

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

Факультет прикладної математики

Кафедра системного програмування і спеціальних комп’ютерних систем

**Розрахунково-графічна робота**

з дисципліни “ Програмування 2. Складні структури ”

**Варіант № 6**

Виконав:

студент 1 курсу ФПМ

групи КВ-41:

Горпинич-Радуженко

Іван Олександрович

**Перевірено:**

Київ 2015

***Постановка задачі***

Створити ігрову програму мовою прогрмування Паскаль.

Розробка і реалізація ігрових програм має вестися з врахуванням графічних та звукових можливостей, що надаються конкретним комп’ютером.

Програма мусить коректно розв’язувати поставлену задачу. Логічно відокремлені частини алгоритма реалізувати за допомогою окремих процедур або функцій.

Також потрібно передбачити та забезпечити виконання всіх можливих розгалужень алгоритма, тобто програма повинна коректно реагувати на будь-які можливі ситуації (наприклад, виникнення помилкових ситуацій, перевірка файлів на порожність, правильність введених з клавіатури значень і т.д.). Передбачити взаємодію з користувачем (наприклад, можливість виводу правил гри, допомоги), таймер, лічильник числа ходів.

***Завдання за варіантом***

«15». На квадратному полі розміром 4\*4 за допомогою генератора випадкових чисел розставлені 15 фішек з номерами від 1 до 15. Є одна вільна позиція. Розставити фішки по зростанням їх номерів по вертикалі. Пересувати фішки можна тільки на сусідню вільну позицію.

***Описання структури програми***

«П'ятнашки» на перший погляд проста гра, але для її реалізації необхідно звернеться практично до всіх розділів програмування в середовищі Турбо Паскаль.

Суть самої гри полягає в наступному:

* Гравець на екрані бачить табло, яке розбите на 16 клітин. У п'ятнадцяти з них розташовані неповторювані цифри, у випадковому порядку від 1 до 15 і одна порожня.

У загальному вигляді дане табло можна представити у вигляді таблиці 1:

Таблиця 1 - Зразок табло.

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | 7 | 3 | 8 |
| 15 | 1 | 13 | 2 |
| 14 | 10 | 6 | 4 |
|  | 9 | 11 | 12 |

* Гравець повинен переміщати по однієї клітини з цифрою на порожнє місце.
* Так відбувається до тих пір, поки користувач не вибудує послідовну комбінацію цифр (Таблиця 2), і лише після цього гравець вважається переможцем.

Таблиця 2 - Правильне заповнення табло.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 5 | 9 | 13 |
| 2 | 6 | 10 | 14 |
| 3 | 7 | 11 | 15 |
| 4 | 8 | 12 |  |

Програма виконана в середовищі програмування Турбо Паскаль. Сама програма - додаток Windows. Для виведення графічних об'єктів використовуються модуль Graph. Програмою обробляються події від натискання клавіш на клавіатурі, вибору відповідних пунктів меню.

Уся програма розбита на процедури, кожна з яких відповідає за певну дію. Всю програму можна розділити на два основні розділи:

* Основні процедури, тобто ті процедури, які відповідають за саму гру і взаємодія з користувачем під час гри;
* Допоміжні процедури. Це ті процедури, які доповнюють дане додаток додатковими і дуже корисними функціями;
* Основна програма.

До основних процедур відносяться:

* Tablo;
* Output;
* direction;
* Search;
* repl;
* check;
* Game15;

До допоміжних процедур входить:

* help;
* autor;
* cheat.

**Основні процедури**

**procedure Tablo;**

Дана процедура формує табло, заповнене випадковими, неповторюванними цифрами від 1 до 15 і одним порожнім кліткою. Процедура реалізована з використанням двох масивів: одномірний і двовимірний.

Одновимірний масив з 16 цілими числами заповнюється випадковим чином, причому, враховується, щоб цифри в даному масиві не повторювалися.

Кожній цифрі з одновимірного масиву присвоюється, аналогічний їй строковий елемент і вводиться в двовимірний масив. Наприклад, цифрі 5, присвоюється строковий елемент '5'. Виняток становить цифра 16, їй присвоюється пробіл.

**procedure Output;**

Процедура виведення на екран табло з цифрами сформоване на момент відображення. Якщо програма тільки занедбана, тоді на екран виводиться таблиця заповнене випадковим чином. Якщо гра вже йде то дана процедура виводить на екран ту комбінацію цифр, яка визначена користувачем під час гри.

**procedure direction;**

У даній процедурі користувачеві, за допомогою курсору, пропонується ввести напрямок переходу. У даній процедурі зчитується код натиснутої клавіші, щоб надалі можна було здійснювати пересування.

**procedure Search;**

У цій процедурі здійснюється пошук порожнього елемента. Це необхідно для того, щоб надалі користувач зміг щодо порожнього елемента зробити свій хід. Процедура зчитує кожен елемент двовимірного масиву і порівнює його з порожнім. Після того як порожній знайдений процедура запам'ятовує координати порожнього елемента, а саме рядок і стовпець.

**procedure repl;**

Програма залежно від вибору напряму здійснює перестановку елементів в двовимірному масиві.

Також в даній процедурі зчитується кожен хід користувача, щоб надалі, якщо користувач розкладе необхідний розклад швидше, ніж гравці, які грали раніше, можна було ввести інформацію про гравця в файл.

**procedure check;**

Після скоєння перестановки цифр, а іншими словами після чергового ходу, програма порівнює розстановку цифр в двовимірної матриці. Якщо поточна комбінація є правильною, тоді гравець вважається переможцем.

**procedure Game15;**

Ця процедура є основною. У неї підключається графічний модуль і відбувається основний процес гри.

Гра буде тривати до тих пір, поки не буде розкладений правильний розклад або до тих пір поки користувач не перерве гру за допомогою клавіші ESC.

**Допоміжні процедури**

**procedure help;**

Виводить на екран інформацію про правила гри.

**procedure autor;**

Виводить на екран інформацію про програму.

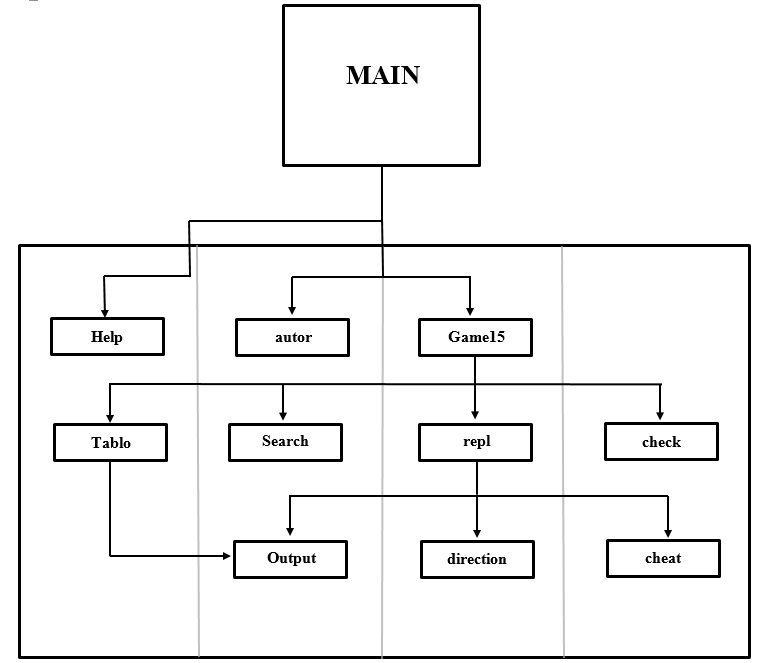
**procedure cheat;**

Маленька хитрість даної програми. Вона призначена для розробника програми і для викладача перевіряючого правильність написання гри. Щоб, при перевірці, викладачеві не була потрібна проходити всю гру, досить натиснути клавішу Кінець на клавіатурі і розклад майже разолжется. Також ця процедура була б дуже корисна розробнику, для налагодження і тестування програми.

**Основна програма**

В основній програмі на екрані виводиться Головне меню і залежно від обраних користувачем дій відбувається взаємодія з процедурами.

**Схема викликів процедур та функцій**



***Текст Програми***

**program** DKR\_Programming;

**uses**

crt, Graph;

**var**

arr: **array**[1..4, 1..4] **of** string;{Двовимірний масив, містить елементи табло}

vect: **array**[1..16] **of** integer; {Масив для заповнення випадковими числами}

men: **array**[1..5] **of** integer; {Масив виводу елементів Головного меню}

i, j: integer; {Змінні для роботи з масивами}

mline, mcolumn: integer; {Координати порожнього елемента}

slc: integer; {Змінна для роботи з Головним меню}

ch: char; {Змінна, якій присвоюється код клавіші клавіші на клавіатурі}

chk: boolean; {Перевірка правильності розкладу}

f: text; {Файлова змінна}

**procedure** Output; {Процедура виведення на екран табло з цифрами сформоване на момент відображення}

**var**

lx, ly: integer; {Координати виведення двомірного масиву}

x, y: integer; {Координати клітин}

j1, i1: integer; {Змінні лічильники, для малювання клітин}

w1, h1: integer; {Ширина і висота клітин}

**begin**

OutTextXY(210, 50, 'For leaving press ENTER');

w1 := 30;

h1 := 30; {Клітка розміром 30 на 30}

**for** i1 := 0 **to** 3 **do** {Цикл, промальовування клітин}

**for** j1 := 0 **to** 3 **do**

**begin**

x := 235 + j1 \* 35; {Зсув клітин по х}

y := 150 + i1 \* 35; {Зсув клітин по у}

setFillStyle(1, 4); {Колір і стиль клітин, колір синій, стиль заповнення поточному кольором}

Bar(x, y, x + w1, y + h1); {Малювання клітини}

**end**;

lx := 245;

ly := 162;

**for** i := 1 **to** 4 **do** {Цикл виведення двомірного масиву по вгору клітин}

**begin**

**for** j := 1 **to** 4 **do**

**begin**

OutTextXY(lx, ly, arr[i, j]); {Виведення тексту на екран}

lx := lx + 35;

**end**;

lx := 245;

ly := ly + 35;

**end**;

line(220, 135, 220, 300); {Малювання рамки}

line(385, 135, 385, 300);

line(220, 135, 385, 135);

line(220, 300, 385, 300);

**end**;

**procedure** Tablo; {Формування табло при першому запуску заповнене випадковими і неповторяющимися цифрами}

**var**

b: integer; {Змінна, якій присвоюється випадкове число}

k, z: integer; {Лічильники для операцій з масивами}

**begin**

randomize;

**for** z := 1 **to** 16 **do**

**begin**

b := random(15); {Вибір випадкового числа}

k := 1;

**while** k <> 17 **do** {Цикл поки не буде заповнений масив з цілими цифрами}

**begin**

**if** vect[k] = b **then**

**begin**

b := random(17);

k := 1;

**end**

**else** k := k + 1;

**end**;

vect[z] := b; {Присвоєння чергового неповторюваного елемента масиву}

**end**;

z := 1;

**for** i := 1 **to** 4 **do** {Заповнення двомірного масиву, замість цифр з одновимірного, присвоюються рядкові елементи}

**begin**

**for** j := 1 **to** 4 **do**

**begin**

**case** vect[z] **of**

1: arr[i, j] := '1 ';

2: arr[i, j] := '2 ';

3: arr[i, j] := '3 ';

4: arr[i, j] := '4 ';

5: arr[i, j] := '5 ';

6: arr[i, j] := '6 ';

7: arr[i, j] := '7 ';

8: arr[i, j] := '8 ';

9: arr[i, j] := '9 ';

10: arr[i, j] := '10';

11: arr[i, j] := '11';

12: arr[i, j] := '12';

13: arr[i, j] := '13';

14: arr[i, j] := '14';

15: arr[i, j] := '15';

16: arr[i, j] := ' ';

**end**;

z := z + 1;

**end**;

**end**;

Output; {Вивід табло на екран}

**end**;

**procedure** Search; {Пошук пустого елемента в табло}

**begin**

**for** i := 1 **to** 4 **do**

**begin**

**for** j := 1 **to** 4 **do**

**begin**

**if** arr[i, j] = ' ' **Then** {Пошук, дорівнює чи поточний елемент пробілу}

**begin**

mline := i; {Якщо дорівнює, то присвоюються координати порожнього елемента}

mcolumn := J

**end**;

**end**;

**end**;

**end**;

**procedure** cheat; {Бонус, для перевірки. При натисканні клавіші END на клавіатурі розклад збирається}

**begin**

arr[1, 1] := '1 ';arr[1, 2] := '5 ';arr[1, 3] := '9 ';arr[1, 4] := '13 ';

arr[2, 1] := '2 ';arr[2, 2] := '6 ';arr[2, 3] := '10 ';arr[2, 4] := '14 ';

arr[3, 1] := '3 ';arr[3, 2] := '7';arr[3, 3] := '11';arr[3, 4] := '15';

arr[4, 1] := '4';arr[4, 2] := '8';arr[4, 4] := '12';arr[4, 3] := ' ';

mline := 4;mcolumn := 3;

**end**;

**procedure** direction; {Введення напрямку переходу}

**begin**

ch := readkey; {Перемінної присвоюється код клавіші користувачем клавіші на клавіатурі}

**end**;

**procedure** repl; {Пересування клітин з цифрами в залежності від вибору користувача}

**begin**

direction;{Процедура, введення напрямки переходу}

**if** ord(ch) = 79 **then** cheat; {Якщо натиснута клавіша END на клавіатурі то розклад сам збирається}

**if** ord(ch) = 75 **then** {Якщо натиснута клавіша вліво}

**begin**

**if** mcolumn <> 4 **then** {Якщо це не останній елемент, що стоїть біля кордону табло}

**begin**

arr[mline, mcolumn] := arr[mline, mcolumn + 1];

arr[mline, mcolumn + 1] := ' ';

mcolumn := mcolumn + 1;

**end**;

**end**;

**if** ord(ch) = 72 **then** {Якщо натиснута клавіша вгору}

**begin**

**if** mline <> 4 **then** {Якщо це не останній елемент, що стоїть біля кордону табло}

**begin**

arr[mline, mcolumn] := arr[mline + 1, mcolumn];

arr[mline + 1, mcolumn] := ' ';

mline := mline + 1;

**end**;

**end**;

**if** ord(ch) = 77 **then** {Якщо натиснута клавіша вправо}

**begin**

**if** mcolumn <> 1 **then** {Якщо це не останній елемент, що стоїть біля кордону табло}

**begin**

arr[mline, mcolumn] := arr[mline, mcolumn - 1];

arr[mline, mcolumn - 1] := ' ';

mcolumn := mcolumn - 1;

**end**;

**end**;

**if** ord(ch) = 80 **then** {Якщо натиснута клавіша вниз}

**begin**

**if** mline <> 1 **then** {Якщо це не останній елемент, що стоїть біля кордону табло}

**begin**

arr[mline, mcolumn] := arr[mline - 1, mcolumn];

arr[mline - 1, mcolumn] := ' ';

mline := mline - 1;

**end**;

**end**;

Output;

**end**;

**procedure** check; {Перевірка правильно розкладено табло}

**var** sd: integer;

**begin**

chk := false;

**if** (arr[1, 1] = '1 ') **and** (arr[1, 2] = '5 ') **and** (arr[1, 3] = '9 ') **and** (arr[1, 4] = '13 ')

**and** (arr[2, 1] = '2 ') **and** (arr[2, 2] = '6 ') **and** (arr[2, 3] = '10 ') **and** (arr[2, 4] = '14 ')

**and** (arr[3, 1] = '3 ') **and** (arr[3, 2] = '7') **and** (arr[3, 3] = '11') **and** (arr[3, 4] = '15')

**and** (arr[4, 1] = '4') **and** (arr[4, 2] = '8') **and** (arr[4, 3] = '12') **and** (arr[4, 4] = ' ')

**then**

**begin**

chk := true;

OutTextXY(245, 100, 'Congratulate!!!');

OutTextXY(245, 330, 'You have won =)');

sd:=100;

**repeat**

sound(sd);

delay(10);

nosound;

inc(sd,10);

**until** (sd>1800);

readln;

**end**;

**end**;

**procedure** Game15;

**var**

grMode: integer; {Режим роботи відеосистеми}

grPath: string; {Шлях до файлу}

grDriver: integer; {Використовуваний програмою драйвер відеоадаптера}

**begin** {Підключення графіки і перехід в режим ІГРИ}

grDriver := VGA;

grmode := 2;

grPath := 'EGAVGA.BGI';

initGraph(grDriver, grMode, grPath); {Ініціалізація графічного режиму}

Tablo; {Формування табло}

Search; {Пошук чистого елемента}

**repeat**

repl; {Пересування в масиві}

check; {Перевірка чи є даний розклад вірним}

**until** (ord(ch) = 13) **or** (chk = true);

closeGraph; {Закриття графічного режиму}

**end**;

**procedure** help; {Перехід в режим довідки}

**var**

f: text;

g1: string;

**begin**

clrscr;

assign(f, 'fhelp.txt');

reset(f);

**while not**(eof(f)) **do begin**

readln(f, g1); writeln(g1); **end**;

writeln;

writeln('For leaving press ENTER');

readln;

close(f);

**end**;

**procedure** autor; {Вивід загальної інформації на екран в розділ опис}

**var**

f: text;

g1: string;

**begin**

clrscr;

assign(f, 'fhelp1.txt');

reset(f);

**while not**(eof(f)) **do begin**

readln(f, g1); writeln(g1); **end**;

writeln;

writeln('For exit press ENTER');

readln;

close(f);

**end**;

**begin**

{Основна програма}

{Виведення на екран головного меню}

{Елементи Головного меню, один з яких зафарбований білим кольором, а решта червоним}

randomize;

men[1] := 15;

men[2] := 12;

men[3] := 12;

men[4] := 12;

**repeat**

clrscr;

mline := 1; {Поточний рядок}

slc := 1;

GoToXY(32, 10);Textcolor(men[1]);writeln('Key');

GoToXY(32, 11);Textcolor(men[2]);writeln('About program');

GoToXY(32, 12);Textcolor(men[3]);writeln('Play');

GoToXY(32, 13);Textcolor(men[4]);writeln('Exit');

ch := readkey; {Вибір напрямку пересування елементів меню}

**if** (ord(ch) = 80) **then** {Якщо вниз тоді поточний стає білим, а нижній стає червоним}

**begin**

**for** i := 1 **to** 4 **do**

**begin**

**if** (men[i] = 15) **and** (mline <> 4) **then**

**begin**

men[mline] := 12;

men[mline + 1] := 15;

**end**

**else** mline := mline + 1;

**end**;

**end**;

**if** ord(ch) = 72 **then** {Якщо вгору, то поточний білим, а верхній червоним}

**begin**

**for** i := 1 **to** 4 **do**

**begin**

**if** (men[i] = 15) **and** (mline <> 1) **then**

**begin**

men[mline] := 12;

men[mline - 1] := 15;

**end**

**else** mline := mline + 1;

**end**;

**end**;

**if** ord(ch) = 13 **then** {Якщо натиснуто ENTER}

**begin**

**for** i := 1 **to** 4 **do**

**begin**

**if** men[i] = 15 **then**

**begin**

**if** slc = 1 **then begin** Help;**break**; **end**; {Перехід в режим довідки}

**if** slc = 2 **then begin** autor;**break**; **end**; {Перехід в режим довідки}

**if** slc = 3 **then begin** Game15;**break**; **end**; {Перехід в режим гри}

**end**

**else** slc := slc + 1;

**end**;

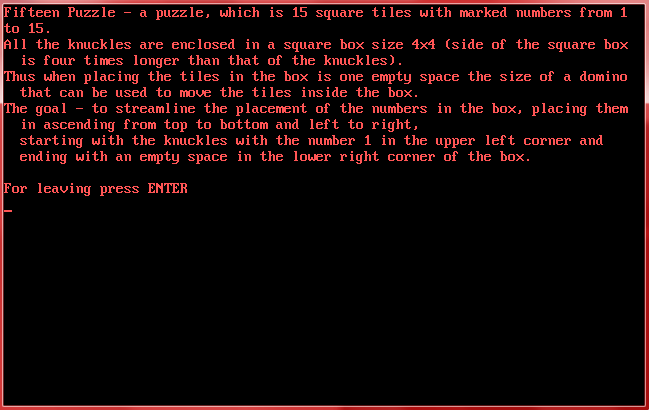
**end**;

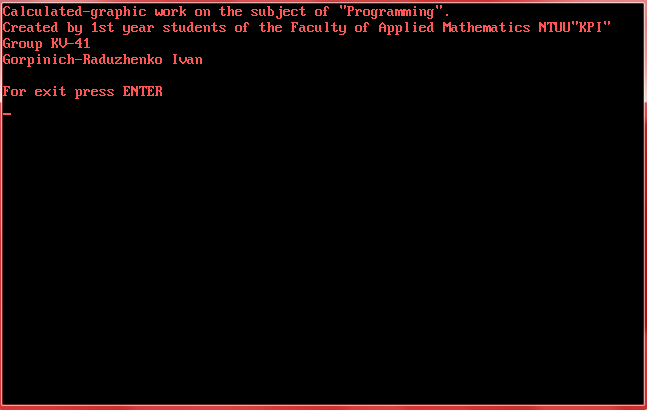
**until** slc = 4 {До тих пір поки не натиснуто пункт EXIT}

**end**.

***Тест Програми***

******

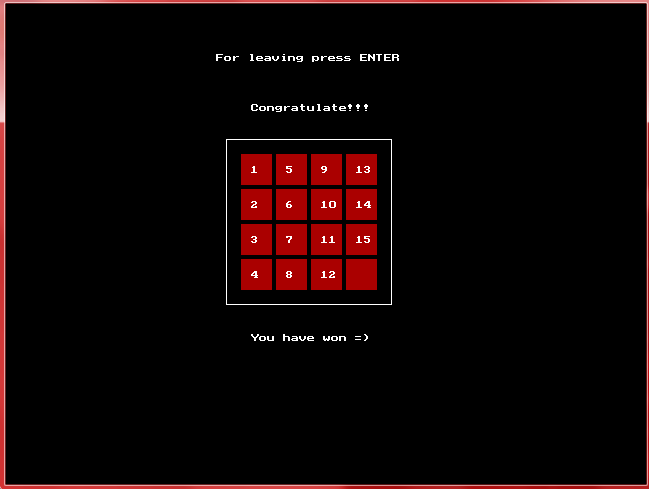
******

******

******

******

******

******