

Национальный исследовательский университет “Высшая Школа Экономики”,
Факультет компьютерных наук
Департамент программной инженерии

**«Программа моделирования задачи об острове
сокровищ»**

Пояснительная записка

Исполнитель:
Студент группы БПИ199
Федченко Всеволод Александрович

Москва 2020

Оглавление

1. Текст задания.....	3
2. Применяемые расчетные методы.....	4
2.1 Алгоритм.....	4
2.2 Входные данные.....	4
2.3 Выходные данные.....	4
3. Тестовые примеры.....	5
4. Список используемых источников.....	6
5. Текст программы.....	7

1. Текст задания

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 не обследованной 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. [1]

2. Применяемые расчетные методы

2.1 Алгоритм

При запуске программы создается массив типа bool, имитирующий участки острова. Далее рандомно выбирается участок, в котором будет находиться сокровище (в массиве по этому индексу устанавливается значение true). В качестве команд пиратов выступают потоки, каждый из которых пытается найти «сокровище» на выделенных ему участках. Всем потокам передается одна функция для исполнения, которая делит участки на группы размером [кол-во участков / кол-во команд], после чего каждый поток осуществляет поиск «сокровища» в своей группе секторов.

2.2 Входные данные

Входные данные отсутствуют, однако при желании можно изменить переменные кол-ва команд пиратов и участков на острове в коде (Кол-во участков должно нацело делиться на кол-во команд пиратов).

2.3 Выходные данные

В консоль выводится количество участков на острове, количество команд пиратов, ищущих сокровища, а так же номер участка на котором зарыто сокровище. Далее выводится информация о результате поиска сокровища каждой командой на отведенной ей территории (для примера вывода см. рис. 1.1, 1.2).

3. Тестовые примеры

Программа выводит в консоль информацию о результате моделирования задачи (см. Рисунок 1.1, 1.2)

```
number of sectors = 1000000
number of teams = 8
treasure index = 128444
Team 0 havent found anything
Team 1 have found treasure in sector 128444
Team 2 havent found anything
Team 3 havent found anything
Team 4 havent found anything
Team 5 havent found anything
Team 6 havent found anything
Team 7 havent found anything
```

Рис 1.1. Консольный вывод. Пример 1

```
number of sectors = 1000000
number of teams = 8
treasure index = 816023
Team 0 havent found anything
Team 1 havent found anything
Team 2 havent found anything
Team 3 havent found anything
Team 4 havent found anything
Team 5 havent found anything
Team 6 have found treasure in sector 816023
Team 7 havent found anything
```

Рис 1.2. Консольный вывод. Пример 2

4. Список используемых источников

[1] Практические приемы построения многопоточных приложений. [Электронный ресурс]. // URL: <http://softcraft.ru/edu/comparch/tasks/t03/> (Дата обращения: 16.11.2020, режим доступа: свободный)

5. Текст программы

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

// Вариант: 26
// Выполнил: Федченко Всеволод Александрович БПИ199

//Вторая задача об Острове Сокровищ.
//Шайка пиратов подпредводительством Джона Сильвера высадились 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 не обследованной 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.

using namespace std;

//Кол-во участков.
const int numOfSectors = 1000000;
//Кол-во команд (должно нацело делить кол-во участков)
const int numOfThreads = 8;

int treasureIndex = 0;

bool islandSectors[numOfSectors];

thread threads[numOfThreads];
int threadOutput[numOfThreads];

void searchForTreasure(int i, void* reslut)
{
    (*(int*)reslut) = -1;
    int numOfSectorsForThread = numOfSectors / numOfThreads;

    for (int j = i * numOfSectorsForThread; j < i * numOfSectorsForThread + numOfSectorsForThread; j++)
    {
        if (islandSectors[j])
        {
            (*(int*)reslut) = j;
        }
    }
}

int main()
{
    //Создание рандомайзера.
    std::random_device rd;
    std::mt19937 rng(rd());
    std::uniform_int_distribution<int> uni(0, numOfSectors);

    auto random_integer = uni(rng);

    // Рандомно задать сектор с сокровищем.
    treasureIndex = random_integer;
    islandSectors[treasureIndex] = true;
    cout << "number of sectors = " << numOfSectors << endl;
    cout << "number of teams = " << numOfThreads << endl;
    cout << "treasure index = " << treasureIndex << endl;

    //Создание потоков
    for (int i = 0; i < numOfThreads; i++)
    {
        threads[i] = thread(searchForTreasure, i, (void*)(threadOutput + i));
    }

    //Ожидания окончания выполнения всех потоков.
```

```

for (int i = 0; i < numOfThreads; i++)
{
    threads[i].join();
}

//Вывод информации.-
for (int i = 0; i < numOfThreads; i++)
{
    if (threadOutput[i] == -1)
        cout << "Team "<< i<< " havent found anything"<< endl;
    else
        cout << "Team " << i<< " have found treasure in sector " << threadOutput[i]<<
endl;
}
}

```