

## Grafický popis algoritmu



### POUZE KONCEPT

#### vývojový diagram

slouží ke grafickému znázornění algoritmu pomocí symbolů a čar (šipek)

pravidla:

algoritmus postupuje shora dolů, zleva doprava, pokud není jiný směr uveden šipkami, ty určují směr v jiném případě

obdélník s popisem - krok algoritmu

kosočtverec - větvení postupu algoritmu podle splnění podmínky(cyklus) ??

obdélník se zaoblenými rohy - počátek nebo ukončení algoritmu

obdélník s čarami po stranách - programy

rovnoběžník(lichoběžník) - vstup, výstup

šestiúhelník - cyklus

druhy algoritmů

document flowcharts - ukazují řízení toků dokumentů

data flowcharts - řízení toků dat

system flowcharts - řízení toků fyzické vrstvy, nebo vrstvy zdrojů

program flowcharts - řízení toků v programu

UML

unified modeling language

grafický jazyk pro vizualizaci a návrhy programových způsobů. Podporuje objektově orientovaný přístup k analýze, ale již nespécifikuje metodiku funkce programu.

účely UML

Kreslení konceptu - do diagramů se nakreslí podstatné věci před programováním

Kreslení detailních návrhů - aby programátor přesně pochopil, co mu analytik zadává

Jako programovací jazyk - z diagramů přímo spustitelný kód, v této souvislosti se často používá pojem MDA (model driven architecture)

Diagramy UML se dělí na dvě základní skupiny:

Diagramy struktury (tříd, objektů, apod.) - zobrazují vnitřní strukturu programu

Diagramy chování (use-case diagramy,...) - zobrazují pouze vnější chování, to, co uvidí běžný uživatel

Use-Case diagramy - doslova „případ užití“ - zobrazují pouze funkcionalitu systému, chování programu tak, jak jej vydí uživatel, nikoliv přesnou realizaci. Diagram ukazuje, co má systém umět, ale nikoliv jak to bude zrealizováno. Základní dva prvky diagramu jsou značky aktérů (actors - uživatelů) a případů užití (use case - jednotlivých možností, které se aktérům nabízejí).

Diagramy tříd - můžeme využít k zobrazení tříd, jejich metod, atributů a vztahů mezi nimi.

Vztahy mezi třídami:

Asociace - vztah informuje o spojení instancí dvou tříd

Agregace - vztah celek - část

Kompozice - silnější verze agregace

Dědičnost - potomek dědí atributy předka

Závislost - změna jedné třídy ovlivní druhou

Realizace - souhrn všech veřejně dostupných metod dané třídy

From:

<https://old.gml.cz/wiki/> - **GMLWiki**

Permanent link:

<https://old.gml.cz/wiki/doku.php/informatika:maturita:17a?rev=1426593091>

Last update: **17. 03. 2015, 12.51**

