

**Вариант № 30**

**Задача 30. 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 управляющим потоком**, моделирующее действия Сильвера и пиратов.

**В приложении была реализована модель**

**Управляющий и рабочие** – модель организации вычислений, при которой существует поток, координирующий работу всех остальных потоков. Как правило, управляющий поток распределяет данные, собирает и анализирует результаты.

Эта парадигма часто применяется в задачах оптимизации и статистической обработки информации, при обработке изображений и других научных вычислениях с итеративными алгоритмами. -

(источник :

[https://studref.com/702392/informatika/paradigmy\\_parallelnyh\\_prilozheniy](https://studref.com/702392/informatika/paradigmy_parallelnyh_prilozheniy) )

Управляющий поток делегирует задачу, которую каждый рабочий поток должен выполнить, путем задания некоторой функции. Вместе с задачей на рабочий поток возлагается и ответственность за ее

выполнение и получение результатов. Кроме того, на этапе получения результатов возможна синхронизация действий с управляющим (или другим) потоком.

Управляющий поток может создавать рабочие потоки в результате запросов, обращенных к системе. При этом обработка запроса каждого типа может быть делегирована рабочему потоку. В этом случае управляющий поток выполняет некоторый цикл событий. По мере возникновения событий рабочие потоки создаются и на них тут же возлагаются определенные обязанности. Для каждого нового запроса, обращенного к системе, создается новый поток. При использовании такого подхода процесс может превысить предельный объем выделенных ему ресурсов или предельное количество потоков. В качестве альтернативного варианта управляющий поток может создать пул потоков, которым будут переназначаться новые запросы. Управляющий поток создает во время инициализации некоторое количество потоков, а затем каждый поток приостанавливается до тех пор, пока не будет добавлен запрос в их очередь. По мере размещения запросов в очереди управляющий поток сигнализирует рабочему о необходимости обработки запроса. Как только поток справится со своей задачей, он извлекает из очереди следующий запрос. Если в очереди больше нет доступных запросов, поток приостанавливается до тех пор, пока управляющий поток не просигналит ему о появлении очередного задания в очереди. Если все рабочие потоки должны разделять одну очередь, то их можно программировать на обработку запросов только определенного типа. Если тип запроса в очереди не совпадает с типом запросов, на обработку которых ориентирован данный поток, то он может снова приостановиться. Главная цель управляю-потока — создать все потоки, поместить задания в очередь и «разбудить» рабочие потоки, когда эти задания станут доступными. Рабочие потоки справляются о наличии запроса в очереди, выполняют назначенную задачу и приостанавливаются сами, если для них больше нет работы. Все рабочие и управляющий потоки выполняются параллельно.

( Источник - [https://mir-knig.com/read\\_564021-36](https://mir-knig.com/read_564021-36) )

## **Ввод/Вывод**

### **Ввод**

**Реализовано две версии :**

1) Через аргументы командной строки (всего 3 аргумента) :  
В данном случае нужно ввести строку, содержащую путь к .exe файлу и два числа типа int (>0), разделенных пробелами  
например - Home\_3.exe 15 15

Остров будет создан, дальше от пользователя будет требоваться ввести числа типа int (>0) для пиратов. Все подсказки (ожидаемые действия от пользователя) будут выведены в консоль.

2) Полное управление через консоль. В этом случае также будут выведены все подсказки в консоль. От пользователя потребуется ввести сначала  
высоту острова (int)  
ширину острова (int)  
и численность пиратов (int), а также групп этих пиратов (int)

### **Вывод**

Весь вывод программы осуществляется в консоль, сразу во время работы.

Каждое действие пользователя прописано в подсказках, которые выводятся в консоль в нужное время. Всё происходящее также выводится в консоль. (В ходе выполнения программа выводит в консоль все происходящие события)

## **Сама программа**

Сначала пользователю предлагается ввести два значения: высоту и ширину острова. После проверки введенных данных на корректность, создается и выводится в консоль остров ( как поле

для морского боя ), каждая ячейка которого обозначается “0”, что символизирует статус ячейки как “неизведанной”. Одна из ячеек хранит в себе сокровище (определяется рандомно), но для пользователя она пока невидима.

Дальше снова идет процедура ввода значений. Сначала общее количество пиратов, которые будут исследовать остров. Далее, пользователю предлагается разбить пиратов на группы (группа может вмещать в себя как 1, так и всех пиратов). Естественно, сумма участников каждой группы не превосходит общее количество пиратов. Таким образом, если общее количество пиратов = 10, то групп может быть от одной (все 10 пиратов) до 10 (10 групп по 1 пирату).

От количество пиратов в группе, зависит время, которое эта группа будет тратить на изучение ячейки острова. Один пират тратит 7 секунд на изучение ячейки. Два - 3.5 секунды и т.д.

Когда все пираты разбиты на группы, создаются потоки, определяющие каждую из команд пиратов. Главный поток отправляет свободные команды на изучение ячеек острова. Каждая команда работает над своей ячейкой, вместе над одной ячейкой команды работать не могут. ( mutex, чтобы ограничивать одновременный доступ ) Таким образом, когда команда освобождается от изучения своей ячейки, она ждёт, пока главный поток переопределяет эту команду на другую свободную ячейку ((использован conditional variable)).

Группы пиратов (подконтрольные потоки) выдают информацию в консоль о своих действиях. Каждый раз при получении нового задания (новой ячейки для исследования), начале или при завершении исследования какой-то ячейки. Команды (потоки) при получении задания от главного потока, приступают к изучению ячейки ( спят определенное количество времени ), затем извещают главный поток об успехе их исследований (найден клад / не найден) и выводят в консоль остров (с обновленными данными по ячейкам). Если в этой ячейке клад **не был** обнаружен, то ячейка обозначается символом “\_”, что символизирует статус ячейки как “исследована, без клада”. Если же в этой ячейке клад **был** обнаружен, то ячейка

обозначается символом “X”, что символизирует статус ячейки как “исследована, клад найден”. Группа(поток), нашедшая клад, выводит в консоль информацию об этом, сам остров, и передаёт данные об успехе главному потоку.

Так продолжается, пока не будет изучен весь остров (если клад находится в самой последней ячейке) или пока не будет найден клад.

Когда клад будет найден, главный поток передает данные об этом во все подконтрольные потоки, что завершает их дальнейшую работу.

Дождавшись окончания работы всех подконтрольных потоков, главный поток выводит в консоль информацию о том, что клад был найден группой №, после чего программа завершается.