

Cvičení

Cíl:

- Vývojové prostředí jazyka Java – NetBeans
- Překlad programu, Makefile
- Běh programu (runtime prostředí)
- Datové typy jazyka Java
- Konstrukce jazyka Java – výrazy, operátory, řídicí konstrukce

Výstup:

- Plnění úkolu 4, 5, 6, 7, 8, 9.
- Plnění testu k tématu.

Zadání:

1. Seznámit se s prostředím NetBeans.
 - a. Konfigurace vývojového prostředí.
 - b. Otevření projektu, import projektu, založení projektu.
 - c. Konfigurace projektu.
 - d. Kompilace projektu a ladění.
2. Instalace jazyka Java, JDK
 - <http://programujte.com/clanek/2007040702-java-tutorial-technologie-1-dil/>
 - <https://www.algoritmy.net/article/21353/Prvni-program-2>
 - a. Kde se nachází kompilátor a interpret jazyka?
3. Vyzkoušet si ruční kompilaci projektu pro jeden soubor a pro více souborů? Které binární soubory jsou kompilovány při rekompilaci?

Vytvořte program *01_typovy_system* (typ konzolová aplikace) s následujícím (statickými) metodami:

4. (Úkol) Vytvořte metodu `void deklarace_promennych()`, ve které se provede deklarace proměnných existujících primitivních datových typů. Všechny proměnné inicializujte. Jaké všechny datové typy byly identifikovány? Vytvořte instanci třídy `Object` a získejte textový popis objektu. Jaká metoda (název metody) třídy `Object` vrací textový popis?
5. (Úkol) S využitím nástrojů Javy vytvořte metodu `void TiskCislo(int cislo)` s celočíselným parametrem. Tato funkce zobrazí hodnotu čísla předaného parametrem v dekadickém tvaru, v binárním tvaru, v hexadecimálním tvaru a v osmičkovém tvaru. Funkci vyzkoušejte na několika číslech zadaných v programu.
6. (Úkol) Vytvořte metodu, která v přehledné formě zobrazí seznam datových typů, u vhodných datových typů pak rozsah hodnot a velikost místa v paměti pro jednu proměnnou (zobrazit ve formátu B a b). Použít vlastnosti `.MAX_VALUE`, `.MIN_VALUE` a `.SIZE` obalovací třídy datového

typu; použít tabulátor a znaky ‘|’ a ‘-’ pro vytvoření přehledné tabulky; vytvořit záhlaví tabulky; použít funkce pro zápis na standardní (konzolový) výstup.

Pozn: Pro přepočítání mezi bajty a bity použijte vámi nadeklarovanou konstantu uchovávající hodnotu počtu bitů připadající na 1 B.

Pozn: Pro definování šířky sloupců přehledové tabulky použijte pevnou šířku sloupců, tabulátor a formátování údaje buňky na předem danou šířku sloupce. Implementujte ve třech funkcích.

Použijte nástroje třídy String a formátování řetězců.

Vytvořte program *01_vstup_dat*

7. Vytvořte třídu *VstupDat* se statickými metodami zajišťující načtení hodnoty primitivního datového typu. Uvažujte chybové stavy indikované mechanismem výjimek. Metodu ověřte jejich voláním například v hlavní funkci, zajistěte ošetření chybových stavů.

Vytvořte program *01_sachovnice_01*

8. (Úkol) s funkcí *hra_posun_po_sachovnici*, která bude zpracovávat klávesy pro pohyb po šachovnici (např. <L> posun vlevo, <R> posun vpravo, <U> posun nahoru, <D> posun dolů a K konec hry) a dle toho nastavovat aktuální souřadnice v rámci velikosti šachovnice (rozměry šachovnice jsou zadány a předány prostřednictvím parametrů implementované metody a není možné programově tuto hodnotu změnit). V rámci pohybu po šachovnici, nebude možné překročit hranice šachovnice a budou ignorovány jiné klávesy než klávesy pro pohyb. Výchozí hodnotou je souřadnice [0;0] a aktuální stav bude zobrazován ve formě souřadnice. Klávesou <K> bude hra ukončena, ale až po dotazu, zda skutečně se má hra ukončit.
 - a. Provedte implementaci na základě strukturovaných principů. Nejdříve provedte v rámci jedné funkce (program *01_sachovnice_01*).
 - b. Vytvořte makefile pro kompilaci projektu - definujte název pro výslednou aplikaci.

Vytvořte program *01_sachovnice_02*

9. (Úkol) Provedte přepracování problému ve třídu *HraSachovnice*.. Třída bude poskytovat aktuální hodnotu souřadnic, bude pracovat s identifikátorem výčtového typu pro směr pohybu. Vytvořte makefile zajišťující kompilaci všech částí projektu.