V3.01

## Содержание

* Глобальные переменные
* Локальные переменные
* Ключевые особенности
* Списки поставщиков/покупателей
* Параметры товара
* Список товаров
* Формула ранжирования товаров
* Режимы процедуры покупки
* Формула ранжирования целевых станций
* Навигация
* Типы контрагентов
* Валидация контрагентов
* Структуры данных
* Особенности работы некоторых команд

## Глобальные переменные

wwsl.T2.LastShipNumber - номер последнего переименованного корабля

wwsl.T2.global - глобальные параметры

## Локальные переменные

wwsl.T2.config - указывает на структуру типа ShipStruct

wwsl.T2.menu.opened - признак открытия корабельного терминала

## Ключевые особенности

1) Нет глупым ограничениям! Если технически ничто не мешает исполнять кораблем работу, то эта работа должна исполняться.

2) Интересный игровой процесс - это думать! Программа избавляет от занудного микроменеджмента и выполняет рутинную работу, но не думает за игрока. На примере шахмат: программа знает как двигать фигуры, но не способна предотвратить ситуацию мата. Игрок планирует деятельность, а программа работает исходя из предположения, что игрок понимает, что он делает. В безвыходной ситуации программа будет пытаться «двигать фигуры» пока ситуация не исправится или параметры не будут откорректированы.

Т2Дроид включает в себя несколько программ:

1) транспорт - базовая единица для решения транспортных задач

2) шеф - позволяет настраивать параметры кораблей в группе

3) старатель - добывает ресурсы из обломков

\*) инженер - размещение объектов (в т.ч. работы по строительству комплексов)

\*) менеджер флота - поставка, оборудование, ремонт и замена кораблей

\*) снабженец - поддерживает заданный уровень обеспечения флота

\*) караван (а нужен?)

X) кто-то был еще

TODO: Расписать. Везде приоритеты отдаются по объему перевозок. Не по стоимости перевозимого груза потому, что задача обеспечить эффективную перевозку грузов, а не заработок.

## Списки поставщиков/покупателей

С этими списками произошли фундаментальные изменения. Списки поставщиков/покупателей больше не определяют приоритетные станции, требующие первоочередного обслуживания. Теперь эти списки используются как одноранговое дополнение контрагентов рабочей области. Однако, если включен признак обслуживания только станций из списков, то программа работает как и раньше: при наличии в списке объектов, в качестве соответствующих контрагентов используются только станции из списка.

Это сделано для того, что бы снизить путаницу и усилить фокус корабля на отдельной задаче. Например, раньше с помощью списков поставщиков и покупателей можно было сначала заполнять свои фабрики ресурсами, а в свободное время торговать с NPC. Это создавало некоторую путаницу, но было оправдано, так как сокращало суммарное время рутинных процедур управления кораблями. Но в новой версии Т2-Дроида это стало не актуально. Спектр решаемых кораблем задач расширился за счет добавления возможности работы со списком товаров, а количество действий по управлению каждым отдельным кораблем сократилось за счет введение новой программы «Шеф». В связи с этим, списки поставщиков/покупателей теперь следует рассматривать исключительно как часть параметров рабочей области.

## Параметры товара

Параметры товара представляют собой набор ключевых значений,

являются ключевым элементом программы, входящих в систему T2 Дроид.

Параметры товара позволяют отключать закупку и/или реализацию товара. В этом случае, соответствующий этап работы с товаром будет пропущен. Закупочные рейсы выполняются до тех пор, пока в рабочей области имеется хотя бы один поставщик и на корабль можно погрузить минимальную партию для закупки. Цикл продаж выполняется до тех пор, пока в трюме корабля есть минимальная партия для продажи, а в рабочей области имеется хотя бы один потребитель.

## Список товаров

Помимо индивидуальных настроек отдельного товара, начиная с версии 3.01 Т2-Дроид поддерживает работу со списком товаров. Ключевой идеей введения списка товаров была реализация такого поведения, при котором программа работает на основании информации о текущем спросе на товар и возможности его (спроса) удовлетворения.

Одновременно программа может работать только с одним товаром, выполняя закупку или продажу в зависимости от того, что сейчас наиболее предпочтительно. Поскольку настройки каждого отдельного товара позволяют определять индивидуальный способ работы («ничего не делать», «только покупать», «только продавать» и «покупать-продавать»), выбирать товар из списка по одинаковому принципу становится невозможно. Например, товар для которого отключены продажи, выбирать по количеству покупателей не имеет смысла. Товар для которого отключены покупки имеет смысл выбирать только при наличии в трюме товара, а в рабочей области покупателя этого товара. Формула ранжирования товаров, подразумевающих и закупку, и реализацию требует для работы обязательного наличия хотя бы одного поставщика и одного покупателя, что делает ее автоматически неприменимой к ситуациям, где один тип цены не указан.

Для того, что бы обеспечить выбор рабочего товара при существующем множестве различных условий, перед осуществлением выбора все товары из списка подразделяются на категории, определяющие приоритет обработки и адекватный способ выбора товара для следующего рейса.

**Категория №1.** Первым делом необходимо распродать имеющиеся товары, так как доступный объем трюма влияет на общую эффективность работы. В эту категорию попадают товары, которые в объеме минимальной партии для продажи имеются в трюме и для которых включена реализация (комбинации «только продавать» и «покупать-продавать»). Эти товары перебираются в порядке уменьшения занимаемого товаром объема (именно объема, а не количества) и первый товар, для которого есть хотя бы один покупатель, выбирается для следующего рейса. Наличие товара в этой категории означает, что следующий рейс будет рейсом реализации товара. Отсутствие товара будет означать, что продавать нечего и необходимо выбрать товар для рейса закупки.

Помимо приоритетного освобождения места в грузовом отсеке, обработка товаров первой категории позволяет организовать такие задачи, как «распродать разнообразное барахло из трюма, по ценам не ниже заданных», «освободить трюм для последующей работы» и т.п.

**Категория №2.** В эту категорию попадают товары, которые нужно забрать у поставщика и доставить покупателю. То есть, товары с полным рабочим циклом («покупать-продавать») при условии, что в трюме достаточно места для приобретения минимальной партии. Для каждого товара этой категории в соответствии с формулой ранжирования товаров (описывается ниже) рассчитывается рейтинг. Товар с наивысшим ненулевым торговым рейтингом выбирается для следующего рабочего цикла. Если товара с ненулевым рейтингом в этой категории нет, то осуществляется переход к следующей категории.

Обработка товаров данной категории позволяет решать такие задачи, как «обслуживание производственных цепочек», «торговля высокотехнологичными товарами» и т. п.

**Категория №3.** В третью категорию с наименьшим приоритетом попадают товары, для которых включена только закупка. Такие товары приводят к постепенному заполнению трюма, что при наличии в списке товаров с другими режимами работы, будет неотвратимо снижать эффективность работы. Что бы максимально отсрочить полное заполнение трюма, товары третьей категории обрабатываются в последнюю очередь. Обязательным условием попадания товара в эту категорию является наличие свободного места для загрузки минимальной партии. Далее, товары вошедшие в эту категорию перебираются в порядке уменьшения объема предложения (именно объема, а не количества) и товар с наибольшим предложением выбирается для следующего рабочего цикла. Отсутствие подходящих товаров на данном этапе означает, что в данный момент делать абсолютно нечего и нужно подождать изменения ситуации.

## Формула ранжирования товаров

Ранжирование товара используется в процессе выбора наиболее подходящего товара для следующего рабочего цикла. Фактически, ранжирование выполняется по критерию объема с учетом текущей ликвидности товара. В общем виде формула выглядит так:

Спрос / Предложение \* Спрос

Где:

Спрос — суммарный спрос в единицах трюма

Предложение — суммарное предложение в единицах трюма

Общий смысл таков: наиболее подходящий тот товар, где больше востребованный объем перевозок. Поскольку SE не поддерживает вещественные значения, а возведение спроса во вторую степень может дать арифметические переполнение, используется формула в адаптированном виде:

Фактор = Спрос \* 100 / Предложение

Рейтинг = (Фактор <= 100 ? Фактор : 100) \* Спрос / 100

Из формулы видно, что нулевой спрос даст нулевой рейтинг, а нулевое предложение делает расчет невозможным.

1) Пока непонятно. На примере баты+челты+растар практически всегда фигачит на батах. Нужно последить, как будет работать при более эффективном выборе станции.

2) Проблема в том, что при достижении товаром наивысшего рейтинга, на него быстро перебрасываются слишком много кораблей. Это создает ситуацию, когда они косяком летят за товаром, а затем начинаются проблемы с закупкой или с реализацией. Тестируется случайный выбор товара с вероятностью равной доле рейтинга в сумме пунктов по всем ранжированным товарам для выбора.

3) Теперь проблема с перепродажей оружия. На примере плазмогенераторов в XTM: в доке 1 место, на станции 8 штук — халява! Но по такой формуле получается рейтинг 0. Нужно ставить единицу в рейтинг при наличии хотя бы одной пары покупатель-поставщик. КСТАТИ: об этом нужно упомянуть в документации, где про смешивание низко- и высокотехнологичных товаров.

4) Часто зависает в такой ситуации: определяет необходимый товар, отправляет несколько рейсов, первый исполняется, последующие зависают. Есть идея добавить в buyware дополнительный аргумент — режим. Типа если указан, то только долететь до станции, но не покупать. Тогда будет шанс повторно проверить востребованность товара по прибытии. А в случае если товар все еще востребован, то повторная попытка купить но уже без «особого» режима приведет к покупке на той станции, на которой находится в момент запуска процедуры.

5) Интересный вариант избавления от товарных пробок: если транспорт под управлением шефа, можно снизить спрос на суммарный объем товара, находящегося в данный момент в других транспортах группы. Это и ценность шефа (группы) поднимает, и делает планирование более интересным. И тут сходу проблема: ведь для щитов и лазеров, возможно ракет это будет работать крайне не эффективно. Нужно еще предварительно проверить, но теоретически, если будет покупать не больше спроса, то будет возить по одной пушке. Нужно игнорить коррекцию спроса для товаров с Maintype 8, 9, 10. При чем, все эти корректировки необходимо делать непосредственно в функции расчета рейтинга. Иначе получается ситуация, что мы локально определили перекупленность, а последующий выбор товара снова выдает нам этот же товар.

6) Неплохо было бы отдавать предпочтение товару с ненулевым рейтингом, если он имеется для покупки в текущем местоположении корабля. Это позволит сократить количество пустопорожних перелетов.

7) Выяснилась проблема с корректировкой спроса: возникла ситуация, когда наличие товара в собственном трюме корректирует спрос в ноль при наличии реального спроса, который мог бы быть удовлетворен имеющейся партией. Возможно имеет смысл не спрос корректировать, а предложение? Не учитывать в корректировке собственную партию в принципе можно... В общем получается, что нужно рейтинг по-разному считать для категорий 1 и 2/3. Для первой корректировать спрос не нужно. Нужно корректировать только при закупке. Поскольку у нас функция подсчета рейтинга принимает маску валидации (которая сейчас тип контрагента игнорит), то можно это без особых усилий аргументировать через эту маску.

8) Плохо работает при огромном спросе и малом предложении: моментами летят косяком к одному поставщику. Но как это разрулить, пока не представляю.

## Режимы процедуры покупки

Закупочные операции осуществляются на основании анализа текущего спроса. Однако, между принятием решения (выбором товара) и непосредственно моментом его покупки может пройти достаточно много времени. В совокупности с предположением о том, что на рейсах будет множество кораблей (первичная цель программы), это значительно повышает вероятность возникновения ситуации перекупленности товара и образования товарных пробок (из кораблей, которые ждут потребителя). Иначе говоря, необходимо выполнять повторную проверку спроса на товар непосредственно перед его покупкой.

Для обеспечения этой возможности, скрипт закупки принимает дополнительный аргумент, значение которого определяет режим покупки. Ниже перечислены доступные режимы:

| Код | Кратко | Описание |
| --- | --- | --- |
| 0, null | fly & buy (default) | Обычный режим при котором осуществляется выбор поставщика, выполняется полет и попытка закупки в соответствии с параметрами. |
| 1 | just fly | Осуществляется поиск поставщика и, если поставщик найден, выполняется рейс на выбранную станцию. По прибытии на станцию товар НЕ покупается, а скрипт завершается с возвратом нулевого значения. |
| 2 | buy or die | Выполняется попытка покупки в текущем местоположении. Если в текущем местоположении купить товар невозможно, то завершается с возвратом кода «поставщик недоступен по прибытии». В случае успешной покупки возвращается купленное количество. |

## Формула ранжирования целевых станций

После того, как был определен тип очередного рейса (закупка или реализация), выполняется поиск всех подходящих для выполнения операции станций. Последующее ранжирование используется для определения наиболее предпочтительной для очередного рейса станции.

В основе формулы лежит принцип обеспечения максимальной эффективности грузоперевозок: приоритет отдается поставщику, предлагающему наибольшую партию, и потребителю, нуждающемуся в наибольшем количестве товара. Дополнительными факторами, влияющими на рейтинг контрагента являются время начала предыдущего рейса (чем меньше времени прошло с момента отправления предыдущего корабля, тем хуже контрагент — повышается шанс неудачного рейса) и расстояние.

КолвоПроц + КолвоПроц \* ВремяПредОтпр / 3600

Где:

КолвоПроц — количество станции в зависимости от типа операции: в наличии для поставщика, потребность для покупателя выраженное в процентах от объема склада;

ВремяПредОтпр — время с момента начала аналогичной операции другим кораблем. Если это значение превышает час ~~или предыдущее отправление на эту станцию совершил этот же корабль~~, то берется значение 3600 (т. е. час).

1) неэффективно работает формула исключительно по количеству. На примере батареек, привозит на более заполненные низкоуровневые фабрики, а высокотехнологичные, с небольшим объемом спроса стоят и ждут. Проверяем на формуле с объемом в процентах.

2) Необходимо усилить влияние малого значения количества.

## Навигация

Целью навигации могут быть сектор или любой объект сектора: корабль, станция, астероид, врата, контейнер и т.п. Для объектов сектора навигация подразумевает два режима: стыковки и следования. Переключение между этими режимами выполняется с помощью дополнительного аргумента - точности следования. Указание этого аргумента означает выбор режима следования, независимо от типа цели. Если точность следования не указана, это означает выбор режима стыковки с целью.

Программа навигации не выполняет никаких проверок цели (это должно осуществляться пользовательским кодом), а лишь пытается выполнить очередной этап навигации к ней. Навигация выполняется поэтапно, предоставляя пользовательскому коду возможность выполнить необходимые процедуры между этапами (например, проверку валидности цели). Если очередной этап завершился достижением цели, то программа завершается с истинным результатом. В противном случае работа программы завершается ложным результатом. В качестве примеров навигационных этапов можно привести: следовать в следующий по маршруту сектор, прыгнуть к ближайшим к цели вратам, стыковаться с целью, следовать за целью и т.п.

Если навигационный этап не завершился достижением цели, это может означать как то, что был выполнен очередной навигационный этап и можно приступить к выполнению следующего, так и то, что цель принципиально не достижима. Второе возможно например, когда цель была уничтожена, или когда в режиме стыковки указан целевой объект, стыковка с которым принципиально невозможна (стыковка с астероидом, стыковка с кораблем, не имеющим стыковочных узлов). Анализ причин завершения навигационной программы ложным результатом полностью лежит на пользовательском коде (усложнять навигационную программу смысла нет, так как все равно без должного анализа ситуации ничто не помешает пользовательскому коду бесконечно вызывать программу навигации для некорректной цели).

Задача навигации считается выполненной, если осуществлена стыковка с целью в режиме стыковки или цель достигнута с указанной точностью в режиме следования. Что касается точности следования, то здесь имеется в виду не фактическая, а удовлетворительная точность по версии движка (то есть, когда соответствующая навигационная команда завершается с результатом «цель достигнута»). Фактическое же расстояние между кораблем и целевым объектом может превышать указанную точность. Но в общем, завершение навигационной программы с истинным результатом можно рассматривать как разрешение на выполнение последующих действий. Например, при следовании за кораблем с целью дистанционного обмена грузом, достижение навигационной цели можно рассматривать как получение технической возможности обмена грузом.

Что делать, если цель это корабль и он пристыкован к другому объекту, к которому инициатор не факт что может пристыковаться? На первом этапе не будем обрабатывать ситуацию, когда цель пристыкована. Нужно добавить в соответствующий валидатор отклонение контрагента, если он пристыкован. В навигационной программе сразу возвращать ложный результат, если цель пристыкована (использовать паузу в несколько секунд).

## Типы контрагентов

Контрагент - это второй (противоположный) участник сделки. Говоря контрагент, мы подразумеваем, что некий субъект (инициатор) находится в состоянии осуществления или поиска сделки и рассматривает другой субъект (контрагента) как потенциального участника сделки. По сути, здесь важен не какой-то конкретный противоположный участник, а любой, подпадающий под наши требования (то есть категория субъектов). Например, если корабль находится в процессе закупки товара, то контрагентом для него будет любой продавец искомого товара. Если требуется продать товар, то контрагентом является любой покупатель товара, который готов пойти на сделку. По отношению к инициатору, контрагентов может быть множество (а может не быть ни одного), но независимо от количества, все контрагенты подразумевают принадлежность к одной из категорий: продавцы или покупатели.

В отличии от стандартной игровой ситуации, когда контрагентами могут выступать только станции, для кораблей под управлением Т2 Дроида, контрагентами могут выступать другие корабли. В качестве контрагента-корабля может выступать абсолютно любой корабль (даже корабль не принадлежащий игроку) с одним обязательным условием - интересующий товар должен быть в списке товаров контрагента, а заданные параметры работы должны удовлетворять условиям совершения операции. Это работает точно так же как и для станций, с тем отличием, что параметры работы с товаром, такие как закупочная цена и/или цена реализации, минимальный объем партии для покупки и/или продажи, задаются особым образом (для кораблей игрока через корабельный терминал). И хотя корабли для Т2 являются полноценными контрагентами, между кораблями и станциями, когда они рассматриваются в качестве контрагентов, существуют фундаментальные отличия, что не позволяет ограничиться простой декларацией возможности и требует пояснения некоторых деталей.

Что бы эффективно организовать работу, следует понимать принципиальную разницу между контрагентом кораблем и контрагентом станцией. Прежде всего, у кораблей, как и у некоторых станций (например верфей) складское пространство является общим для всех товаров, хранящихся на складе. Для таких контрагентов расчет объема занимаемого места выполняется по фактическому количеству, а не по стандартному для игры как суммарно-занятое пространство. Это важно понимать, потому что это является триггером: куда вести товар (аналогично и в процессе подсчета спроса).

Прям сейчас работаем: нельзя просто так всем подряд кораблям засовывать в трюм что тебе хочется. Как минимум, это делает любой корабль «подходящим» как для покупки, так и для продажи,что приводит к известному глюку. Настройку списка товаров нужно сделать доступным для любого корабля, даже если на нем не выбрана программа. И через этот список потенциальным контрагентам-кораблям нужно задавать параметры товара. И если товара нет в списке товаров контрагента, или если параметры не подходят, то он такой контрагент должен считаться неподходящим для этой операции.

Кроме этого, при рассмотрении в качестве контрагента корабля, выполняется дополнительная проверка текущего местоположения проверяемого контрагента в рабочей области инициатора. Это логично, ведь корабли могут перемещаться. И если корабль ушел за пределы рабочей области, значит он более недоступен для осуществления операции.

Рассматривать корабль в качестве подходящего контрагента возможно в двух случаях:

1) На одном из кораблей (инициаторе или контрагенте) установлен транспортер;

2) Инициатор может пристыковаться к контрагенту

Эти проверки интегрированы в функции ???531??? проверки возможности стыковки.

Здесь еще можно поработать над опциональным переключением способа влияния времени последнего рейса к контрагенту. В отличии от станций, для кораблей может быть повышенная необходимость в последовательном, а не параллельном распределенном заполнении грузом. Но это совсем «на-потом».

## Валидация контрагентов

Для решения некоторых задач требуется выполнять проверку соответствия контрагента определенным требованиям. Комбинации требований различаются от задачи к задаче, но общее количество требований невелико. Поскольку в условиях X3 SE создавать новый валидатор под каждую ситуацию иррационально, была разработана отдельная процедура, принимающая комбинацию требований в качестве параметра.

Набор требований валидации определяется комбинацией включенных битов дополнительного аргумента, который называется маской валидации. Для формирования маски валидации соответствующие коды объединяются с помощью побитового OR (логического ИЛИ ) в единое значение. Ниже приведена таблица, в которой расшифровываются доступные десятичные коды, соответствующие конкретным битам маски валидации.

| Код | Требование |
| --- | --- |
| 1 | Тип контрагента. Включение этого значения в маску означает проверку (или поиск) поставщика, отсутствие — проверку покупателя. Здесь проверяется разрешена ли соответствующая операция для выполнения на текущем корабле (включена ли покупка/продажа). |
| 2 | Контрагент в принципе продает/покупает (здесь и далее в зависимости от типа контрагента) товар. Поверхностная валидация нужна для отбора потенциальных целей. Например, что бы оценить потенциал того или иного товара в регионе не нужно выполнять проверку рейтинга или возможности стыковки. В этом случае достаточно просто отобрать станции которые соответственно продают или покупают. Контрагенты-корабли проходят данную проверку, если товар имеется в их списке товаров и для него (товара) разрешена соответствующая операция. Корабль-контрагент так же должен иметь техническую возможность транспортировать указанный товар. |
| 4 | Контрагент принципиально доступен для осуществления соответствующей операции. Рабочая валидация нужна для отбора потенциальных контрагентов по отношению к владельцу корабля. То есть, список таких станций, на которые владельцу корабля доступ для операций разрешен в принципе: владелец имеет достаточный рейтинг для стыковки и, в случае поиска поставщика, достаточный рейтинг для покупки указанного товара. Такой фильтр требуется например для решения задач по оценке реально-доступного потенциала товара или поиска станции для конкретной операции. В случае с кораблями, данный тест проверяет принадлежность обоих кораблей одному владельцу. |
| 8 | Проверка текущих параметров контрагента. Пред- и послерейсовая валидация подразумевает соответствие контрагента установленным требованиям по цене и объему. Используется как для проверки контрагента перед отправлением корабля в рейс и после прибытия на станцию, так и для формирования списка соответствующих контрагентов в моменте. Если контрагент корабль, то его настройки объемов и минимальной партии в оценке ситуации не учитываются (эти настройки предназначены только для ограничения собственных рейсов). |
| 16 | Проверка технической возможности осуществления операции конкретным кораблем. Сюда так же входит проверка способности корабля транспортировать товар. В случае двух кораблей проверяется возможность стыковки с кораблем-контрагентом или наличие транспортера на одном из кораблей. Если в процедуру валидации был передан список секторов, контрагент-корабль должен находиться в одном из указанных секторов. Если секторы не указаны, то проверка не выполняется. |
| 32 | Проверка лимитов (мин. партия для покупки/продажи). В случае проверки потребителя выполняется проверка наличия на счете контрагента средств, для покупки минимальной партии. А в случае поставщика - наличие средств для покупки минимальной партии самим кораблем. Наличие средств не выполняется, если контрагент является станцией приписки корабля. |

Таким образом, маска со всеми включенными проверками соответствует значению 63 для поставщика и 62 для потребителя.

## Структуры данных

Базовые структуры данных представлены одномерными массивами. Высокоуровневые структуры данных могут содержать вложенные структуры. Ниже описаны существующие структуры данных с расшифровкой значений элементов.

**WareStruct** - Параметры работы с товаром

0 - WareStruct-3.01

1 - товар

2 - закупочная цена (null - закупка отключеня)

3 - цена реализации (null - реализация отключена)

4 - лимит мин. партии для покупки (% от суммарного места под товар в

трюме\*, null - отключен)

5 - лимит мин. партии для продажи

6 - необходимая для начала рейса минимальная заполненность склада поставщика

(% от склада станции, null - проверка отключена)

7 - максимальная заполненность склада потребителя

\* Суммарное место под товар в трюме определяется как объем, уже занятый товаром,

плюс свободный объем трюма за вычетом объема, необходимого для полной дозаправки

**WorkspaceStruct** - Параметры рабочей области

0 - WorkspaceStruct-3.01

1 - центральный сектор

2 - радиус рабочей области в прыжках

3 - флаги\*

4 - массив дополнительных секторов

5 - массив запрещенных секторов

6 - массив станций-поставщиков

7 - массив станций-потребителей

8 - массив запрещенных станций

\* флаги

1 - работать только со станциями из списков

**StatStruct** - Cтатистика

0 - StatStruct-3.01

1 - временная метка (интерпретируется в зависимости от контекста)

2 - суммарное время работы (здесь и далее секунд)

3 - суммарное время простоя

4 - время на закупках/загрузках

5 - время на продажах/разгрузках

6 - время ожидания поставщика

7 - время ожидания потребителя

8 - сумма закупки

9 - сумма реализации

10 - затраты на обслуживание

**ShipStruct** - Конфигурация корабля

0 - ShipStruct-3.01

1 - код программы

2 - флаги\*

3 - руководитель (если корабль под руководством другого)

4 - текущий статус работы\*\*

5 - WorkspaceStruct

6 - WareStruct

7 - StatStruct (сводная статистика по работе)

8 - массив StatStruct

9 - В зависимости от программы:

Шеф - массив кораблей в управлении

10 - массив WareStruct

\* флаги

1 - журналирование

2 - разрешить апгрейд

4 - разрешить ремонт

8 - переименован

\*\* статус работы

0 - работает

1 - приостановлен

2 - завершить и сообщить

**WareRatingStruct** — статистика и рейтинг товара в регионе

0 — WareRatingStruct-3.01

1 — товар

2 — количество поставщиков

3 — количество потребителей

4 — суммарное предложение в единицах объема

5 — суммарный спрос в единицах объема

6 — пункты рейтинга

7 — строка рейтинга в формате, пригодном для сортировки товаров по рейтингу с дополнительной информацией

## Особенности работы некоторых команд

Команда ***Приостановить*** заставляет целевой корабль ожидать на месте до возобновления или прерывания работы. Корабль остается в том окружении в котором был на момент приостановки: если был в пути, то ожидает в космосе; если был пристыкован, то ожидает на станции.

Команда ***Завершить работу*** для программы Транспорт выполняется следующим образом: в соответствии с установками распродаются все товары в трюме, после чего работа прекращается.

## Программа Старатель

Рабочая область определяет секторы добычи и реализации минералов. Список поставщиков не используется и не отображается в интерфейсе. В список товаров могут быть добавлены только минералы. Добыча минерала может быть включена или отключена путем переключения соответствующей опции (фактически выставляет в качестве закупочной цены минимум или null).

Перед началом работы трюм освобождается по схеме, аналогичной транспорту: определяется товар первой категории, после чего выполняется цикл продаж и так до тех пор, пока в трюме имеется товар первой категории.

В случае отсутствия товара первой категории, определяется наиболее востребованный товар из списка: для каждого товара рассчитывается статистика, после чего товары сортируются по спросу и выбирается товар с наибольшим спросом. Если в списке нет товара с ненулевым спросом, то корабль некоторое ожидает время изменения ситуации. Если подходящий товар выбран, то корабль следует в случайный сектор рабочей области и начинает добычу.