```
#include <iostream>
#include <vector>
#include <thread>
#include <random>

using namespace std;

/// <summary>
/// Класс ячейки
/// У каждой ячейки своя ценность
/// </summary>
class Cell
{
public:
	Cell(){
		cost = rand() % 10000;
		is_alive = true;
	}
	int get_cost(){
		return cost;
	}
	bool get_is_alive(){
		return is_alive;
	}
	void set_is_alive() {
		is_alive = false;
	}
private:
	int cost;
	bool is_alive;
};
/// <summary>
```

```
/// Класс поля страны
/// </summary>
class Field
{
public:
	Field(int high, int wight, int cost){
		sum_cost = 0;
		spended_money = 0;
		cost_of_ammo = cost;
		this->high = high;
		this->wight = wight;
		field.resize(high);
		for (int i = 0; i < high; i++){
			field[i].resize(wight);
			for (int j = 0; j < wight; j++){
				field[i][j] = Cell();
				sum_cost += field[i][j].get_cost();
			}
		}
	}

	int get_high(){
		return  high;
	}

	int get_widht(){
		return wight;
	}

	int get_sum_cost(){
		return  sum_cost;
	}
```

```
        void take_gamage(int x, int y){
                field[x][y].set_is_alive();
        }

        bool spend(int max){
                spended_money += cost_of_ammo;
                return (max > spended_money);
        }

        bool destroyed(){
                for (int i = 0; i < high; i++){
                        for (int j = 0; j < wight; j++){
                                if (field[i][j].get_is_alive())
                                        return false;
                        }
                }
                return true;
        }
private:
        vector<vector<Cell>> field;
        int cost_of_ammo;
        int spended_money;
        int sum_cost;
        int wight, high;
};

/// <summary>
/// Метод атаки, выполняется пока соблюдаются условия
/// Ведения огня, если кто-то победил, то программа завершается сразу
/// С информирующим сообщением
/// </summary>
```

```
/// <param name="attacker">Атакующая сторона</param>
/// <param name="defender">Атакуемая сторона</param>
/// <param name="name">Название атакующей страны</param>
void attack(Field attacker, Field defender, string name){
	while (attacker.spend(defender.get_sum_cost()) && !defender.destroyed())
	{
		int x = rand() % defender.get_high(); //Случайно выбирается цель
		int y = rand() % defender.get_widht();
		defender.take_gamage(x, y);
	}

	if(defender.destroyed()){
		cout << name << " победила " << endl; //В случае победы
		exit(0);
	}
	cout << name << " прекратила стрельбу, так как это стало невыгодно" << endl; //В случае прекращения стрельбы
}
int main()
{
	setlocale(LC_ALL, "Russian");
	int x, y, cost;
	cout << "Добро пожаловать в симулятор войны:\nВам будет предложено задать размеры государств\n" <<
		"Для баланса они будут использованы для обоих стран\nИ цену снарядов" << endl;
	cout << "Введите высоту страны(целое число)" << endl;
	cin >> x;
	cout << "Введите ширину страны(целое число)" << endl;
	cin >> y;
	cout << "Введите цену снаряда(целое число)" << endl;
	cin >> cost;
	Field Anch = Field(x, y, cost);
	Field Tar = Field(x, y, cost);
```

```
    cout << "Война началась" << endl;

    thread th1 = thread(attack, Anch, Tar, "Анчуария");
    thread th2 = thread(attack, Tar, Anch, "Тарантерия");
    th1.join();
    th2.join();

    cout << "Диктаторы пришли к перемирию" << endl;
}
```