Задача. Створити об’єкт «Тир». У масиві зберігаються 1, якщо у цьому квадраті є заєць і 0 в іншому випадку.

|  |  |
| --- | --- |
| Тир |  |
| Поля(властивості) | Масив, у якому зберігається поле з зайцями |
| Методи (дії) | Метод пострілу (задається позиція пострілу)  Виведення ігрового поля |

#pragma once

class Tir

{

private:

//Опис полів

int rabbitsCount;

int bulletsCount;

int positionsCount;

int\* gameField;

//Опис методів доступу

void set\_rabbitsCount(int value);

void set\_bulletsCount(int value);

void set\_positionsCount(int value);

public:

//Конструктори

Tir(int rabbitsCount, int bulletsCount, int positionsCount);

Tir();

//Інші методи

bool shoot(int shootPosition);

bool play(); //true - виграш, false -програш

void printGameField();

~Tir();

};

#include "stdafx.h"

#include "Tir.h"

#include <stdlib.h>

#include <iostream>

void Tir::set\_rabbitsCount(int value)

{

if (value > positionsCount)

throw "Error! Rabbits count > positions count";

rabbitsCount = value;

//------- Очищаємо ігрове поле (всюди 0)

for (int i = 0; i < positionsCount; i++)

{

gameField[i] = 0;

}

//------- Випадковим чином розміщуємо зайців

for (int i = 0; i < rabbitsCount; i++)

{

int randomIndex;

do

{ //Випадковим чином вибираємо індекс комірки для розміщення зайця

randomIndex = rand() % positionsCount;

} while (gameField[randomIndex]==1);

gameField[randomIndex] = 1; //Розміщуємо зайця

}

//Підхід №2 для розміщення зайців

//int\* freeCellNumbers = new int[positionsCount];

//for (int i = 0; i < positionsCount; i++)

//{

// freeCellNumbers[i] = i;

//}

//int freePositionsCount = positionsCount;

//for (int i = 0; i < rabbitsCount; i++)

//{

// int randomIndex = rand() % freePositionsCount; //Вибираємо номер коробки

// gameField[freeCellNumbers[randomIndex]] = 1; //freeCellNumbers[randomIndex] - номер вільної кімнати

// freeCellNumbers[randomIndex] = freeCellNumbers[freePositionsCount - 1]; //Коробку з кінця ставимо у позицію randomIndex

// freePositionsCount--; //Кількість коробок зменшується на 1

//}

//delete[] freeCellNumbers;

}

void Tir::set\_bulletsCount(int value)

{

if (value < rabbitsCount)

throw "error";

bulletsCount = value;

}

void Tir::set\_positionsCount(int value)

{

if (value < 5)

throw "error (<5)";

positionsCount = value;

gameField = new int[positionsCount];

}

Tir::Tir(int rabbitsCount, int bulletsCount, int positionsCount)

{

set\_positionsCount(positionsCount);

set\_rabbitsCount(rabbitsCount);

set\_bulletsCount(bulletsCount);

}

Tir::Tir():Tir(5,7,10)

{

}

bool Tir::shoot(int shootPosition)

{

if (gameField[shootPosition] == 1)

{

gameField[shootPosition] = -1;

return true;

}

else

return false;

}

bool Tir::play()

{

do

{

printf("--- Bullets count: %d, Rabbits count%d ---\n",

bulletsCount, rabbitsCount);

printGameField();

printf("Shoot position: ");

int shootPosition;

std::cin >> shootPosition;

bulletsCount--;

if (shoot(shootPosition))

{

rabbitsCount--;

printf("OK\n");

}

else

printf("Bad \n");

} while (bulletsCount>0 && rabbitsCount>0);

return rabbitsCount==0;

}

void Tir::printGameField()

{

for (int i = 0; i < positionsCount; i++)

{

switch (gameField[i])

{

case 0:printf("|%3c%c%3c",' ',' ',' ');

break;

case 1:printf("|%3c%c%3c", ' ', '\*', ' ');

break;

case -1:printf("|%3c%c%3c", ' ', '-', ' ');

break;

}

}

printf("|\n");

}

Tir::~Tir()

{

delete[] gameField;

}

// ConsoleApplication30.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include "Tir.h"

#include <iostream>

using namespace std;

int main()

{

Tir\* tir = new Tir(3, 5, 5);

if (tir->play())

printf("Won");

else

printf("Lose");

system("pause");

return 0;

}