

# Uživatelská příručka

Autor: Martin Fiala

Vzhledem k tomu, že navržený program nefunguje samostatně a jedná se pouze o část implementovanou do *pluginu* BJ2NB vyvíjeného na Vysoké škole ekonomické, je nutné pro správnou funkci programu, nainstalovat celý *plugin*. Následuje návod, jaké kroky je třeba učinit pro instalaci pluginu a spuštění generátoru kopenogramů.

## 1. Spuštění pluginu jako projektu v NetBeans

V případě zájmu je možno současnou verzi pluginu spustit i přímo jako projekt z NetBeans a dostat se tak ke zdrojovým kódům pluginu:

- a) Rozbalte rar, který jste stáhli ze stránek [java.vse.cz/4it101/Kopenogramy](http://java.vse.cz/4it101/Kopenogramy)
- b) Rozbalený projekt přidejte v *Netbeans* pomocí menu *File* a položky *Open Project*.
- c) Spusťte projekt kliknutím pravého tlačítka myši na něj a zvolením položky *Run*.
- d) Otevře se nová instance *Netbeans*, v ní pak stačí otevřít libovolný projekt, otevřít třídu, a v ní kliknout pravým tlačítkem na název metody, kterou chceme zobrazit a pod položkou „Refactor“ kliknout na „Create kopenogram“.

## 2. Instalace pluginu

Před začátkem tohoto postupu se předpokládá stažení souboru s *pluginem*. Jednotlivé kroky návodu, popsané v této kapitole, se vztahují na anglickou jazykovou verzi prostředí *Netbeans*:

- a) Spusťte vývojové prostředí *Netbeans*.
- b) Na hlavním panelu vyberte menu „Tools“ a v něm položku „Plugins“.
- c) Otevře se dialogové okno, ve kterém je několik záložek. Vyberte záložku *downloaded*.
- d) Klikněte na tlačítko „Add Plugins...“ a v okně, které se objeví, najděte soubor s *pluginem* (koncovka „.nbm“).
- e) Klikněte na tlačítko „Install“ ve spodním levém rohu okna.
- f) Potvrďte dialogové okno, které se vám objeví pro pokračování.
- g) Následně je nutné odsouhlasit licenční podmínky pro používání *pluginu*.

h) V dalším kroku se *plugin* nainstaluje.

i) Po instalaci *pluginu* je nutné restartovat vývojové prostředí *Netbeans*. Po opětovném spuštění bude vývojové prostředí běžet s nainstalovaným *pluginem*.

### 3. Používání generátoru kopenogramů jako součásti prostředí NetBeans

Jakmile je založen nový projekt, je třeba mít v něm vytvořenu alespoň jednu třídu. Tato třída se bude zobrazovat v okně navigátoru v levé části vývojového prostředí. Ve třídě je třeba mít deklarovanou metodu, kterou chce uživatel vykreslit pomocí generátoru kopenogramů. Po kliknutí pravým tlačítkem na název metody se objeví kontextové menu, a v něm pod položkou *Refactor*, klikne uživatel na položku *Create kopenogram*. V novém okně se objeví výsledný kopenogram.

Pokud by si uživatel chtěl nastavit jiné, než předdefinované barvy kopenogramů, nebo použít další možná nastavení generátoru, klikne v nástrojové liště vývojového prostředí na položku „*Tools*“, a v menu, které se objeví, na položku „*Options*“. V zobrazeném okně *Options* klikne uživatel na záložku „*Source code Vizualization*“ a zde zvolí požadovaná nastavení generátoru kopenogramů.

### 4. Nastavení programu – menu Options

Třída *Properties* umožňuje vytvořit soubor, do kterého může uživatel lehce zasáhnout a definovat si vlastní hodnoty konstant, které jsou v programu využívány. Je to tedy jakési nastavení programu. Ve vytvářeném programu je tato funkce využívána pro nastavení barev všech prvků, které diagram vytváří, mezer mezi prvky a dalších prvků, které budou následně popsány.

Soubory *properties* se obvykle ukládají na snadno přístupných místech, nebo místech, která jsou k tomu předem určena. Uživatel poté může tyto soubory snadno najít a upravit, ručně v textovém editoru. Je však také možné vytvořit program, který tyto soubory upravuje. V případě generátoru kopenogramů, se tento soubor automaticky upravuje na základě voleb v menu *Options* generátoru kopenogramů.

### 5. Nastavení generátoru kopenogramů

V této části jsou podrobně popsány všechny proměnné, které je možno změnit v souboru *Properties* a tak ovlivnit grafické prvky diagramu, který vytváří.

## 5.1. Zobrazování klíčových slov

*KopenogramKeywords* – v původní definici kopenogramů jsou veškeré bloky příkazů zobrazovány bez klíčových slov. Při vytváření nástroje, byla klíčová slova (např. *if*, *switch*, *for*) do diagramů zahrnuta. Generátor dosud zpracovává pouze metody v Javě, proto nevzniká problém s terminologií různých jazyků. Ale z toho důvodu, aby se uživatel mohl rozhodnout, zda chce nebo nechce klíčová slova v diagramu zobrazit, je zde tento checkbox, který při zaškrtnutí udává zobrazení klíčových slov a naopak nezaškrtnutý udává nezobrazování klíčových slov.

## 5.2. Zobrazování dlouhých příkazů

*NumberOfSymbolsInExpression* – tato proměnná souvisí s tím, že ve vytvářených programech se setkáme s mnoha příkazy různé délky. Existují krátké příkazy jako definice proměnné, nebo příkaz „*Return*“. Na druhé straně se stává, že používání tříd a metod vytvoří jeden dlouhý příkaz na několik řádků. Takto dlouhé příkazy by v diagramu působily nevhodně a narušovaly by jeho strukturu. Problém je možné vyřešit dvěma způsoby:

- a) rozdělit příkaz na více řádků,
- b) omezit příkaz na určitý počet znaků.

Z důvodu omezeného času, vyhrazeného na implementaci zmíněného problému byla zvolena druhá možnost. Díky této proměnné je možné definovat počet znaků, které může program mít. Základní nastavená hodnota je 35, ale lze nastavit libovolný počet znaků a zobrazit tak dlouhé příkazy v úplnosti, ovšem přinejmenším za cenu neesteticky formátovaného diagramu.

## 5.3. Velikost písma v diagramech

*FontSize* – je proměnná, která nastavuje velikost písma v diagramech. Písmo je možné pro lepší čitelnost zvětšit. Základní velikost je 20 – udává se v pixelech.

## 5.4. Mezery

Jednotlivé prvky diagramu, až na výjimky, mají pro lepší odlišení mezi sebou mezery.

V následujících dvou proměnných je možné tyto mezery regulovat.

- c) *VerticalGap* – mezera mezi prvky, které jsou umístěny pod sebou. Je možné ji nastavit – diagram se roztáhne na délku. Základní nastavení je 15 pixelů.
- d) *HorizontalGap* - mezera mezi prvky, které jsou umístěny vedle sebe. Je možné ji nastavit - diagram se roztáhne na šířku. Neovlivňuje však prvky, které jsou spojeny *Vertical Containerem*. Základní nastavení je 5 pixelů.

## 5.5. Barvy

Nastavení barev v diagramu je velice důležité. Základní barvy, které diagram obsahuje, jsou samozřejmě dány definicí kopenogramů. Existuje však také možnost, aby si uživatel zvolil barvy, více odpovídající jeho představám a záměru.

I v případě, že by uživatel chtěl dodržovat definované barvy kopenogramů, může nastat situace, kdy bude vhodné barvy změnit. Typickým příkladem takové vhodné změny je promítání diagramů na projektoru. Projektory totiž obvykle nemají barevný rozsah srovnatelný s monitory počítačů, a tak se barvy na monitoru často jeví jinak, než je tomu na projektoru. Světlé odstíny různých barev, které jsou v kopenogramech hojně využívány, se mohou všechny zobrazovat jako bílá barva. Z toho důvodu je důležité před prezentací na projektoru všechny barvy vyzkoušet a případně upravit.

V menu *Options* lze nastavit barvy všech prvků kopenogramu. Na výběr je z několika předdefinovaných barev a nebo si uživatel může vybrat svou barvu možností *custom*.

Jednotlivé části kopenogramů je možno označit těmito barvami:

*Block Body Color*(světle žlutá barva) – barva těla bloku, je stejná jako barva těla metody.

*Break Color* (sytě červená barva) – barva, která podbarvuje příkaz *break*, rovněž toto nastavení změní i barvu příkazu *return*.

*Catch Head Color*(tmavě fialová barva) – barva hlavičky bloku s odchycením výjimky.

*Catch Body Color*(světle fialová barva) – barva těla bloku s odchycením výjimky.

*Continue Color* (tmavě zelená barva) – barva prvku *continue*.

*Cycle Head Color* (zelená barva) – barva hlavičky cyklů. Vztahuje se na všechny

cykly: *for*, *foreach*, *while*, *do/while*.

*Cycle Body Color* (světle zelená barva) – barva těla cyklů, je nastavena jako světlejší odstín zelené barvy hlavičky.

*Expression Color* (světle červená barva) – barva, která podbarvuje všechny příkazy, jako jsou definice proměnných, výpis do konzole atd.

*Finally Head Color* (tmavě fialová barva) – barva hlavičky bloku *finally*.

*Finally Body Color* (světle fialová barva) – barva těla bloku *finally*.

*Labeled Statement Color* (tmavě oranžová barva) – barva prvku *label*.

*Method Head Color* (žlutá barva) – barva hlavičky metody.

*Method Body Color* (světle žlutá barva) – barva těla metody.

*Synchronized Head Color* (tmavě šedá barva) – barva hlavičky bloku *synchronized*

*Synchronized Body Color* (světle šedá barva) – barva těla bloku *synchronized*.

*Switch Head Color* (modrá barva) – barva hlavičky přepínačů, tedy příkazů *if aswitch*.

*Switch Body Color* (světle modrá barva) – barva podbarvující tělo přepínačů.

*Throw Color* (tmavě červená barva) – barva prvku *try*.

*Try Head Color* (tmavě fialová barva) – barva hlavičky bloku *synchronized*

*TryBody Color* (světle fialová barva) – barva těla bloku *try*.

Pokyny i informace obsažené v uživatelské příručce by měly být dostatečným vodítkem pro instalaci a užívání nástroje, sloužícího k tvorbě kopenogramů. Věřím, že bude dobrým pomocníkem při grafickém zobrazování algoritmů.