

Programování pomocí grafických symbolů

Bc. Tomáš Horák, Ing. Tomáš Martinec, Ph.D.

Abstrakt

Příspěvek obsahuje popis využití grafických symbolů pro tvorbu jednoduchých algoritmů. Byl vytvořen grafický programovací jazyk, který vznikl jako podmnožina symbolů vývojových diagramů definovaných normou. Vzniklo vývojové prostředí, které pomocí tohoto jazyka umožní tvorbu algoritmů a následně funkčních konzolových či grafických aplikací, případně aplikací pro vývojovou desku používanou při výuce předmětu Počítačový hardware a rozhraní.

Úvod

Grafické symboly jsou hojně využívány pro programování v oblasti automatizační techniky a nejrůznějších robotů. Existuje několik programovacích jazyků, založených na použití grafických symbolů. Příkladem může být diagram funkčních bloků, využívaný hlavně pro definování elektrických schémat, stavový diagram pro popis stavových strojů, či žebříkový diagram využívaný pro definování funkcí některých programovatelných automatů. Velice názorně jsou algoritmy popsány také pomocí vývojových diagramů, které našly využití při programování například robotů, ale hlavně se velice často využívají při výuce.

Právě pro svou názornost byly pro účel definování jednoduchých algoritmů vybrány právě vývojové diagramy. Při výuce se vývojové diagramy zpravidla tvoří na papíře, což má jednu nevýhodu. Na papíře je velice těžké dělat v již vytvořených diagramech změny. Tato nevýhoda je odstraněna, pokud je pro tvorbu vývojového diagramu použit počítač s nějakým vhodným vývojovým prostředím.

Metody

Jelikož norma definující vývojové diagramy popisuje velké množství symbolů, bylo nutné vybrat ty, které definují samotný algoritmus nějakého výpočtu. Těmito symboly jsou symbol pro start a konec algoritmu, zpracování dat, vstup a výstup dat, podmínka, cyklus a podprogram.

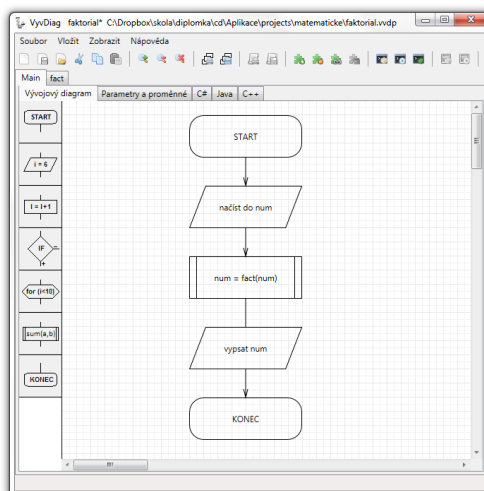
Poté, co byl definován jazyk, bylo nutné stanovit možnosti budoucího vývojového prostředí. Základní funkcí samozřejmě musí být možnost graficky sestavit vývojový diagram. Většina algoritmů pro svou funkci potřebuje proměnné, bylo nutné stanovit způsob tvorby proměnných, dostupné datové typy a jejich použití v diagramu. Výsledkem snažení uživatele bude funkční aplikace, která ale vznikne klasickým způsobem, kompilací ze zdrojových kódů. Další fází tedy bylo navržení způsobu generování zdrojových kódů z vývojových diagramů a také vybrání vhodného programovacího jazyka.

Požadavkem bylo také umožnit vizuální simulaci průběhu algoritmu. Byl navrhnut jednoduchý simulátor algoritmů s vizualizací průběhu a zobrazením aktuálních hodnot proměnných.

Výsledky a diskuze

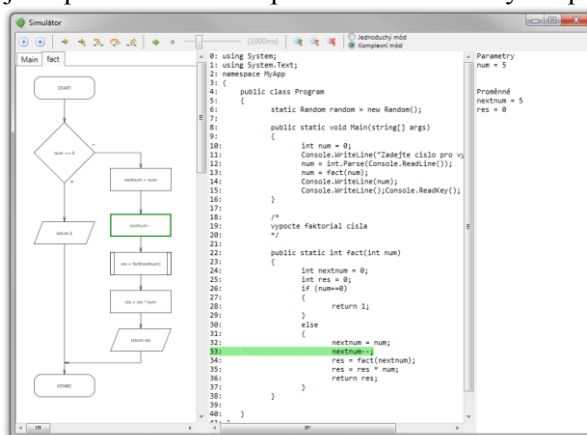
Vzniklo vývojové prostředí, které umožňuje tvorbu algoritmů pomocí vývojových diagramů. Z takto vytvořených diagramů umožňuje vygenerovat zdrojové kódy v jazycích C#, Java, C++ a C. Z vygenerovaných zdrojových kódů umožňuje vytvořit funkční konzolové aplikace, aplikace s grafickým uživatelským rozhraním, jehož návrh je umožněn také a aplikace vhodné pro nahrání do vývojové desky používané při výuce na TUL.

Rozšířený Abstrakt



Obrázek 1. Vývojový diagram použití funkce pro výpočet faktoriálu vytvořený v prostředí

Prostředí také umožňuje simulaci průběhu algoritmu. Samotný simulátor je postaven nad knihovnou Mdbg poskytující aplikační rozhraní pro ladění konzolových aplikací na platformě .NET.



Obrázek 1. Simulátor vytvořených algoritmů

Závěr

Výstupem práce je vývojové prostředí, které může usnadnit začátky programování studentům při výuce a graficky jim nastítnit události vznikající v průběhu algoritmu nějakého výpočtu. Možnosti prostředí jsou velice rozsáhlé, a proto by bylo vhodné provést jeho testování na větším okruhu lidí. Nedostatkem se může zdát omezení jednoho symbolu diagramu pouze na jednu operaci, takže některé složitější výpočty vyžadují velké množství symbolů. Tento přístup byl zvolen z důvodu zachování názornosti funkcí symbolů s ohledem na to, že tvořeny budou z větší části pouze jednoduché algoritmy.

Reference

- [1] ČSN ISO 5807. 1996. Zpracování informací: Dokumentační symboly a konvence pro vývojové diagramy, diagramy toku dat, programu a systému, síťové diagramy systému a diagramy zdrojů systému. První vydání. Praha: Český normalizační institut.
- [2] PETZOLD, Charles. Mistrovství ve Windows Presentation Foundation. Vyd. 1. Brno: Computer Press, 2008, 928 s. ISBN 978-80-251-2141-2.
- [3] MICROSOFT. CLR Managed Debugger (mdbg) Sample 4.0. Microsoft download center [online]. 31.1.2011 [cit. 2014-03-02]. Dostupné z: <http://www.microsoft.com/en-us/download/details.aspx?id=2282>.