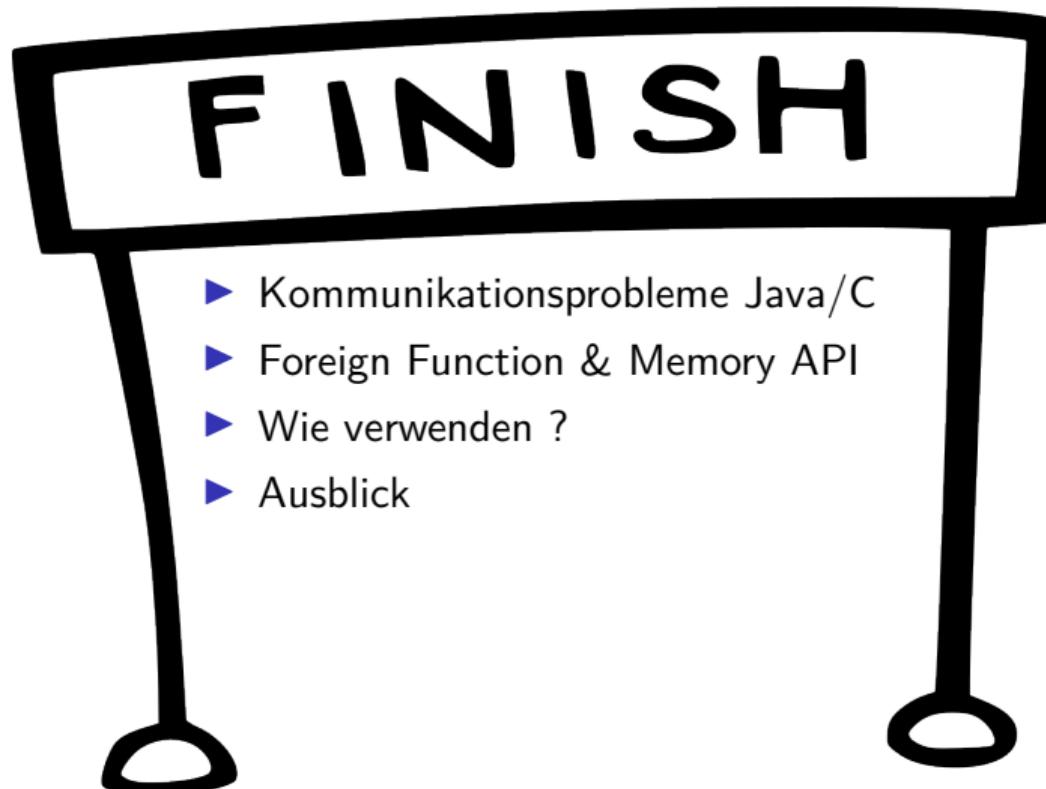


# Javas neue Gesprächskultur – ganz wie in Panama –

Bernd Müller  
Ostfalia





## Vorstellung Referent

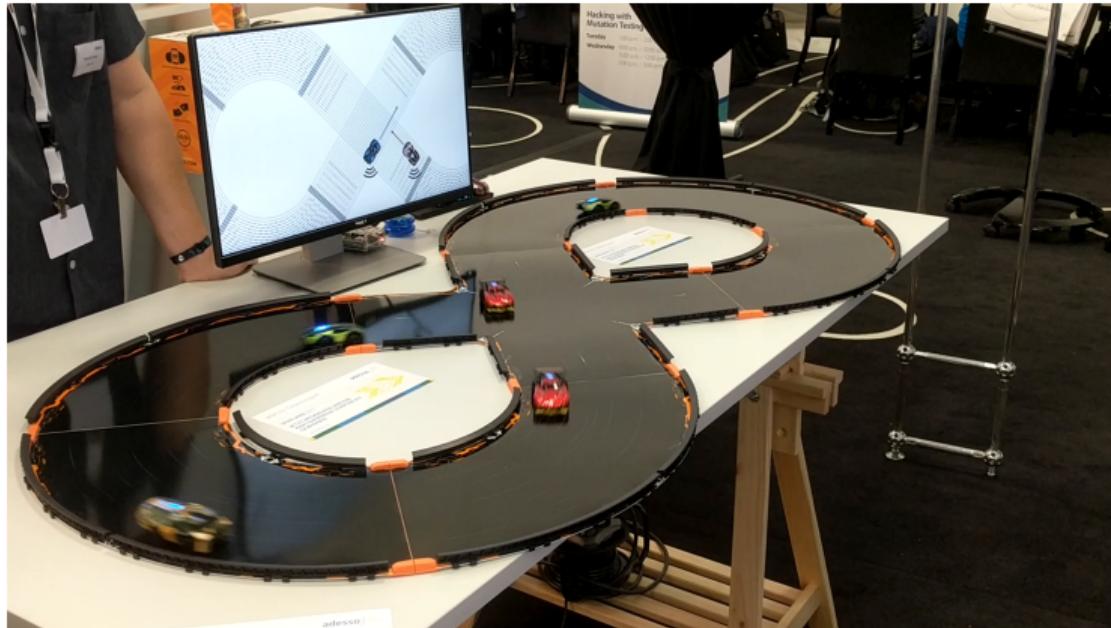
- ▶ Prof. Informatik (Ostfalia, HS Braunschweig/Wolfenbüttel)
- ▶ Buchautor (JSF, JPA, Seam, ...)



- ▶ Mitglied EGs JSR 344 (JSF 2.2) und JSR 338 (JPA 2.1)
- ▶ Geschäftsführer PMST GmbH
- ▶ JUG Ostfalen (Mitorganisator)
- ▶ Beirat der Java aktuell
- ▶ bernd.mueller@ostfalia.de
- ▶ @berndmuller
- ▶ BerndMuller

## Motivation

## JavaLand 2017, Adesso-Stand



## GitHub: adessoAG/anki-drive-java

A Java port of the Anki Drive SDK

java vehicle java-library anki  
anki-drive drive-sdk anki-vehicles

Readme  
MIT License  
16 stars  
7 watching  
27 forks

Releases 1

First official release Latest  
on 4 Apr 2017

File	Description	Date
gradle/wrapper		5 years ago
src/main		5 years ago
.gitignore		5 years ago
build.gradle	Support automated builds via Travis CI (#19)	5 years ago
gradlew	Fix link in README (#16)	5 years ago
gradlew.bat	Initial commit	5 years ago
taskpane.json		5 years ago

## JSR 82: Review Ballot 2000, FR3 2010

The screenshot shows the Java Community Process website with a specific focus on the JSR-000082 page. The page title is "JSR-000082 Java(TM) ME for Bluetooth 1.1.1 Final Release 3". Below the title, there is a message: "Thank you for accepting the software license agreement; you may now download the software." A large red oval is drawn across the top half of the page, covering the title and the download instructions. The download section includes a table with one item:

File Description and Name	Size
JSR-000082_Bluetooth 1.1.1 Final Release 3 bluetooth-1.1.1-mrel2-javadoc.zip	480.28 KB

At the bottom of the page, there is a note: "If you need assistance with downloads, please contact Customer Service. For all other JCP related questions, please see our Frequently Asked Questions (FAQ)."

## JAnki

The screenshot shows a GitHub repository page for the user BerndMuller named JAnki. The repository is public and contains 1 branch and 0 tags. The master branch has 33 commits by BerndMuller, mostly minor cleanups. The repository is described as a simple library to use Anki Overdrive with Java. It includes links to Readme, MIT License, and statistics like 4 stars and 3 watching.

BerndMuller / JAnki Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code About

BerndMuller Minor cleanups d440d00 on 13 Jan 2021 33 commits

lib	Minor cleanups	13 months ago
src	Minor cleanups	13 months ago
.gitignore	notifications refactored to package notifications, added C...	4 years ago
LICENSE.txt	license added	4 years ago
README.md	license added	4 years ago
backlog.txt	Minor cleanups	13 months ago
pom.xml	Minor cleanups	13 months ago
run-cli.sh	tracking added	4 years ago
run-tracking.sh	tracking added	4 years ago

About A simple library to use Anki Overdrive with Java

Readme MIT License 4 stars 3 watching 0 forks

Releases No releases published Create a new release

Packages

## TinyB von Intel

The screenshot shows the GitHub repository page for `intel-iot-devkit / tinyb`. The repository is public and has 33 watchers, 105 forks, and 229 stars. The main navigation tabs include Code, Issues (82), Pull requests (4), Actions, Projects, Wiki, Security, and three more options. Below the tabs, there's a dropdown for the branch (master) and buttons for Go to file, Add file, and Code. A red oval highlights the **About** section, which contains a summary of the project: "TinyB exposes the BLE GATT API for C++, Java and other languages, using BlueZ over DBus." To the right of the About section is a sidebar with links to Readme, MIT License, stars (229), watching (33), forks (105), and a Release section. Another red oval highlights the **Releases** section, which shows one release: v0.5.1 (Latest), released on 19 Jan 2018.

Code

Issues 82

Pull requests 4

Actions

Projects

Wiki

Security

...

master

Go to file

Add file

Code

About

vkołotov and petreetime Implementing discovery filter... on 14 Oct 2017 131

api Implementing a generic method to set discover fi... 4 years ago

cmake Add additional version information, including in j... 6 years ago

examples examples: list\_mfg also lists advertised service ... 5 years ago

include Adding support for setting RSSI discovery filter 4 years ago

java Implementing discovery filter by UUIDs (java) an... 4 years ago

src Implementing discovery filter by UUIDs (java) an... 4 years ago

.gitignore .gitignore: Now all build\* directories are excluded 6 years ago

.travis.yml Add checkinit which tests if library loads ok and ... 6 years ago

TinyB exposes the BLE GATT API for C++, Java and other languages, using BlueZ over DBus.

Readme

MIT License

229 stars

33 watching

105 forks

Releases 1

v0.5.1 Latest

on 19 Jan 2018

Selbst ist der Mann  
und das Ergebnis ist immer ein Erlebnis ;-)



Ziel: Keine Bibliothek, nur JDK + BlueZ

## Verallgemeinerung

Keine Bibliothek, nur JDK + <your choice>

## Das Problem

## Das Problem – und hoffentlich die Lösung

- ▶ Manchmal genügt JDBC, HTTP, NIO, UNIX-Domain-Sockets, ... nicht
- ▶ Sogenannter *Off-Heap-Speicher* (außerhalb JVM) soll direkt zugegriffen werden
- ▶ Machen z.B. Tensorflow, Ignite, Lucene, Netty und vielen anderen
- ▶ Es gibt: JNI, Byte-Buffer-API, sun.misc.Unsafe, ...
- ▶ Macht aber auf Dauer nicht glücklich ...
- ▶ JEPs 191, 370, 383, 389, 393, 412, 419, 424
- ▶ Mit Project Panama: Interconnecting JVM and native code

## JEP 419: Foreign Function & Memory API

**OpenJDK**

[OpenJDK FAQ](#)  
[Installing](#)  
[Contributing](#)  
...  
...

### JEP 419: Foreign Function & Memory API (Second Incubator)

Owner Maurizio Cimadamore

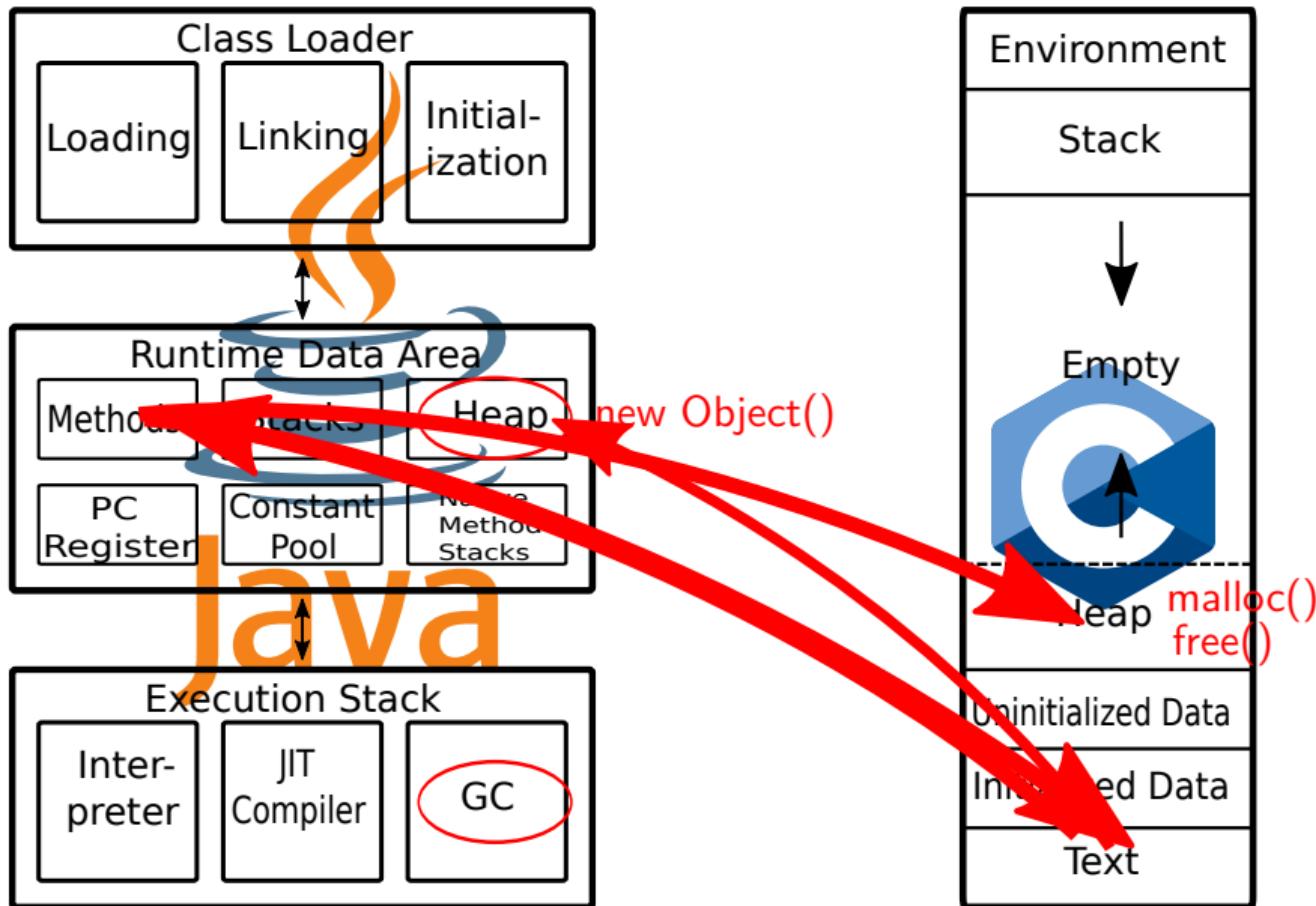
#### Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.

**Groups**  
(overview)  
[Adoption](#)  
[Build](#)  
[Client Libraries](#)  
[Compatibility & Specification Review](#)  
[Compiler](#)  
[Conformance](#)  
[Core Libraries](#)  
[Governance Board](#)

#### Summary

Introduce an API by which Java programs can interoperate with code and data outside of the Java runtime. By efficiently invoking foreign functions (i.e., code outside the JVM), and by safely accessing foreign memory (i.e., memory not managed by the JVM), the API enables Java programs to call native libraries and process native data without the brittleness and danger of JNI.



Native Method Interface + Libraries

## Lösungsversuch FFM: Konzepte und Klassen (Java 17)

- ▶ Allokation von Speicher  
MemorySegment, MemoryAddress, SegmentAllocator, ...
- ▶ Lesen und Schreiben von (strukturiertem) Speicher  
MemoryLayout, VarHandle (java.lang.invoke), ...
- ▶ Regeln des Lebenszyklus von Ressourcen  
ResourceScope
- ▶ Aufruf von Funktionen  
SymbolLookup, CLinker

## Beispiel Memory Segment

- ▶ Zusammenhängender Speicherbereich
- ▶ On- oder Off-Heap
- ▶ Nativ, mapped (mmap), Array oder Buffer
- ▶ Garantien für räumliche, zeitliche sowie Thread-bezogene Beschränkungen
- ▶ ...

## Beispiele

## Prozess-Id (C)

Prozess-Id des Prozesses: keine Parameter, Integer-Rückgabe (jps-Clone)

```
#include <unistd.h>

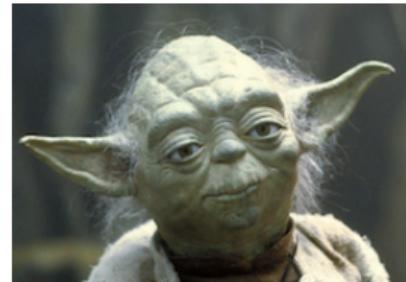
pid_t getpid(void);
```

## Prozess-Id (Java)

```
MethodHandle getpid = CLinker.getInstance().downcallHandle(
    CLinker.systemLookup().lookup("getpid").get(),
    MethodType.methodType(int.class),
    FunctionDescriptor.of(CLinker.C_INT));

System.out.println("Process Id: " + (int) getpid.invokeExact());
```

## Demo Time



Sleepy from slides, we are !

## Quicksort

```
#include <stdlib.h>

void qsort(void *base, size_t nmemb, size_t size,
           int (*compar)(const void *, const void *));
```

## Vergleichsmethode/-klasse (à la Comparator#compare())

```
static class Comparator {  
  
    static int compare(MemoryAddress addr1, MemoryAddress addr2) {  
        int v1 = MemoryAccess.getIntAtOffset(  
            MemorySegment.globalNativeSegment(),  
            addr1.toRawLongValue());  
        int v2 = MemoryAccess.getIntAtOffset(  
            MemorySegment.globalNativeSegment(),  
            addr2.toRawLongValue());  
        return v1 - v2;  
    }  
}
```

## Method-Handle auf qsort

```
MethodHandle qsort = CLinker.getInstance().downcallHandle(
    CLinker.systemLookup().lookup("qsort").get(),
    MethodType.methodType(void.class, MemoryAddress.class, long.class,
                          long.class, MemoryAddress.class),
    FunctionDescriptor.ofVoid(C_POINTER, C_LONG, C_LONG, C_POINTER)
);
```

## Method-Handle auf Vergleichsmethode

```
MethodHandle comparHandle = MethodHandles.lookup()  
  
    .findStatic(Comparator.class, "compare",  
  
    MethodType.methodType(int.class, MemoryAddress.class,  
                          MemoryAddress.class));
```

## Allokation des Arrays und eigentlicher Aufruf

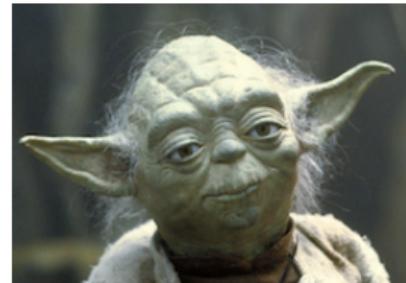
```
try (ResourceScope scope = ResourceScope.newConfinedScope()) {
    MemoryAddress comparFunc = CLinker.getInstance().upcallStub(
        comparHandle,
        FunctionDescriptor.of(C_INT, C_POINTER, C_POINTER), scope);

    MemorySegment array = SegmentAllocator.ofScope(scope)
        .allocateArray(C_INT, new int[] {0, 9, 3, 4, 6, 5, 1, 8, 2, 7});

    qsort.invokeExact(array.address(), 10L, 4L, comparFunc);

    int[] sorted = array.toIntArray();
}
```

## Demo Time



Sleepy from slides, we are !

## Werkzeugunterstützung

## JExtract

- ▶ Reduktion des Overheads durch
  - ▶ Analyse einer Include-Datei
  - ▶ und Generierung der entsprechenden Methoden und Typen für den Aufruf
- ▶ Nicht im JDK, evtl. später separat verfügbar

## getpid ohne jextract

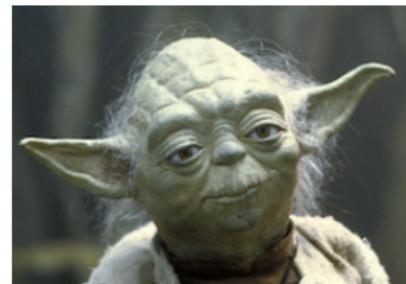
```
MethodHandle getpid = CLinker.getInstance().downcallHandle(
    CLinker.systemLookup().lookup("getpid").get(),
    MethodType.methodType(int.class),
    FunctionDescriptor.of(CLinker.C_INT));

System.out.println("Process Id: " + (int) getpid.invokeExact());
```

## getpid mit jextract

```
System.out.println("Process Id: " + getpid());
```

## Demo Time



Sleepy from slides, we are !

Es geht weiter . . . (Java 17 ⇒ 18 ⇒ 19)

## Quicksort mit Java 17 (Ausschnitt)

```
MethodHandle qsort = CLinker.getInstance().downcallHandle(
    CLinker.systemLookup().lookup("qsort").get(),
    MethodType.methodType(void.class, MemoryAddress.class,
        long.class, long.class, MemoryAddress.class),
    FunctionDescriptor.ofVoid(C_POINTER, C_LONG, C_LONG, C_POINTER)
);
```

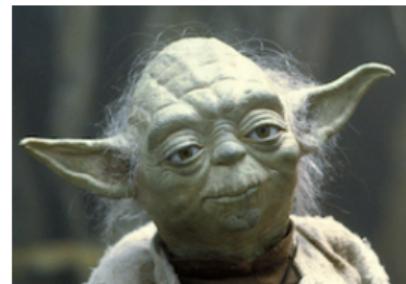
## Quicksort mit Java 18 (Ausschnitt)

```
MethodHandle qsort = CLinker.systemCLinker().downcallHandle(  
    CLinker.systemCLinker().lookup("qsort").get(),  
    FunctionDescriptor.ofVoid(ADDRESS, JAVA_LONG, JAVA_LONG, ADDRESS)  
) ;
```

## Quicksort mit Java 19 (Ausschnitt)

```
MethodHandle qsort = Linker.nativeLinker().downcallHandle(  
    Linker.nativeLinker().defaultLookup().lookup("qsort").get(),  
    FunctionDescriptor.ofVoid(ADDRESS, JAVA_LONG, JAVA_LONG, ADDRESS)  
) ;
```

## Demo Time



Sleepy from slides, we are !

## Fazit und Ausblick

### Persönliche – und damit subjektive – Einschätzung

- ▶ Eine wirklich gute Idee
- ▶ Versuche JAnki zu migrieren. Schwierigkeit: Komplexität des D-Bus-Interface und eigene Unzulänglichkeiten ;-)
- ▶ Wenn GraalVMs Native-Image-Erzeugung FFM unterstützt: 😊
- ▶ und was machen andere ?

## Tomcat SSL

Tomcat 9.0.55 (remm)

2021-11-10

Coyote

① Improve performance of Connector shutdown - primarily to reduce the time it takes to run the test suite. (markt)

🔧 Refactor the APR/native connector shutdown to reduce the possibility of a JVM crash during the connector shutdown. (markt)

🔧 [#457](#): Add a `toString()` method to `MimeHeader` to aid debugging. (dblevins)

🔧 Add experimental OpenSSL support through the Panama API incubating in Java 17, with support for OpenSSL 1.1. This no longer requires tomcat-native or APR. Please refer to the `openssl-java17` module from the `main` branch for more details. (remm)

🔧 Fix APR connector stop so it correctly waits for the sendfile thread, if any, to exit. (markt)

🔧 Do not ignore the error condition if the APR connector is not able to open a server socket as continuing in this case will trigger a JVM crash. (markt)

## Elasticsearch

### Some scattered feedback on the Foreign Linker API

Chris Hegarty [chegar999@gmail.com](mailto:chegar999@gmail.com)

Wed Dec 22 09:19:40 UTC 2021

- Previous message (by thread): [Calling the Port Audio C API via Panama FFI APIs](#)
- Next message (by thread): [Some scattered feedback on the Foreign Linker API](#)
- **Messages sorted by:** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#)

---

Hi,

As part of a recent prototyping effort, we evaluated replacing the usage of JNA in the core of the Elasticsearch server with the incubating Foreign Linker API.

TL;DR things just worked, perf improved and we really like the restricted native access.

Our usage (at least in this particular case) is not really performance sensitive, in that we reach into native to setup syscall filtering and check resource limits - but hey, faster is always better! We measured approximate perf improvements between 8 and 20 times faster for simple downcalls. The reason the numbers vary so much is more to do with JNA rather than Panama since the Panama times from our to our user

## JDK-Integration



Chris Vest  
@chvest

...

It's happening! \*jazz hands\*

[Tweet übersetzen](#)

openjdk/panama-foreign

#633 **8280527: Move  
jdk.incubator.foreign to  
java.lang.foreign**



2 comments 0 reviews 271 files +2488 -3462



FrauBoes · January 24, 2022 · 6 commits



github.com

8280527: Move jdk.incubator.foreign to java.lang.foreign by FrauBoes · Pull Req...

To prepare for preview, this change removes the jdk.incubator.foreign module and moves the public classes to java.base/java.lang.foreign, implementation classe...

8:16 nachm. · 24. Jan. 2022 · Twitter Web App

## Fragen und Anmerkungen



## Vortrag und Code

<https://github.com/BerndMuller/panama-jfs-2022>

