FUN WITH JSHELL

Anton Epple

Anton Epple

Trainer and Consultant

Eppleton IT Consulting

Twitter: @monacotoni

www.eppleton.de

AGENDA

- **Was ist die JShell?**
- Wozu kann man sie verwenden?
- Wie funktioniert die JShell API?
- Hacken
- JShell API
- Wie kann man mit der JShell interaktiv Anwendungen Debuggen?
- Fragen

REPL

***R**ead **E**val **P**rint **L**oop*

READ

EVAL

PRINT

LOOP

API

TOOL

TEIL DES JDK

AGENDA

- Was ist die JShell?
- **Wozu kann man sie verwenden?**
- Wie funktioniert die JShell API?
- Hacken
- JShell API
- Wie kann man mit der JShell interaktiv Anwendungen Debuggen?
- Fragen

JAVA LERNEN

APIS AUSPROBIEREN

ENTWICKELN

DEBUGGEN

OHNE JSHELL

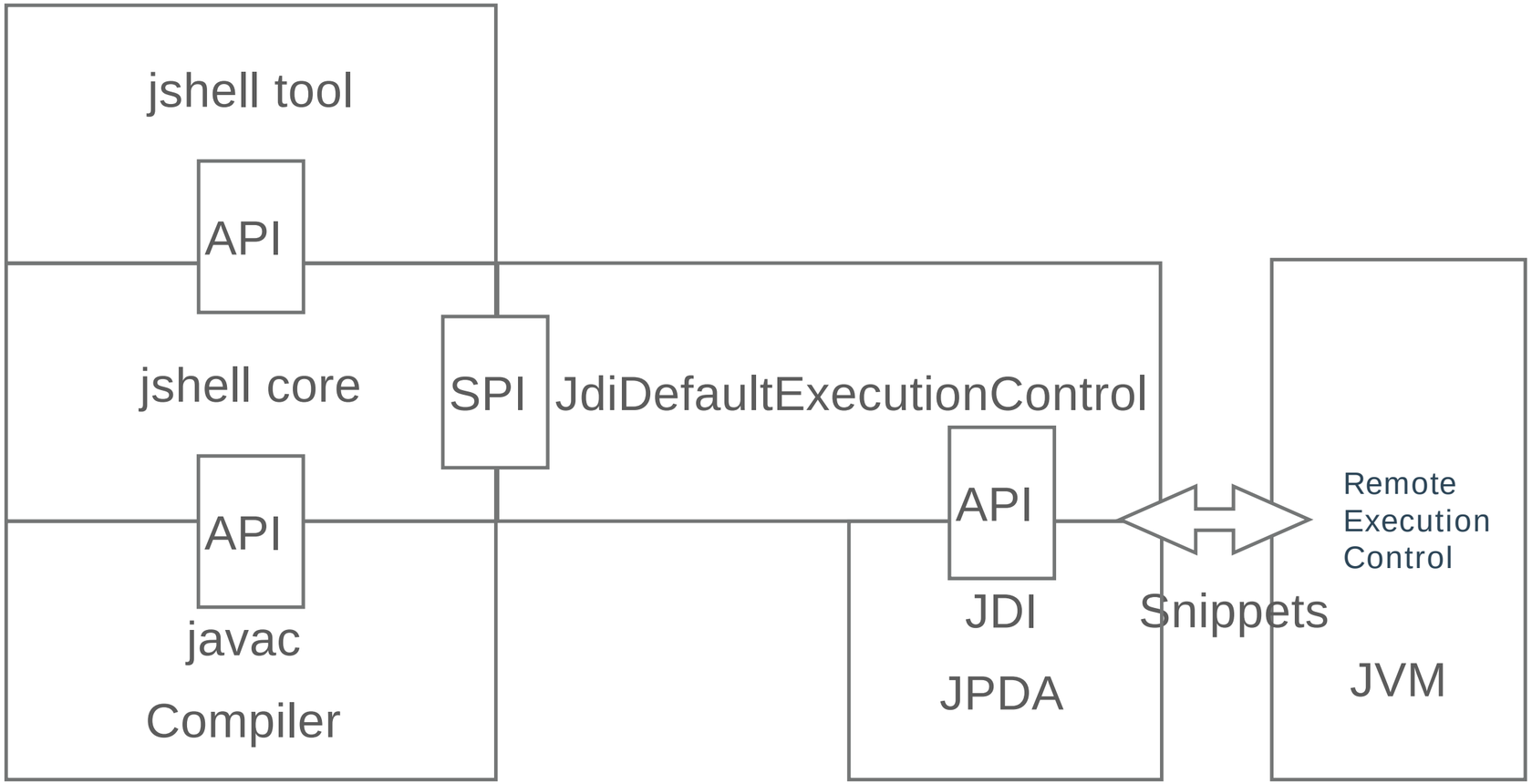
- Anlegen eines Projektes (oder einer Java Datei auf der Kommandozeile)
- Anlegen einer Java Datei mit einer Klasse und einer Methode
- Testcode schreiben
- Kompilieren
- Ausführen
- Ändern & wiederholen

=>Umständlich & langsam für kleine Experimente, Test einzelner Methoden

AGENDA

- Was ist die JShell?
- Wozu kann man sie verwenden?
- **Wie funktioniert die JShell API?**
- Hacken
- JShell API
- Wie kann man mit der JShell interaktiv Anwendungen Debuggen?
- Fragen

WIE FUNKTIONIERT DAS?



=> IMMER AKTUELL

AGENDA

- Was ist die JShell?
- Wozu kann man sie verwenden?
- Wie funktioniert die JShell API?
- **Hacken**
- JShell API
- Wie kann man mit der JShell interaktiv Anwendungen Debuggen?
- Fragen

JSHELL

```
$ jshell
|  Welcome to JShell -- Version 9.0.1
|  For an introduction type: /help intro
jshell>
```

TAB COMPLETION

```
jshell> Sys
```

TAB COMPLETION

```
jshell> Sys<TAB>
```

TAB COMPLETION

```
jshell> System
```

TAB COMPLETION

```
jshell> System.<TAB>
```

TAB COMPLETION

```
jshell> System.
Logger                LoggerFinder          arraycopy(
class                 clearProperty(        console()
currentTimeMillis()   err                   exit(
gc()                  getLogger(            getProperties()
getProperty(          getSecurityManager()  getenv(
identityHashCode(     in                    inheritedChannel()
[…]
```

TAB COMPLETION

```
jshell> System.o<TAB>
```

TAB COMPLETION

```
jshell> System.out
```

TAB COMPLETION

```
jshell> System.out.p<TAB>
```

TAB COMPLETION

```
jshell> System.out.print
```

TAB COMPLETION

```
jshell> System.out.print<TAB>
```

TAB COMPLETION

```
jshell> System.out.print

print(     printf(    println(
```

TAB COMPLETION

```
jshell> System.out.printl<TAB>
```

TAB COMPLETION

```
jshell> System.out.println(<TAB>
```

TAB SHOW SIGNATURES

```
jshell> System.out.println(<TAB>

Signatures:

void PrintStream.println()

void PrintStream.println(boolean x)

void PrintStream.println(char x)

void PrintStream.println(int x)

void PrintStream.println(long x)

[…]

<press tab again to see documentation>
```

TAB SHOW JAVADOC

```
jshell> System.out.println(<TAB><TAB>
Signatures:
void PrintStream.println()
void PrintStream.println(boolean x)
void PrintStream.println(char x)
void PrintStream.println(int x)
void PrintStream.println(long x)
[…]
<press tab again to see documentation>
```

TAB JAVADOC

```
jshell> System.out.println(

void PrintStream.println()

Terminates the current line by writing the line separator string.The line

separator string is defined by the system property line.separator , and is not

necessarily a single newline character ( '\n' ).


<press tab to see next documentation>
```

OHNE STRICHPUNKT

```
jshell> System.out.println(„Hello World“)<ENTER>
```

OHNE STRICHPUNKT

```
jshell> System.out.println(„Hello World“)

Hello World



jshell>
```

EINFACHE AUSDRÜCKE AUSWERTEN

```
jshell> System.out.println(„Hello World“)

Hello World



jshell> 1 + 1
```

EINFACHE AUSDRÜCKE AUSWERTEN

```
jshell> System.out.println(„Hello World“)

Hello World



jshell> 1 + 1

$1 ==> 2



jshell>
```

VARIABLEN

```
jshell> System.out.println(„Hello World“)
Hello World


jshell> 1 + 1
$1 ==> 2


jshell> $1 * 4
$2 ==> 8
```

/vars

```
jshell> /vars
```

/vars

```
jshell> /vars
|    int $1 = 2
|    int $2 = 8


jshell>
```

/vars

```
jshell> /vars
|    int $1 = 2
|    int $2 = 8


jshell> int myInt = 7
myInt ==> 7


jshell>
```

/drop

```
jshell> /drop 1
```

/set

```
jshell> /set
|  /set editor -default
|  /set start -default
|  /set feedback normal
|
|  Available feedback modes:
|     concise
|     normal
|     silent
|     verbose
|
|  To show mode settings use '/set prompt', '/set truncation', ...
|  or use '/set mode' followed by the feedback mode name.
```

/! revaluiere letzten Ausdruck

```
jshell> /!

add(1,2)

$3 ==> 3
```

PARTIELLE AUSWERTUNG

```
jshell> public int mult(int a, int b){<ENTER>
```

JSHELL

```
jshell> public int mult(int a, int b){
   ...>     return a*b; <ENTER>
```

JSHELL

```
jshell> public int mult(int a, int b){
   ...>     return a*b;
   ...> };<ENTER>
```

JSHELL

```
jshell> public int mult(int a, int b){
   ...>     return a*b;
   ...> };
|  created method mult(int,int)
```

/save

```
jshell> /save -history ~/snippets.txt
```

/open

```
jshell> /save -history ~/snippets.txt

jshell> /open ~/snippets.txt
```

JSHELL

```
jshell> interface Printer{
   ...> void print(String toPrint);
   ...> }
|  created interface Printer
jshell> new Printer(){void print(String toPrint){System.out.println(toPrint);} }
$9 ==> 1@5f2108b5
jshell> $9.print("hallo");
hallo
```

/types

```
jshell> /types
|    interface Printer
|    class PrinterImpl


jshell>
```

/exit

```
jshell> /types
|    interface Printer
|    class PrinterImpl


jshell>/exit
|  Goodbye
```

AUFRUFPARAMETER

```
jshell <params> (files)
```

- https://docs.oracle.com/javase/9/tools/jshell.htm
- --add-modules module[,module…] // root Modules
- --module-path
- -Cflag // Compiler Flags
- --class-path path
- --feedback mode // silent, verbose…
- --help
- --startup file
- …

AUFRUFPARAMETER

```
$ jshell -class-path lib/JavaDukeAdventure.jar commands.txt
```

https://goo.gl/ExpAvg

/imports

```
$ jshell -class-path lib/JavaDukeAdventure.jar commands.txt
|  Welcome to JShell -- Version 9.0.1
|  For an introduction type: /help intro
jshell>/imports
```

/imports

```
jshell>/imports
|    import java.io.*
|    import java.math.*
|    import java.net.*
|    import java.nio.file.*
|    import java.util.*
|    import java.util.concurrent.*
[…]
|    import javadukeadventure.*
```

Import

```
jshell> import java.nio.ByteBuffer


jshell> ByteBuffer buf = ByteBuffer.allocate(48);

buf ==> java.nio.HeapByteBuffer[pos=0 lim=48 cap=48]
```

USING IMPORTED CLASSES

```
jshell> JavaDukeAdventure.start()
```

USING IMPORTED CLASSES

```
jshell> JavaDukeAdventure.start()

This is an open field west of a white house, with a boarded front door.

There is a small mailbox here.

A rubber mat saying 'Welcome to Oracle!' lies by the door.

You see:

door

mailbox

Exits are to:

EAST
```

/methods

```
jshell>/methods
```

/methods

```
jshell>/methods
|    void goWest()
|    void goSouth()
|    void goNorth()
|    void goEast()
|    void look()
```

/methods

```
jshell>look()
```

/methods

```
jshell>look()

This is an open field west of a white house, with a boarded front door.

There is a small mailbox here.

A rubber mat saying 'Welcome to Zork!' lies by the door.

You see:

door

mailbox

Exits are to:

EAST
```

/methods

```
jshell>goEast()
```

/methods

```
jshell>goEast()

You bump into the boarded front door.

jshell>
```

/methods

jshell>goEast()

You bump into the boarded front door.

jshell>goWest()

Redefine methods

```
jshell>goEast()

You bump into the boarded front door.

jshell>goWest()

This method doesn't do anything yet, try to redefine it! (hint: JavaDukeAdventure has useful methods)


jshell>
```

/list

jshell>goEast()

You bump into the boarded front door.

jshell>goWest()

This method doesn't do anything yet, try to redefine it! (hint: JavaDukeAdventure has useful methods)

jshell>/list goEast

/list - Methoden anzeigen

jshell>goEast()

You bump into the boarded front door.

jshell>goWest()

This method doesn't do anything yet, try to redefine it! (hint: JavaDukeAdventure has useful methods)

jshell>/list goEast

5 : void goEast({JavaDukeAdventure.go(Location.Direction.EAST);}

Methoden redefinieren

```
jshell>void goWest(){JavaDukeAdventure.go(Location.Direction.WEST);}
```

Methoden redefinieren

```
jshell>void goWest(){JavaDukeAdventure.go(Location.Direction.WEST);}
|  modified method goWest()


jshell>
```

Methoden redefinieren

```
jshell>void goWest(){JavaDukeAdventure.go(Location.Direction.WEST);}
|  modified method goWest()


jshell>goWest()
```

Methoden redefinieren

```
jshell>void goWest(){JavaDukeAdventure.go(Location.Direction.WEST);}
|  modified method goWest()


jshell>goWest()
There is no exit to WEST
You are here: West of House
[…]
Exits are to:
EAST
```

/open - Skript-Dateien laden

```
jshell>/open morecommands.txt
```

/open

```
jshell>/open morecommands.txt


jshell>/methods
```

/open

```
jshell>/open morecommands.txt


jshell>/methods
|    void goWest()
|    void open(String)
[…]
|    void lieDown()
|    void inventory()
|    void help()
```

//KOMMENTARE

jshell>// Spielgegenstände werden einer Methode als String-Parameter übergeben: open(„mailbox“)

jshell> // Wende Deine Java-Kenntnisse an, um zum Beispiel fehlende Klassen zu importieren, oder Interfaces zu implementieren…

jshell>/save -history saved.txt; // <- so kannst Du speichern

jshell>/exit // wenn es nicht mehr weitergeht

jshell>/open saved.txt

jshell> // Es können Exceptions auftreten. Das sind keine Bugs, sondern Herausforderungen

jshell> // Googlen hilft

jshell>// Es gibt bestimmt mehr als eine Lösung, viel Spass!

https://goo.gl/ExpAvg

AGENDA

- Was ist die JShell?
- Wozu kann man sie verwenden?
- Wie funktioniert die JShell API?
- Hacken
- **JShell API**
- Wie kann man mit der JShell interaktiv Anwendungen Debuggen?
- Fragen

JSHELL

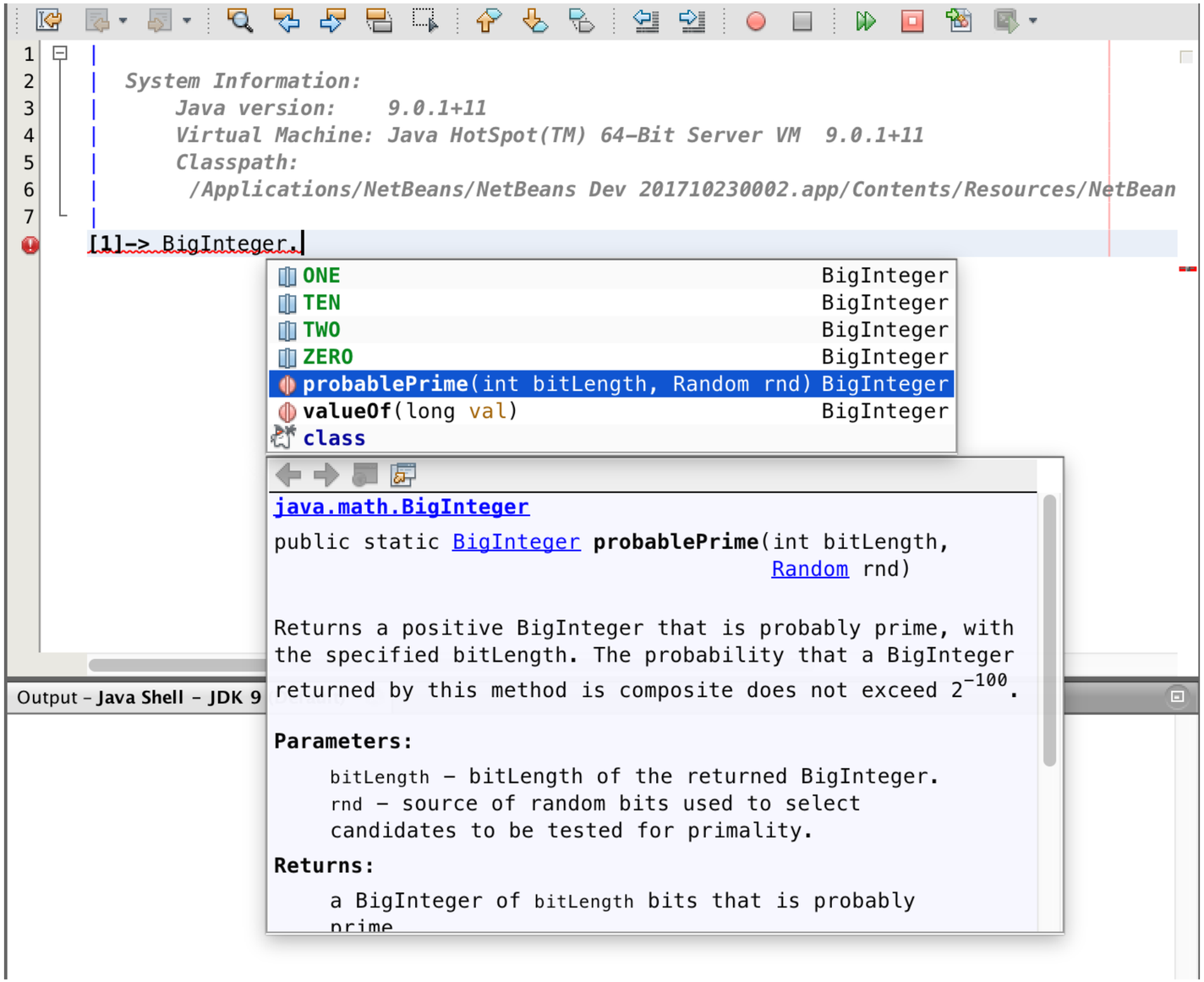
```
jshell> import jdk.jshell.*;

jshell> JShell sh = JShell.create()

sh ==> jdk.jshell.JShell@4206a205


jshell> sh.eval("1+1")

$4 ==> [SnippetEvent(snippet=Snippet:VariableKey($1)#1-
1+1,previousStatus=NONEXISTENT,status=VALID,isSignatureChange=true,causeSnippetnullvalue=2)]


jshell> $4.get(0).value();

$5 ==> "2"
```

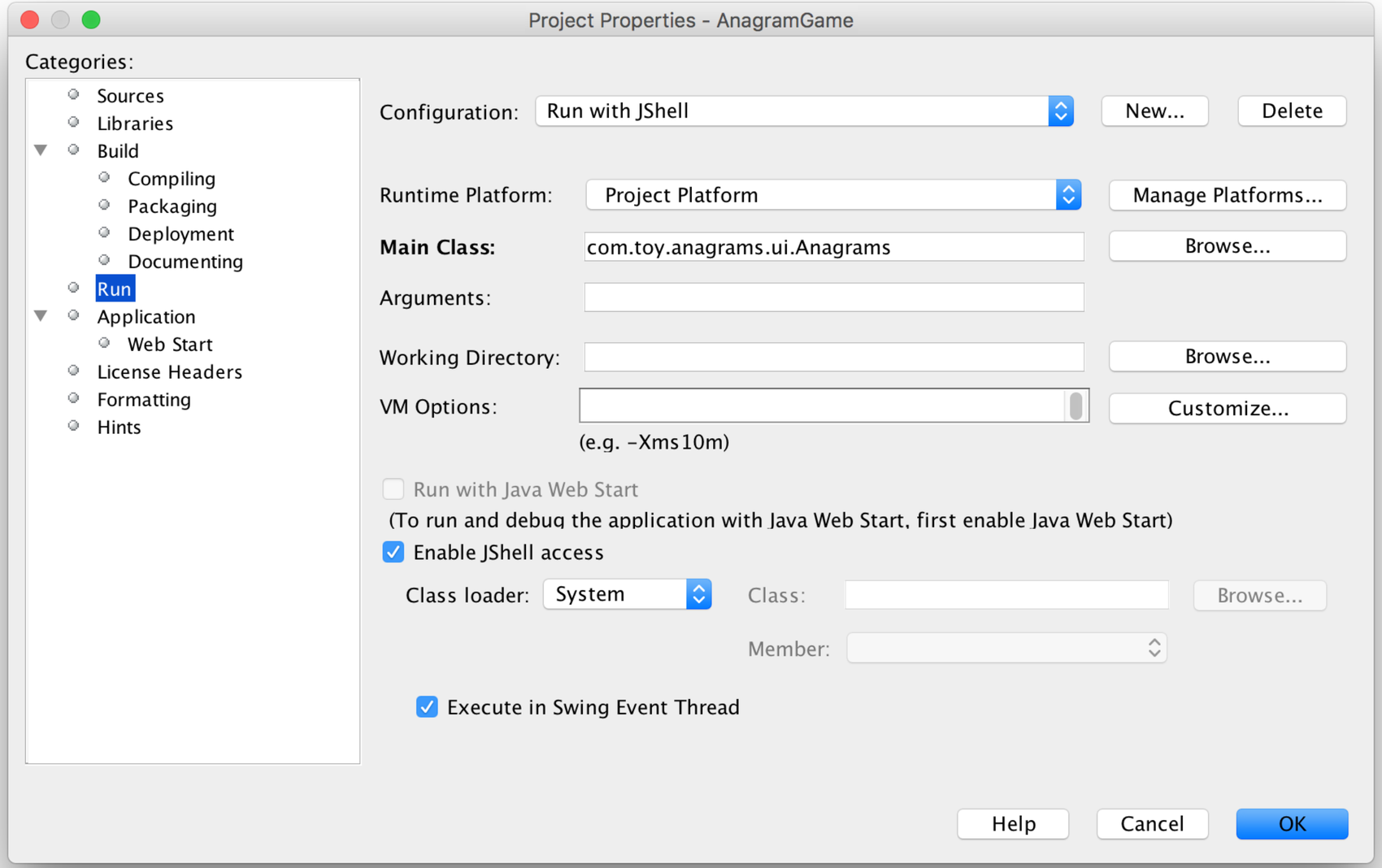
AGENDA

- Was ist die JShell?
- Wozu kann man sie verwenden?
- Wie funktioniert die JShell API?
- Hacken
- JShell API
- **Wie kann man mit der JShell interaktiv Anwendungen Debuggen?**
- Fragen

INTERAKTIV ANWENDUNGEN DEBUGGER

- NetBeans Dev
- http://bits.netbeans.org/download/trunk/nightly/latest/
- https://github.com/apache/incubator-netbeans
- https://builds.apache.org/job/incubator-netbeans-linux/
- https://cwiki.apache.org/confluence/display/NETBEANS/NetBeans+9.0+-+New+and+Noteworthy


1
2
3
4
5
6
7
System Information:
Java version: 9.0.1+11
Virtual Machine: Java HotSpot(TM) 64-Bit Server VM 9.0.1+11
Classpath:
/Applications/NetBeans/NetBeans Dev 201710230002.app/Contents/Resources/NetBean
[1]-> BigInteger.
ONE BigInteger
TEN BigInteger
TWO BigInteger
ZERO BigInteger
probablePrime(int bitLength, Random rnd) BigInteger
valueOf(long val) BigInteger
class
java.math.BigInteger
public static BigInteger probablePrime(int bitLength,
Random rnd)
Returns a positive BigInteger that is probably prime, with
the specified bitLength. The probability that a BigInteger
returned by this method is composite does not exceed 2-100.
Parameters:
bitLength - bitLength of the returned BigInteger.
rnd - source of random bits used to select
candidates to be tested for primality.
Returns:
a BigInteger of bitLength bits that is probably
prime
Output - Java Shell - JDK 9


Project Properties - AnagramGame
Categories:
Sources
Libraries
Build
Compiling
Packaging
Deployment
Documenting
Run
Application
Web Start
License Headers
Formatting
Hints
Configuration:
Run with JShell
New...
Delete
Runtime Platform:
Project Platform
Manage Platforms...
Main Class:
com.toy.anagrams.ui.Anagrams
Browse...
Arguments:
Working Directory:
Browse...
VM Options:
Customize...
(e.g. -Xms10m)
Run with Java Web Start
(To run and debug the application with Java Web Start, first enable Java Web Start)
Enable JShell access
Class loader:
System
Class:
Browse...
Member:
Execute in Swing Event Thread
Help
Cancel
OK

DEMO

OHNE NETBEANS

- Bei einfachen Anwendungen:
  - Statische Entität als Einstieg
  - Main-Methode aufrufen

- Alternativ:
  - https://github.com/cinquin/attaching_jshell

JAVA 9 BOOTCAMP

25.April, München, www.eppleton.de

JAVA 9 MIGRATION

26. April, München, www.eppleton.de

VIELEN DANK