

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии
Дисциплина: «Архитектура вычислительных систем»

Домашнее задание № 3

Вариант 25

Пояснительная записка

Выполнила: Резуник Людмила
студент гр. БПИ198

Москва
2020

Содержание

1. Текст задания.....	2
2. Постановка задачи.....	2
3. Описание алгоритма	2
3.1. Формирование массива ячеек «острова».....	2
3.2. Разделение пиратов на группы	3
3.3. Распределение территорий	3
3.4. Работа с потоками.....	3
4. Описание входных и выходных данных	3
5. Тестирование программы	4
Список использованных источников	8

1. Текст задания

Вариант 25

Первая 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 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 управляющим потоком, моделирующее действия Сильвера и пиратов.

2. Постановка задачи.

Разработать консольную программу, которая моделирует описанную выше ситуацию при помощи потоков.

3. Описание алгоритма

Работу алгоритма можно разделить на 3 этапа. Первый этап – формирование массива, содержащего все ячейки «острова». Второй этап – разделение всех пиратов на группы. Третий этап – распределение территорий между группами пиратов. Четвёртый этап – работа с потоками, моделирование действий каждой группы пиратов и вывод результатов «экспедиции».

3.1. Формирование массива ячеек «острова»

На начальном этапе пользователь вводит значения ширины и высоты острова. Программа осуществляет проверку вводимых значений (они должны быть больше 0). В случае некорректных данных ввод повторно запрашивается до тех пор, пока он не будет корректным. Далее, осуществляется вывод всех ячеек острова на экран (значения заменены на символ *). На этом этапе используется функция `start()`, вызываемая из метода `main()`.

3.2. Разделение пиратов на группы

За разделение пиратов на группы отвечает метод `createCrew()`. Разделение осуществляется следующим образом: пользователь указывает количество пиратов в каждой группе последовательно, пока пиратов без группы не останется. Программа осуществляет также проверку вводимых значений.

3.3. Распределение территорий

Распределение территорий между командами пиратов инициализируется методом `partCrew()`. Пользователь последовательно в цикле вводит размер исследуемой территории для каждой группы (целое число). Оно в то же время не должно превышать значения, которое вычисляется по формуле $(width * height) / numberGroups$, где `width` и `height` – ширина и высота острова, а `numberGroups` – количество групп. Это сделано для того, чтобы не произошло ситуаций, в которых хотя бы одна из групп осталась без своего участка территории. В то же время, если пользователь не распределил всю территорию между группами, оставшаяся её часть переходит к последней группе.

3.4. Работа с потоками

Данный этап начинается с заполнения вектора `threads` потоками, которые дальше будут осуществлять работу метода `groupAction`. В этом методе каждый поток проходится по массиву в цикле и проверяет, находится ли сокровище (то есть символ «1») среди проверяемых ячеек. Если нет, то команда не нашла сокровище, и происходит вывод части карты острова, которая была проверена. В этом случае будут выведены нули. Если же команда нашла сокровище, то в выводе карты части острова будет присутствовать символ «1». После вывода сообщений на экран работа с потоками завершается.

4. Описание входных и выходных данных

Входные данные представляют собой целые положительные числа не равные нулю.

Выходные данные представляют собой строки с сообщениями пользователю, уведомляющие его об изменениях, результатах работы программы, или же запрашивающие ввод.

Также, все вводимые данные корректно обрабатываются программой: выводятся соответствующие сообщения пользователю и запрашивается повторный ввод.

5. Тестирование программы

Далее будут представлены скриншоты работы программы на нескольких тестах.

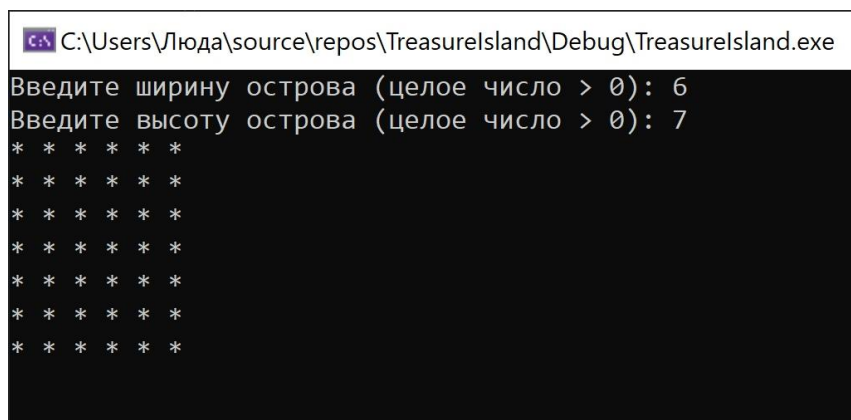


Рисунок 1 – Начало работы. Ввод параметров острова.

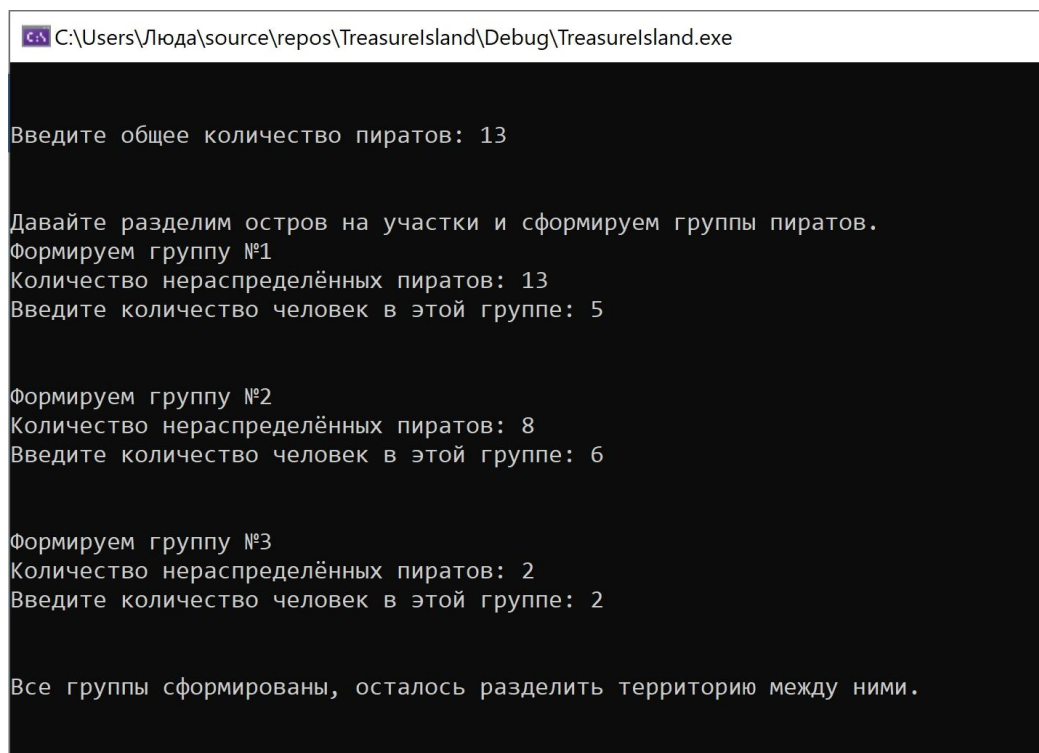


Рисунок 2 – Деление пиратов на группы

```

C:\Users\Людa\source\repos\TreasureIsland\Debug\TreasureIsland.exe
Не распределено ещё 42 квадрат. м. острова.
Назначьте группе №1 размер участка острова (1 целое число): 5

Не распределено ещё 37 квадрат. м. острова.
Назначьте группе №2 размер участка острова (1 целое число): 7

Последняя группа берёт на себя всю оставшуюся территорию острова - 30 кв. м.

```

Рисунок 3 – Распределение территории между группами пиратов

```

C:\Users\Людa\source\repos\TreasureIsland\Debug\TreasureIsland.exe
Группа №2 вернулась из поисков.
Она не нашла сокровища.
Группа отметила исследованную зону на карте:
* * * * * 0
0 0 0 0 0 0
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

Группа №1 вернулась из поисков.
Она не нашла сокровища.
Группа отметила исследованную зону на карте:
0 0 0 0 0 *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

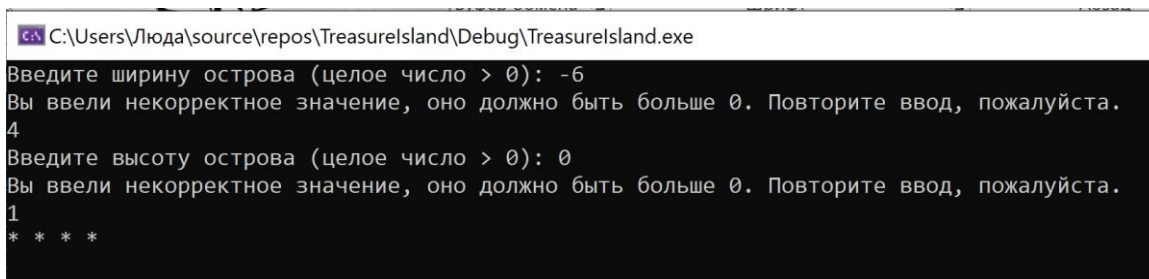
Группа №3 вернулась из поисков.
Она нашла сокровища!
Группа отметила исследованную зону на карте:
* * * * *
* * * * *
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 1
0 0 0 0 0 0
0 0 0 0 0 0

```

Рисунок 4 – Вывод результатов

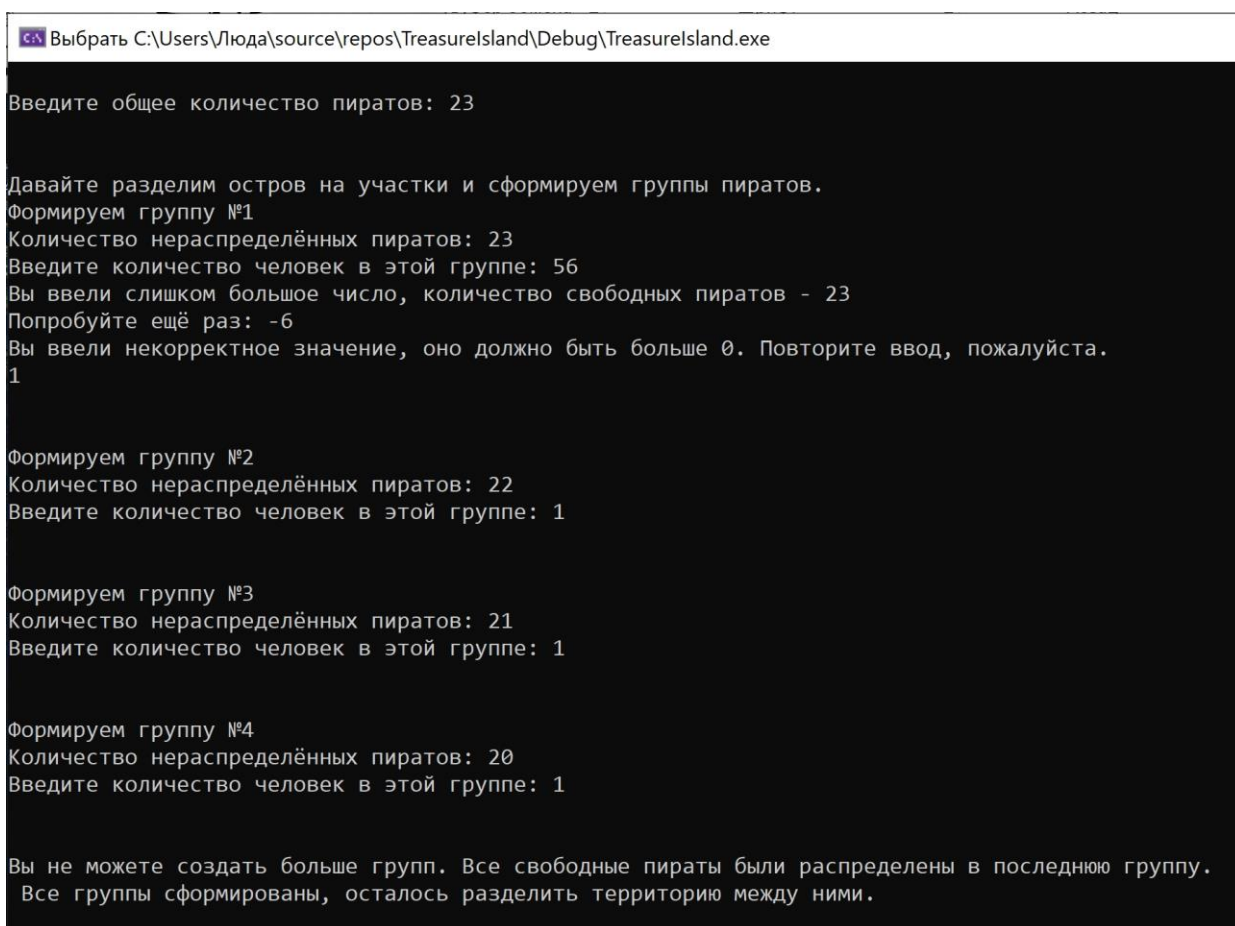
Можно заметить, что потоки выводят информацию не по порядку, однако это не влияет на моделирование данной ситуации, так как группы могут затрачивать на поиск разное время.

Далее будет рассмотрен пример работы программы с некорректными данными.



```
C:\Users\Людa\source\repos\TreasureIsland\Debug\TreasureIsland.exe
Введите ширину острова (целое число > 0): -6
Вы ввели некорректное значение, оно должно быть больше 0. Повторите ввод, пожалуйста.
4
Введите высоту острова (целое число > 0): 0
Вы ввели некорректное значение, оно должно быть больше 0. Повторите ввод, пожалуйста.
1
* * * *
```

Рисунок 5 – Пробуем задать острову отрицательные или нулевые параметры



```
Выбрать C:\Users\Людa\source\repos\TreasureIsland\Debug\TreasureIsland.exe
Введите общее количество пиратов: 23

Давайте разделим остров на участки и сформируем группы пиратов.
Формируем группу №1
Количество нераспределённых пиратов: 23
Введите количество человек в этой группе: 56
Вы ввели слишком большое число, количество свободных пиратов - 23
Попробуйте ещё раз: -6
Вы ввели некорректное значение, оно должно быть больше 0. Повторите ввод, пожалуйста.
1

Формируем группу №2
Количество нераспределённых пиратов: 22
Введите количество человек в этой группе: 1

Формируем группу №3
Количество нераспределённых пиратов: 21
Введите количество человек в этой группе: 1

Формируем группу №4
Количество нераспределённых пиратов: 20
Введите количество человек в этой группе: 1

Вы не можете создать больше групп. Все свободные пираты были распределены в последнюю группу.
Все группы сформированы, осталось разделить территорию между ними.
```

Рисунок 6 – При формировании групп пытаемся вводить некорректное количество пиратов, пытаемся создать больше групп чем частей острова.

```

Выбрать C:\Users\Людя\source\repos\TreasureIsland\Debug\TreasureIsland.exe

Не распределено ещё 4 квадрат. м. острова.
Назначьте группе №1 размер участка острова (1 целое число): 2
Вы ввели слишком большое число. Замечание: группе нельзя назначить больше 1 кв. м.
Попробуйте ещё раз: 1

Не распределено ещё 3 квадрат. м. острова.
Назначьте группе №2 размер участка острова (1 целое число): 1

Не распределено ещё 2 квадрат. м. острова.
Назначьте группе №3 размер участка острова (1 целое число): 1

Последняя группа берёт на себя всю оставшуюся территорию острова - 1 кв. м.

```

Рисунок 7 – Ограничение на размер участка территории

```

Выбрать C:\Users\Людя\source\repos\TreasureIsland\Debug\TreasureIsland.exe

Группа №3 вернулась из поисков.
Она не нашла сокровища.
Группа отметила исследованную зону на карте:
* * 0 *

Группа №1 вернулась из поисков.
Она нашла сокровища!
Группа отметила исследованную зону на карте:
1 * * *

Группа №2 вернулась из поисков.
Она не нашла сокровища.
Группа отметила исследованную зону на карте:
* 0 * *

Группа №4 вернулась из поисков.
Она не нашла сокровища.
Группа отметила исследованную зону на карте:
* * * 0

```

Рисунок 8 – Вывод результатов

ПРИЛОЖЕНИЕ 1**Список использованных источников**

1. Multithreading in C++ [Электронный ресурс] // URL: <https://www.geeksforgeeks.org/multithreading-in-cpp/> (дата обращения: 15.11.2020)
2. Методика разработки многопоточных приложений [Электронный ресурс] // URL: <https://rdsn.org/article/baseserv/RUThreadingMethodology.xml> (дата обращения: 17.11.2020)
- 3.