

Национальный исследовательский университет "Высшая школа экономики".  
Факультет компьютерных наук.

Пояснительная записка к домашнему заданию № 3 по предмету  
"Архитектура вычислительных систем".

Выполнила студентка  
2 курса группы БПИ192  
Цимбалистая Ульяна Игоревна

17 ноября 2020 г.

## Содержание

1. [Текст задания](#)
2. [Алгоритм решения](#)
3. [Текст программы](#)
4. [Примеры работы](#)
5. [Дополнение](#)

## 1. Текст задания

Вариант 26.

Вторая задача об Острове Сокровищ. Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Несмотря на добытую карту старого Флинта, местоположение сокровищ по-прежнему остается загадкой, поэтому искать клад приходится практически на ощупь. Так как Сильвер ходит на деревянной ноге, то самому бродить по джунглям ему не с руки. Джон Сильвер поделил остров на участки, а пиратов на небольшие группы. Каждой группе поручается искать клад на нескольких участках, а сам Сильвер ждет на берегу. Группа пиратов, обшарив одну часть острова, переходит к другой, еще необследованной части. Закончив поиски, пираты возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляющим потоком, моделирующее действия Сильвера и пиратов. При решении использовать парадигму портфеля задач.

## 2. Алгоритм решения

Остров разбивается на 15 участков, каждый из которых также разбивается на 15 частей, в которых может быть спрятан клад. Для этого формируется двумерный массив типа `bool` размером 15 на 15, где на случайной позиции стоит `true` — это и есть клад.

Создаются 5 групп пиратов, т. е. 5 потоков. По очереди пираты отправляются на поиски сокровищ, начиная с участка № (номер группы - 1). Таким образом, 1 группа отправляется на участок № 0, 2 группа на № 1 и так далее до 5 группы.

В методе `treasureSearch` происходит поиск по переданной строке массива, и если клад находится, то программа завершится. Если же в переданной строке нет элемента `true`, то группа пиратов (поток) снова переходит в этот метод, но со строкой равной (номер строки + количество групп). Проверку участков пиратами можно изобразить следующим образом:

Номер участка (строки)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Номер группы (потоки)	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5

Таким образом потоки пройдут по всем строкам массива и элемент `true` гарантированно будет найден.

### 3. Текст программы

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

using namespace std;

// Количество групп пиратов.
const int groups = 5;
// Количество участков.
const int DIM = 15;
// Массив, представляющий собой остров, где количество строк – участки,
// а столбцы – места, где могут лежать сокровища.
bool place[DIM][DIM];
// Флаг, сигнализирующий о нахождении кладе.
bool flag = false;

// Заполнение массива, в случайную ячейку помещается true (сокровище).
void initPlaces() {
    int dim1 = rand() % 15;
    int dim2 = rand() % 15;

    for (int i = 0; i < DIM; ++i) {
        for (int j = 0; j < DIM; ++j) {
            place[i][j] = false;
        }
    }

    place[dim1][dim2] = true;
}

// Поиск кладе по cell участку. Выполняется пока клад не будет найден.
void treasureSearch(int cell, int n) {
    if (!flag) {
        bool done = false;
        for (int i = 0; i < DIM; ++i) {
            if (place[cell][i]) {
                done = true;
                flag = true;
                cout << "Group of pirates # " << n << " found treasure at the " << cell <<
                    " place.\n" << endl;
                exit(0);
            }
        }

        cell = cell + groups;
        // Если клад еще не найден и обследованы не все участки,
        // то пираты переходят к другому участку.
        if (!done && cell < DIM) {
            treasureSearch(cell, n);
        }
    }
}

int main() {
    srand(time(0));

    initPlaces();

    cout << "The pirates, led by John Silver, have landed on Treasure Island.\n"
        << "Leader of the pirate gang sent groups of pirates to find treasure deep to jungles of
            island.\n\n";

    // "Портфель" потоков.
    vector<thread> threads;
    threads.reserve(groups);

    // Номер участка.
    int cell = 1;
    // Номер группы пиратов.
    int n = 1;

    // Отправляем пиратов на поиски, пока не найдены сокровища.
    // Каждая группа отправляется на поиски с участка под номером (номер группы - 1) и позже
    // перемешаются по участкам (номер участка + количество групп).
    // Таким образом пираты проходят по всем участкам.
```

```

for (int i = 0; i < groups; ++i) {
    cout << "Group of pirates # " << n << " start their searches
           from " << i << " place. \n" << endl;
    threads.emplace_back(treasureSearch, i, n);
    n++;

    if (flag)
        exit(0);
}

// Ждем выполнения.
for (auto &th : threads)
    th.join();
threads.clear();

return 0;
}

```

## 4. Примеры работы

```
/Users/ulyanatsimbalistaya/CLionProjects/c++Seminars/abchw3/cmake-build-debug/abchw3
The pirates, led by John Silver, have landed on Treasure Island.
Leader of the pirate gang sent groups of pirates to find treasure deep to jungles of island.

Group of pirates # 1 start their searches from 0 place.

Group of pirates # 2 start their searches from 1 place.

Group of pirates # 3 start their searches from 2 place.

Group of pirates # 4 start their searches from 3 place.

Group of pirates # 5 start their searches from 4 place.

Group of pirates # 5 found treasure at the 9 place.

Process finished with exit code 0
```

```
/Users/ulyanatsimbalistaya/Desktop/hw_3/abchw3/cmake-build-debug/abchw3
The pirates, led by John Silver, have landed on Treasure Island.
Leader of the pirate gang sent groups of pirates to find treasure deep to jungles of island.

Group of pirates # 1 start their searches from 0 place.

Group of pirates # 2 start their searches from 1 place.

Group of pirates # 3 start their searches from 2 place.

Group of pirates # 4 start their searches from 3 place.

Group of pirates # 5 start their searches from 4 place.

Group of pirates # 1 found treasure at the 5 place.

Process finished with exit code 0
```

## 5. Дополнение

Код программы находится на репозитории [https://github.com/UlyanaTsi/avs\\_hw](https://github.com/UlyanaTsi/avs_hw) в файле `avs_hw/hw_3/abchw3/main.cpp`