

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Dokumentace semestrální práce

9.12.2023

Matěj Vohradský

vohry8@students.zcu.cz

Zadání úlohy.....	3
Ortogonální latinské čtverce.....	3
Výchozí podmínky.....	3
Analýza úlohy.....	3
Možné postupy.....	3
Algoritmem.....	3
Bázi dat.....	3
Pravidly.....	3
Použité řešení.....	3
Popis použitého algoritmu.....	5
Programová dokumentace.....	5
Třídy.....	5
Main.....	5
Inferenční modul.....	5
Komunikační modul.....	5
Vysvětlovací modul.....	6
Báze znalostí.....	7
Báze dat.....	7
Uživatelská dokumentace.....	8
Spuštění.....	8
Obsluha.....	8
Menu.....	8
Čtverec.....	8
Zhodnocení výsledků.....	8
Závěr.....	8

Zadání úlohy

Ortogonalní latinské čtverce

Latinský čtverec řádu n obsahuje čísla 1 až n , každé n krát, rozmístěné tak, že v žádném sloupci ani řádku se žádné číslo nevyskytuje dvakrát. Ortogonalní latinský čtverec obsahuje dvojice čísel $\langle i, j \rangle$, $1 \leq i, j \leq n$, každou jedenkrát tak, že v první i druhé složce tvoří latinský čtverec. Naprogramujte generování ortogonalního latinského čtverce daného řádu tak, aby se co nejdříve vyloučily nevyhovující možnosti (tj. ne jednoduchý backtracking, ale například dopředná kontrola).

Výchozí podmínky

S vyučujícím předmětu jsme se domluvili, že stačí, aby program uměl generovat čtverce do řádu deset.

Analýza úlohy

Možné postupy

Algoritmem

Řešení, které by tvořilo ortogonalní latinské čtverce pro všechny řády n , pro které by to šlo. Tento algoritmus je postavený na jednoduchém nastavení prvních řádků hodnot a posouvání těchto hodnot, aby vytvořené dvojice tvořily ortogonalní latinský čtverec. Tento algoritmus je velice jednoduchý pro liché řády, ale pro sudé je velice obtížné ho vymyslet.

Bázi dat

Toto řešení je velice jednoduché. Pro každý řád by v bázi dat byl vytvořen jeden ortogonalní latinský čtverec, který by se použil při vytváření nového čtverce. Nejhorší na tomto řešení je nutnost ručně zapsat ortogonalní čtverce do báze paměti. To by pro větší čtverce mohlo zanechat chybu do programu.

Pravidly

Řešení, kde se generování čtverců řídí pravidly. Pravidla by bylo dobré použít pro generování latinských čtverců, ale zaručení ortogonality by bylo mnohem složitější.

Použité řešení

Pro tuto úlohu bylo vybráno řešení algoritmem pro liché řády a pro sudé bylo použito řešení pomocí báze dat. Hlavní důvody jsou jednoduchost implementace a rychlá generace ortogonalních latinských čtverců.

Popis použitého algoritmu

Inferenční modul se stará o správný běh programu a používá komunikační modul pro získání příkazů od uživatele. Po obdržení příkazu se zkontroluje, zda je příkaz platný, a poté se ho pokusí splnit. Ke splnění použije bázi znalostí, kde jsou metody, které používá ke splnění příkazu. Pro vytvoření odpovědi pro uživatele použije vysvětlovací modul a odpoví pomocí komunikačního modulu. Čtverce ukládá do báze dat, odkud je možné je znovu načíst.

Programová dokumentace

Třídy

Main

Spouštěcí třída, zavolá metodu `spustit()` v inferenčním modulu.

Inferenční modul

Metoda `spustit()` - Obsluhuje menu pomocí zadaných příkazů, na které umí reagovat. V případě špatně zadaného příkazu tuto informaci vypíše uživateli. Po vybrání nebo vytvoření čtverce zavolá metodu `obsluhuj_ctverec(ctverec)`. Po ukončení aktualizuje bázi dat aktuálními daty.

Metoda `obsluhuj_ctverec(ctverec)` - Obsluhuje čtverec pomocí zadaných příkazů, na které umí reagovat. V případě špatně zadaného příkazu tuto informaci vypíše uživateli. Po ukončení se vrátí do metody `spustit()`.
`ctverec` - Obsluhovaný čtverec.

Komunikační modul

Metoda `vstup()` - Vyžádá uživatele o zadání zprávy. Odpověď vrátí.

Metoda `spatny_vstup()` - Vypíše zprávu o špatném formátu zprávy.

Metoda `spatny_vstup_nazev()` - Vypíše zprávu o špatném formátu nového názvu čtverce.

Metoda `vstup_novy_nazev()` - Vyžádá uživatele o nový název čtverce v případě uložení čtverce bez názvu. Vrátí odpověď.

Metoda `vstup_ulozeni()` - Vyžádá uživatele o odpověď na otázku, zda má přepsat již existující čtverec novým. Odpověď vrátí.

Metoda `odpoved(text)` - Vypíše odpověď na zadaný příkaz.
`text` - odpověď na příkaz.

Metoda `vypis_menu()` - Vypíše příkazy pro menu.

Metoda `vypis_ctverec()` - Vypíše příkazy pro čtverec.

Metoda `vypis_uvod()` - Vypíše úvod do menu.

Metoda `vypis_uvod_ctverec()` - Vypíše úvod do práce se čtvercem.

Metoda `vypis_konec()` - Vypíše konec a bude čekat na potvrzení.

Vysvětlovací modul

Metoda `ctverce()` - Připraví zprávu o názvech uložených čtverců.

Metoda `ctverec_odstranen(nazev)` - Připraví zprávu o odstranění čtverce.

`nazev` - Název odstraněného čtverce.

Metoda `spatna_velikost(velikost)` - Připraví zprávu o špatně zadané velikosti.

`velikost` - Špatně zadaná velikost.

Metoda `spatny_nazev(nazev)` - Připraví zprávu o špatně zadaném názvu.

`nazev` - Špatně zadaný název.

Metoda `vytvor_ctverec(ctverec)` - Připraví čtverec na vypsání.

`ctverec` - Čtverec.

Metoda `nazev(nazev)` - Připraví zprávu o názvu čtverce.

`nazev` - Název čtverce.

Metoda `velikost(velikost)` - Připraví zprávu o velikosti čtverce.

`velikost` - Velikost čtverce.

Metoda `zmena_nazvu(nazev)` - Připraví zprávu o změně názvu.

`nazev` - Nový název čtverce.

Metoda `ctverec_ulozen(nazev)` - Připraví zprávu o uložení čtverce.

`nazev` - Název, pod kterým byl čtverec uložen.

Metoda `vytvor_data()` - Vytvoří nové data pro zapsání do báze dat.

Metoda `akce_ukoncena()` - Připraví zprávu o ukončení akce.

Metoda `otazka_ano_ne()` - Připraví zprávu o špatné odpovědi na otázku ano/ne.

Metoda `nazev_nezadan()` - Připraví zprávu o špatně zadaném názvu.

Metoda `sloupce_prohozeny(sloupec1, sloupec2)` - Připraví zprávu o prohození sloupců.

`sloupec1` - Číslo sloupce.

`sloupec2` - Číslo sloupce.

Metoda `radky_prohozeny(radek1, radek2)` - Připraví zprávu o prohození řádků.
`radek1` - Číslo řádku.
`radek2` - Číslo řádku.

Metoda `spatna_velikost_ctverec(velikost)` - Připraví zprávu o špatně zadané velikosti čtverce.
`velikost` - Velikost čtverce.

Báze znalostí

Metoda `vytvor_ctverec(velikost)` - Vytvoří nový čtverec o zadané velikosti.
`velikost` - Velikost nového čtverce.

Metoda `kontrola_vstupu_cislo(velikost)` - Kontroluje velikost. Vrací True/False.
`velikost` - Zadaná hodnota.

Metoda `kontrola_nazvu(nazev)` - Kontroluje název. Vrací True/False.
`nazev` - Zadaná hodnota.

Metoda `kontrola_vstupu_2cisla(cislo1, cislo2, velikost)` - Kontroluje vstup dvou velikostí.
`cislo1` - Zadaná hodnota.
`cislo2` - Zadaná hodnota.
`velikost` - Velikost Čtverce.

Metoda `prohod_sloupce(sloupec1, sloupec2, ctverec)` - Prohodí sloupce.
`sloupec1` - Číslo sloupce.
`sloupec2` - Číslo sloupce.
`ctverec` - Čtverec.

Metoda `prohod_radky(radek1, radek2, ctverec)` - Prohodí řádky.
`radek1` - Číslo řádku.
`radek2` - Číslo řádku.
`ctverec` - Čtverec.

Báze dat

Třída pro uchování dat. Data jsou uchovávána ve struktuře `ctverce {}` ve tvaru
`název: [velikost, [ctverec]]`.

Další struktura `sude {}` použitá pro vytváření sudých čtverců.

`název: [velikost, [ctverec]]`.

Báze dat je vždy po ukončení programu aktualizována podle práce, kterou uživatel vykonal.

Uživatelská dokumentace

Spuštění

Program se spustí spuštěním souboru main.py.

Obsluha

Menu

S menu se pracuje předepsanými příkazy:

Seznam dostupných zpráv		volání zpráv
Vytvořit nový čtverec		1 <velikost>
Vypsát názvy existujících čtverců		2
Načte existující čtverec		3 <název_čtverce>
Odstranit existující čtverec		4 <název_čtverce>
Ukončit program		k

Například pro vytvoření nového čtverce o velikosti 5 zadáte:

1 5

Čtverec

Se čtvercem se pracuje předepsanými příkazy:

Seznam dostupných zpráv		volání zpráv
Vypsát název čtverce		1
Změnit název čtverce		2 <nový_název>
Vypsát velikost čtverce		3
Prohodit sloupce		4 <Číslo_sloupce> <Číslo_sloupce>
Prohodit řádky		5 <Číslo_řádku> <Číslo_řádku>
Uložit čtverec		6
Ukončit režim čtverce		k

Například pro změnu názvu čtverce zadáte:

2 Čtverec_1

Zhodnocení výsledků

Program pracuje pro zadané podmínky dobře. Výhodou je spolehlivost a rychlost.

Nevýhodou je řešení pro sudé řády čtverce, které nedoporučuji použít pro čtverce vyšších řádů. Řešení by bylo sice rychlejší, ale mohlo by dojít k chybě při zadávání.

Závěr

Zadání bylo splněno. Program by mohl být rozšířen o algoritmus, který by byl schopen generovat i čtverce vyšších sudých řádů.