

JAVA

FUNDAMENTALS

CLASSLOADERS

Toomas Rõmer
@toomasr

March 27, 2017

AGENDA

- **Classes and Objects**
- **Classloaders**
- **ClassLoader Delegation**
- **Object Initialization**
- **Homework!**

WANT TO RUN EXAMPLES?

- <https://github.com/JavaFundamentalsZT>
- Repo: java-fundamentals
- Sub project: classloaders

CLASSES AND OBJECTS

```
1 class Foo {
2     private String firstName;
3     private String lastName;
4
5     public String getFullName() {
6         return firstName + " " + lastName;
7     }
8 }
```

CLASSES AND OBJECTS

```
1 public static void main(String[] args) {  
2     Foo foo = new Foo();  
3 }
```

CLASSES AND OBJECTS

Where does the instance come from?

CLASSES AND OBJECTS

- From the filesystem
- From some sort of memory
- From the interwebs
- Bytes generated on the fly
- ...


```
0000000 ca fe ba be 00 00 00 31 02 57 07 00 02 01 00 1f
0000010 6f 72 67 2f 7a 65 72 6f 74 75 72 6e 61 72 6f 75
0000020 6e 64 2f 7a 69 70 2f 5a 69 70 73 54 65 73 74 07
0000030 00 04 01 00 18 6a 75 6e 69 74 2f 66 72 61 6d 65
0000040 77 6f 72 6b 2f 54 65 73 74 43 61 73 65 01 00 06
0000050 3c 69 6e 69 74 3e 01 00 03 28 29 56 01 00 04 43
0000060 6f 64 65 0a 00 03 00 09 0c 00 05 00 06 01 00 0f
0000070 4c 69 6e 65 4e 75 6d 62 65 72 54 61 62 6c 65 01
0000080 00 12 4c 6f 63 61 6c 56 61 72 69 61 62 6c 65 54
0000090 61 62 6c 65 01 00 04 74 68 69 73 01 00 21 4c 6f
00000a0 72 67 2f 7a 65 72 6f 74 75 72 6e 61 72 6f 75 6e
00000b0 64 2f 7a 69 70 2f 5a 69 70 73 54 65 73 74 3b 01
00000c0 00 17 74 65 73 74 44 75 70 6c 69 63 61 74 65 45
00000d0 6e 74 72 79 41 74 41 64 64 01 00 0a 45 78 63 65
00000e0 70 74 69 6f 6e 73 07 00 11 01 00 13 6a 61 76 61
00000f0 2f 69 6f 2f 49 4f 45 78 63 65 70 74 69 6f 6e 07
0000100 00 13 01 00 0c 6a 61 76 61 2f 69 6f 2f 46 69 6c
0000110 65 08 00 15 01 00 20 73 72 63 2f 74 65 73 74 2f
0000120 72 65 73 6f 75 72 63 65 73 2f 64 75 70 6c 69 63
0000130 61 74 65 2e 7a 69 70 0a 00 12 00 17 0c 00 05 00
```

LETS HAVE A LOOK!

CLASSLOADERS

CLASSLOADER

- Classloader is an Object
- `java.lang.ClassLoader`
- Every class is loaded by a Classloader
- You can subclass a Classloader!

CLASSLOADER

- So every class is loaded by a Classloader
- The Classloader is a class
- Oh, wait ...
- ... who loads the Classloader then?

CLASSLOADER

- Every class has a classloader
- Even the classloader itself
- Classloaders have a parent classloader
- Can be null

CLASSLOADER

- System Classloader
- Bootstrap Classloader
- Extension Classloader

CLASSLOADER

```
public class ClassLoader {  
    protected Class loadClass(String name){  
        - is class loaded already? Return that!  
        - Parent classloader, please load it!  
        - Okay, I'll do it myself!  
        findClass(name);  
        defineClass(name);  
    }  
}
```


CLASSLOADER

```
1 public class ClassLoader {
2     protected Class findClass(String name) {
3         File clazzFile = findTheFile(name);
4         byte[] bytes = readBytes(clazzFile);
5         return defineClass(name, bytes);
6     }
7 }
```

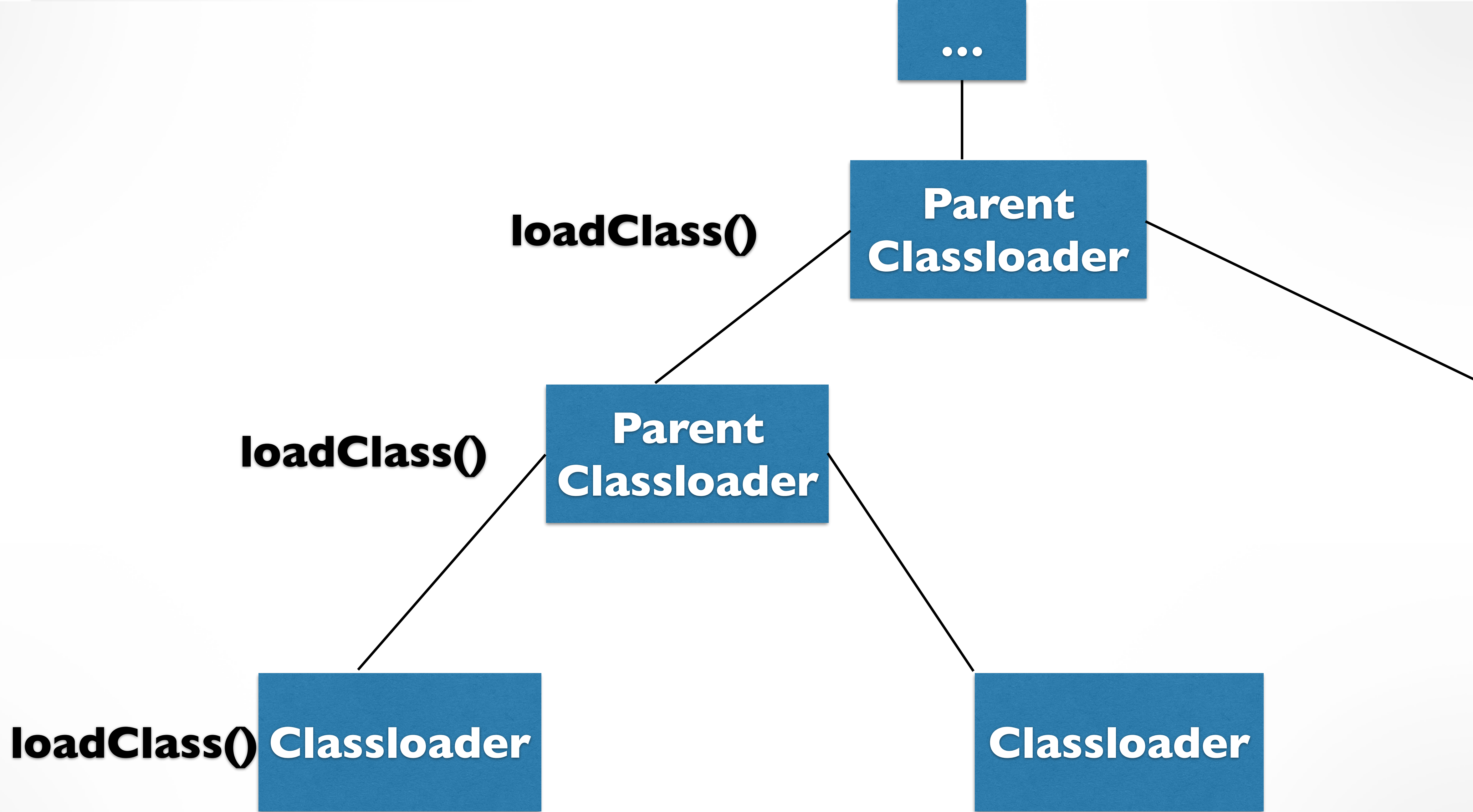
LETS HAVE A LOOK!

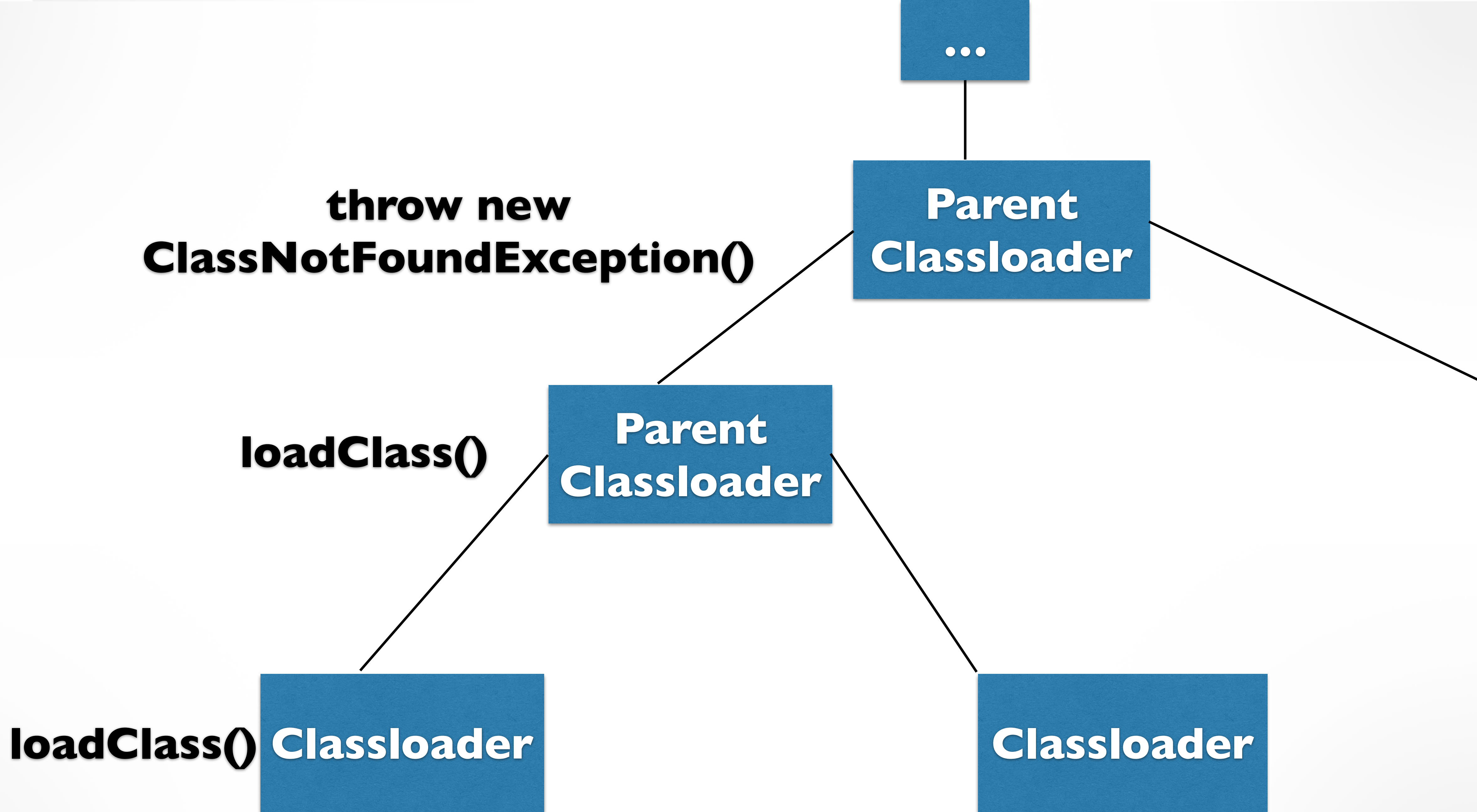
WHAT ABOUT RESOURCES?

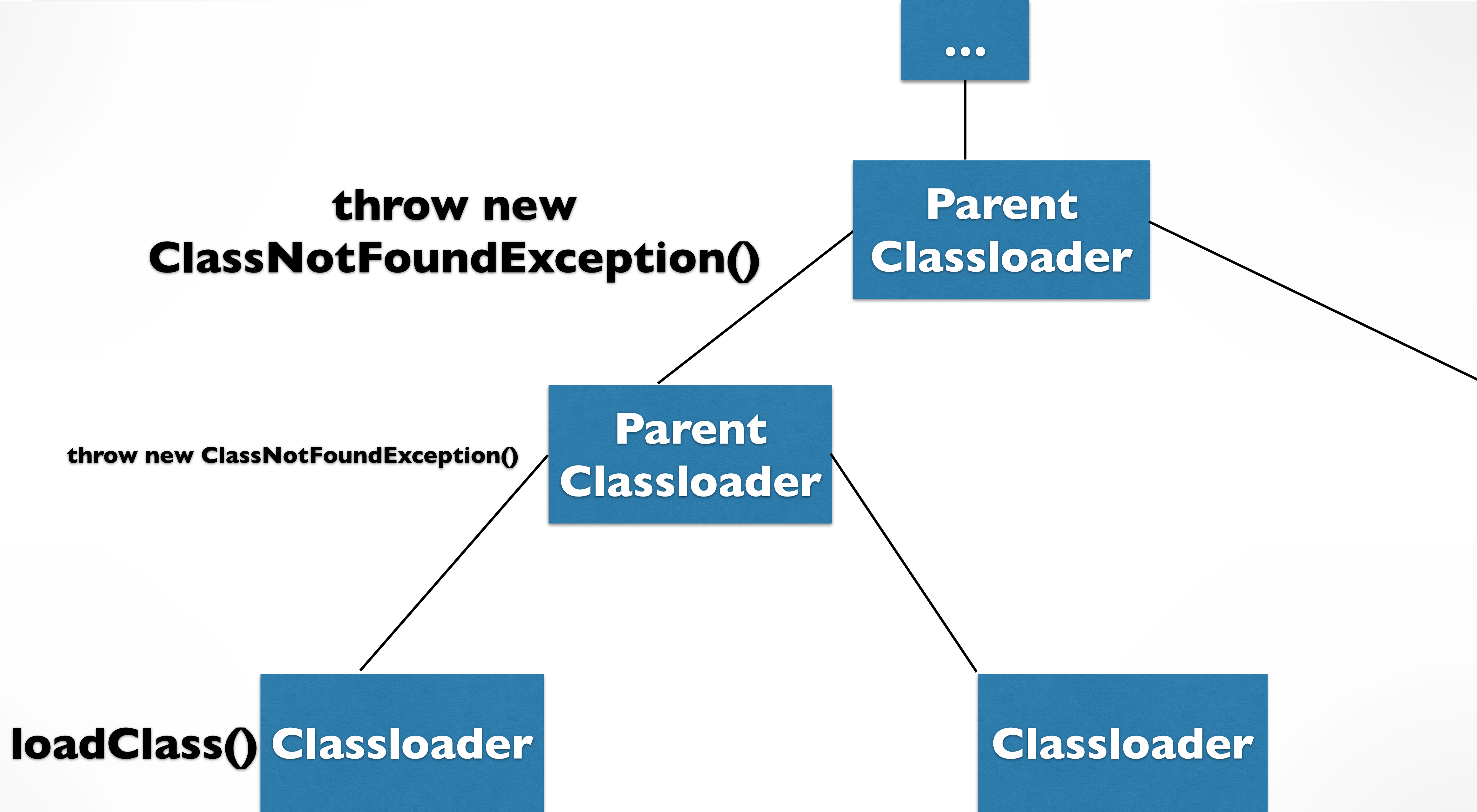
- `public URL getResource(String name);`
- `public Enumeration getResources(String name);`

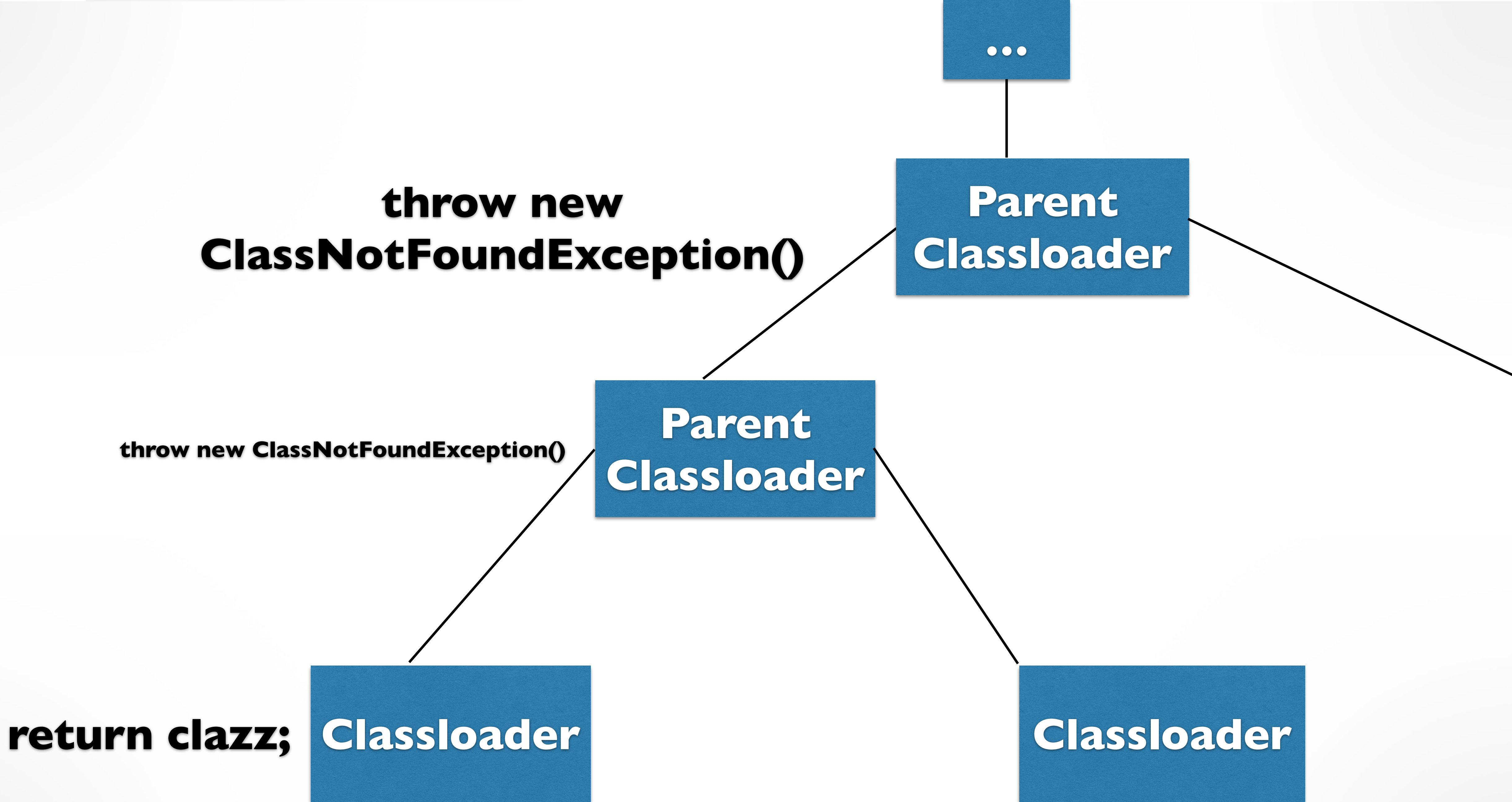
CLASSLOADER DELEGATION MODELS

Classical Model









DELEGATION MODELS

- Don't delegate to parent
- Delegate to parent
- Based on package name
- Your random approach goes here

LOOKING AT CLASSLOADERS

- `java.lang.ClassLoader`
- `java.net.URLClassLoader`
- Tomcat Webapp ClassLoader
- JBoss classloaders

OBJECT INITIALIZATION

- Instance variables
- Local variables
- Object initialization

DEFAULT VALUES

Type	Default value	Size
boolean	FALSE	undefined
byte	(byte) 0	1 byte
short	(short) 0	2 byte
int	0	4 byte
long	0L	8 byte
char	\u0000	2 byte
float	0.0f	4 byte
double	0.0d	8 byte
object reference	null	

CONSTRUCTORS

- They are not methods
- `this/super`
- default constructor
- initializers (final, forward references)
- Naming variables

INITIALIZATION AND INHERITANCE

- Instance variables
 - from Object to MyObject
- Instance initializers
 - from MyObject to Object

HOMEWORK!

<https://github.com/JavaFundamentalsZT/jf-hw-classloaders>