

Задача 6. (10 т.) Задачата да се реши на езика C++ или Java. В началото на вашето решение посочете кой език сте избрали.

Нека GameBoard е предварително дефинирана квадратна матрица от цели числа с размери $N \times N$, представляваща игрова дъска. Всеки елемент в матрицата има стойност 0 („земя“), 1 („огън“) или 2 („вода“). За две позиции в матрицата (i, j) и (i', j') казваме, че са съседни, ако $|i - i'| \leq 1$ и $|j - j'| \leq 1$.

А) Да се дефинира структура Point, описваща позиция на игровата дъска. Да се дефинира абстрактен клас (или интерфейс) GamePlayer, който описва играч на игровата дъска със следните операции:

- `getPosition()` – Връща позицията на играча на дъската;
- `allowedMoves()` – Връща списък (колекция) с всички възможни позиции, до които играчът може да достигне с един ход.

Б-1) Да се дефинира клас Knight, наследник на GamePlayer, описващ „сухопътен рицар“. Рицарят може да се придвижва само в такава съседна позиция, която е „земя“ и не е в съседство с „огън“. Пример за достижими позиции за рицаря К е показан на диаграмата вдясно.

1	0	1
0	К	2
0	0	2

Б-2) Да се дефинира клас SeaMonster, наследник на GamePlayer, описващ „морско чудовище“. Морското чудовище може да се придвижва с произволен брой позиции по хоризонтала или по вертикала, но само по „вода“. Пример за достижими позиции за чудовището S е показан на диаграмата вдясно.

1	1	0	2	0
0	2	1	0	2
2	S	2	2	1
1	1	2	2	0
2	2	1	1	1

В) „Война“ наричаме такава подредба на играчите по дъската, при която на някоя от съседните позиции на всеки играч има друг играч. Да се дефинира функция

`allMoves ([подходящ тип] players[, ...])`

която по даден списък (колекция) `players`, съдържащ произволен брой разнородни играчи, извежда на стандартния изход всеки възможен ход на играч от `players` такъв, че след изпълнението му списъкът с играчи да описва война. Информацията за ходовете да съдържа типа на играча, старата позиция и новата позиция.

Пример:

`Knight (0,0) -> (1,1)`

`SeaMonster (2,2) -> (5,2)`

Забележка: реализирайте всички конструктори и други операции, които смятате, че са необходими на съответните класове.