

**Правительство Российской Федерации
Национальный исследовательский университет
«Высшая школа экономики»**

Факультет компьютерных наук
Департамент программной инженерии

Домашняя работа “Task 03”
Вариант 25

Выполнил
Студент группы БПИ193
Цыдыпов Агван
avtsydygov@edu.hse.ru

Москва 2020

Оглавление

1. Задание	3
2. Документация.....	4
2.1 Пример работы	4
2.2 Методы.....	4
2.3 Формат ввода-вывода.....	4
3 Текст программы	5

1. Задание

Первая задача об Острове Сокровищ. Шайка пиратов под предводительством Джона Сильвера высадилась на берег Острова Сокровищ. Не смотря 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 одном из участков, а сам Сильвер ждет на берегу. Пираты, обшарив свою часть острова, возвращаются к Сильверу и докладывают о результатах. Требуется создать многопоточное приложение с управляемым потоком, моделирующее действия Сильвера и пиратов.

2. Документация

По условию: Требуется создать многопоточное приложение.

О программе : программа симулирует остров (двумерный массив, размеры которого вводятся с клавиатуры) и 2 пиратские команды(2 потока), пираты ищут сокровища на 2 частях карты, для симуляции времени поиска была подключена библиотека “Chrono”, где 1 час равен 0.333 секундам.

2.1 Пример работы

Результат работы программы:

Программа выводит в консоль информацию о том, какая команда пиратов нашла сокровище.

2.2 Методы

Методы используемые в программе:

1. PirateTeam1: симулирует первую пиратскую команду, которая ищет клад на верхней части карты.
2. PirateTeam2: симулирует вторую пиратскую команду, которая ищет клад на нижней части карты.

2.3 Формат ввода-вывода

Формат ввода:

Вводятся 2 числа:

- 1) Ширина карты
- 2) Высота карты

Формат вывода:

Выводит номер команды, которая нашла сокровище

3 Текст программы

```
#include <iostream>
#include <thread>
#include <vector>
#include <string>
#include <chrono>
#include <ctime>
void PirateTeam1(std::vector<std::vector<int>> vec, int x, int y)
{
    int b = 0;
    for (int i = 0; i < y / 2; i++)
        for (int j = 0; j < x; j++)
    {
        {
            std::this_thread::sleep_for(std::chrono::milliseconds(333));
            std::cout << "TEAM1 + 1hour" << std::endl;
            if (vec[i][j] == 1)
            {
                b = 1;
                std::cout << "Team1Gold" << std::endl;
                return;
            }
        }
    }
    if (b == 1)
        std::cout << "team1 wins!" << std::endl;
}
void PirateTeam2(std::vector<std::vector<int>> vec, int x, int y)
{
    int b = 0;
    for (int i = y / 2; i < y; i++)
        for (int j = 0; j < x; j++)
    {
        {
            std::this_thread::sleep_for(std::chrono::milliseconds(333));
            std::cout << "TEAM2 + 1hour" << std::endl;
            if (vec[i][j] == 1)
            {
                b = 1;
                std::cout << "Team2Gold" << std::endl;
                return;
            }
        }
    }
    if (b == 1)
        std::cout << "team2 wins!" << std::endl;
}

int main()
{
    int x;
    std::cout << "enter field width: " << std::endl;
    std::cin >> x;
    std::cout << "width = " << x << std::endl;
    int y;
    std::cout << "enter field height: ";
    std::cin >> y;
    std::cout << "height = " << y << std::endl;
    std::vector<int> Gold(2);
    std::srand(std::time(0));
    Gold[0] = rand() % y;
    Gold[1] = rand() % x;
    std::cout << Gold[0] << "\n";
```

```
std::cout << Gold[1] << "\n";  
  
std::vector<std::vector<int>> field(y, std::vector<int>(x)); // Объявление вектора на  
у строк по x элементов  
  
for (int i = 0; i < y; i++)  
    for (int j = 0; j < x; j++)  
    {  
        field[i][j] = 0; // Заполнение вектора или массива  
    }  
field[Gold[0]][Gold[1]] = 1;  
for (int i = 0; i < y; i++)  
{  
    for (int j = 0; j < x; j++)  
        std::cout << field[i][j];  
    std::cout << "\n";  
}  
std::thread th(PirateTeam1, field, x, y);  
std::thread th1(PirateTeam2, field, x, y);  
th.join();  
th1.join();  
  
return 0;  
}
```