**Логика работы программы:**

При запуске программа попросит пользователя ввести кол-во портов РАЗГРУЗКИ, затем нужно ввести кол-во портов ПОГРУЗКИ, затем кол-во грузовиков которые должны быть разгружены.

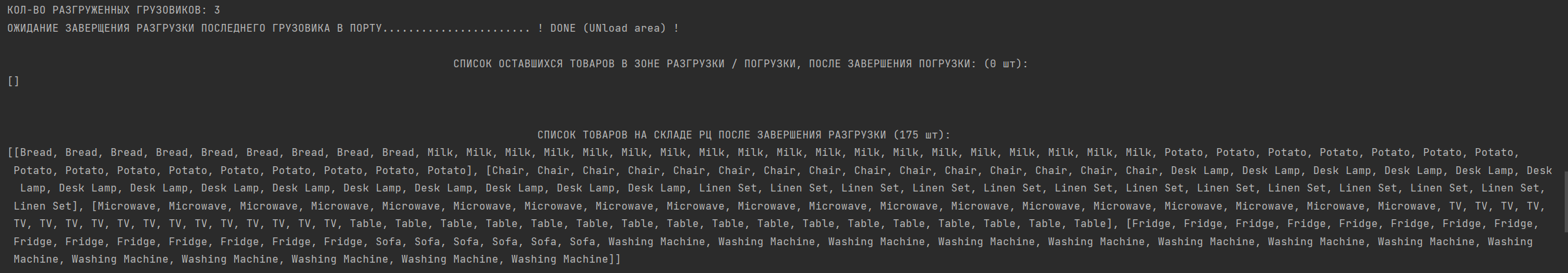
Грузовики с товарами попадают в свободный порт РАЗГРУЗКИ и постепенно переносят товар из кузова в ЗОНУ РАЗГРУЗКИ ТОВАРА (Время, затраченное на перенос товара из кузова в ЗР зависит от типа грузовика и веса товара. Грузовики генерируются случайного типа и содержат случайное кол-во товара в кузове. Вес товара так же определяется случайным образом. Кол-во товара в кузове не может превышать грузоподъёмность грузовика, но грузовик может быть пустым – без товара).

Как только грузовик попадает в зону разгрузки, он автоматически считается разгруженным (счетчик разгруженных грузовиков увеличивается на 1). НО! Разгрузка происходит в соответствии с заданием – каждый товар один за одним перемещается из кузова в ЗОНУ РАЗГРУЗКИ. Т.е, если грузовик не пустой, он пробудет в порту какое-то время.

Программа перейдет в стадию завершения, когда в порты РАЗГРУЗКИ попадет столько грузовиков, сколько ввел пользователь. Например, пользователь ввел 2 порта РАЗГРУЗКИ и 3 грузовика. 2 первых грузовика начнут разгружаться одновременно (так как 2 порта разгрузки), как только один из портов разгрузит грузовик, в него попадет 3-ий грузовик, а программа перейдет в стадию завершения. При этом на экране появится надпись:



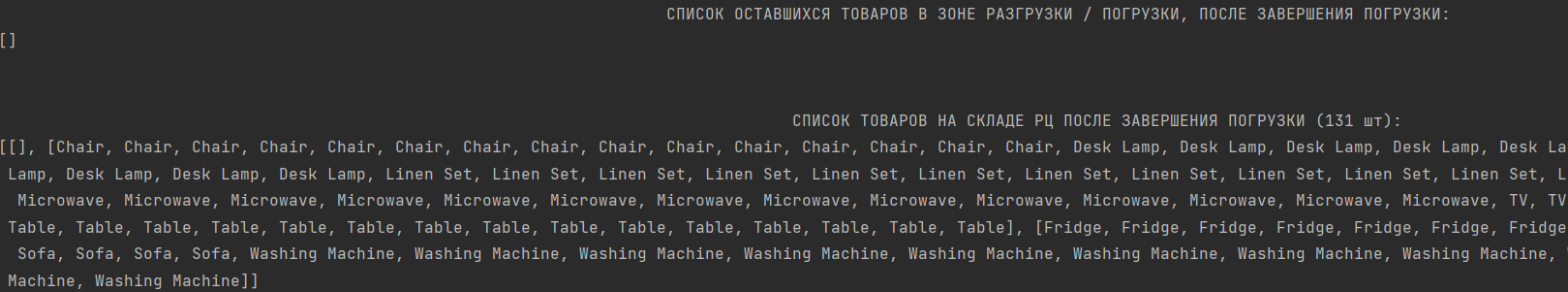
В стадии завершения освободившиеся порты больше НЕ ПРИНИМАЮТ новые грузовики, а все ЗАНЯТЫЕ порты должны завершить свою работу. В качестве индикатора прогресса печатаются точки. Как только порты РАЗГРУЗКИ завершат свою работу (разгрузят столько грузовиков, сколько указал пользователь – 3 в нашем примере), на экран выведется информация о том, сколько товаров осталось в ЗОНЕ ВЫГРУЗКИ, затем будет выведен список товаров на складе РЦ:



Как только товар попадает из кузова грузовика в ЗОНУ ВЫГРУЗКИ, этот товар переносится на СКЛАД РАСПРЕДЕЛИТЕЛЬНОГО ЦЕНТРА (РЦ) и кладется в определенное место (сортируется). Далее, со склада РЦ товар попадает в порт ПОГРУЗКИ, где загружается в пустой грузовик. ПОГРУЗКА И ПЕРЕМЕШЕНИЕ ТОВАРОВ НА СКЛАД РЦ ТОВАРОВ происходит СИНХРОННО. Т.е порт ПОГРУЗКИ не дожидается завершения работы портов РАЗГРУЗКИ, а работает независимо от них.

После того, как порты РАЗГРУЗКИ завершили свою работу, порты ПОГРУЗКИ останавливают свою работу (новые грузовики перестают приниматься этими портами, а те, что уже в них, дожидаются погрузки товаров). Во время погрузки может произойти ситуация, когда в кузов грузовика положили товар, и грузоподьемность грузовика стала отрицательной (перегруз). В этом случае происходит выгрузка последнего добавленного товара (который вызвал перевес) в зону РАЗГРУЗКИ. Оттуда его, так же, как и в портах РАЗГРУЗКИ, подхватывает корутина РЦ и переносит на склад РЦ.

Во время завершения работы портов ПОГРУЗКИ, по аналогии с портами РАЗГРУЗКИ, выводится сообщение о завершении работы и печатаются точки. После того, как последний грузовик покинул РЦ в консоль выводится информация о кол-ве товаров в зоне разгрузки и оставшиеся товары на складе РЦ:



На этом программа останавливается.

**Первый запуск:**

Во время работы, программа выводит много информации в консоль, в которой легко потеряться. Если запустить программу с определенными настройками, то она из асинхронной будет работать как КАК-БЫ СИНХРОННАЯ. Информация в консоль будет выводиться последовательно, понять логику работы будет намного проще.

*Для того, чтобы понять логику работы рекомендую провести первый запуск со следующими параметрами*:

Кол-во портов разгрузки:3

Кол-во портов погрузки: 1

Кол-во разгружаемых грузовиков: 3

***Так же, в объекте LoadArea на 24-ой строке раскомментировать delay(20000) сек.!!!!!!!!!!!!***

Данный делей отсрочит запуск портов ПОГРУЗКИ на 20 сек. Скорее всего, за это время порты РАЗГРУЗКИ успеют завершить свою работу, и информация из портов ПОГРУЗКИ не будет печататься в консоль во время работы портов РАЗГРУЗКИ. Через 20с порты разгрузки начнут работать и выводить свою информацию, а порты РАЗГРУЗКИ уже остановлены => они больше не выводят свою информацию во время работы портов ПОГРУЗКИ.

Таким образом информация в консоле будет последовательной, что и нужно для первого знакомства. Если убрать этот делай работа программы будет параллельной и информация в консоле будет в перемешку от 2-ух разных источников (портов РАЗГРУЗКИ и портов ПОГРУЗКИ).

Так как грузовики имеют рандомное время разгрузки (информация о времени разгрузки выводится в консоле), то даже задержки в 20 сек до запуска портов ПОГРУЗКИ может быть недостаточно (если время разгрузки хотя бы одного из сгенерированных грузовиков будет больше 20 сек). Но, так как генерируется всего 3 машины, и они сразу же обрабатываются портами РАЗГРУЗКИ (так как создано 3 порта), такая ситуация возникает не часто. В случае промаха, рекомендую просто запустить программу еще раз!

После изучения информации из консоли нужно закомментировать данный делай ***delay(20000*** и запустить программу с любыми вводными параметрами. После завершения работы разобраться в консоле будет намного проще!

**Структура программы:**

Программа состоит из 3-ех классов, 4-ех объектов и фаила main. В main происходи вызов объекта DistributionCenter.

Объект DistributionCenter содержит в себе основную логику работы программы. В блоке init данного объекта происходит запуск 3-ех сопрограмм:

1 – запускает объект UnloadArea. В данном объекте запускаются свои сопрограммы, которые генерируют грузовики с товарами (За создание грузовика и заполнением его кузова товарами отвечают классы **Transport, Truck, Good и объект ListWithGoods**). Так же генерируются порты РАЗГРУЗКИ. Полный грузовик попадает в порт РАЗГРУЗКИ и товары из кузова попадают в ЗОНУ РАЗГРУЗКИ (а не на склад РЦ!) выгрузка каждой единицы товара из кузова в ЗОНУ РАЗГРУЗКИ определяется задержкой (она зависит от типа товара и типа грузовика). При этом товары, попавшие в ЗОНУ РАЗГРУЗКИ обрабатываются сопрограммой №3 (сортируются).

2 – запускается объект LoadArea (запуск этого объкта можно отсрочить, см раздел *первый запуск*). В данном объекте запускаются свои сопрограммы, которые генерируют пустые грузовики и списки товаров, которые должны бать загружены в эти грузовики (За создание грузовика отвечают классы **Transport, Truck, Good и объект ListWithGoods**). Так же генерируются порты ПОГРУЗКИ. Пустой грузовик попадает в порт ПОГРУЗКИ и товары со склада РЦ (а не из зоны разгрузки!) попадают в кузов один за другим с определенной задержкой (определяется типом товара и типом грузовика). При этом товары, попавшие в кузов, сразу же удаляются со склада РЦ

3 – запускается сопрограмма (корутина) внутри объекта DistributionCenter. Данная корутина мониторит зону РАЗГРУЗКИ ТОВАРА и переносит товар, попавший в эту зону из грузовика, на склад РЦ (в определенное место. Т.е происходит сортировка товара на складе РЦ). Данная корутина завершается только тогда, когда свою работу завершили 2 пред идущие сопрограммы. Нет смысла мониторить зону РАЗГРУЗКИ, если порты РАЗГРУЗКИ и ПОГРУЗКИ остановлены, а другого способа получить товар нет. ХОЧУ ОТМЕТИТЬ, что при ПОГРУЗКЕ товара со склада РЦ в грузовик, может возникнуть перевес (перегруз грузовика). В таком случае, последний добавленный в кузов товар (вызвавший перевес), убирается из кузова в зону РАЗГРУЗКИ, а сопрограмма №3 возвращает его на склад РЦ в определенное место.

**Завершение работы программы:**

После разгрузки определенного кол-ва грузовиков (устанавливает пользователь) программа переходит в стадию заверения: новые грузовики не принимаются на РАЗГРУЗКУ, далее дожидаемся завершения РАЗГРУЗКИ грузовиков, которые уже разгружались. Далее в порты ПОГРУЗКИ передается сигнал (по средствам отмены Job корутин в UnloadArea) о том, что порты РАЗГРУЗКИ остановлены. Далее порты ПОГРУЗКИ перестают принимать новые грузовик на ПОГРУЗКУ и ожидают заверения ПОГРУЗКИ грузовиков, которые уже ГРУЗИЛИСЬ. Как только порты ПОГРУЗКИ остановлены, происходит отмена Job корутин в LoadArea. Корутина №3 это видит и завершает свою работу -> программа завершена.