

Zaimplementujemy prostą planszę szachową (założmy, że dowolnego rozmiaru podawanego jako argument uruchomienia programu) na której wstawimy pionki, które będą mogły poruszać się tylko w górę o 1 pole.

Założmy, że plansza ma postać macierzy kwadratowej NxN, gdzie pole 0 oznacza, że nie znajduje się na nim pionek, zaś 1, że na danym polu stoi pionek. Ruch pionka oznacza, że pole na którym stał pionek zmienia się z 1 na 0, zaś to na które pionek się przemieści zmienia się z 0 na 1. Założmy, że nasza domyślna plansza będzie następującej postaci:

```
Rozmiar planszy: 8
  0 1 2 3 4 5 6 7
-----
0| 0 0 0 0 0 0 0 0
1| 0 0 0 0 0 0 0 0
2| 0 0 0 0 0 0 0 0
3| 0 0 0 0 0 0 0 0
4| 0 0 0 0 0 0 0 0
5| 0 0 0 0 0 0 0 0
6| 0 1 0 1 0 1 0 1
7| 1 0 1 0 1 0 1 0
```

Tworzymy klasę ChessPiece (figura szachowa).

Klasa powinna zawierać następujące pola:

```
static int PieceCounter; //licznik ilosci figur, domyslnie 0
```

```
int fX, fY; //pozycja figury na planszy
```

konstruktor:

- przyjmujący trzy parametry ustawiające pole fName, fX oraz fY. Zwiększa również licznik ilości figur.

destruktor:

- zmniejsza liczniki figur oraz pionków

oraz:

- void SetPosition(int x, int y); //ustawia pozycje figury
- int GetPositionX(); //zwraca pozycje x
- int GetPositionY(); //zwraca pozycje y
- void Move(int **tab, int size); //Przyjmuje plansze i jej rozmiar, przesuwa pionek w gore planszy o 1 (z wiersza o większym numerze na wiersz o mniejszym numerze)

Poniżej znajduje się zawartość pliku **main.cpp**. W pliku tym będą znajdować się również 3 funkcje globalne.

```
void PrintTable(int **tab, int size)
{
    //Funkcja wypisuje tablice na ekran
}

void SetPiece(ChessPiece **pieceTab, int **tab, int size)
{
    /* Metoda SetPiece przyjmuje przyjmuje tablice 2D, jej rozmiar oraz tablice
    wskaznikow na pionki. Jej zadaniem jest stworzenie obiektu typu ChessPiece w
    miejscu gdzie na planszy znajduje się 1. */
}

int main()
{
    /*
    1) Tworzymy plansze-tablice 2D
    2) Następnie wypisujemy wczytana plansze na ekran, PrintTable 2p.
    3) Tworzymy tablice wskaznikow na pionki, np. ChessPiece *pieceTab[100];
    a następnie ustawiamy pionki za pomoca funkcji SetPiece 2p.
    4) Przesuwamy pionek pieceTab[1] oraz pieceTab[4] metoda Move 0.5p.
    5) Ponownie wypisujemy plansze na ekran
    6) Usuwamy plansze-tablice 2D oraz tablice wskaznikow na obiekt ChessPiece
    0.5p
    */
```

```

    return 1;
}

```

Przykładowy wynik działania programu (**pogrubionym** zaznaczono pionki 1 oraz 4, które zostały przesunięte):

```

Rozmiar planszy: 8
  0 1 2 3 4 5 6 7
-----
0| 0 0 0 0 0 0 0 0
1| 0 0 0 0 0 0 0 0
2| 0 0 0 0 0 0 0 0
3| 0 0 0 0 0 0 0 0
4| 0 0 0 0 0 0 0 0
5| 0 0 0 0 0 0 0 0
6| 0 1 0 1 0 1 0 1
7| 1 0 1 0 1 0 1 0

```

Wynik po ruszeniu dwóch pionków

```

Rozmiar planszy: 8
  0 1 2 3 4 5 6 7
-----
0| 0 0 0 0 0 0 0 0
1| 0 0 0 0 0 0 0 0
2| 0 0 0 0 0 0 0 0
3| 0 0 0 0 0 0 0 0
4| 0 0 0 0 0 0 0 0
5| 0 0 0 1 0 0 0 0
6| 1 1 0 0 0 1 0 1
7| 0 0 1 0 1 0 1 0

```

Uwaga!

1. Sposób alokowania pamięci tablicy dwuwymiarowej (dla tablicy typu `int`):

```

int rozmiar = 100;
int **tab = new int*[rozmiar];
for(int i=0; i<rozmiar; i++)
    tab[i] = new int[rozmiar];

```

2. Sposób zwalniania pamięci tablicy dwuwymiarowej:

```

for(int i=0; i<rozmiar; i++)
    delete[] tab[i];
delete[] tab;

```