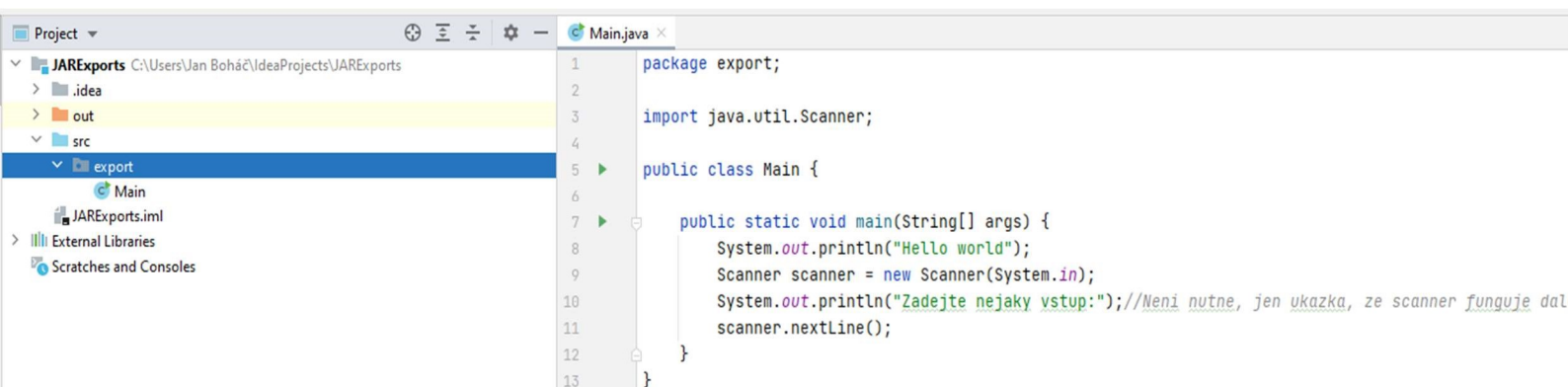


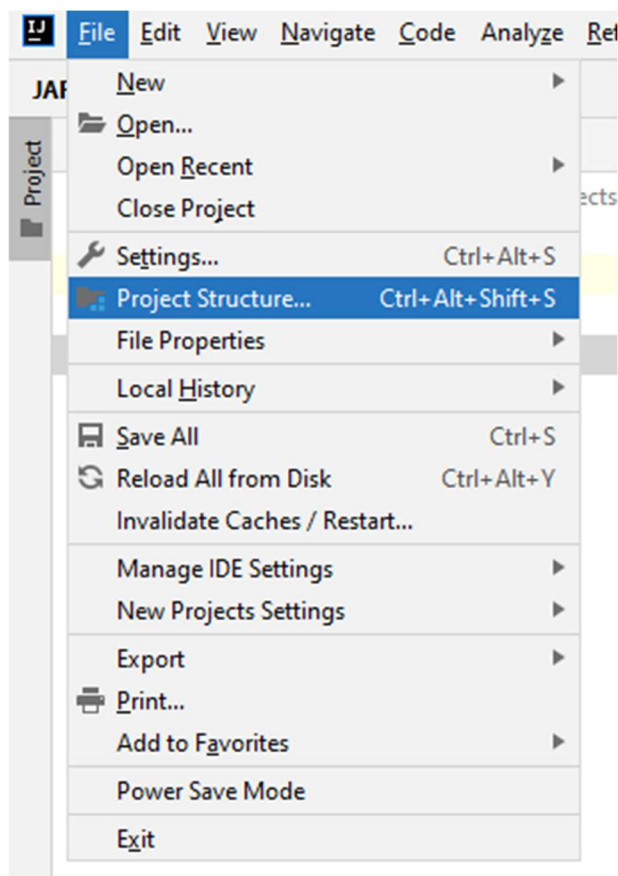
## Tvoření JAR v IntelliJ IDEA

Pro ukázkou vytvořím projekt s metodou main a následným kódem:



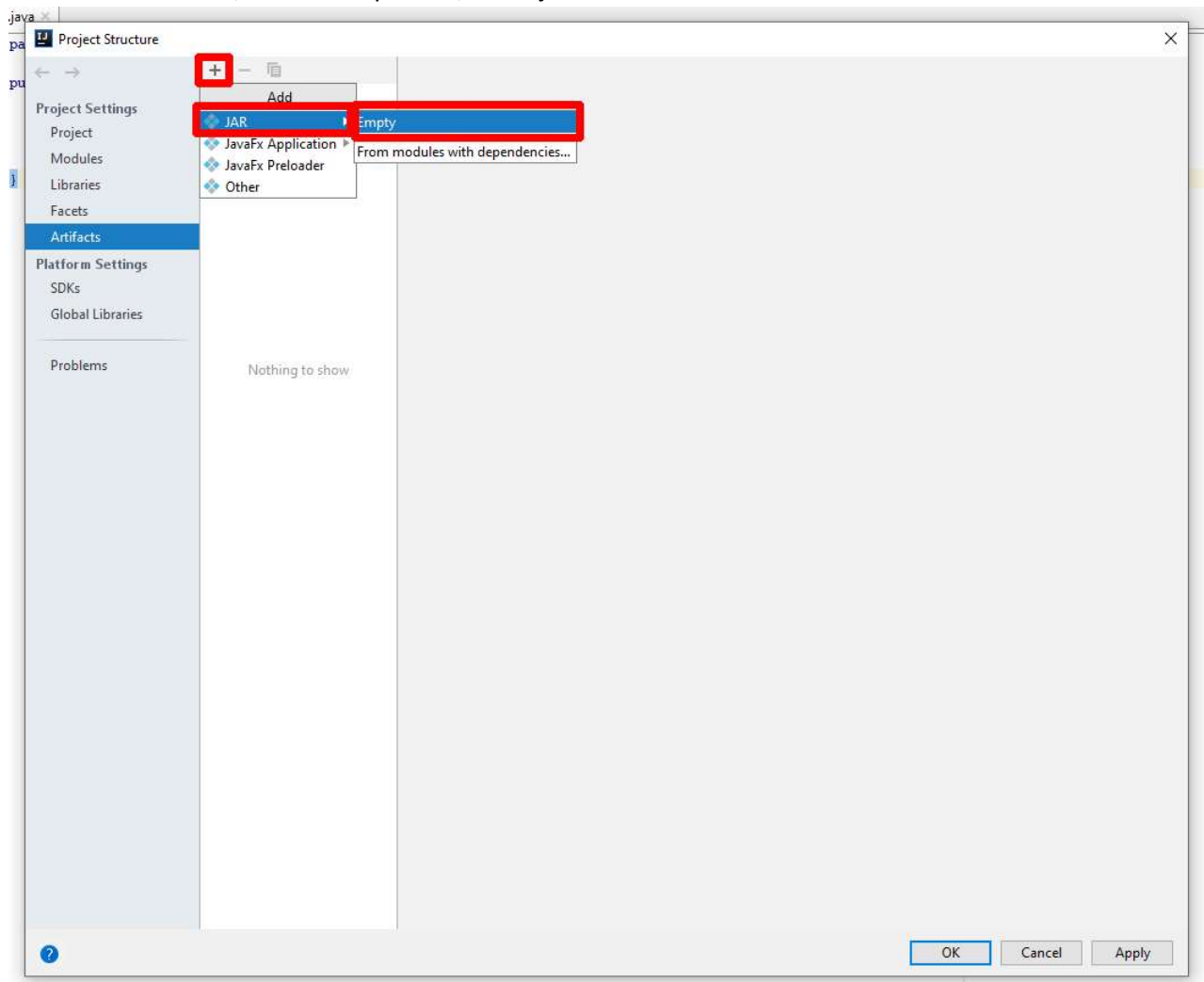
Kód v první řadě zkusím spustit v IntelliJ, abych zjistil, že v něm nejsou chyby.

Zkratkou `ctrl+alt+shift+s` nebo v položce File z horní lišty programu se dostanu do menu *Project Structure*.



V menu se zaměřím na postranní seznam, konkrétně mě zajímá *Artifacts* pod položkou *Project Settings*.

Vidím postranní lištu s orientací pod *Project structure* dál mě zajímá druhé (poprvé prázdné) menu. Konkrétně tlačítko +, které něco přidává, nás zajímá možnost JAR.



Takto by tedy mělo vypadat naše menu bez úprav, pouze s pojmenováním výstupního JARu. Mimo jiné lze i přepsat Output directory, kam se JAR vygeneruje, ideálně nechávat ve složce s projektem.

Name:  Type: JAR

Output directory:

☐ Include in project build

Output Layout   Pre-processing   Post-processing

jmenoProgrmu.jar

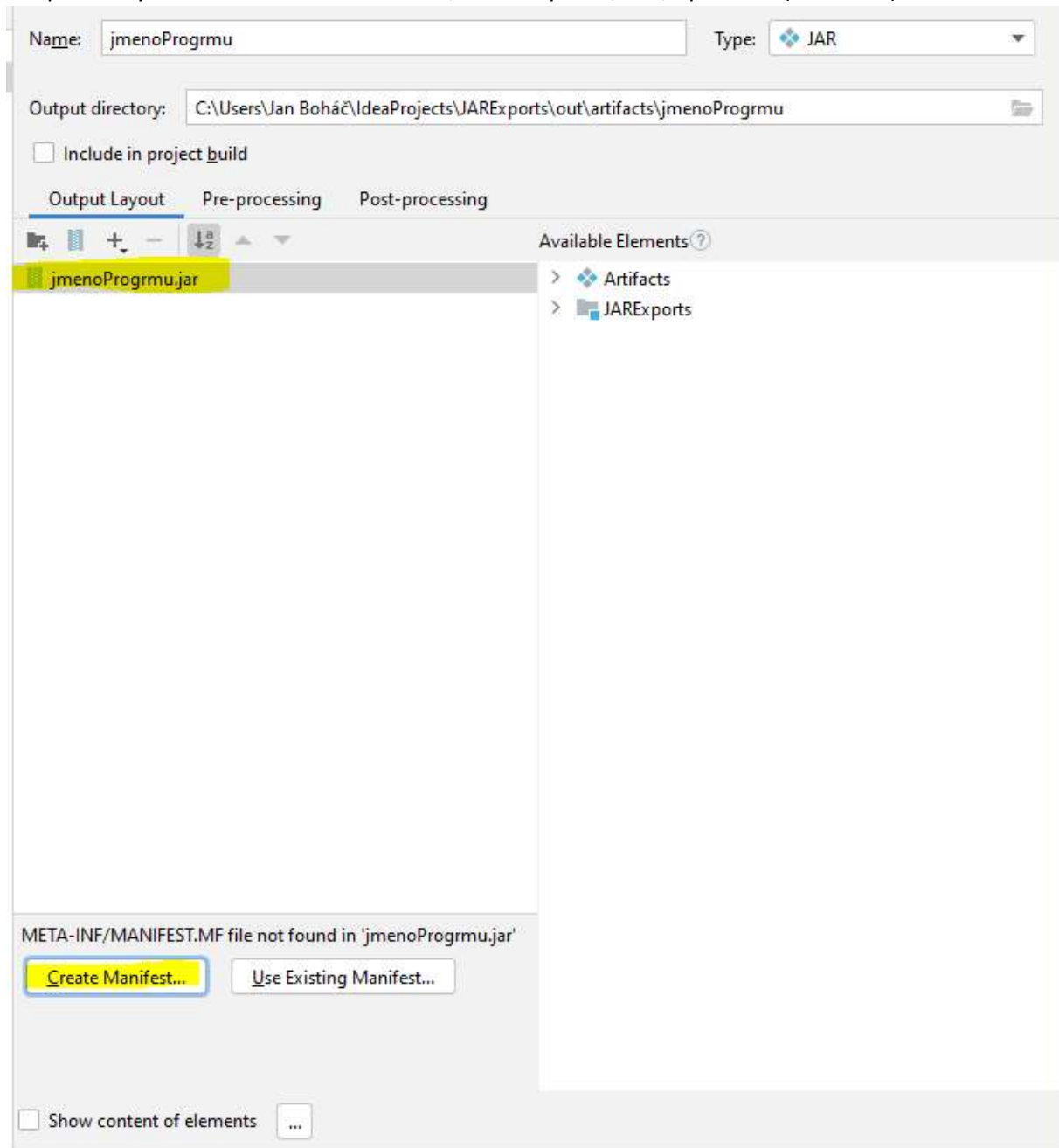
Available Elements ?

- > Artifacts
- > JARExports

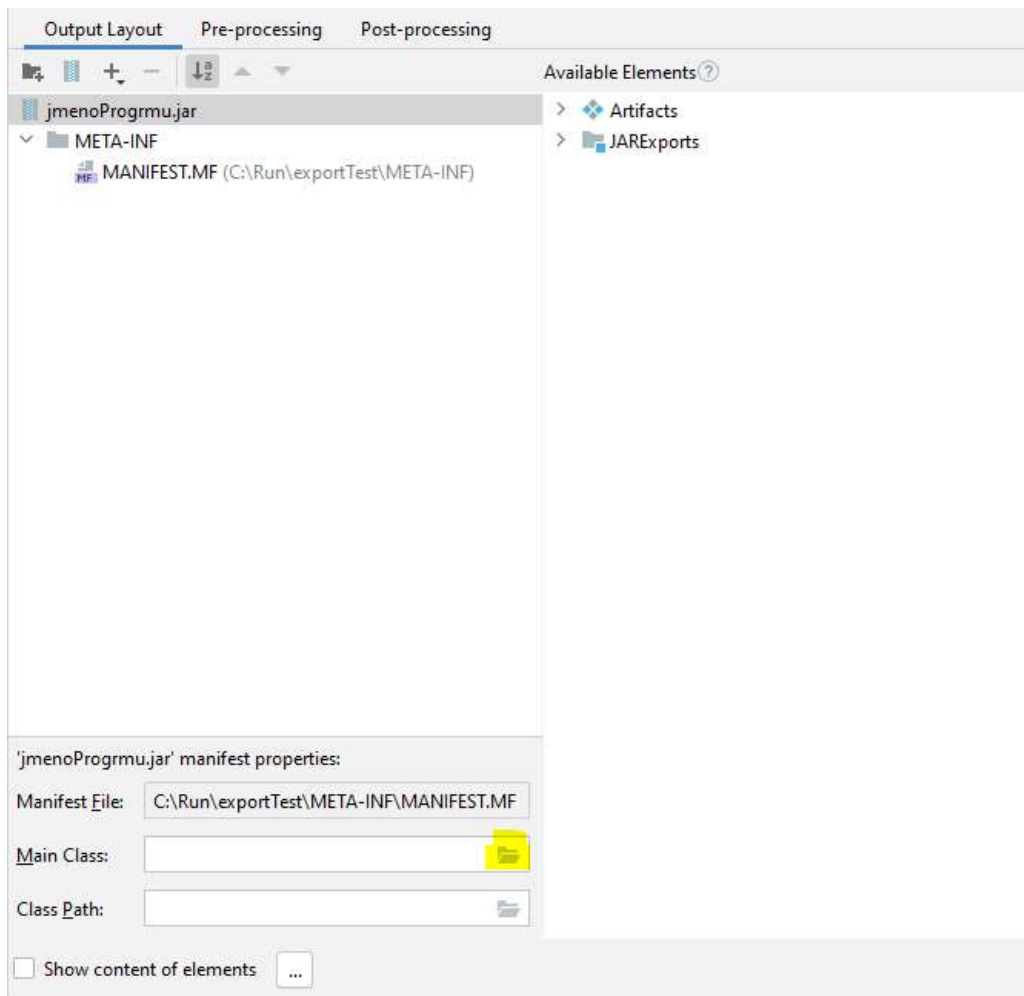
☐ Show content of elements ...

OK Cancel Apply

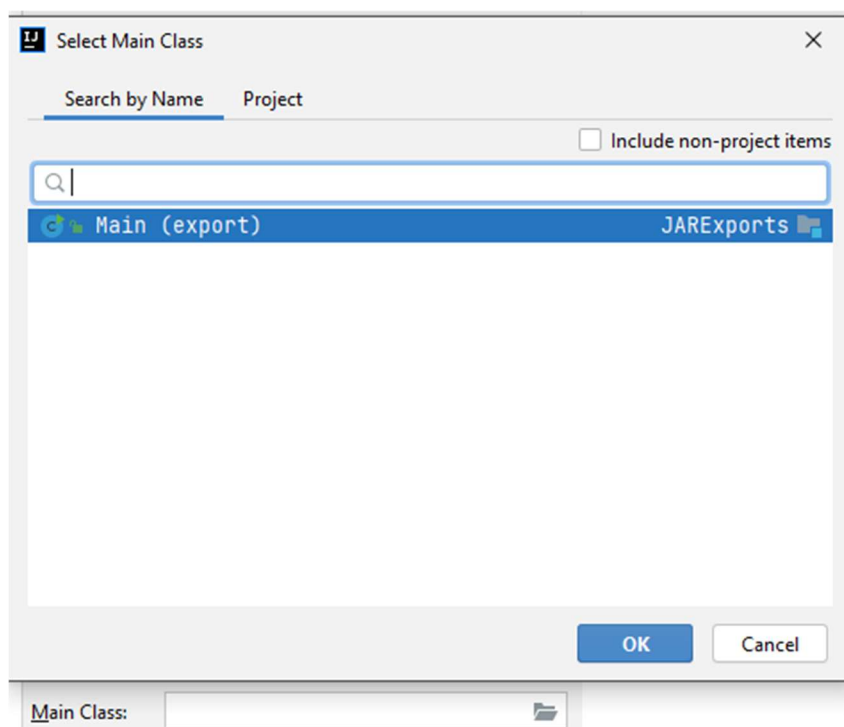
Jeden z nejpodstatnějších kroků je vytvoření Manifestu. Pro vytvoření si klikněte na zvýrazněný jar. poté se dole objeví menu ohoedně manifestu, vás zajímá Create Manifest...Až si budete vybírat, kam umístíte, tak klidně do složky s projektem, nebo si někde udělejte složku, kde budete mít všechny JAR soubory pohromadě... ale nezapomínejte, že to slovo Manifest se bude bít, takže byste museli mít pro každý JAR a manifest vlastní složku, takže např. C:\Run\exportTest (konkretni)



Po zvolení bude menu vypadat následovně: Vás bude zajímat definovat hlavní třídu, resp. třídu s metodou main, která se spustí s defaultním během programu. Pro přidání klikněte na zvýrazněnou část



Poté zvolíme metodu main v následujícím menu (může být více metod main, nás zajímá ta, která se bude spouštět, jakmile to jednou exportujeme, tak už to nezměníme, aniž bychom exportovali znovu)



Menu tedy vypadá takto:

Name:  Type: JAR

Output directory:

☐ Include in project build

Output Layout   Pre-processing   Post-processing

Available Elements ?

- jmenoProgrmu.jar
  - META-INF
    - MANIFEST.MF (C:\Run\exportTest\META-INF)

> Artifacts

> JARExports

'jmenoProgrmu.jar' manifest properties:

Manifest File:

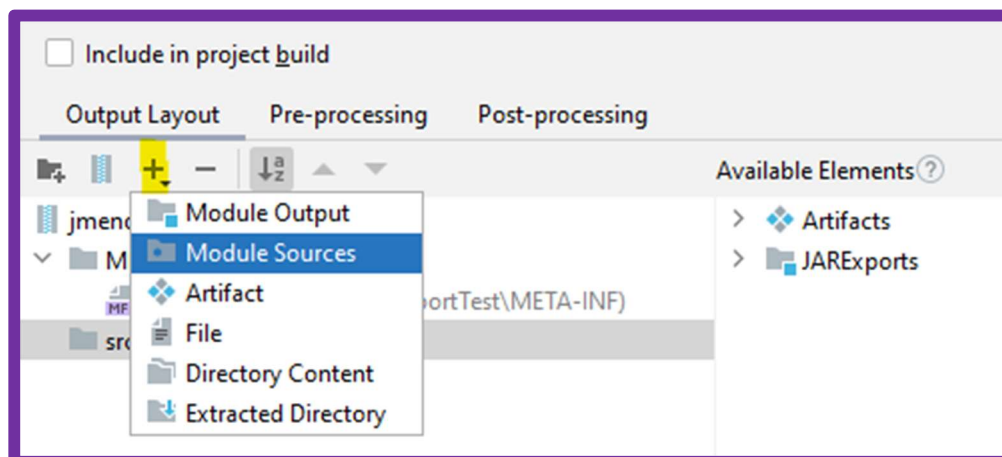
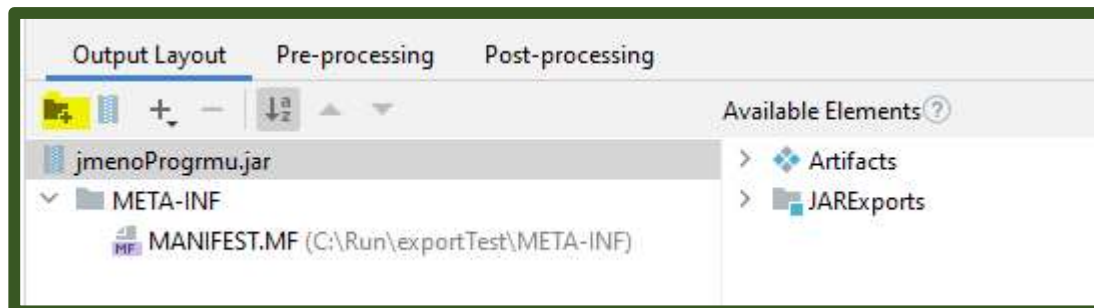
Main Class:

Class Path:

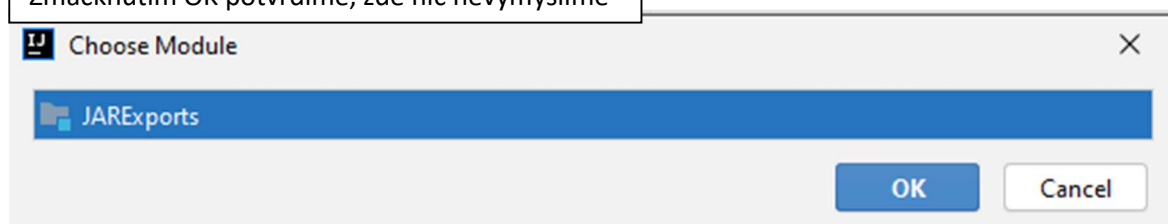
☐ Show content of elements ...

OK Cancel Apply

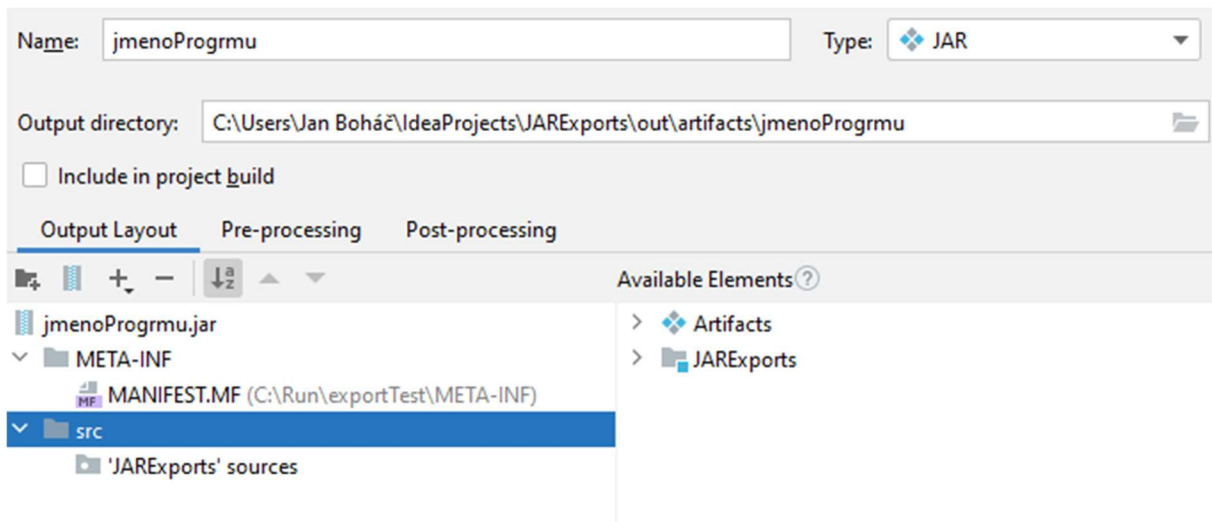
Další kroky jsou v rámci kontroly tak, abyste odeslali i zdrojové kódy. **Vytvoříme pomocí tlačítka nový adresář, pojmenujeme ho src, klikneme na něj a když ho máme označený, přirádíme do něj moduly, ve vašem případě to znamená, že tam dáváme zdrojáky.**



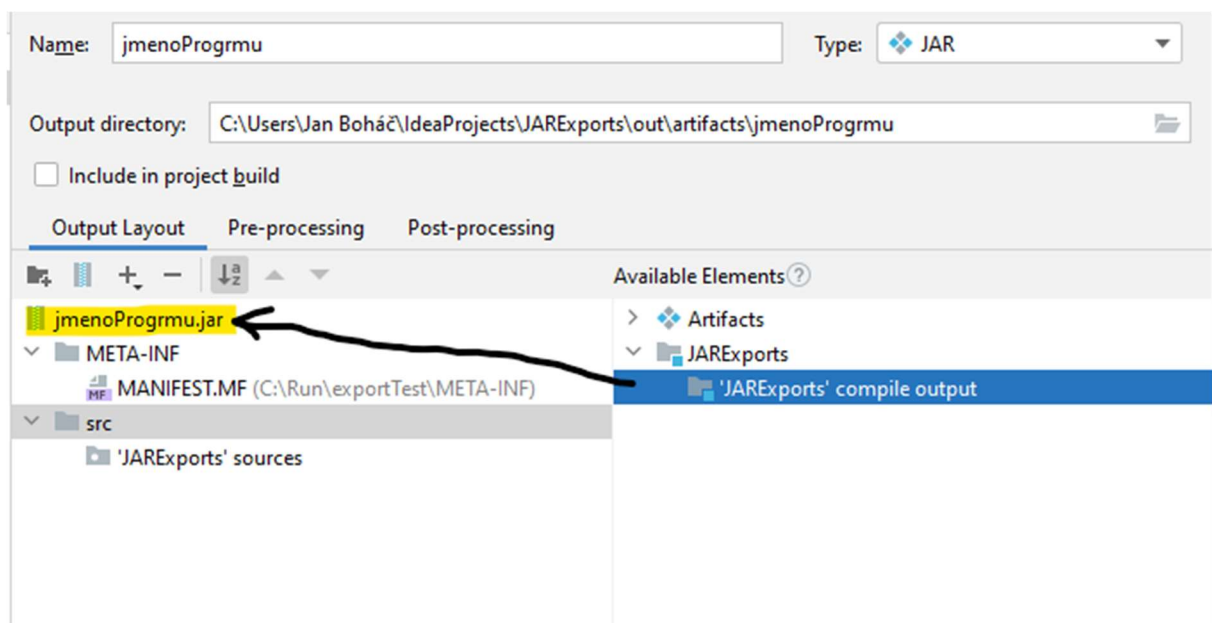
Zmáčknutím OK potvrdíme, zde nic nevymýšlíme



Menu by tedy mělo být v tomto stavu, resp. u vás by neměla být vpravo vidět možnost Artifacts, pokud je toto prvně, co jar nastavujete (pokud jste to dělali předtím, tak tam pravděpodobně bude, pokud ano, ignorujeme ji)

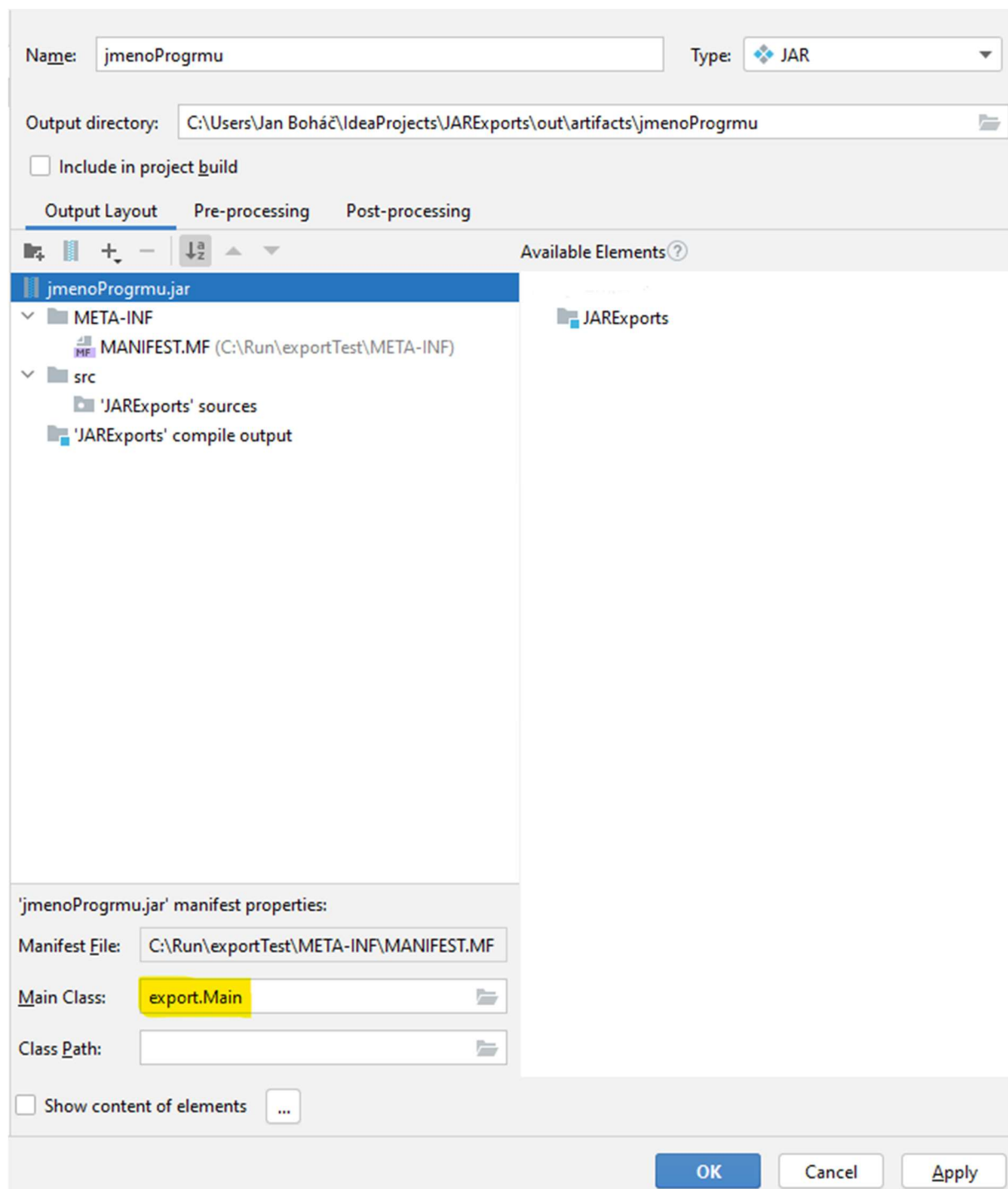


Zbývá poslední úprava. V pravé část *Available Elements* je složka JARExports, kterou si otevřeme, resp. rozbalíme, abychom viděli vnitřek, konkrétně *'JARExports' compile output*. Tuto položku myší podržíme a přetáhneme na levou část přímo na .jar.

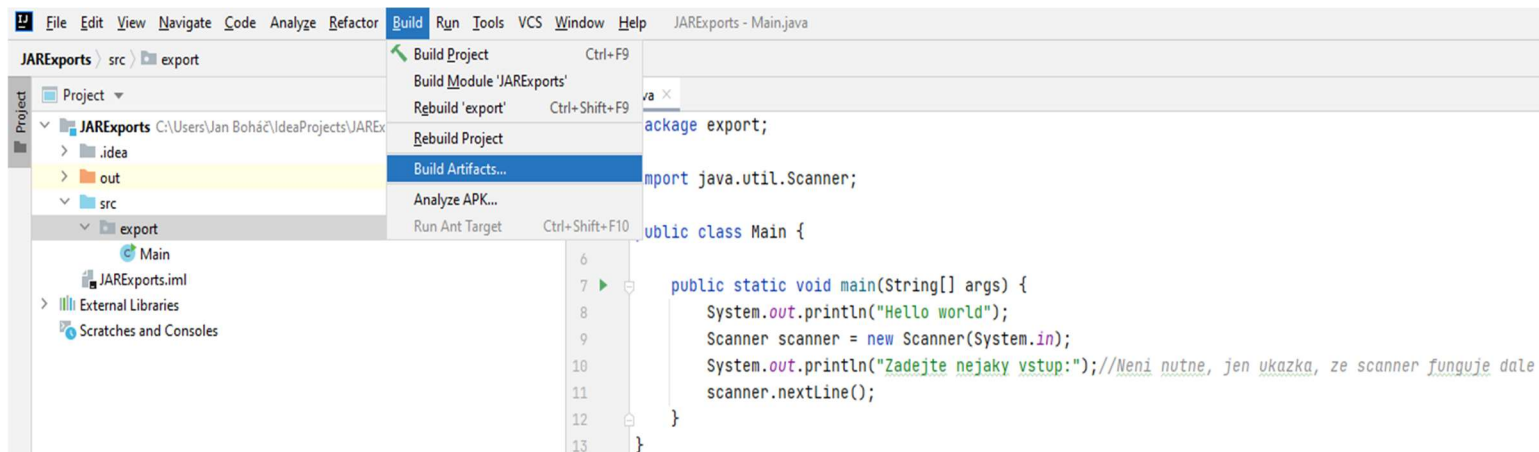




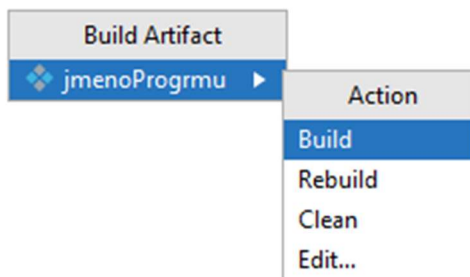
Pokud jsme postupovali správně, tak menu můžeme v klidu potvrdit tlačítkem OK popř apply+ok. Samotné menu by mělo vypadat tedy takto, zkontrolujte si, že odpovídá tomuto obrázku a zároveň je při kliknutí na jmeno.jar vidět v dolním menu manifest a hlavní třída.



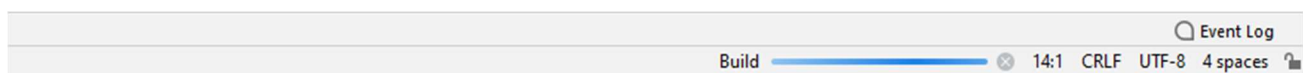
Pro spuštění tvorby JAR souboru si v horní liště IDE zvolíte možnost *Build* a logicky když jsem dělali v menu *Artifacts*, tak vás i teď zajímá *Build Artifacts*




Po zvolení u vás někde v prostředku kódu vyskočí menu s nabídkou vybuildění nějakého konkrétního Artifactu, jelikož tento je první, víc by jich tam být nemělo, v podsložce nás zajímá kolonka Build, popř. Rebuild, pokud jste vybuildili a chcete hned vytvořit další JAR s nějakou změnou, ale nepohnuli jste s původním.



Po stisknutí byste měli u sebe v dolním rohu vidět process build (možná bude tak rychlý, že se nestihne... pokud jste udělali nějakou chybu v programu (program se testovaně přeloží), tak se JAR nevytvoří a ukáže vám program chybu (například kdybyste někde vynechali středník))

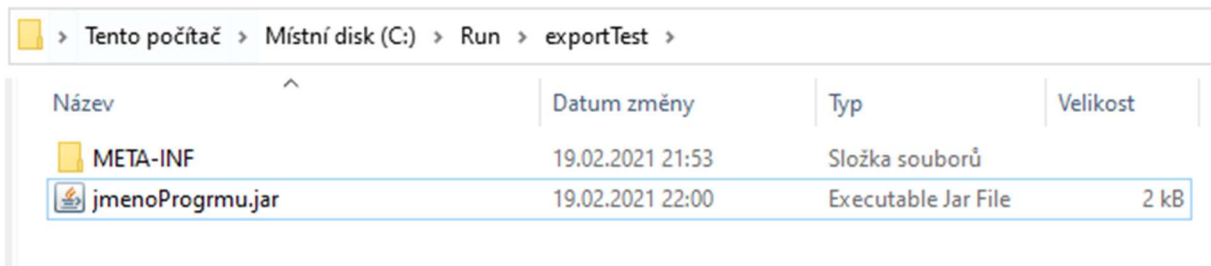


Ve složce projektu by měl být v tento moment vygenerovaný .jar soubor

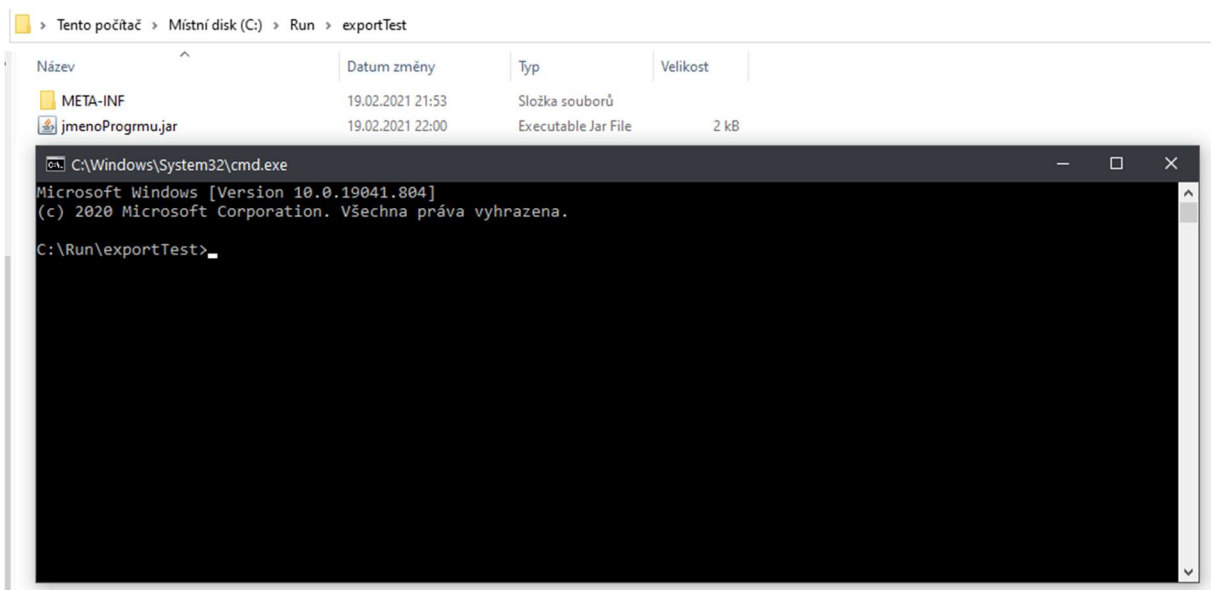
C:\Users\Jan Boháč\IdeaProjects\JARExports\out\artifacts\jmenoProgrmu			
Název	Datum změny	Typ	Velikost
 jmenoProgrmu.jar	19.02.2021 22:00	Executable Jar File	2 kB

## Spuštění JAR

Námi vytvořený jar si dame do složky, ze které ho budeme spouštět, není od věci, aby to byla nějaká separovaná složka, kde je uložena i složka s Manifestem.



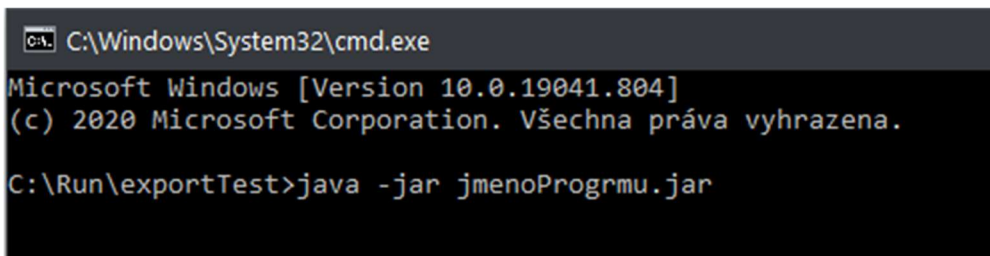
V této složce spustíme příkazový řádek (buď ho zadáte ve vyhledávači a cmd nastartuje rovnou v této složce, nebo ho klidně spustíte klasicky a pomocí cd se dostanete do složky)



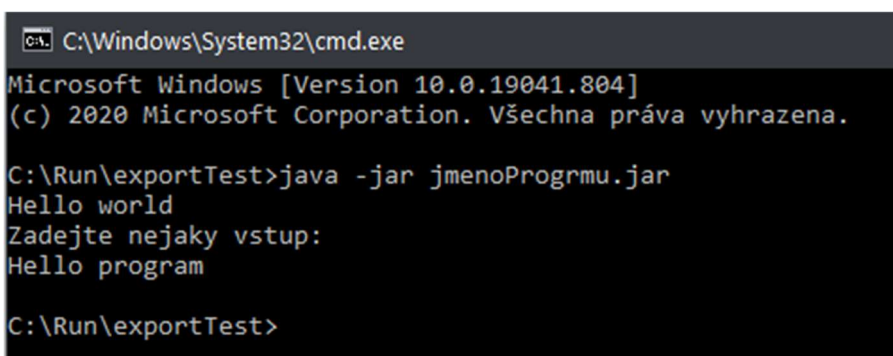
Samotný program lze supstit následujícím textem:

***java -jar jmeno.jar***

Příkazy, bez -jar nebudou fungovat, jelikož by to musel jinak být celý build projektu.

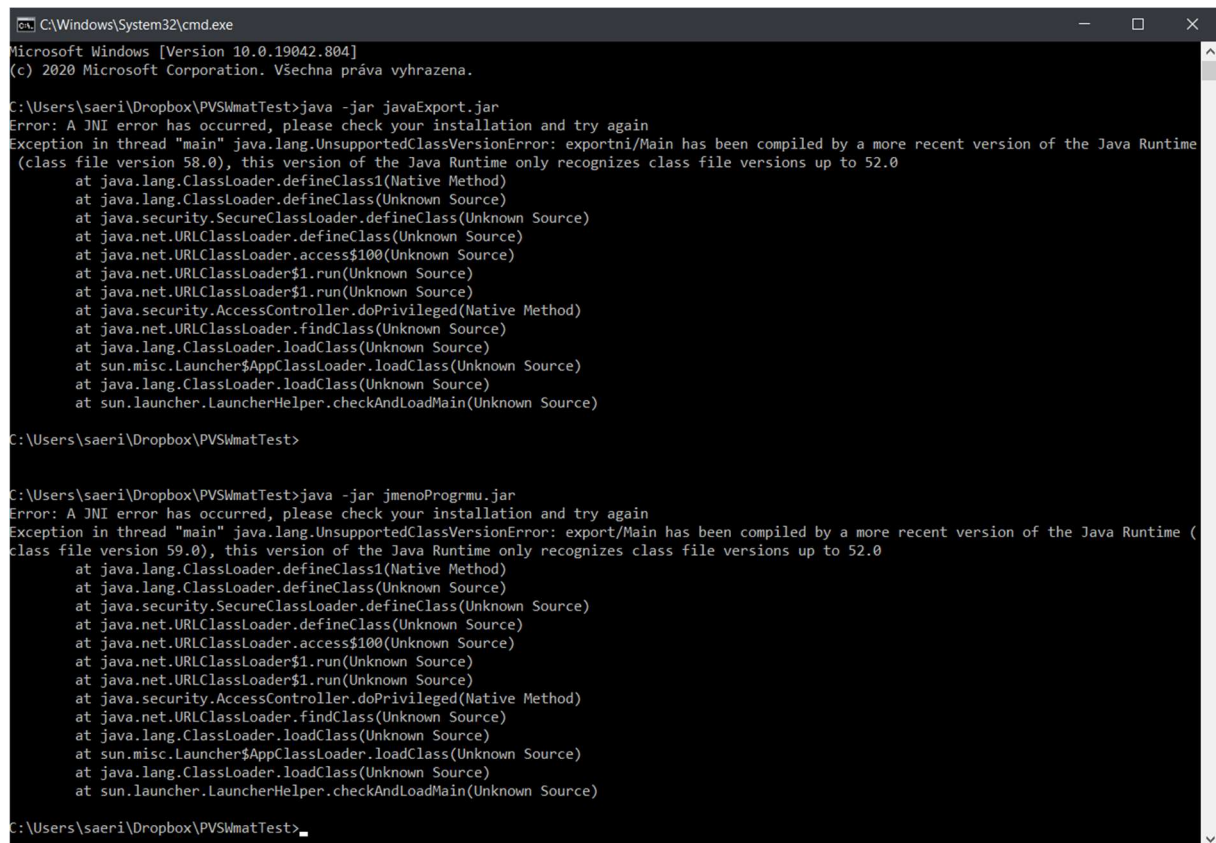


To, co tento příkaz vypíše se bude lišit u každého z vás. Pokud vám nevyskočila žádná výjimka, měli byste se dostat do tohoto stavu a gratuluji, máte hotovo:



## Řešení nekompatibilních verzí javy

Trochu nudné teorie, ale abyste vůbec věděli, proč to řešíme: Celkový zdroj problémů je java 8, která je takovým standardem, ta podporuje nějaké verze, konkrétně vše do 52, tu pravděpodobně máme minimálně všichni, možná máte někteří více. Veškeré budoucí verze se nějak přezývají, ale tady konkrétně osmička si s vyššími moc nekamarádí, proto příkládám obrázek, všimněte si čísla 52, to symbolizuje verzi javy version 52, kam až zvládne tato cmd zpracovat kód. Cokoliv and už bude házet error, viz:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\saeir\Dropbox\PVSWmatTest>java -jar javaExport.jar
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.UnsupportedClassVersionError: exportni/Main has been compiled by a more recent version of the Java Runtime
(class file version 58.0), this version of the Java Runtime only recognizes class file versions up to 52.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(Unknown Source)
    at java.security.SecureClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.access$100(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)

C:\Users\saeir\Dropbox\PVSWmatTest>

C:\Users\saeir\Dropbox\PVSWmatTest>java -jar jmenoProgramu.jar
Error: A JNI error has occurred, please check your installation and try again
Exception in thread "main" java.lang.UnsupportedClassVersionError: exportni/Main has been compiled by a more recent version of the Java Runtime (
class file version 59.0), this version of the Java Runtime only recognizes class file versions up to 52.0
    at java.lang.ClassLoader.defineClass1(Native Method)
    at java.lang.ClassLoader.defineClass(Unknown Source)
    at java.security.SecureClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.defineClass(Unknown Source)
    at java.net.URLClassLoader.access$100(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.net.URLClassLoader$1.run(Unknown Source)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.misc.Launcher$AppClassLoader.loadClass(Unknown Source)
    at java.lang.ClassLoader.loadClass(Unknown Source)
    at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)

C:\Users\saeir\Dropbox\PVSWmatTest>
```

Dokonce je k vidění, že problémem je ta možnost překládat and 52.0.

Je nutné si uvědomit, co to znamená, není to, jak váš projekt tento projekt zpracuje (to se dá nastavit v projektu), jde ale o to, zda Java, kterou používá váš počítač pro hraní si, je systémově nastavená na nějakou verzi (čili v IntelliJ můžu klidně jet na javě 15, ale můj komp prostě javu bere od osmičky)

Abychom si to ověřili, tak si pojdme zadat dva příkazy do cmd, které by nám měly ledacos napovědět:

```
Příkazový řádek
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\saeri>java -version
java version "1.8.0_181"
Java(TM) SE Runtime Environment (build 1.8.0_181-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.181-b13, mixed mode)

C:\Users\saeri>javac -version
javac 1.8.0_162


C:\Users\saeri>
```

Tohle mi říká, že i když si v IntelliJ hraju s Javou 15, tak 1.8 odpovídá. Existují různé praktiky, jak si pohrát se systémovými proměnnými konkrétně JAVA\_HOME a PATH, ale toho se teď zdržíme, jelikož nevím, jaká máte vlastní individuální nastavení a místo toho se shodneme, že naším úkolem bude si tuto Javu aktualizovat.

Abychom k tomu došli, stáhneme si Javu 15 z tohoto webu, kde je ta distribuce ok. Zkusíme si stáhnout tuto verzi, jelikož má svojí výhodu v rámci installeru.

<https://www.oracle.com/java/technologies/javase-jdk15-downloads.html>

Na tomto odkazu nás zajímá možnost pro Windows (nebo mac) hlavně **Installer** tento installer totiž všechny ty záležitosti s nastavováním systémových proměnných spolehlivě, aniž byste tuto část musle riskovat sami, ono je tam docela velkého prostoru pro chybu.

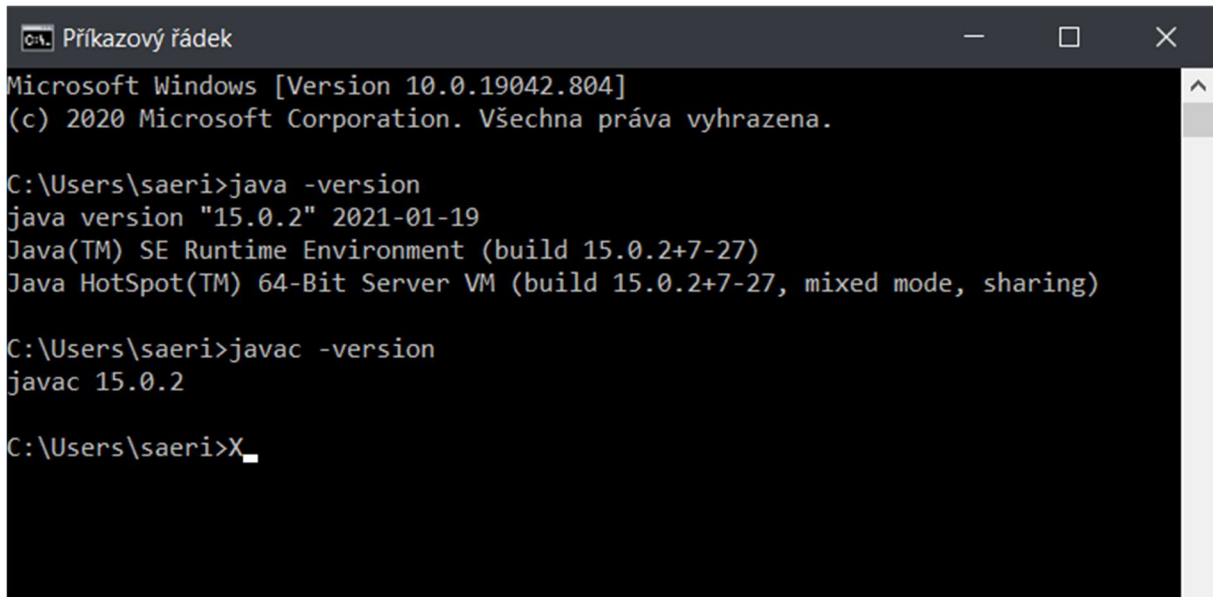
Windows x64 Installer	159.71 MB	 <a href="#">jdk-15.0.2_windows-x64_bin.exe</a>
-----------------------	-----------	--

Takže opakuji – chci installer, kdybych vzal jen archiv, tak si to vše budu muset nastavit sám, nechci.

Pohodlně spustíme installer a proklikáme se tak, že mu na všechno kývneme a necháme ho, ať to vše nastaví za nás. Zavřete si všechna cmd, co vám běží v pozadí, automaticky by se nemusely aktualizovat, v případě chyb doporučuji i restartovat PC.

## Už je to funkční?

Tak si to jdeme ozkoušet, víme, že ke spuštění těch jarů potřebuju něco víc než Javu 8, resp více než prostředí 1.8.0.... po resetu znovu zadáme příkazy, abychom si zobrazili verze:



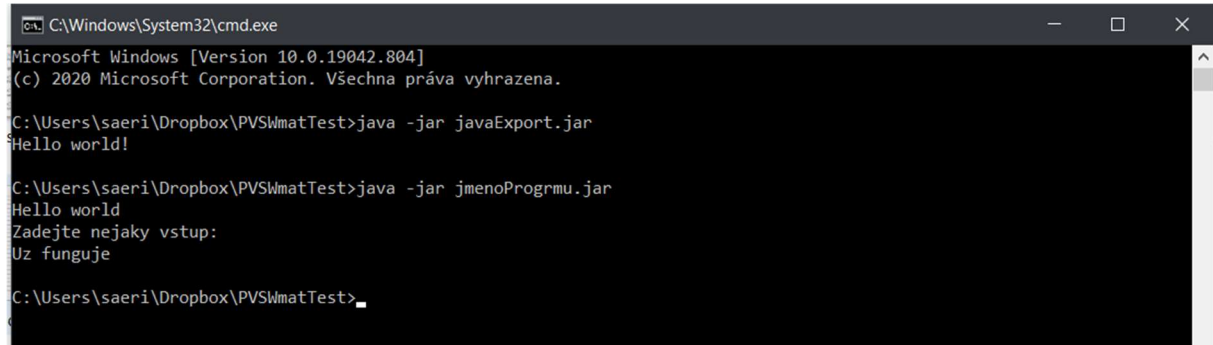
```
C:\Users\saeri>java -version
java version "15.0.2" 2021-01-19
Java(TM) SE Runtime Environment (build 15.0.2+7-27)
Java HotSpot(TM) 64-Bit Server VM (build 15.0.2+7-27, mixed mode, sharing)

C:\Users\saeri>javac -version
javac 15.0.2

C:\Users\saeri>X_
```

...což přesně to, co chceme, verze vypadají OK. Ted' už tedy vše vypadá fajn, doporučuji ozkoušet na jarech:

No a můžeme zaplesat:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. Všechna práva vyhrazena.

C:\Users\saeri\Dropbox\PVSWmatTest>java -jar javaExport.jar
Hello world!

C:\Users\saeri\Dropbox\PVSWmatTest>java -jar jmenoProgrmu.jar
Hello world
Zadejte nejaky vstup:
Uz funguje

C:\Users\saeri\Dropbox\PVSWmatTest>_
```