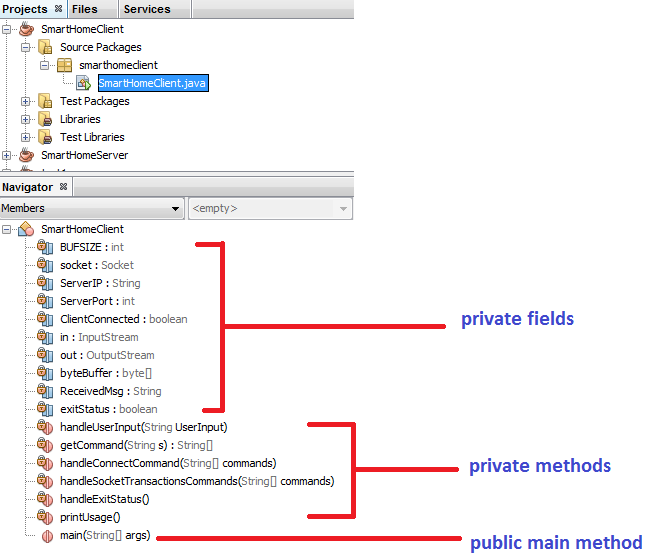
# Introduction

In the class structure section below, there is a screenshot of the Netbeans project and member window showing all the fields and methods in the class.

The main method of SmartHomeClient calls several private methods to keep the code modular, clean and readable. So, the flow chart section has flow-charts of main and all the other private methods.

# Class Structures

## package SmartHomeClient -> class SmartHomeClient



# Flow Chart of SmartHomeClient

## Legends of colour code

Green – start of a method Red – end of a method Dark Orange - exception

Yellow – decision making Blue – processing/execution

## Method: public static void main (String[] args)



## Method: private static void checkArguments(String[] args)



## Method: private static void handleUserInput(String UserInput)



## Method: private static String[] getCommand (String s)



## Method: private static void handleConnectCommand (String[] commands)



## Method: private static void handleSocketTransactionsCommands (String[] commands)



## Method: private static void handleExitStatus()



## Method: private static void printUsage()

