Javička

# Základní příkazy v CLI

* java –version
* javac <JAVA FILE>
* java <CLASS FILE>
* jar cvfm <NEW JAR FILE NAME>.jar manifest.txt \*.class
* jar tf/xf <JAR FILE>.jar
* java –jar <JAR FILE>.jar

## java –version

Tento příkaz v podstatě i slouží k tomu, abyste si zkontrolovali, že máte nainstalovanou Javu v systému a můžete používat základní příkazy a spouštět java aplikace.

K tomu potřebujete mít nainstalovanou Javu (stačí JRE) a nastavenou ji správně v systémových proměnných. K systémovým proměnným se dostanete třeba přes:

1. **Tento počítač**
2. **kliknout pravým myšítkem někde ve složce Tento počítač** (ne na ikonu Tento počítač!!)
3. zvolit **vlastnosti**
4. v levém menu zvolte poslední položku **Upřesnit nastavení systému**
5. klikněte na tlačítko **Proměnné prostředí…**
6. A v druhém podokně (Systémové proměnné) najděte položku **Path** a tu si rozklikněte dvojklikem
7. No a tady musí být v seznamu kdekoliv cesta k Java **JDK**.

Standardně po nainstalování Javy bývá dneska už tato cesta: **C:\Program Files (x86)\Common Files\Oracle\Java\javapath**. V tomto souboru pak bývá cesta k aktuální verzi Javy.

Aby vám fungoval i následující příkaz, musí být striktně v seznamu cest (Path) uvedena i přímo cesta k vašemu JDK, takže kupříkladu přidejte řádek: **C:\Program Files\Java\jdk1.8.0\_261**

## javac

Příkaz, který zkompiluje .java soubor do bytekódu (.class souboru). Bytekód je to, co umí JVM přečíst a spustit. Aby vám tento příkaz fungoval, musí být nastavena cesta k JDK v systémových proměnných (viz předchozí odstavec – sekce Path). Pokud vám píše příkaz javac „javac command not found“, tak to znamená, že nemáte správně nastavenou systémovou cestu k JDK.

## java

Tento příkaz spouští .class soubory. Není třeba psát příponu. Pokud máte k dispozici .class soubor, ve kterém je zkompilovaná main metoda, tak jej tímto příkazem spustíte.

## jar cvfm

Příkazem „**jar cvfm <název>.jar manifest.txt \*.class**“ vytvoříte spustitelný JAR soubor. Podmínkou je, abyste měli k dispozici zkompilovanou třídu s main metodou, která spustí aplikaci + obyčejný textový soubor s názvem manifest, ve kterém je uvedeno, která třída bude spouštěcí třída aplikace. Prostě v tom souboru bude pouze jeden řádek s tímto příkazem: **Main-Class: <název hlavní třídy bez přípony>**

**Co znamenají jednotlivé přepínače za příkazem jar:**

* c (create) už podle názvu znamená vytváření JAR souboru
* v (verbose output) výpis do konzole o tom, jestli se podařilo nebo nepodařilo vytvoření JARka
* f (filename) přepínač, kterým určujete, že zadáte název souboru
* m (manifest file) přepínač, kterým určujete, že bude přidán manifest.txt soubor.

Na konkrétním příkladu na konci tohoto dokumentu uvidíte podrobný postup, jak si vytvořit a spustit velice jednoduchý java příklad.

## jar tf/xf

Tento příkaz slouží k rozbalení JAR souboru. Přepínač **xf** slouží pro rozbalení všech souborů. Pokud vám stačí pouze výpis toho, co je v JAR souboru, tak použijte přepínač **tf**.

## java –jar

Příkaz pro spuštění jar souboru.

# Demonstrativní příklad

Budeme používat GIT BASH v domnění, že je vše nainstalováno a nastaveno správně.

1. Otevřete si git bash (třeba na ploše nebo kdekoliv)
2. Příkazem „touch“ vytvořte nějaký soubor s příponou .java

*touch Soubor.java*

1. Příkazem „vim“ otevřete super editor☺ (pokud neumíte s vimem, tak jste hodně špatní, ale ok, tak si to otevřete v nějakém jiném editoru)

*vim soubor.java*

1. Napište jednoduchý prográmek pro vytvoření malého okýnka

*import javax.swing.\*;*

*public class Soubor {*

*public static void main(String[] args) {*

*JFrame frame = new JFrame();*

*frame.setSize(400, 400);*

*frame.setLocationRelativeTo(null);*

*frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);*

*frame.setVisible(true);*

*}*

*}*

1. Zkompilujeme program pomocí příkazu javac

*javac Soubor.java*

1. Pro zkoušku zkusíme spustit aplikaci pomocí příkazu java

*java Soubor*

1. Nyní vytvoříme manifest soubor (a použijeme rychlejší cestu, takže rovnou i spustíme editor)

*vim manifest.txt*

1. A napíšeme do něj mocný příkaz, kterým určíme main class

*Main-Class: Soubor*

1. Teď vytvoříme jar soubor pomocí příkazu jar cvfm

*jar cvfm Testovaci.jar manifest.txt \*.class (nebo by se dalo napsat jar cvfm Testovaci.jar manifest.txt Soubor.class, protože máme jen jeden class soubor)*

1. No a spustíme si jarko

*java –jar Testovaci.jar*

Pokud byste chtěli přibalit do JAR souboru další soubory (třeba výše uvedený Soubor.java), tak stačí jednoduše tento soubor připsat

*jar cvfm Testovaci.jar manifest.txt Soubor.class Soubor.java*

Prostě pak už je to na vás, co si tam všechno narvete. Jestli složku, určité soubory, všechny soubory, to je prostě na vás ☺