Cvičení

Cíl:

- Třídy a objekty.
- Definice třídy, vytvoření objektu třídy, práce s objektem.
- Konverze datových typů, přetypování objektů, spojování řetězců

Výstup:

- Plnění úkolu 1, 2 a 3.

Zadání:

Konverze datových typů.

a. Jednoduchá konverze mezi číselnými datovými typy a mezi typem char a celočíselným číselným datovým typem.
b. Konverze primitivního datového typu na řetězec, konverze řetězce na primitivní datový typ.
c. Uplatnění konverze v objektech a jeho metodách (*.clone*, *.equals*)

Konverze datových typů, rozsah hodnot, spojování řetězců

1. (Úkol) Konverze datových typů. Vytvořte program s následujícím:
    a. Do proměnné typu byte se zapíše hodnota 0. Poté se provede dekrementace proměnné. Jaká hodnota je v proměnné obsažena? Odůvodněte.
    b. Do proměnné typu byte se zapíše hodnota +127. Po té se provede inkrementace proměnné. Jaká hodnota je v proměnné obsažena? Odůvodněte.
    c. Program pro zadaný znak zobrazí ASCII kód.
    d. Použijte třídu *Trojuhelnik* z přednášky. Vytvořte objekt třídy trojúhelník a odkaz na objekt uložte do deklarované proměnné typu *Trojuhelnik*. Přetypujte objekt na třídu Object a uložte do proměnné příslušného typu *Object*. Tuto instanci přetypujte zpět na *Trojuhelnik* a formálně ověřte, že lze přetypování provést.
    e. Vyzkoušejte chování metody *.ToString*. Pro objekt třídy *Trojuhelnik*, co je výsledkem příkazu *println(tr);* kde tr je instance třídy *Trojuhelnik*. Co tiskne metoda *.ToString*, pokud není ve třídě definována?

Pro plněné úkoly vyvoďte závěry a proveďte shrnutí v textovém dokumentu

2. (Úkol) Vytvořte program (konzolová aplikace) se třídou Znaky. Tato třída bude mít dvě metody. Metoda *tisk_ASCII() : void* zobrazí ASCII kódy pro písmena abecedy a pro znaky číslic na standardní výstup konzolové aplikace. Metoda *get_ASCII() : string* poskytne přehledovou tabulku (viz tisk_ASCII) v textové proměnné.

Projděte si studijní podklady – definice třídy, vytvoření objektu třídy, klonování objektů.

3. (Úkol): Přepracujte realizaci hry z předchozího cvičení tak, aby byla řešena na principech objektového programování. Navrhněte a definujte třídu *Sachovnice* a *Hra_Sachovnice*, která bude realizovat celý proces pohybu po šachovnici. Jaké atributy bude mít třída *Sachovnice*? Pro směry pohybu vlevo, vpravo, nahoru, dolů nadefinujte výčtový typ SMER_POHYBU. Výchozí souřadnice bude [0; 0].

    Třída *Sachovnice(byte size_x, byte size_y, byte actual_x, byte actual_y)*
    Jaký konstruktor třídy bude definován? Co bude v konstruktoru nutně provedeno?
    Metoda *test() : bool* ověří platnost šachovice, resp. její rozměry.
    Metody pro získání aktuálních hodnot souřadnice na šachovnici a velikosti šachovnice.
    Metoda pro test pozice na okraji šachovnice: *can_left() : boolean*, *can_right() : boolean*, *can_up() : boolean*, *can_down() : boolean.*
    Metoda *can_move(SMER_POHYBU direction) : boolean* pro test na pohybu v žádaném směru.

Metoda *move(SMER_POHYBU direction) : boolean* pro uskutečnění pohybu.
Třída *Hra_Sachovnice*
Metoda *nacti_direction() : char* pro načtení symbolu (znaku) udávající směr pohybu (klávesy, viz minulé cvičení).
Metoda *tisk() : void* pro zobrazení aktuálního stavu. Metoda zobrazí aktuální souřadnice a zobrazí šachovnici.
Zajistěte, aby bylo možné uchovat stav objektu *Sachovnice* (hluboká kopie instance), - metoda *.clone*() *: object*. Vyzkoušejte vytvoření kopie. Pozn: Po vytvoření kopie změňte originál a po té se podívejte, zda v kopii jsou původní hodnoty.

Které z výše implementovaných metod mohou být public?