

# Language & Library Improvements

---



**Sander Mak**

FELLOW & SOFTWARE ARCHITECT

@Sander\_Mak

# HttpClient



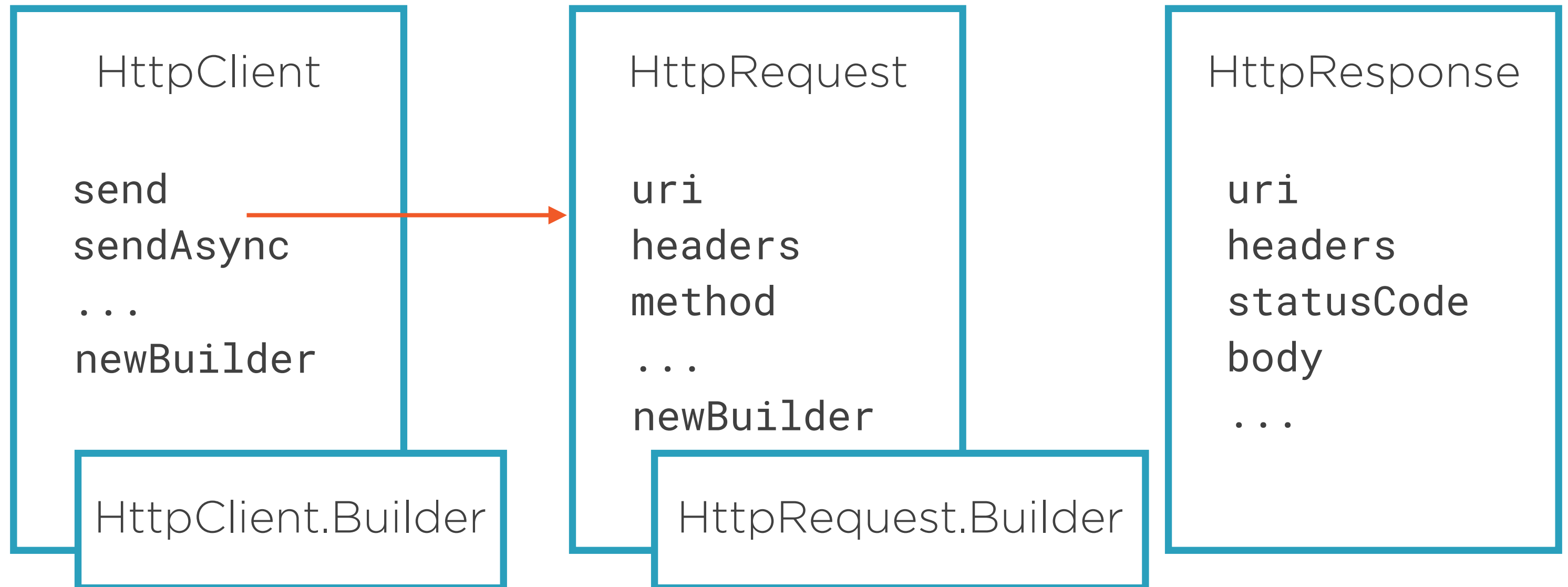
Replacement for `java.net.HttpURLConnection`

Supports HTTP/2, WebSocket

Reactive Streams integration

**Goal:** easy to use in common cases,  
powerful enough for complex ones

# HttpClient: Important Types



Demo

Using HttpClient

IntelliJ Community Edition 2018.2



[jetbrains.com/idea/download/](https://jetbrains.com/idea/download/)

# Library Additions

String

repeat

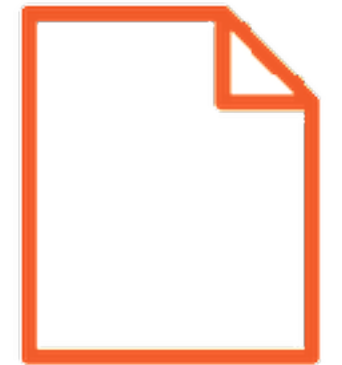
isBlank

strip

lines

# Library Additions

## Files



```
String readString(Path path)
```

```
String readString(Path path, Charset cs)
```

```
Path writeString(Path path, CharSequence cs,  
                OpenOption options)
```

```
Path writeString(Path path, CharSequence cs,  
                Charset cs, OpenOption options)
```

# Library Additions

Optional::isEmpty

```
var opt =  
    Optional.ofNullable(null);  
opt.isEmpty() == true
```

Predicate::not

```
Stream<String> strings = ...;  
strings  
    .filter(Predicate.not(String::isBlank))  
    .forEach(System.out::println);
```

# Unicode 10

Upgrade from Unicode 8 in Java 10

Adds 16,000+ new characters

Ten new scripts



`\u20BF`



# Java 10: Local-variable Type Inference

```
public void aMethod() {  
    var name = "Sander";  
}
```



# Local-variable Syntax for Lambda Parameters

```
(String a, String b) -> a.concat(b)
```

```
(a, b) -> a.concat(b)
```

```
(var a, var b) -> a.concat(b)
```

```
(@NonNull var a,  
 @Nullable var b) -> a.concat(b)
```

# Limitations

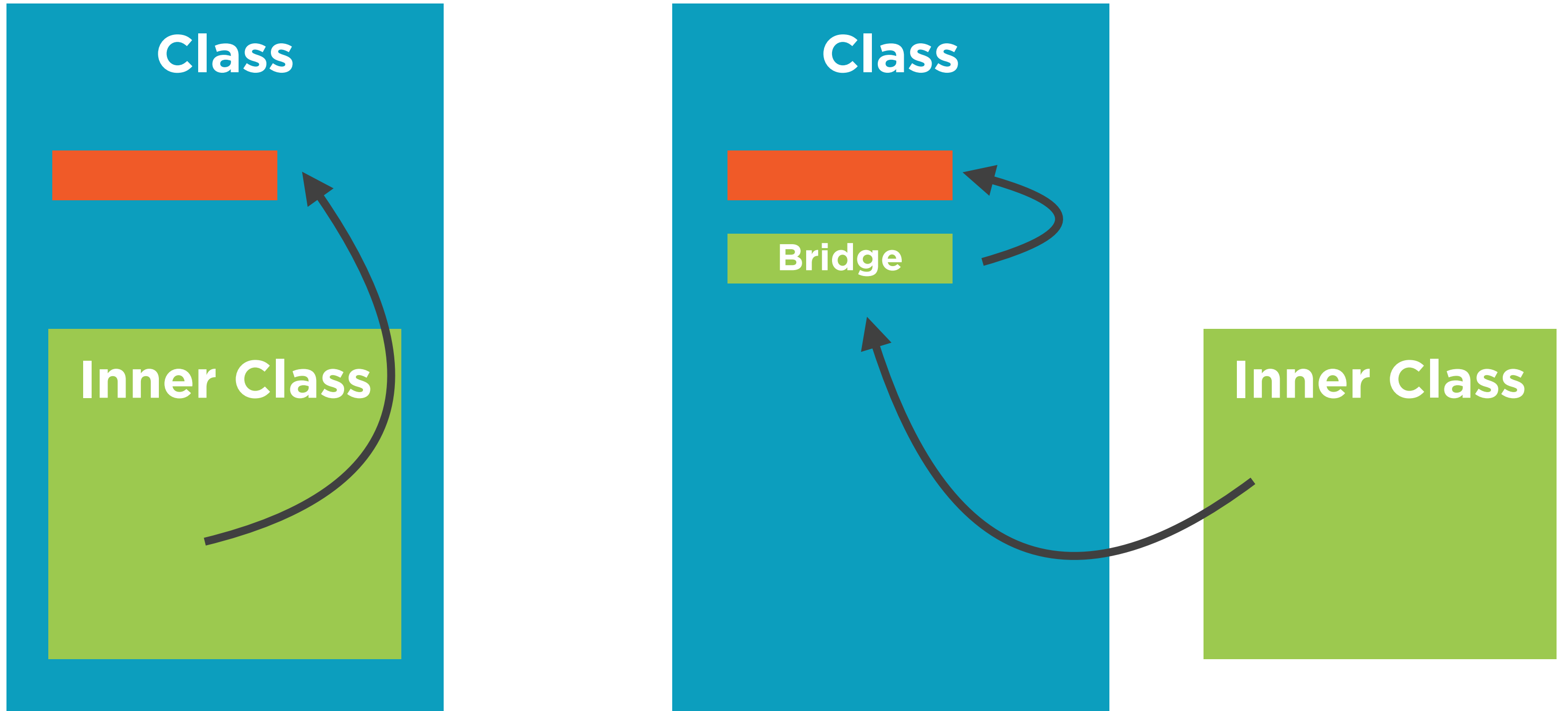


```
(var a, b) -> a.concat(b)
```

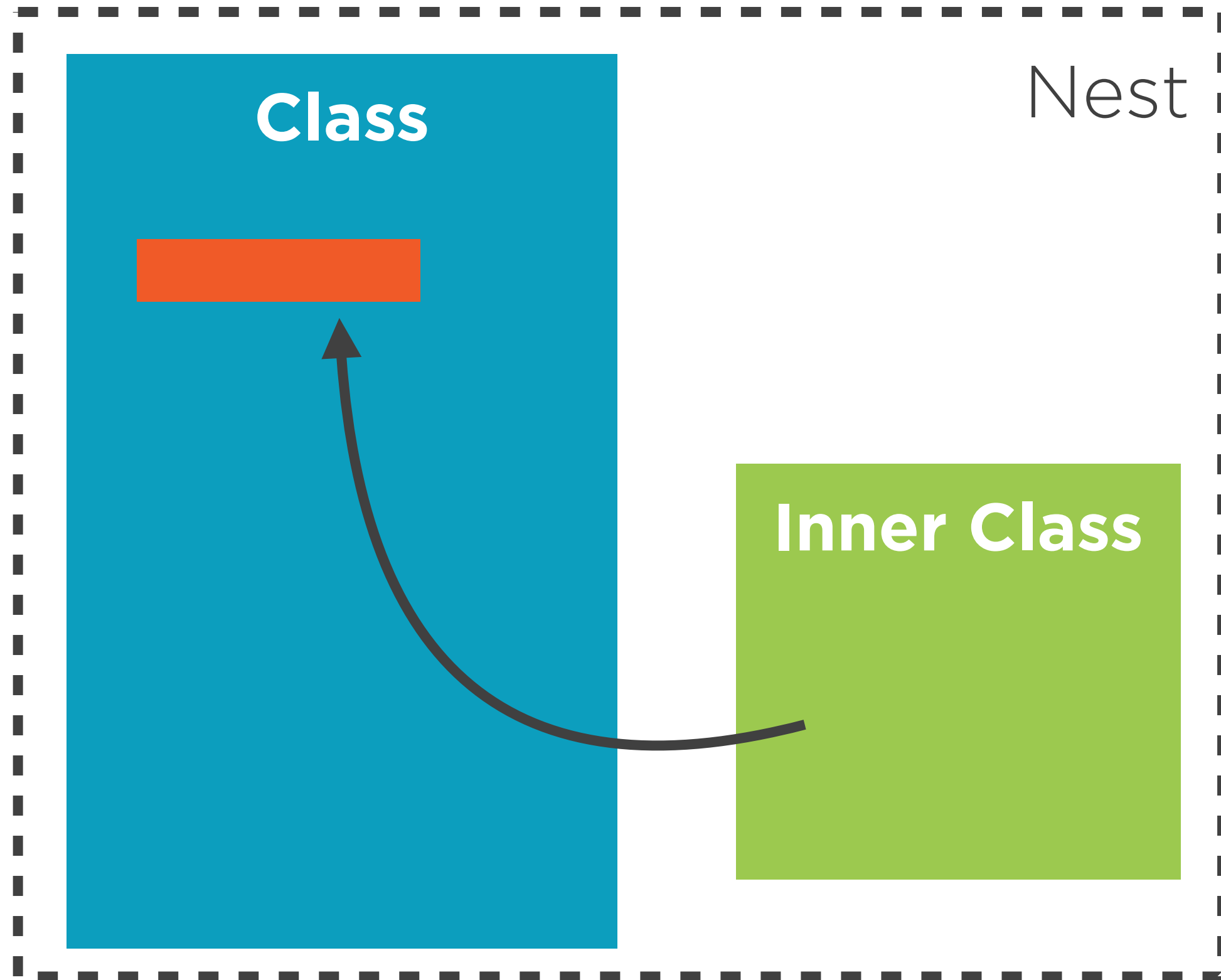
```
(var a, String b) -> a.concat(b)
```

```
var a -> a.toUpperCase()
```

# Nest-based Access-control



# Nest-based Access-control



# One More JVM Change



Dynamic Class-file Constants

**JEP 309:** [openjdk.java.net/jeps/309](https://openjdk.java.net/jeps/309)

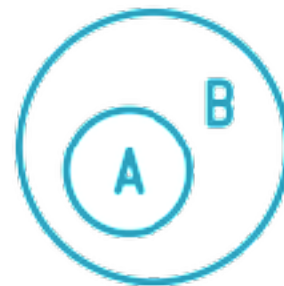
# Summary



New HttpClient API



var in lambdas



Nest-based Access-control