Sada č. 1b

3. Realizace operace načtení údaje ze standardního vstupu. Obsluha chybových stavů a ošetření chybového stavu.

... vytvořte modul (třídu) NacteniHodnot se statickými metodami, které zajistí načtení hodnoty určitého typu ze standardního vstupu. Definujte metody pro načtení hodnoty typu: short, int, float, double, String a char.

Metody pojmenujte: nacti\_short() : Short, nacti\_int() : Integer, …

Metoda bude indikovat chybové stavy (FormatException a NumberFlowException). Metody vyzkoušejte a ověřte zachycení chybových stavů v konzolové aplikaci. Program napíše informaci o chybě: 'Špatný formát údaje' nebo 'Překročen rozsah údaje'.

Pozn: Metody třídy nebudou informaci zobrazovat, informace o chybě je zobrazena až v programu.

Vyzkoušejte funkčnost metod tak, že se hodnota uchová do proměnné datového typu. Jaký je využití toho, že třída vrací proměnnou typu obalovací třídy?

---------------------------------------------------------------

Sada č.2

Vytvořte program v konzolové aplikaci, který bude simulovat pohyb po šachovnici. Realizace bude provedena na objektových principeh. Definujte třídu HraPosunPoSachovnici, která bude realizovat problém a bude nezávislá na typu uživatelského rozhraní. Třída RealizaceHryKonzola bude realizovat hru v konzolové aplikaci.

Nejdříve proveďte návrh tříd a následně jejich implementaci. Vstupem bude velikost šachovnice (výška = počet řádků nx, šířka = počet sloupců ny). Aktuální umistění na šachovnici je dáno atributy x, y. Bude umožněno realizovat pohyb vlevo, vpravo, nahoru a dolů. Pokud bude aktuální poloha na okraji šachovnice, nebude možné se posunout ve směru přes okraj.

Ve výchozím stavu bude poloha [0;0]. Bude zobrazována aktuální poloha ve formě souřadnice a pohyb (historie) bude uchováván v poli. Kdykoliv bude možné získat aktuální historii pohybu.

V konzolové aplikaci pak bude realizace posunu na základě stisku kláves <L>, <R>, <U>, <D>, Historie pohybu bude vytisknuta při každém posunu. Klávesou <K> bude "hra" ukončena.

Využijte modul NacteniHodnot realizovaný v předchozím úkolu.

Pozn: Využijte a použijte vámi definovaný výčtový typ - indikace směru pohybu {left,right,up,down}.

Jaké bude mít třída HraPosunPoSachovnici atributy a metody?

Jaké bude mít třída RealizaceHryKonzola atributy a metody?

—---------------------------------------------

Sada č. 3

Vytvořte plnohodnotou třídu KomplexniCislo tak, aby bylo mozno sčítat, odčítat a násobit komplexní čísla.

KomplexniCislo + KomplexniCislo, KomplexniCislo + 5

KomplexniCislo

Plus(KomplexniCislo) : KomplexniCislo, Plus(Double) : KomplexniCislo

—--------------------------------------------------------

Sada č. 4

U1:

Řešte představené problémy:

a)

Uvažujme barevné modely RGB a CMYK, a barvy vyjádřené barevnými složkami. Barva má intenzitu. Model RGB má více možností výpočtu intenzity barvy.

Barva v barevném modelu RGB je dána hodnotami složky červené (r), zelené (g) a modré (b), datový typ byte.

Intenzita barvy je dána i) průměrem těchto složek, ii) todo . Hodnota barvy je v rozsahu <0; 255>, použijme datový typ byte :);. Intenzita

barvy je reálným číslem.

Definujte třídu BarvaRGB na objektových principech dle uvedeného popisu a potomky třídy realizující konkrétní způsob výpočtu intezity barvy

BarvaRGB1 a BarvaRGB2. Hodnotu barvy lze měnit. Objekt třídy poskytuje stav objektu ve formátu r:HH;g:HHůb:HH (nebo #HHHHHH),

kde HH znamená hexadecimální vyjádření hodnoty barvy.

Jak se jednoduše změní definice třídy (úpravou z BarvaRGB), pokud již nepůjde hodnotu barvy měnit? Zamyslete se nad řešením.

Přidáme třídu CMYK, která má společné vlastnosti a metody s třídou RGB. Vytvořte třídu BarvaCMYK a třídu Barva.

Identifikujte pro každou třídu vlastnosti a metody a definujte rodičovskou třídu Barva.

Objektový model:

Barva -> BarvaRGB -> BarvaRGB1, BarvaRGB2

-> BarvaCMYK

Vyzkoušjte vytvoření objektů baren a výpočet intezitu. Přetypujte objekty barev na třídu Barva.

----

b) Vytvořte (uživatelskou) aplikaci s grafickým uživatelským rozhraním, která bude umět pracovat s nádrží a bude umožňovat přidávat do nádrže

a odebírat z nádrže.

Objekt "nádrže" má stav nádrže, maximální objem (kapacita nádrže), typ obsahu, je umožněno odebírat z nádrže a plnit nádrž.

Při odběru nelze odebrat více než je stav nádrže (nelze jít do mínusu), lze doplňovat nádrž v rámci povolené kapacity a vzhledem k aktuálním

stavu nádrže.

Definujte třídu Nadrz na objektových principech a dle uvedeného popisu. Pro typ obsahu použijme vámi definovaný výčtový typ. Objekt nádrže poskytuje stav objektu ve formátu 'stav/kapacita(míra zaplnění),typ nádrže',

kde míra zaplnění je procentuální naplnění nádrže, např. 25 %. Předpokládejme, že kapacita nádrže a aktuální stav nádrže je celým číslem.

Adekvátně definujte všechny potřebné metody. Pokus o přeplnění je indikován výjimkou

MyException\_PlnaNadrz a pokus o nadměrné odebrání je indikován výjimkou MyException\_PrazdnaNadrz. Textem výjimky je informace o dotčené nádrži a informace o požadované operaci.

Otestujte funkčnost objektu nádrže.

—------------------------------------------------------------

Sada č.5

Knihovny v jazyku Java

Vytvořte program (v konzolové aplikaci) s následujícími metodami:

- metoda mocnina\_cisla spočítá pro zadanou hodnotu reálného čísla druhou a třetí mocninu, druhou a třetí odmocninu. Výsledky zobrazí.

- metoda goniometricke\_funkce určí a zobrazení pro úhel (ve stupních) hodnotu goniometrických funkcí sin, cos, tg a cotg. Výsledky zobrazí.

- metoda trojuhelnik\_pravouhly pro zadané odvěsny pravoúhlého trojúhelníka dopočítá přeponu a určí hodnoty úhlů. Výsledky zobrazí.

- metoda trojuhelnik\_obecny pro zadaný obecný trojúhelník o stranách a, b, c dopočítá obvod, obsah a hodnoty úhlů.

- metoda platnost\_datumu ověří platnost dvou zadaných datumů. Validita datumu, druhý datum pozdější než první datum (např. pro od-do). Datumy jsou zadány uživatelem.

- metoda aktualni\_datum zobrazí aktuální datum a čas, využitelné například pro zobrazení času spuštění programu.

- metoda, která zobrazí obsah složky na disku. Složka včetně cesty je zadána.

- metoda, která zajistí provedení operace zkopírování/odstranění/přejmenování souboru/adresáře na disku. Pozor na smazání vlastních dat!

—------------------------------------------------------------

Sada č. 6

1. Třídu NacteniHodnot vyexportuje do knihovny VstupniOperace. Knihovnu použije v projektu a ověřte funkčnost použití knihovny.

Ve vámi vytvořené knihovně např. MojeMatematickaKnihovna :)

2. Vytvořte statickou třídu KvadratickaRovniceVypocty. Tato třída bude umět v metodách

- počítat f(x) pro zadanou kvadratickou rovnici,

- určit počet kořenů pro zadanou kvadratickou rovnici,

- určit hodnoty kořenů pro zadanou kvadratickou rovnici.

Kvadratická rovnice ax^2 + bx + c je dána hodnoty parametrů a, b, c. Metody přebírají tyto hodnoty prostřednictvím parametrů nebo prostřednictvím objektu KvadratickaRovnice.

Nadefinujte třídu KvadratickaRovnice, která umožňuje vytvořit objektu kvadratické rovnice s hodnotami parametrů a, b, c. Dále umožňuje získat hodnoty parametru a změnit hodnotu parametru.

—------------------------------------------------------------

Sada č. 7

K úkolu "Sachovnice"

- historie umístění [x, y]

uchování dvojice hodnot x, y: třídu Souradnice s atributy x : int, y : int, pozn: pouze parametrický konstruktor, ....

třída ToolHistoriePohybu

s metodou spoctiVzdalenost(pole s prvky typu Souradnice) : double; metrika: geometrická vzdálenost mezi body

Použité nástroje: knihovna Math

s metodu sestavVyslednouCestu(pole s prvky typu Souradnice) : String

a) použít "klasické pole": Jaké má toto řešení omezení?

b) použít ArrayList

—-----------------------------------------------------------

Sada č. 8

Úkol 1

a) Vytvořte funkci parseJmenoPrijmeni1(string):void. Funkce pro zadaný textový údaj obsahující jméno a příjmení zjistí údaj jména a údaj

příjmení. Tyto údaje uloží do samostatných proměnných a hodnoty proměnné zobrazí v apostrofech.

Předpokládá se, že jméno a příjmení je oddělen mezerou. Údaj obsahující jméno a příjmení bude do metody předán parametrem.

b) Vytvořte funkci parseJmenoPrijmeni2(string):void. Rozšiřte funkci kód funkce u1a tak, že se předpokládá, že jméno a příjmení je odděleno

libovolným počtem mezer a před jménem a za příjmením mohou být taktéž mezery v libovolném počtu.

Bude funkce (metoda) parseJmenoPrijmeni2 správně pracovat i pro jméno a příjmení zadané ve formátu specifikovaném pro úkol 1a?

c) Přepracujte řešení funkce parseJmenoPrijmeni2(string) v metodu třídy tak, že výsledky vrátí návratovou hodnotou.

Jaký bude typ návratové hodnoty. Použijte v programu.

d). Vytvořte funkci parseLogin(string):string[]. Funkce pro zadaný textový údaj obsahující loginy oddělené středníkem zjistí

jednotlivé loginy a ty vrátí v poli. Použijte klasické pole.

e1) Vytvořte funkci parseCislaInt(string):int[]. Funkce pro zadaný textový údaj obsahující celá čísla oddělená středníkem vrátí

hodnoty jednotlivých čísel.

e2) Vytvořte funkci parseCislaDouble(string):double[]. dtto