

Další dimenze – více rozměrů

Uchování složitějších dat – například stav hracího pole.

Úlohy

Manipulace s maticemi

1. Vytvořte třídu `MatrixTools`, ve které budou metody pro některé operace s maticemi a to pro:
 - a) Součet dvou matic reálných čísel.
 - b) Součin dvou matic reálných čísel.
 - c) Převod matice na normovaný tvar. Normovaný tvar vznikne z původní matice tak, že každý její prvek vydělíme absolutní hodnotou prvku matice s maximální absolutní hodnotou.
2. Zapište hlavní program pro realizaci základních operací s maticemi. Program může nabízet komunikaci s uživatelem formou řádkového menu s položkami pro jednotlivé akce: sčítání matic, násobení matic, převod matice do normovaného tvaru. Pro načtení matice a výpis matice vytvořte samostatné metody v hlavní třídě. Pro realizaci jednotlivých akcí (součet, součin, převod) použijte metody ze třídy `MatrixTools` (viz bod 2).
3. Zapište program, který bude testovat, zda zadaná čtvercová matice celých čísel je symetrická. Symetrii lze testovat podle hlavní diagonály, podle vedlejší diagonály, dle vertikální osy, horizontální osy.
4. Zapište program, který bude testovat, zda zadaná matice je stochastická. Stochastická matice je taková matice, která obsahuje pouze nezáporné prvky a zároveň součet prvků v každém řádku je roven jedné.
5. Zapište program, který bude testovat, zda zadaná čtvercová matice řádu N obsahuje všechny hodnoty $1, 2, 3, \dots, N^2$.
6. Zapište program, který bude testovat, zda součty ve všech řádcích, ve všech sloupcích a v obou diagonálách zadané čtvercové matice jsou stejné.
7. Zapište program, který bude testovat, zda zadaná matice je horní trojúhelníková.
8. Zapište program, který bude testovat, zda jednotlivé diagonální řady v matici tvoří konstantní posloupnosti.

Pexeso

Jedním ze základních algoritmů při realizaci hry „Pexeso“ je definovat počáteční rozložení karet. Vytvořme prostředky pro tyto účely.

9. Zapište a otestujte metodu, která bude generovat pole délky n s náhodnou permutací hodnot $1..n$, kde n je celé kladné číslo, jehož hodnota nepřesáhne 1000 (realizujte dvě různé verze algoritmu / dvě různé metody):
10. Zapište a otestujte metodu, která bude generovat matici pro rozložení karet pro hru „Pexeso“.
11. Zapište a otestujte metodu, která bude testovat, zda zadaná matice je maticí pro rozložení karet pro hru „Pexeso“. Matice splňuje požadovanou podmínku, právě když každá z hodnot $1, 2, \dots, N/2$, kde N je celkový počet prvků v matici, se v ní vyskytuje právě dvakrát. (Tato metoda slouží pro automatizované testování výsledků z předchozí úlohy)
12. Při „generování matice pro Pexeso“ můžeme postupovat různým způsobem. Navrhněte kritérium pro porovnání „rozházenosti“ výsledné matice. Implementujte postup v metodě. Metodu použijte pro automatizované hodnocení jednotlivých postupů generování. Vytvořte testovací program, ve kterém budete sledovat jak, dobrá je „rozházenost“ a vedle toho časová náročnost při použití různých metod generování matice.

Píškorky

Jedním z algoritmů, který je třeba implementovat při realizaci hry „Píškorky“ je zjistit, zda byla vytvořena souvislá pěťice (při zobecnění n -tice) symbolů (v řádku, ve sloupci, v některém z diagonálních směrů). Nová pěťice může vzniknout vždy pouze na místě (v okolí místa), kam byl přidán nový symbol.

13. Zapište a otestujte metodu, která na vstupu obdrží aktuální stav rozložení symbolů při hře „piškvorky“ a pozici naposledy vloženého symbolu. Metoda by tedy měla mít tři parametry: matici a dvě celá představující číslo řádku a číslo sloupce naposledy umístěného symbolu. Metoda má určit jak dlouhá je souvislá řádková n -tice, jejíž součástí je symbol, který je na zadané pozici. Metoda by měla vrátet počet (bude vždy větší než 0). Vytvořte obdobné metody pro sloupce a oba diagonální směry.
14. Realizovaný algoritmu použijme v komplexnějším programu. Zapište program, který bude v řádkovém režimu realizovat hru „piškvorky“. Vstupními parametry programu budou velikost hrací plochy n a číslo k jako délka souvislé řady stejných symbolů nutných pro ukončení hry.

Program bude pracovat v cyklu. V každém kroku hracího cyklu program zobrazí hrací plochu (textově jako čtverec sestavený ze znaků, jednotlivé symboly budou reprezentovány například znaky -, o, x; znak - představuje doposud neobsazené políčko). Dále program bude načítat od uživatele pozici (řádek, sloupec) nově vkládaného symbolu (v sudých krocích umísťujeme symbol o v lichých krocích cyklu symbol x). Po načtení pozice je nutné otestovat obsazenost příslušného pole. Pokud je políčko obsazené, opakovaně žádat o zadání správné pozice. Pokud je políčko volné, umístit na něj symbol a otestovat případné ukončení hry. Pokud není dosaženo pěti stejných symbolů, pokračovat dalším krokem hracího cyklu. Program může zároveň počítat počet tahů uskutečněných do ukončení hry.

Komunikaci programu s uživatelem můžete realizovat formou řádkového menu s položkami pro zahájení hry, pokračování rozehrané neukončené hry, ukončení programu. Po zahájení hry by potom měl program mimo jiné umožnit i její dočasné přerušení s návratem do hlavního menu.

Soustava rovnic

15. Zapište obecné prostředky (metody, například ve třídě `MatrixTools`) pro výpočet inverzní matice a soustavy lineárních rovnic Gaussovou (dopřednou i zpětnou tj. Gauss-Jordanovou) eliminační metodou.

Vytvořte hlavní program, který bude uživateli umožňovat zadávat soustavu lineárních rovnic a poskytovat jejich řešení. Uživatelské rozhraní řešte formou řádkového menu. Pro vlastní výpočty použijte dříve připravené metody.