**Блокировки**

**Блокировка** — это информация о том, что данный ресурс захвачен (заблокирован) «кем-то», для выполнения какого-то действия.

**Существует два вида блокировок:**

1. **Объектные блокировки**

Предназначенение: обеспечить конкурентный доступ пльзователей к данным в терминах объектов информационной базы. Здесь мы блокируем не запись в таблице базы даных, а объект информационной базы (документ, справочник и т.д.)

1. **Транзакционные блокировки**

Предназначение: обеспечить конкурентный доступ пользователей к данным в транзакциях в терминах таблиц базы данных.

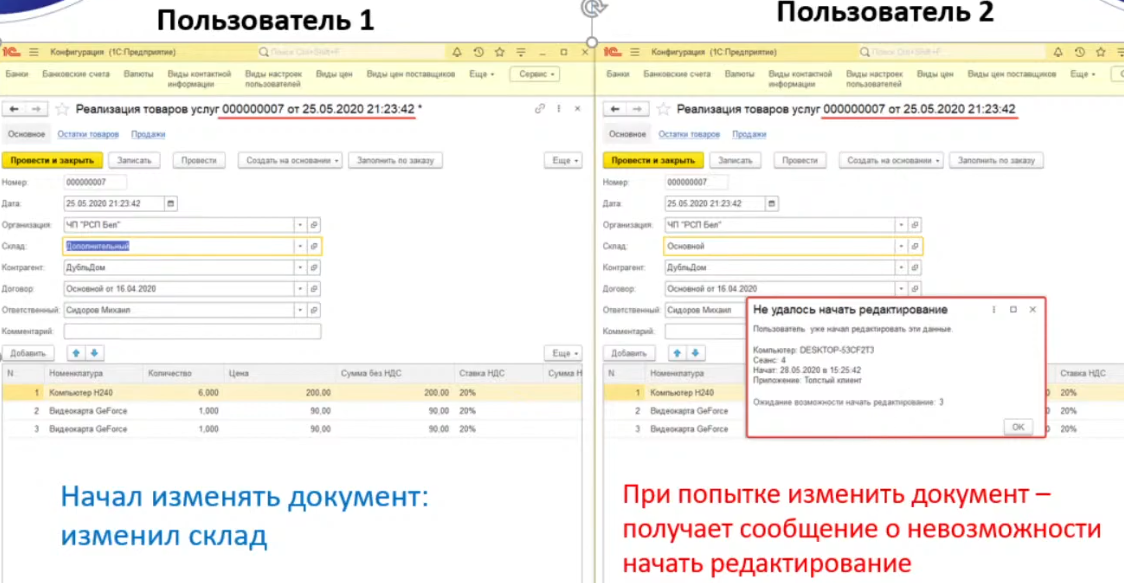
Применяется только в транзакциях.

**Объектные блокировки**

**Виды объектной блокировки:**

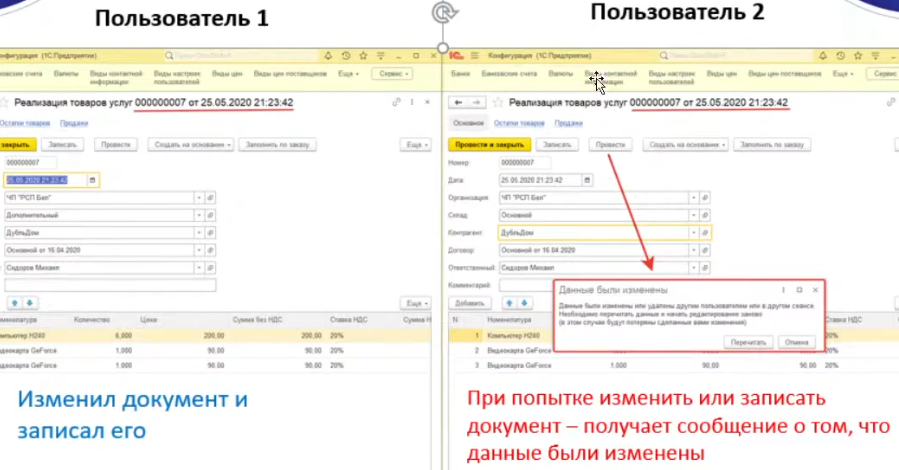
* **Пессиммистичечкая** — предназначена для того чтобы запретить пользователю изменять объект, если этот объект параллельно изменяется другим пользователем.

Срабатывает у Пользователя 2 тогда когда Пользователь 1 начал изменять объект, но еще **не записал** его в базу данных.



* **Оптимистическая** — предназначена для того, чтобы запретить запись объекта в базу данных, если после считывания объекта он был изменен в базе данных другими сеансами или другими программными объектами этого же сеанса. Гарантирует, что пользователь изменяет актуальню версию объекта, а не устаревшую.

Срабатывает у Пользователя 2 тогда когда Пользователь 1 изменил объект и **записал** его в базу данных. Пользователю 2 нужно перечитать данные, чтобы работать с актуальной версией данных.



**Важно:**

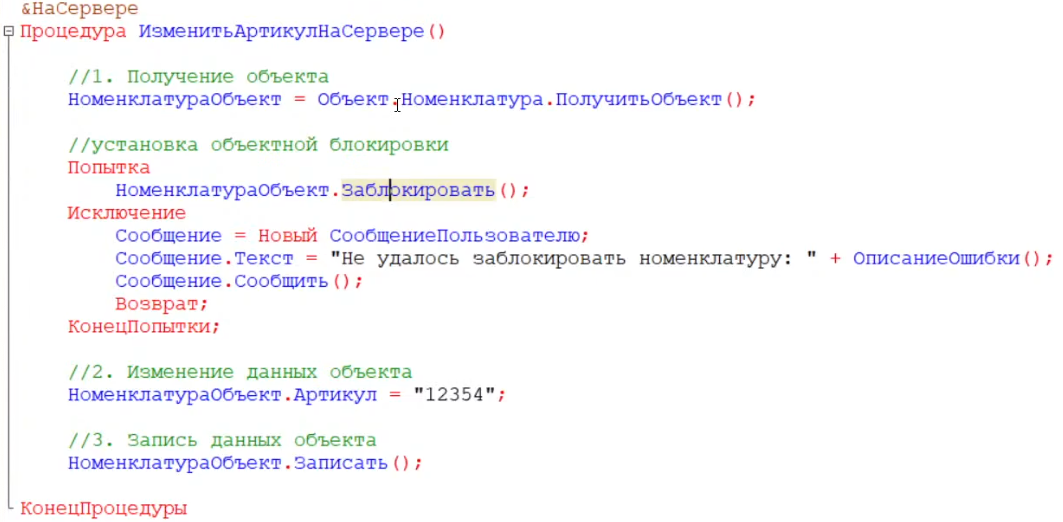
Прежду чем изменять данные объекта из кода нужно предварительного его заблокировать (установить объектную блокировку). В противном случае (если не попытаться заблокировать) будет такая ситуация когда алгоритм в коде отработает, а пользователь увидит сообщение о том, что нужно перечитать новые данные и данные которые пользователь уже внес в документ **не сохраняться**.

**Исключения:**

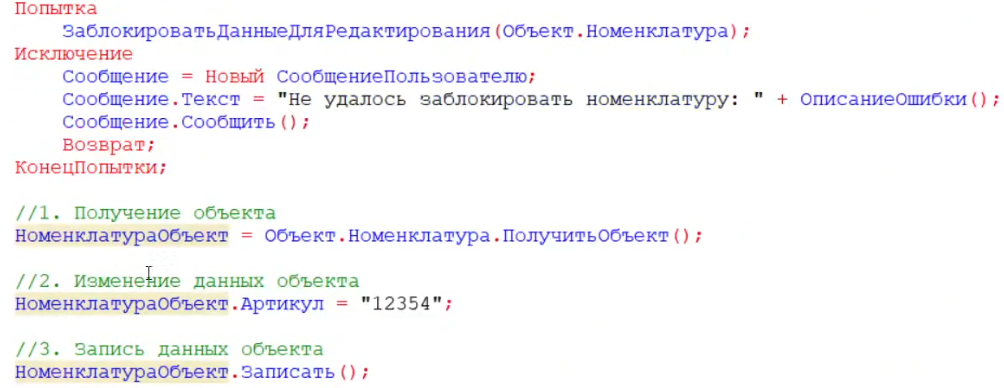
* Изменение данных происходит в монопольном режиме, т. е. При выполнении кода гарантированно не может произойти изменение данных объекта в другом сеансе.
* Операции, которые по логике имеют более выскоий приоритет по сравнению с интерактивными дейсвиями пользователя.

Чтобы заблокировать объект используется 2 метода:

1. **Заблокировать()** — в этом случае сначала считываем данные объекта, а потом пытаемся заблокировать объект. Тоесть сначала нужно выполнить метод ПолучитьОбъект().



1. **ЗаблокироватьДанныеДляРедактирования()** — в этом случае мы сначала пытаемся установить блокировку, а уже потом считываем данные (используем метод ПолучитьОбъект()).



**Транзакционные блокировки**

Механизм **транзационных** блокировок позволяет осуществлять конкурентный доступ пользователей к данным СУБД в терминах базы данных **внутри транзакции**.

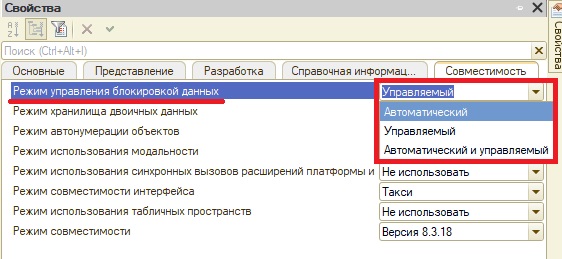
Транзакционные блокировки существуют **только внутри транзакции**.

  
**Как это должно работать:**

Транзакция 1 перед тем как прочитать остатки будет устанавливать блокировку, затем она прочитает остатки, а затем она уменьшит остатки и на этом её транзакция закончится. Если вдруг параллельно с Транзакцией 1 другой пользователь захочет прочитать остатки, то он тоже должен будет установить блокировку, но поскольку одна блокировка уже установлена, другой пользователь не сможет её установить и в этом момент возникнет так называемое «ожидание на блокировки». В итоге Транзакция 2 будет ожидать пока не закончится Транзакция 1, как только Транзакция 1 закончится тогда Транзакция 2 считает корректные остатки.

**Где устанавливаются блокировки ?**

На это влияет свойство конфигурации «Режим управления блокировкой данных»



**Автоматический (считается устаревшим)** — работает менеджер блокировкок на стороне сервера СУБД.

**Управляемый** — помимо менеджера блокировок на стороне сервера СУБД работает менеджер блокировок на стороне сервера 1С:Предприятие.

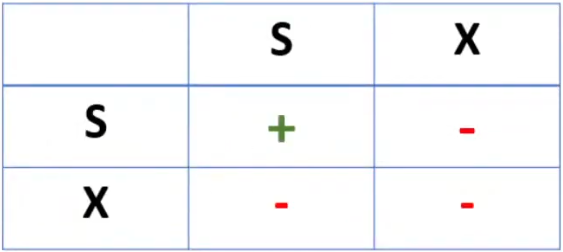
В явном виде в коде на встроенном языке 1с можн устанавливать только управляемые блокировки.

Блокировки на уровне СУБД устанавливаются автоматически.

**Типы управляемых блокировок:**

1. **Разделяемая (Shared)(S)** — ползволяет конкурирующему процессу установить **разделяемую** блокировку по этому же условию, но не позволит установить **исключтельную** блокировку.
2. **Исключительная (Exclusive)(X)** — не позволит конкурирующему процессу установить **разделяемую** или **исключительную** блокировку по этому же условию.

**Совместимость блокировок**

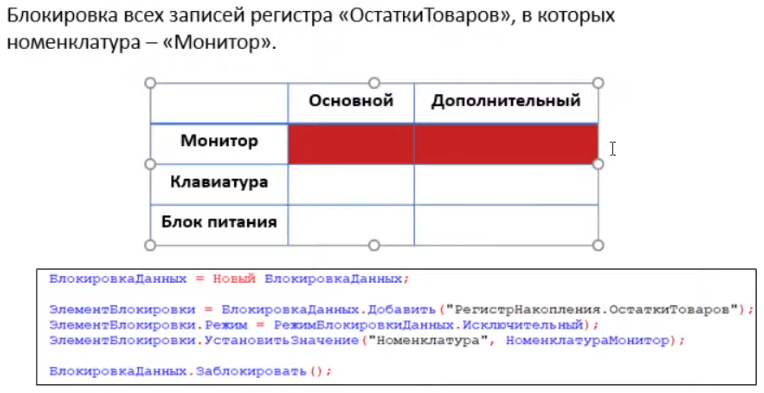
  
  
  
  
  
  
  
  
  
  
  
  
**Когда нужно устанавливать явную управляемую блокировку ?**В случаях, когда сначала выполняется чтение данных, а потом изменение этих же данных.

Блокировку нужно устанавливать **ДО** чтения (выполнения запроса), тип блокировки — **исключительная**.

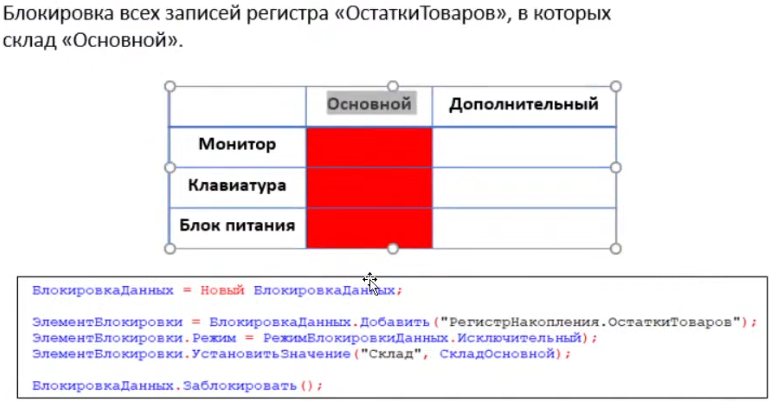
Рисунок с учетом исключительной блокировки

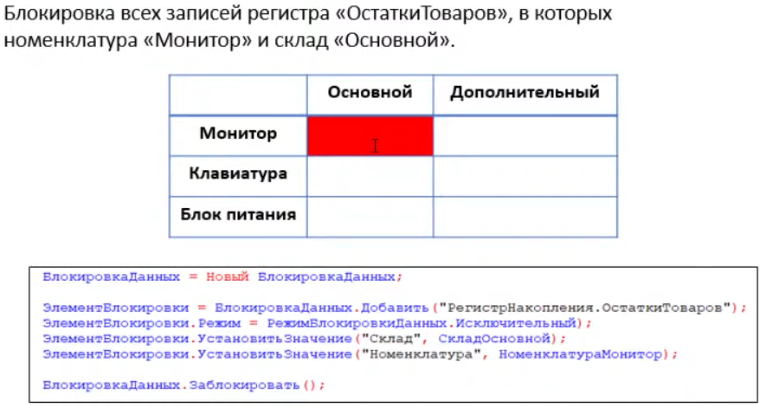
  
  
  
  
  
  
  
  
  
  
  
  
  
Примеры блокировки:

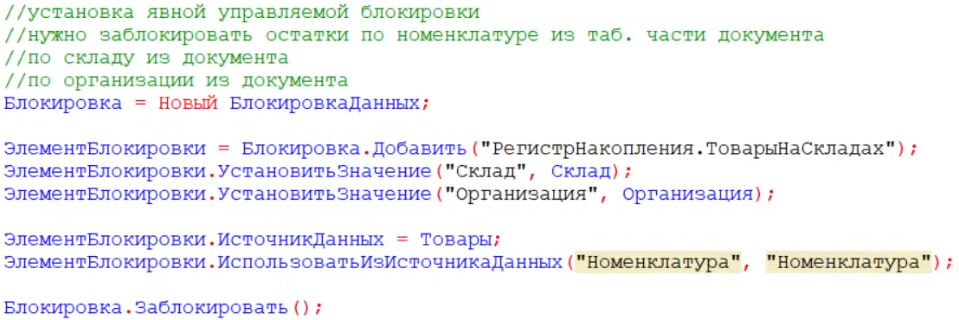
1. **Заблокируются остатки номенклатураы«Монитор» для всех складов.**



1. **Заблокируются все записи регистра в которых склад «Основной».**



1. **Заблокируются все записи в которых номенклатура «Монитор» и склад «Основной».**  
   
2. **В регистре накопления ТоварыНаСкладах заблокируются записи с выбранным складом, организицией и номенклатурой из табличной части Товары.**

****