

## Лабораторная работа № 7

### Создание информационной системы для регистрации продаж в студенческом киоске

#### 1. Задание:

Заказчик просит создать небольшую информационную систему для регистрации продаж в студенческом киоске.

- В киоске продают канцелярские принадлежности, литературу и булочки. Нужно отобразить только факт продажи товаров в киоске.
- В результате выполнения лабораторной работы должен получиться отчет вида:

Товар	Продано
Пончик	3
Кола-кола	5

#### 2. Подготовка:

- создать новую информационную базу;
- открыть информационную базу в режиме «Конфигуратор»;
- открыть окно конфигурации.

#### 3 Выполнение:

В данном справочнике будет храниться перечень товаров, продаваемых в киоске. Создать новый справочник «Товары», как показано на рисунке 1.

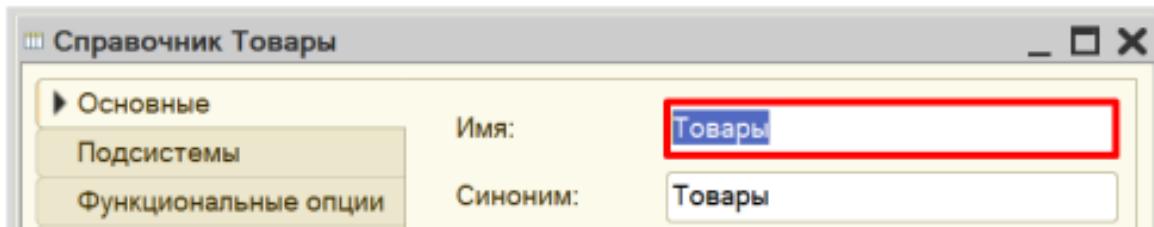


Рисунок 1 – Создание справочника

Открыть справочник в режиме «1С:Предприятие» и добавить несколько товаров, например, «Ручка», «Линейка», «Пончик», как показано на рисунке 2.

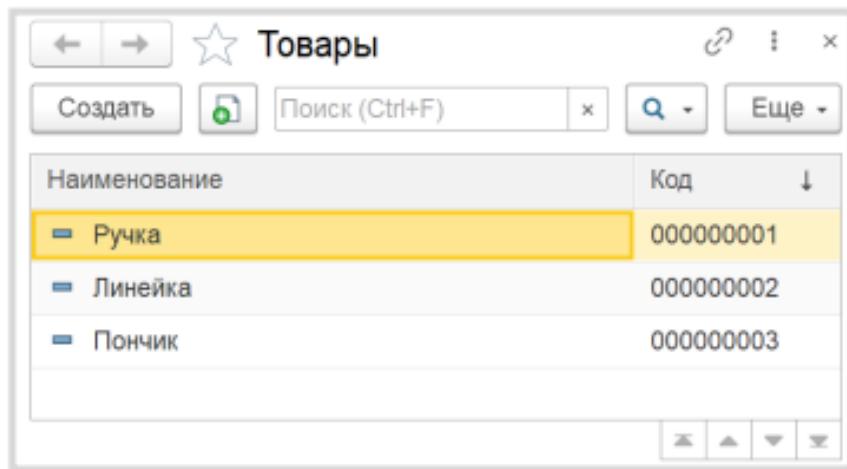


Рисунок 2 – Добавление товаров в справочник

Создав справочник, мы организовали хранение объектов аналитики. Далее следует ответить на вопрос: «Как мы будем регистрировать продажи?».

Для данной цели подойдет объект конфигурации документ.

Добавим новый документ «Продажа», проверить результат, как показано на рисунке 3.

Поля «Номер» и «Дата» система создала автоматически при добавлении нового документа. Это стандартные реквизиты документа. Ознакомиться с перечнем всех стандартных реквизитов можно на вкладке «Данные» окна редактирования документа.

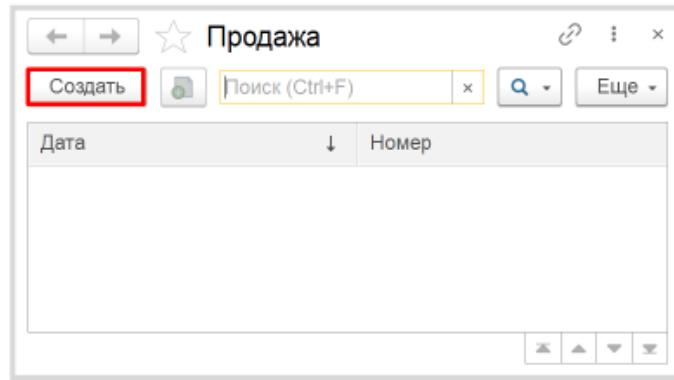


Рисунок 3 – Запуск в режиме 1С:Предприятие

Далее следует определиться со структурой документа. Мы хотим, чтобы в одном документе можно было регистрировать продажу сразу нескольких товаров. В этом нам поможет табличная часть документа. Создадим табличную часть, как показано на рисунке 4.

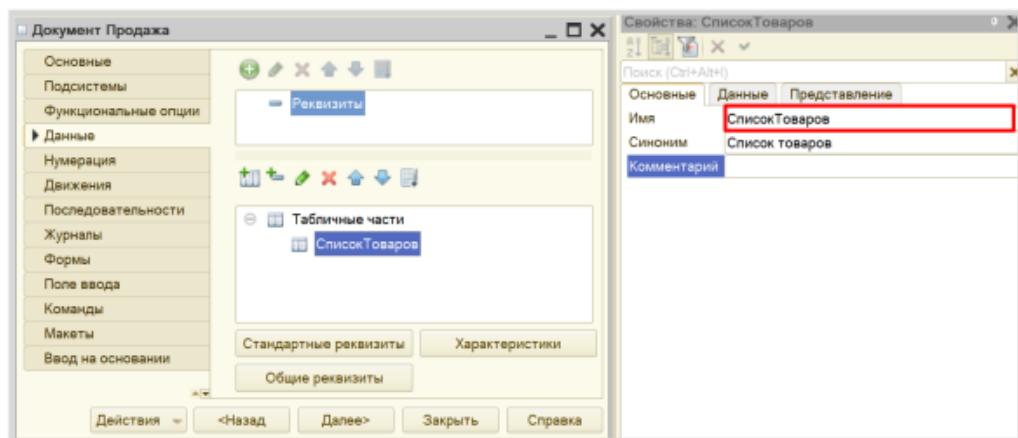


Рисунок 4 – Создание табличной части документа

Далее нужно добавить реквизиты табличной части – колонки таблицы. Для поставленной задачи создайте колонки «Товар» и «Количество». Для добавления реквизита воспользуйтесь кнопкой «Добавить реквизит».

Для каждого из реквизитов нужно определить имя и тип данных:

- реквизит «Товар» должен иметь тип «СправочникСсылка.Товары», чтобы хранить ссылку на элемент справочника «Товары»;
- реквизит «Количество» должен быть числовым.

Проверить работу системы, добавив 5-10 товаров, как показано на рисунке 5.

N	Товар	Количество
1	Пончик	4
2	Ручка	5
3	Линейка	2

Рисунок 5 – Добавление товаров

Можно ли теперь на основе таких документов построить отчет по продажам? Можно, но для этого придется прибегнуть к грубому перебору всех существующих документов. Данный вариант является неправильным, потому что, если таких документов окажется очень много, система будет требовать большого количества ресурсов и времени. Для решения данной проблемы и ускорения процесса извлечения данных создадим еще один объект – регистр накопления.

Регистр накопления – это такая итоговая таблица, которая может автоматически считать какие-либо элементы, например, денежные средства, материалы.

Данный регистр должен быть оборотным, чтобы накапливать данные о продаже товаров, как показано на рисунке 6.

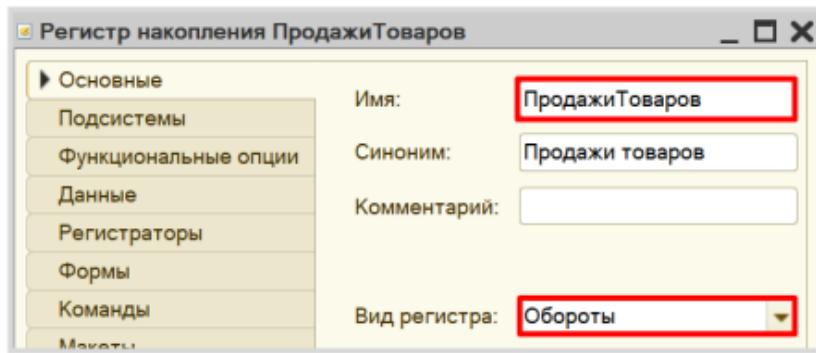


Рисунок 6 – Создание регистра накопления

Переходим к описанию структуры регистра накопления. Для этого переходим на вкладку «Данные».

Заполнение данного окна всегда проще всего начинать с добавления ресурса. Чтобы понять, что использовать в качестве ресурса, необходимо задать вопрос: «Что нам нужно считать?». Нам нужно считать количество. Следовательно, количество и будет являться ресурсом. Тип данного реквизита – «Число».

Чтобы разобраться с измерением, нужно понять, в разрезе чего мы хотим считать количество. Мы хотим считать количество (чего?) товаров. Значит, в качестве измерения следует добавить реквизит «Товар». Тип данного реквизита – «СправочникСсылка.Товары».

Чтобы регистр накопления заработал, нужно сделать следующее:

1. Определить источники данных, которые должны попадать в регистр (определить документы-регистраторы).
2. Описать, каким образом данные из документа-регистратора должны попадать в регистр.

В нашем случае, данные в регистр накопления должны попадать из документа «Продажа», следовательно, данный документ и будет являться регистратором. Для определения регистратора регистра накопления необходимо выбрать нужный документ на вкладке «Регистраторы», как показано на рисунке 7.

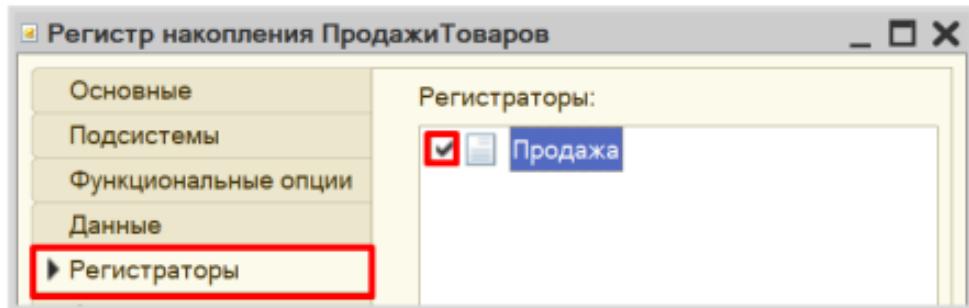


Рисунок 7 – Создание регистратора

Чтобы определить правила передачи данных из регистратора в регистр накопления, откроем вкладку «Движения» окна редактирования документа «Продажа» и воспользуемся конструктором движений.

Если все было сделано правильно, имена и типы реквизитов совпадают, то при нажатии на кнопку «Заполнить выражения» реквизиты регистра заполняются автоматически. Если этого не произошло, то заполните поле «Выражение» вручную, путем выбора соответствующих реквизитов документа. При нажатии на кнопку «OK» система сформирует программный код, который при успешном проведении документа произведет движения в регистр накопления, то есть скопирует данные из документа в регистр накопления, как показано на рисунке 8.

```
Процедура ОбработкаПроведения (Отказ, Режим)
//||_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр ПродажиТоваров
Движения.ПродажиТоваров.Записывать = Истина;
Для Каждого ТекСтроКаСписокТоваров Из СписокТоваров Цикл
    Движение = Движения.ПродажиТоваров.Добавить ();
    Движение.Период = Дата;
    Движение.Товар = ТекСтроКаСписокТоваров.Товар;
    Движение.Количество = ТекСтроКаСписокТоваров.Количество;
КонецЦикла;

///|_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Рисунок 8 – Код модуля объекта «Продажа»

Проверим работу регистра накопления. В первую очередь, необходимо перепровести (провести заново) созданный нами ранее документ, чтобы он сформировал движение в регистр накопления. Чтобы посмотреть на движения в регистре накопления воспользуемся функциями для технического специалиста, как показано на рисунке 9.

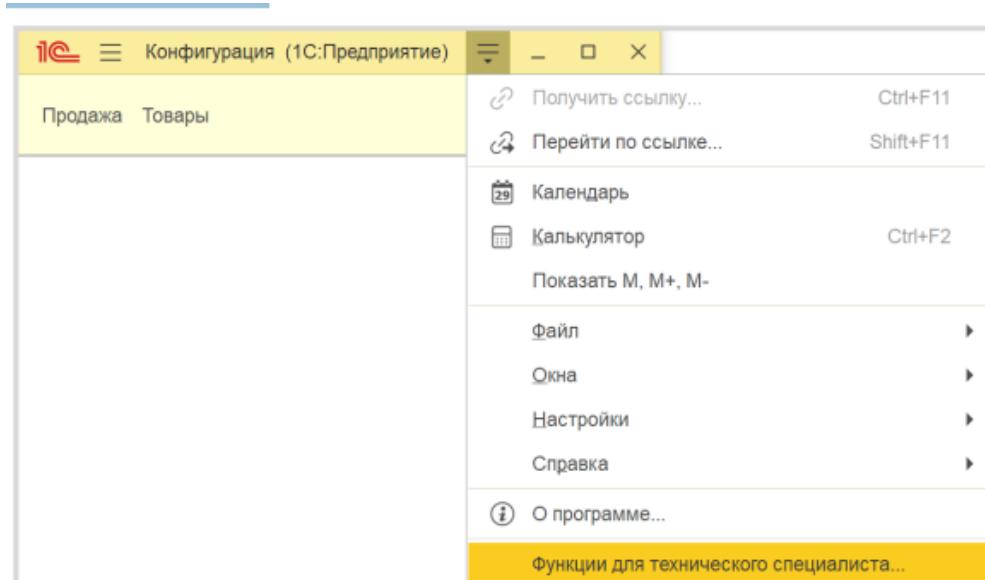


Рисунок 9 – Функции технического специалиста

Найдем наш регистр, откроем его и посмотрим на движения, как показано на рисунке 10, 11.

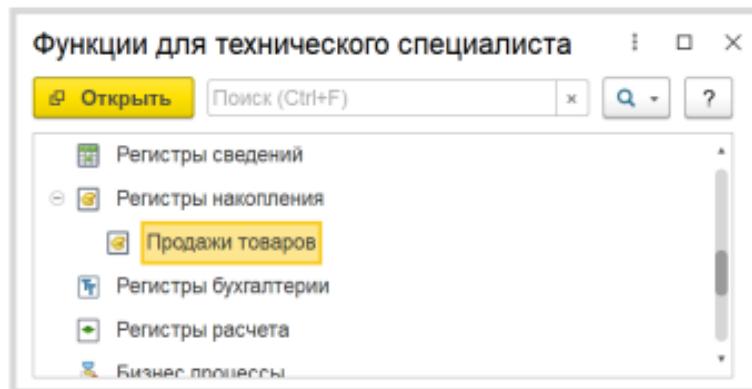


Рисунок 10 – Регистр накопления «Продажа товара»

Период	Регистратор	Номер ...	Товар	Количество
• 28.08.2020 18:27:52	Продажа 000000001 от 28.08.2020 ...	1	Линейка	3
• 28.08.2020 18:27:52	Продажа 000000001 от 28.08.2020 ...	2	Ручка	6

Рисунок 11 – Регистр накопления «Продажа товара»

Таким образом, регистр накопления является своеобразной итоговой таблицей, куда заносятся данные из документа «Продажа». Теперь можно строить отчет.

Создадим новый отчет «ОтчетПоПродажам». Для наполнения отчета воспользуемся конструктором схемы компоновки данных. Все созданные нами объекты конфигурации представляют собой таблицы базы данных. В режиме «1С:Предприятие» мы заполняем эти таблицы данными. Чтобы получить эти данные для отображения в отчете, нужно сформировать запрос к базе данных.

Затем воспользуйтесь конструктором запроса.

Данные будем брать не из регистра накоплений напрямую, а из виртуальной таблицы, которую создает этот регистр автоматически. Эта виртуальная таблица позволит получить уже просуммированные значения по всем документам.

В результате данное окно должно быть заполнено следующим образом, как показано на рисунке 12.

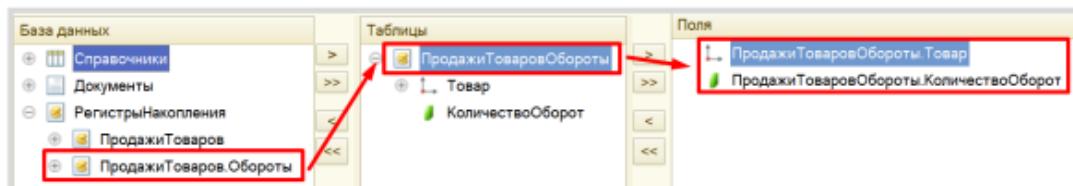


Рисунок 12 – Заполнение конструктора запросов

Запрос к базе данных сформирован. Теперь необходимо настроить внешний вид отчета. Для этого нужно перейти на вкладку «Настройки».

Для настройки внешнего вида отчета воспользуемся конструктором настроек отчета.

Наш отчет будет иметь форму списка.

Далее нужно выбрать поля, которые будут отображены в отчете. Обратите внимание на порядок полей в правом столбце, именно в таком порядке они будут отображены в отчете, как показано на рисунке 13.

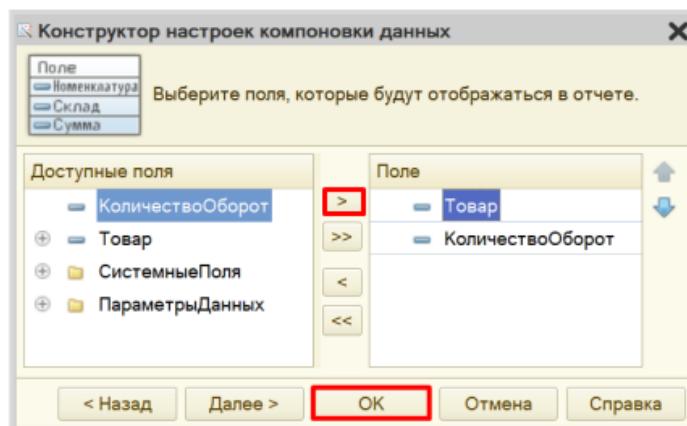


Рисунок 13 – Настройка отчета

Чтобы удостовериться, что отчет работает корректно, можно создать, наполнить и провести еще несколько документов «Продажа», а затем посмотреть на результаты в отчете. Если вы все сделали верно, то товары, указанные в документах, должны быть просуммированы, как показано на рисунке 14.

Товар	Количество Оборот
Пончик	12
Ручка	15
Линейка	6

Рисунок 14 – Итоговый отчет

## **Лабораторная работа № 8**

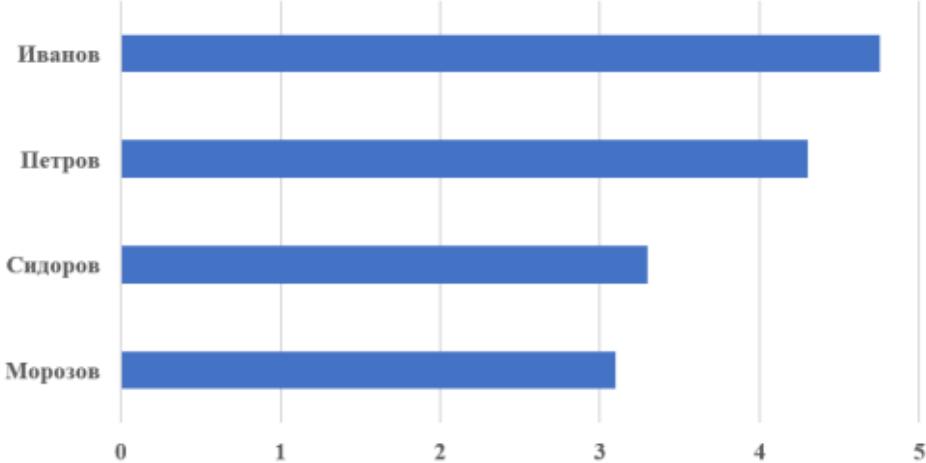
### **Разработка конфигурации для учета работы студентов на занятиях**

#### **1 Задание:**

Заказчик просит разработать конфигурацию для учета работы студентов на занятиях. Учет ведется в разрезе дисциплин.

- В системе необходимо регистрировать Занятия студентов. В конце занятия пользователь в шапке документа указывает название дисциплины, а в табличной части – какие студенты какой балл получили.
- Нужно построить Отчет по текущей успеваемости студентов.

Форма отчета:



Отчет строится по среднему арифметическому баллу студента по указанной дисциплине.

#### **2 Подготовка:**

- создать новую информационную базу;
- открыть информационную базу в режиме «Конфигуратор»;
- открыть окно конфигурации.

### **3 Выполнение:**

Из условия следует, что необходимо хранить информацию о студентах и посещаемых ими дисциплинах. Для решения этой задачи нам понадобятся справочники «Студенты», «Дисциплины».

Откроем программу в режиме «1С:Предприятие» и добавим в каждый справочник несколько элементов: 5 студентов, 5 дисциплин.

Для регистрации занятий студентов следует воспользоваться объектом конфигурации документ. Создайте документ «ЗанятияСтудентов».

Под шапкой документа подразумеваются данные, хранящиеся в верхней части документа до табличной части. Добавим реквизит «Дисциплина». Тип: «СправочникСсылка.Дисциплины».

Сделаем реквизит обязательным для заполнения. Проверка заполнения: «Выдавать ошибку». С помощью такой настройки пользователь не сможет сохранить документ, пока не заполнит поле «Дисциплина».

Далее, исходя из условия, нам понадобится добавить табличную часть «СписокСтудентов». Теперь добавим два реквизита табличной части: «Студент» (тип – СправочникСсылка.Студенты) и «Оценка» (тип – «Число»).

Запустим режим «1С:Предприятие» и попробуем создать несколько документов, результат заполнения представлен на рисунках 1, 2.

N	Студент	Оценка
1	Любимов	4
2	Марковский	3
3	Малиновский	3
4	Федоренко	2

Рисунок 1 – Заполнение документа занятия студентов

Дата	Номер	Дисциплина
09.09.2020 12:00:00	00000005	Высшая математика
10.09.2020 12:00:00	00000001	Операционные системы
11.09.2020 12:00:00	00000002	Операционные системы
12.09.2020 12:00:00	00000003	Операционные системы
13.09.2020 20:05:51	00000004	Высшая математика

Рисунок 2 – Заполнение документа занятия студентов

Можно ли теперь на основе таких документов построить отчет об успеваемости? Можно, но для этого придется прибегнуть к грубому перебору всех существующих документов. Данный вариант является неправильным, потому что, если таких документов окажется очень много, система будет требовать большого количества ресурсов и времени.

Для решения данной проблемы и ускорения процесса извлечения данных создадим еще один объект – регистр накопления.

Добавим новый регистр накопления «УспеваемостьСтудентов». Вид данного регистра – «Обороты».

Как и в случае с документами, для формирования структуры переходим на вкладку «Данные».

Структура регистра накопления отличается от структуры документа. Заполнение данного окна проще всего начинать с добавления ресурса. Чтобы понять, что использовать в качестве ресурса, следует задать вопрос: «Что мы хотим накапливать/считать в данном регистре?». Мы хотим считать оценки. Следовательно, оценка и будет являться ресурсом. Тип данного реквизита – «Число» (неотрицательное).

Чтобы разобраться с измерением, необходимо понять, в разрезе чего мы хотим считать оценки. Мы хотим считать оценки (кого?) студентов в разрезе (чего?) дисциплин. Значит, в качестве измерений нужно добавить реквизиты

«Студент» (тип – «СправочникСсылка.Студенты») и «Дисциплина» (тип – «СправочникСсылка.Дисциплины»). Поставить галочку «Запрет незаполненных значений», «Использование в итогах».

Чтобы регистр накопления заработал, необходимо сделать следующее: 1. Определить источники данных регистра (определить документы-регистраторы). 2. Описать, каким образом данные из документа-регистратора должны попадать в регистр. В нашем случае на оценку студента будет влиять единственный документ. Определим его в качестве документа-регистратора на вкладке «Регистраторы».

Далее для нашего документа необходимо описать процедуру копирования данных в регистр накопления. Откроем окно редактирования данного документа на вкладке «Движения». Воспользуемся конструктором движений.

Если все было сделано правильно, имена и типы реквизитов совпадают, то при нажатии на кнопку «Заполнить выражения» реквизиты регистра заполняются автоматически. Если этого не произошло, то заполните поле «Выражение» вручную, путем выбора соответствующих реквизитов документа.

При нажатии на кнопку «OK» система сформирует программный код, который при успешном проведении документа произведет движения в регистр накопления, то есть скопирует данные из документа в регистр накопления, как показано на рисунке 3.

The screenshot shows a window titled "Документ ЗанятияСтудентов: Модуль объекта". Inside, there is a code editor with the following text:

```
Процедура ОбработкаПроведения(Отказ, Режим)
//{{_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
//  Данный фрагмент построен конструктором.
//  При повторном использовании конструктора, внесенные вручную
//  изменения будут потеряны.
// регистр УспеваемостьСтудентов
Движение.УспеваемостьСтудентов.Записывать = Истина;
Для Каждого ТекСтрокаСписокСтудентов Из СписокСтудентов Цикл
    Движение = Движения.УспеваемостьСтудентов.Добавить();
    Движение.Период = Дата;
    Движение.Студент = ТекСтрокаСписокСтудентов.Студент;
    Движение.Дисциплина = Дисциплина;
    Движение.Оценка = ТекСтрокаСписокСтудентов.Оценка;
КонецЦикла;
//}}_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Рисунок 4 – Код модуля объекта занятия студентов

Откроем систему в режиме «1С:Предприятие» и проверим работу регистра накопления.

В первую очередь, необходимо перепроверить (проводить заново) созданный документ «Занятия студентов». Без проведения документов данные не будут скопированы в регистр накопления.

Обратите внимание, что на главной странице системы не создала кнопку открытия регистра накопления. Это связано с тем, что все расчеты в регистрах накопления происходят в фоновом режиме, «за кадром», то есть пользователю о них знать не нужно вовсе. Поэтому по умолчанию регистры накопления настраивают так, чтобы пользователи не имели к ним доступа.

Но мы, будучи разработчиками, можем обратиться к любому объекту конфигурации. Для этого воспользуемся функциями для технического специалиста.

В открывшемся списке найдем созданный нами регистр накопления и откроем его, результат представлен на рисунке 5.

Период	↓	Регистратор	Номер строки	Студент	Дисциплина	Оценка
• 11.09.2020 12:00:00		Занятия студентов ...	4	Федоренко	Операционные си...	3
• 12.09.2020 12:00:00		Занятия студентов ...	1	Любимов	Операционные си...	5
• 12.09.2020 12:00:00		Занятия студентов ...	2	Марковский	Операционные си...	4
• 12.09.2020 12:00:00		Занятия студентов ...	3	Малиновский	Операционные си...	3
• 12.09.2020 12:00:00		Занятия студентов ...	4	Федоренко	Операционные си...	2
• 13.09.2020 20:05:51		Занятия студентов ...	1	Любимов	Высшая математ...	4
• 13.09.2020 20:05:51		Занятия студентов ...	2	Марковский	Высшая математ...	3
• 13.09.2020 20:05:51		Занятия студентов ...	3	Малиновский	Высшая математ...	3
• 13.09.2020 20:05:51		Занятия студентов ...	4	Федоренко	Высшая математ...	2

Рисунок 5 – Регистр накопления

Таким образом, регистр накопления является некоторой итоговой таблицей. Сюда заносятся данные из документов-регистраторов по определенным правилам. Мы реализовали движение информации об оценках студентов для последующего расчета среднего балла.

Построим отчет. Для этого воспользуемся соответствующим объектом конфигурации.

Добавим отчет «УспеваемостьСтудентов». Воспользуемся схемой компоновки данных.

Добавим новый запрос к базе данных.

Для формирования запроса воспользуемся конструктором запроса.

Данные будем брать из регистра накоплений напрямую, чтобы иметь возможность рассчитывать средний балл. Чтобы перенести данные из одного окна в другое, просто перетащите нужные поля с помощью мыши либо воспользуйтесь стрелочками, расположенными между окнами. В результате данное окно должно быть заполнено следующим образом, как показано на рисунке 6.

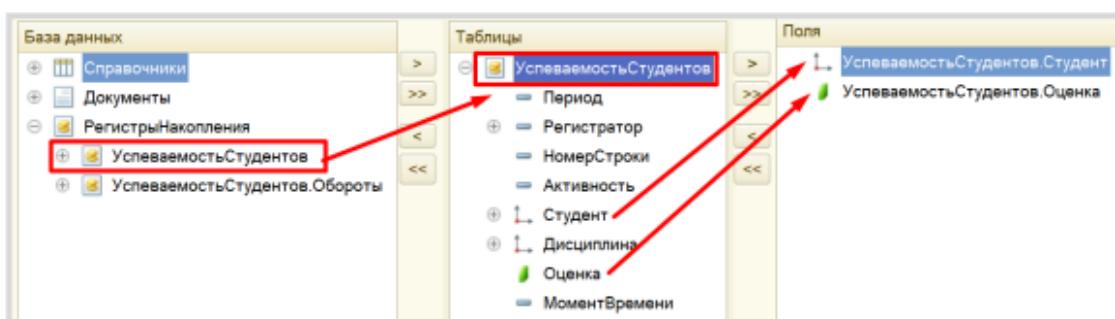


Рисунок 6 – Заполнение конструктора запроса

Чтобы иметь возможность получать различные отчеты в зависимости от выбранной дисциплины – перейдем на вкладку «Условия». Перетащите измерение «Дисциплина» в правую область открывшегося окна и убедитесь, что условие выглядит так же, как на картинке, при необходимости исправьте вручную, как показано на рисунке 7.

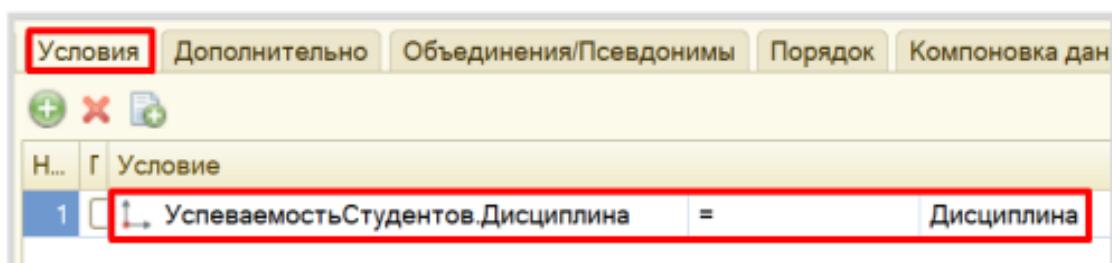


Рисунок 7 – Заполнение условия

Чтобы отчет получился красивым – установим псевдоним для поля «Оценка» и завершим составление запроса. Для этого следует перейти на вкладку «Объединения и псевдонимы» и изменить имя поля с «Оценка» на «СреднийБалл». Для этого дважды щелкните по имени, должна появиться возможность для редактирования имени. После изменения псевдонима данное окно должно быть заполнено следующим образом, как показано на рисунке 8.

Имя поля	Запрос 1
Студент	УспеваемостьСтудентов.Студент
СреднийБалл	УспеваемостьСтудентов.Оценка

Рисунок 8 – Изменение псевдонима

Нажмите на кнопку «OK». Система должна сформировать следующий запрос, как показано на рисунке 9.

```

Запрос:
ВЫБРАТЬ
    УспеваемостьСтудентов.Студент КАК Студент,
    УспеваемостьСтудентов.Оценка КАК СреднийБалл
ИЗ
    РегистрНакопления.УспеваемостьСтудентов КАК УспеваемостьСтудентов
ГДЕ
    УспеваемостьСтудентов.Дисциплина = &Дисциплина

```

Рисунок 9 – Запрос

Следующим этапом будет расчет среднего балла для студента. Для этого перейдем на вкладку «Ресурсы» и установим поле «СреднийБалл» в качестве ресурса. Данное поле должно высчитываться по следующему выражению: Окр(Среднее(СреднийБалл), 2) С помощью метода «Окр(\*, 2)» мы сможем округлить полученное выражение до сотых. Кроме того, следует указать, что ресурс должен рассчитываться по студенту, как показано на рисунке 10.

Доступные поля	Поле	Выражение	Рассчитывать по...
- СреднийБалл	СреднийБалл	Окр(Среднее(СреднийБалл),2)	Студент
Студент			

Рисунок 10 – Установление ресурса

Далее на вкладке «Параметры» нам нужно отключить (снять галочку) ограничение доступности выбора дисциплины в отчете, как показано на рисунке 11.

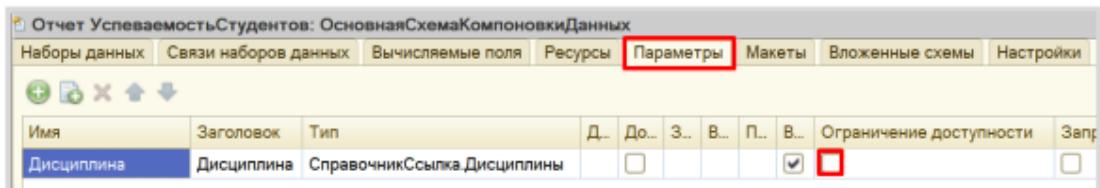


Рисунок 11 – Редактирование параметров

Теперь переходим на вкладку «Настройки» для оформления внешнего вида отчета. Воспользуемся конструктором настроек отчета.

Построим отчет в виде горизонтальной диаграммы.

На данном этапе нужно выбрать поля, которые будут отображаться в отчете, как показано на рисунке 12.

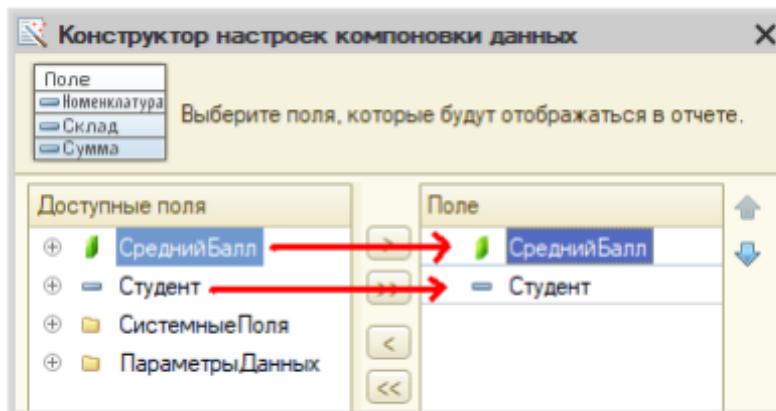


Рисунок 12 – Конструктор настроек компоновки данных

Нажмите на кнопку «Далее».

Теперь нужно определить оси X и Y нашей будущей диаграммы: ось X – это точки, Y – серии. Пусть по оси Y будет указан средний балл, а по оси X – студенты.

Нажмите на кнопку «Далее». Чтобы сделать список студентов в отчете по алфавиту – добавим упорядочивание по полю «Студент».

Нажмите на кнопку «Далее».

Среди приложенных вариантов диаграмм нужно выбрать вариант «Гистограмма горизонтальная», как показано на рисунке 13.

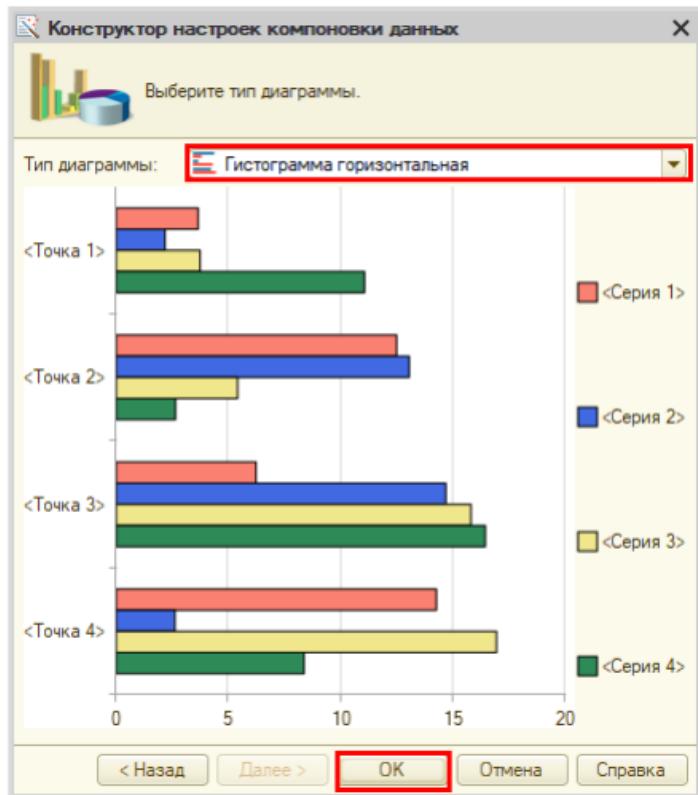


Рисунок 13 – Конструктор настроек компоновки данных

Чтобы у пользователя была возможность выбирать дисциплину, по которой он хочет построить отчет, необходимо включить параметр «Дисциплина» в пользовательские настройки, как показано на рисунках 14, 15.

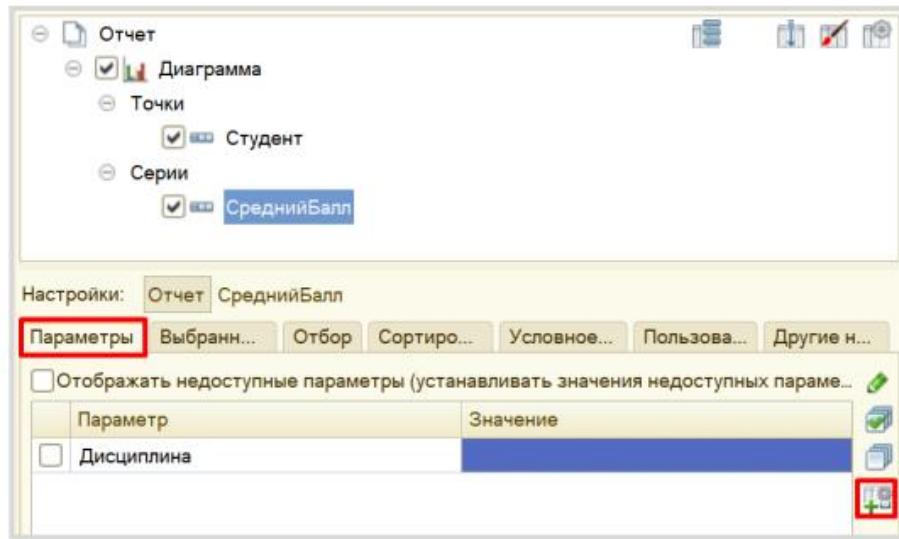


Рисунок 14 – Редактирование параметров

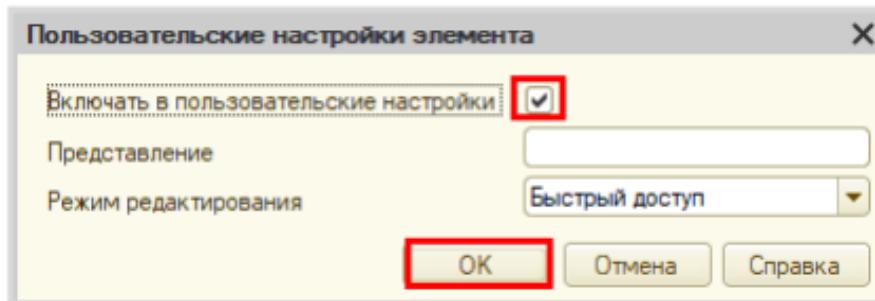


Рисунок 15 – Редактирование параметров

Используем условное оформление для того, чтобы сделать отчет более понятным для пользователя. Для этого следует открыть вкладку «Условное оформление». Здесь можно задать оформление, которое будет применено к отчету или его части, когда происходит определенное событие. Добавьте новое условное оформление.

Сначала нужно выбрать оформление. Мы будем выделять различными цветами состояние успеваемости студентов, как показано на рисунке 16.

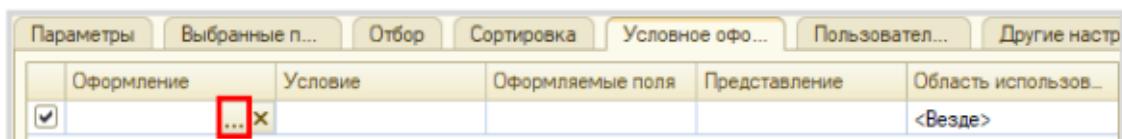


Рисунок 16 – Выбор оформления

В открывшемся окне нас интересует свойство «Цвет в диаграмме». Установим флажок и выберем значение цвета, как показано на рисунке 17.

Параметр	Значение
<input type="checkbox"/> Цвет фона	[0, 0, 0]
<input type="checkbox"/> Цвет текста	[0, 0, 0]
<input checked="" type="checkbox"/> Цвет в диаграмме	[0, 0, 0]
<input type="checkbox"/> Цвет границы	[0, 0, 0]

Рисунок 17 – Редактирование цветов в диаграмме

Чтобы различать успеваемость студентов будем использовать четыре цвета:

- Зеленый – для отличников;
- Оранжевый – для хорошистов;
- Желтый – для троичников;
- Красный – для двоечников.

Мы добавили оформление для отличников (RGB: 51, 153, 102 – зеленый цвет). Аналогичным образом добавьте еще три оформления: для хорошистов, троичников и двоечников.

Заключительным этапом будет установка для каждого оформления условия, поскольку в зависимости от разного среднего балла шкала должна окрашиваться разными цветами. Начнем с первого оформления для отличников, как показано на рисунке 18.

Настройки: Отчет СреднийБалл

Параметры	Выбранные п...	Отбор	Сортировка	Условное офор...	Пользовател...	Другие наст...
Оформление	Условие	<input type="button" value="..."/>				
<input checked="" type="checkbox"/> Цвет в диаграмме		<input type="button" value="..."/>				<Везде>
<input checked="" type="checkbox"/> Цвет в диаграмме						<Везде>
<input checked="" type="checkbox"/> Цвет в диаграмме						<Везде>
<input checked="" type="checkbox"/> Цвет в диаграмме						<Везде>

Рисунок 18 – Редактирование условия

Нужно настроить отбор по среднему баллу. Добавим новый отбор, как показано на рисунке 19.

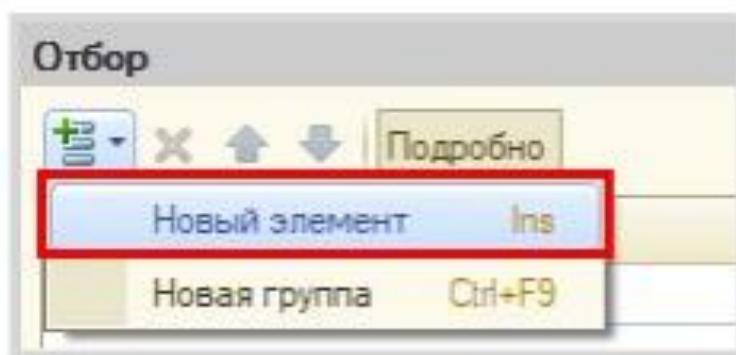


Рисунок 19 – Создание нового элемента

В качестве левого сравниваемого значения выберем поле «СреднийБалл», как показано на рисунке 20.

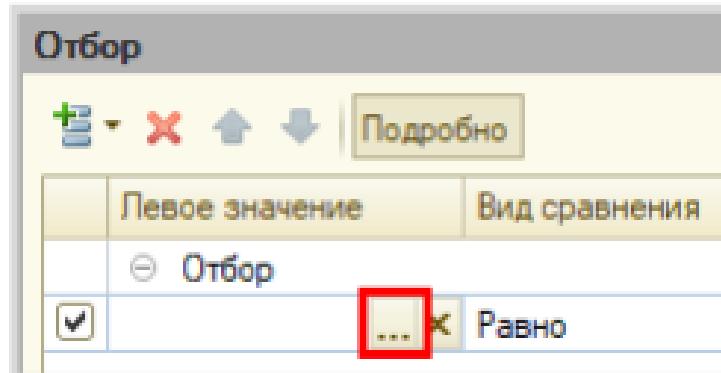


Рисунок 20 – Настройка сравнения

Вид сравнения установим в значение «больше или равно».

В качестве правого значения выберем значение 4,5.

Чтобы в дальнейшем понимать, для кого сформировано условие – установим представление «Отличник», как показано на рисунке 21.

	Левое значение	Вид сравнения	Правое значение	Представление
⊕ Отбор				
<input checked="" type="checkbox"/>	СреднийБалл	Больше или равно	4,5	Отличник

Рисунок 21 – Настройка сравнения

После чего нажмем на кнопку «OK». Таким образом, мы настроили условие для первого оформления, как показано на рисунке 22.

Параметры					
Выбранные п...					
Отбор					
Оформление	Условие	Оформляемые поля	Представление	Область использо...	
<input checked="" type="checkbox"/> Цвет в диаграмме	Отличник			<Везде>	
<input checked="" type="checkbox"/> Цвет в диаграмме				<Везде>	
<input checked="" type="checkbox"/> Цвет в диаграмме				<Везде>	
<input checked="" type="checkbox"/> Цвет в диаграмме				<Везде>	

Рисунок 22 – Условное оформление

Табличная часть справочника должна выглядеть следующим образом, как показано на рисунке 12.

Далее заполните остальные условия по аналогии. Хорошист представлен на рисунке 23, троичник на рисунке 24, двоичник на рисунке 25, итоговое оформление на рисунке 26.

	Левое значение	Вид сравнения	Правое значение	Представление
⊕ Отбор				
<input checked="" type="checkbox"/>	СреднийБалл	Больше или равно	3,5	Хорошист
<input checked="" type="checkbox"/>	СреднийБалл	Меньше	4,5	Хорошист

Рисунок 23 - Хорошист

	Левое значение	Вид сравнения	Правое значение	Представление
⊕ Отбор				
<input checked="" type="checkbox"/>	СреднийБалл	Больше или равно	2,5	Троечник
<input checked="" type="checkbox"/>	СреднийБалл	Меньше	3,5	Троечник

Рисунок 24 – Троичник

	Левое значение	Вид сравнения	Правое значение	Представление
⊕ Отбор				
<input checked="" type="checkbox"/>	СреднийБалл	Меньше	2,5	Двоичник

Рисунок 25 – Двоичник

Параметры	Выбранные п...	Отбор	Сортировка	Условное офор...	Пользовател...	Другие настр...
	Оформление	Условие	Оформляемые ...	Представление	Область использо...	
<input checked="" type="checkbox"/>	Цвет в диаграмме	Отличник			<Везде>	
<input checked="" type="checkbox"/>	Цвет в диаграмме	Хорошист И Хорошист			<Везде>	
<input checked="" type="checkbox"/>	Цвет в диаграмме	Троичник И Троичник			<Везде>	
<input checked="" type="checkbox"/>	Цвет в диаграмме	Двоичник			<Везде>	

Рисунок 25 – Заполнение условного форматирования

Отчет готов. Запустим систему в режиме «1С:Предприятие». В зависимости от выбранной дисциплины будет меняться отчет. Результат представлен на рисунках 26, 27.

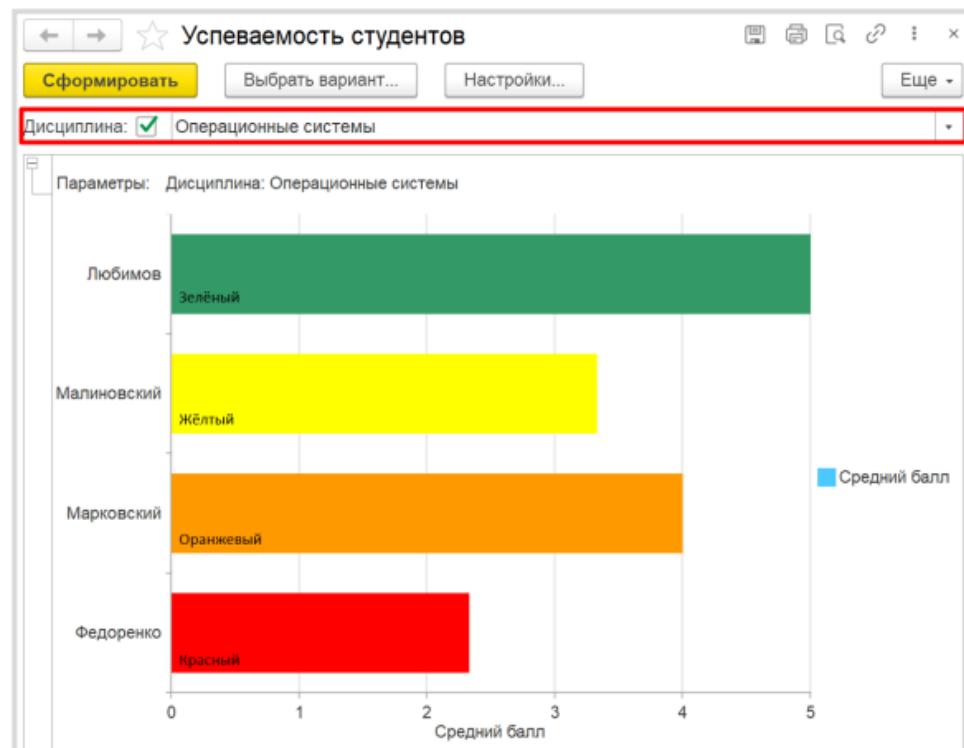


Рисунок 26 – Готовый отчет

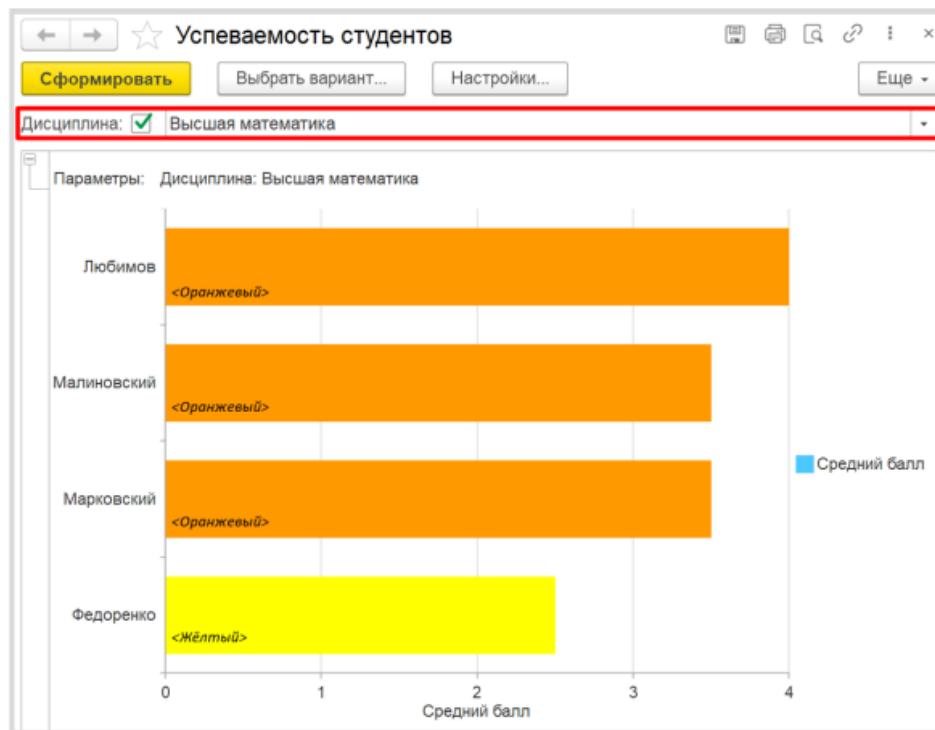


Рисунок 27 – Готовый отчет

## **Лабораторная работа № 9**

### **Автоматизация системы пункта проката электросамокатов в учебном заведении**

#### **1 Задание:**

Заказчик просит автоматизировать систему пункта проката электросамокатов в учебном заведении. Нужно фиксировать в информационной системе, какой студент забрал или вернул самокат. Выдача и возврат должны быть фиксироваться отдельно, причем количество самокатов учитывать нет необходимости, поскольку каждый студент может арендовать (и, соответственно, сдать) не более одного самоката. В результате выполнения лабораторной работы должен получиться отчет вида:

Арендатор
Иванов Иван Иванович
Петров Петр Петрович

#### **2 Подготовка:**

- создать новую информационную базу;
- открыть информационную базу в режиме «Конфигуратор»;
- открыть окно конфигурации.

#### **3 Выполнение:**

Создадим справочник, в котором будет храниться список студентов, которые могут арендовать электросамокат, и назовем его «Студенты».

Для решения данной задачи другие объекты аналитики нам не понадобятся. Запустим программу в режиме «1С:Предприятие» и добавим несколько студентов.

Таким образом, было организовано хранение в системе списка студентов.

Далее нужно реализовать закупку электросамокатов для точки проката. Для данной цели используем объект конфигурации документ. Создадим документ «Закупка».

Для формирования структуры документа переходим на вкладку «Данные». Все самокаты в нашем прокате совершенно одинаковы, поэтому хранить данные о самих самокатах нам не нужно. Важно лишь учитывать, сколько таких самокатов было закуплено для проката. Для этого добавим новый реквизит документа «Количество» с помощью кнопки «Добавить». Тип данных – «Число».

Откроем программу в режиме «1С:Предприятие» и зафиксируем закупку нескольких электросамокатов.

Легко заметить, что система сгенерировала для документа другие стандартные реквизиты: «Номер» и «Дата». Оба поля заполняются автоматически, дата может быть изменена.

Таким образом, документ для регистрации закупки электросамокатов был успешно реализован. Следующий шаг – это регистрация операции по выдаче электросамокатов в аренду. Создадим документ «Сдача В Аренду».

Переходим на вкладку «Данные» для формирования структуры документа. В данном документе нужно фиксировать дату и время, а также ФИО студента, арендовавшего самокат. Поле «Дата» будет создано системой автоматически, а вот поле для ввода ФИО студента отсутствует. Добавим реквизит «Студент» с типом «СправочникСсылка.Студенты». Данный реквизит будет содержать ссылку на объект справочника «Студенты».

Проверим работоспособность документа в режиме «1С:Предприятие», как показано на рисунке 1.

Дата	Номер	Студент
04.09.2020 19:06:16	000000001	Иванов
04.09.2020 19:06:20	000000002	Сидоров
04.09.2020 19:17:39	000000003	Петров

Рисунок 1 – Заполнение документа

Теперь информационная система способна регистрировать дату и ФИО студента, арендовавшего самокат. Регистрация возврата самокатов будет выглядеть аналогично регистрации сдачи в аренду. Создадим новый документ «Возврат», копируя документ «Сдача В Аренду». Для этого выберем нужный документ в окне конфигурации и нажмем на кнопку «Добавить копированием», как показано на рисунке 2.

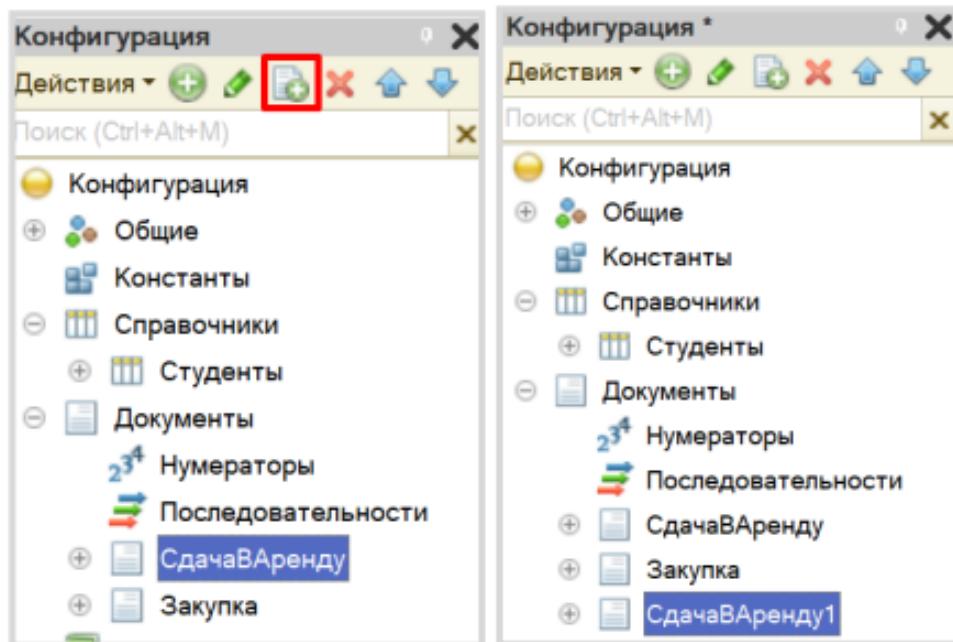


Рисунок 2 – Добавление копированием

Откроем окно редактирования документа «Сдача В Аренду 1» и изменим название документа на «Возврат».

Чтобы убедиться в том, что все было сделано правильно, перейдите на вкладку «Данные» и проверьте, что у данного документа существует реквизит «Студент». Созданная нами информационная система может хранить информацию о закупке электросамокатов, факте сдачи самоката в аренду конкретному студенту, а также факт возврата самоката студентом.

Предположим, что с помощью документа «Закупка» мы зарегистрировали в системе 12 электросамокатов. Если Иванов возьмет один из самокатов в аренду, то в прокате должно остаться 11. По возвращении самоката в прокате снова должно быть 12 самокатов.

Но на деле все введенные нами документы никак между собой не связаны. Нам следует использовать такой объект, который сможет связать между собой данные документы, а также будет накапливать итоговые значения для расчета остатков. Такой объект называется регистром накопления.

Добавим новый регистр накопления «Остатки Самокатов» вида «Остатки». Вид регистра «Остатки» позволяет настроить данный регистр таким образом, что какие-то объекты будут вносить в него данные, а какие-то, наоборот, вычитать из него. Таким образом и получается хранение остатков.

Открываем вкладку «Данные» для формирования структуры регистра накопления.

Добавим ресурс. Что мы хотим считать с помощью данного регистра? Мы хотим считать количество. Следовательно, количество и будет являться ресурсом. Тип данного реквизита – «Число».

Необходимо создать еще один регистр накопления «Аренда Самокатов» вида «Остатки». Он уже будет учитывать, кто взял или вернул самокат.

Открываем вкладку «Данные» для формирования структуры регистра накопления.

В данном регистре накопления (по аналогии с предыдущим) в качестве ресурса выступает реквизит «Количество», тип – «Число».

Чтобы разобраться с измерением, следует понять, в разрезе чего мы хотим считать количество. Мы хотим учитывать, какому студенту сдали самокат в аренду. Значит, в качестве измерения нужно добавить реквизит «Студент». Тип данного реквизита – «СправочникСсылка.Студенты».

Обратите внимание, что типы данных этих регистров совпадают с типами данных регистров, которые были добавлены в документы. Взаимосвязь созданных документов и регистров накопления должна выглядеть следующим образом, как показано на рисунке 3.

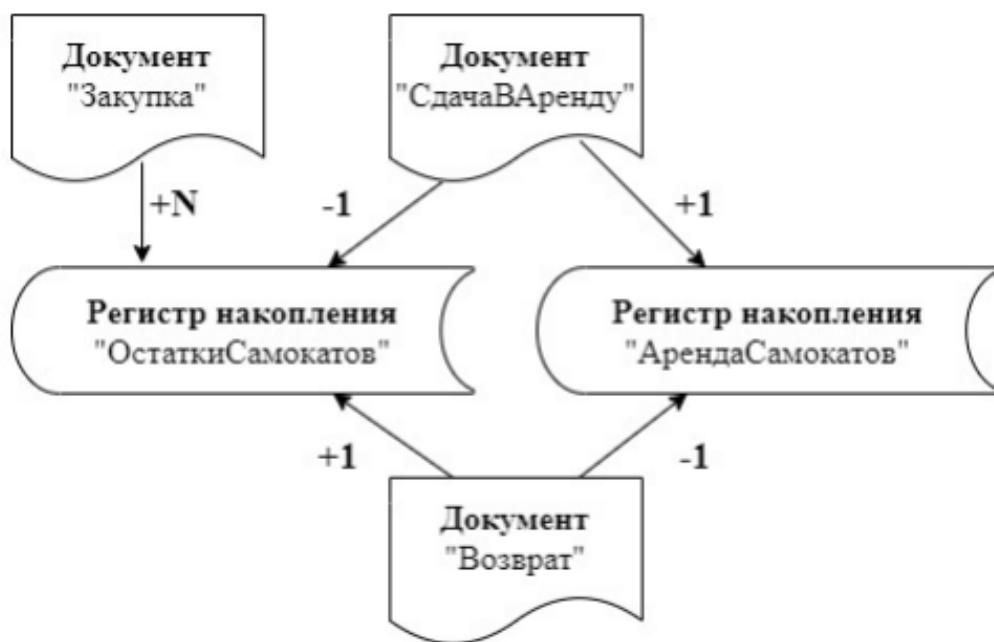


Рисунок 3 – Взаимосвязь созданных документов

При проведении документа «Закупка» в регистр накопления «ОстаткиСамокатов» будет передано то количество самокатов, которое будет указано в документе.

При проведении документа «СдачаВАренду» будет производиться вычет одного самоката из регистра накопления «ОстаткиСамокатов», но при этом студент, взявший самокат, будет внесен в регистр «АрендаСамокатов» в качестве арендатора.

Документ «Возврат» будет делать все наоборот: добавлять единицу в «ОстаткиСамокатов» и убирать студента из списка должников из реквизита «АрендаСамокатов».

Таким образом, количество электросамокатов в регистре накопления «ОстаткиСамокатов» должно соответствовать текущему количеству самокатов на складе. А регистр накопления «АрендаСамокатов» должен содержать список должников.

Чтобы регистры накопления заработали, нужно сделать следующее:

1. Определить, какие данные будут попадать в каждый регистр (определить документы-регистраторы).
2. Описать, каким образом данные из каждого документа-регистратора должны попадать в регистры.

Начнем с документа «Закупка». Откроем окно редактирования данного документа и перейдем на вкладку «Движения». Отметим регистр накопления «ОстаткиСамокатов», поскольку данные о количестве закупленных самокатов необходимо передавать только в этот регистр. Воспользуемся конструктором движений, как показано на рисунка 4, 5, 6.

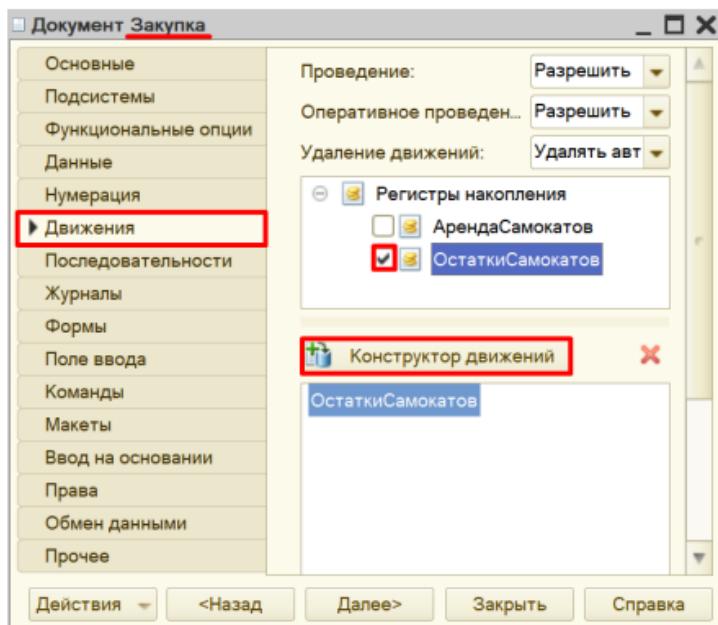


Рисунок 4 – Редактирование документа «Закупка»

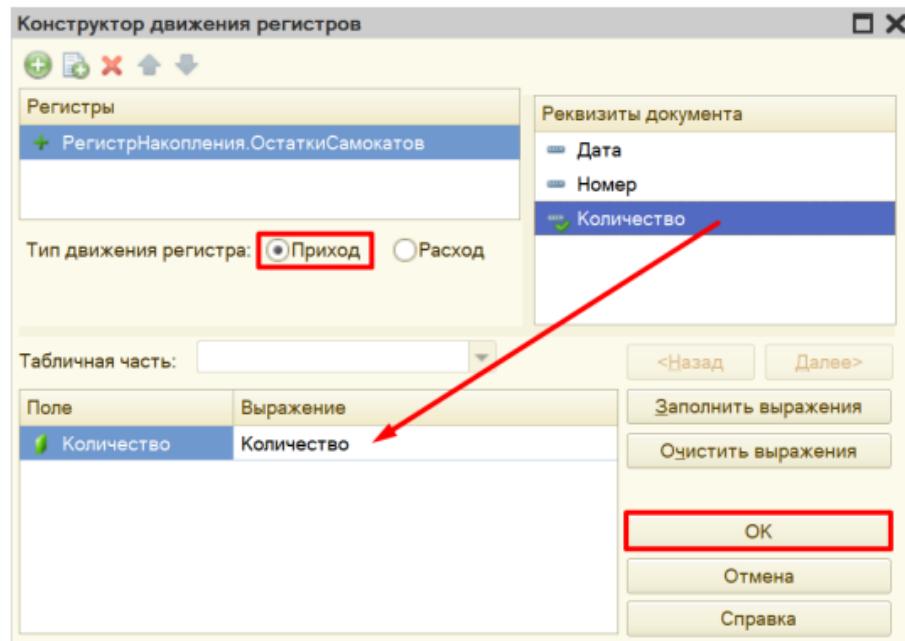


Рисунок 5 – Редактирование документа «Закупка»

```

Документ Закупка: Модуль объекта

Процедура ОбработкаПроведения(Отказ, Режим)
//{{ Конструктор_движений_регистров
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенный
// регистр ОстаткиСамокатов Приход
Движение.ОстаткиСамокатов.Записывать = Истина;
Движение = Движение.ОстаткиСамокатов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Количество = Количество;

//}})_конструктор_движений_регистров
КонецПроцедуры

```

Рисунок 6 – Редактирование документа «Закупка»

Данный документ фиксирует количество закупленных электросамокатов. Следовательно, он должен в регистр накопления приходить со знаком «+» (плюс). Для этого нужно выбрать тип движения регистра «Приход». Если все было сделано правильно, имена и типы реквизитов совпадают, то при нажатии на кнопку «Заполнить выражения» реквизиты регистра заполняются автоматически.

Если этого не произошло, то заполните поле «Выражение» вручную, путем выбора соответствующих реквизитов документа.

При нажатии на кнопку «OK» система сформирует программный код, который при успешном проведении документа произведет движения в регистр накопления, то есть скопирует данные из документа в регистр накопления.

Программный код может быть отредактирован вручную, если возникли проблемы при работе с конструктором. Переходим к настройке документа «СдачаВАренду». Данный документ должен делать движения в оба регистра накоплений, как показано на рисунках 7, 8, 9.

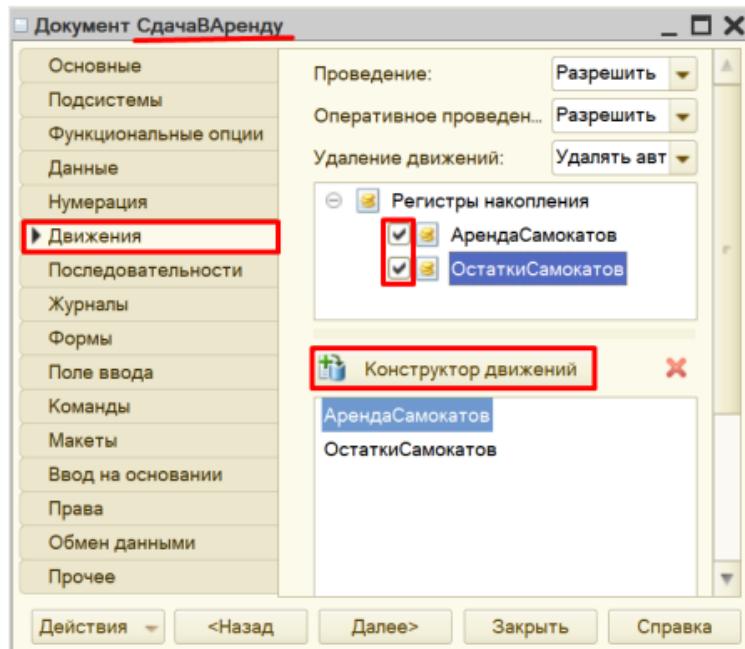


Рисунок 7 – Редактирование документа «СдачаВАренду»

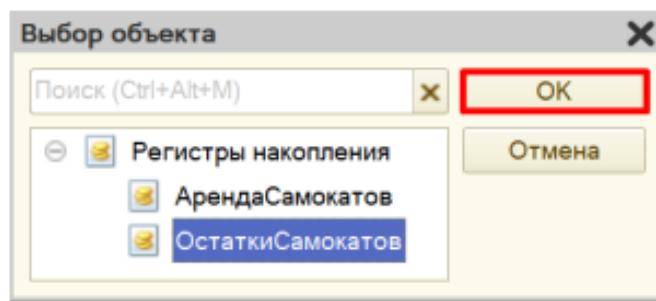


Рисунок 8 – Редактирование документа «СдачаВАренду»

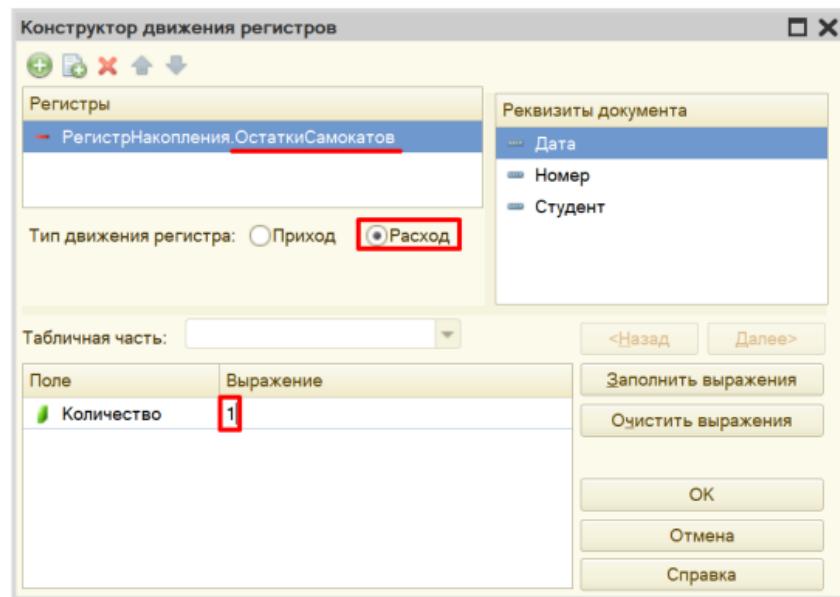


Рисунок 9 – Редактирование документа «Сдача В Аренду»

Теперь опишем взаимодействие с другим регистром накопления. Для этого нужно нажать на кнопку «Добавить» и выбрать регистр накопления «Аренда Самокатов», как показано на рисунках 11.

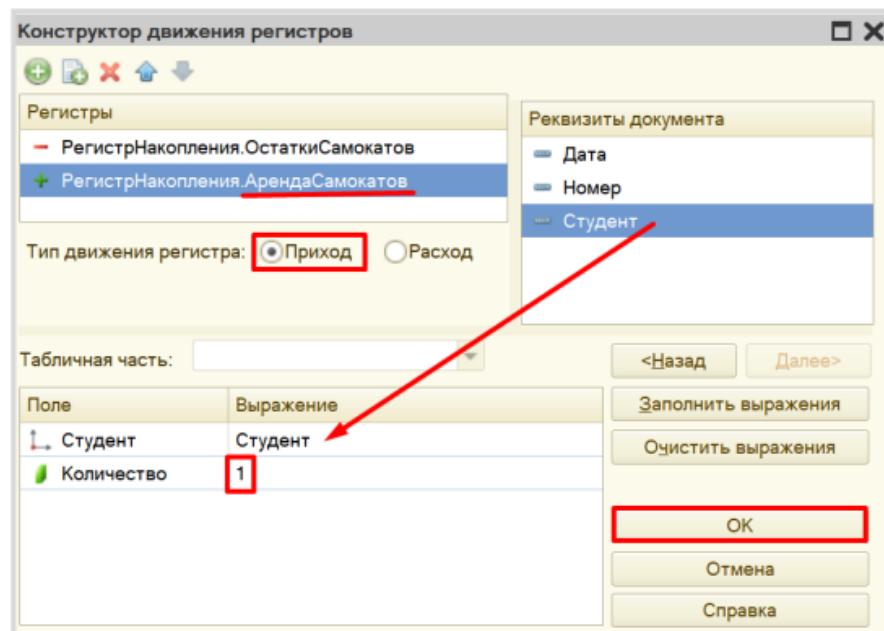


Рисунок 10 – Конструктор движения регистра

```
// Процедура ОбработкаПроведения(Отказ, Режим)
//{(_КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
//  // Данный фрагмент построен конструктором.
//  // При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
//
//  // регистр ОстаткиСамокатов Расход
//  Движение.ОстаткиСамокатов.Записывать = Истина;
//  Движение = Движения.ОстаткиСамокатов.Добавить();
//  Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
//  Движение.Период = Дата;
//  Движение.Количество = 1;
//
//  // регистр АрендаСамокатов Приход
//  Движение.АрендаСамокатов.Записывать = Истина;
//  Движение = Движения.АрендаСамокатов.Добавить();
//  Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
//  Движение.Период = Дата;
//  Движение.Студент = Студент;
//  Движение.Количество = 1;
//
//}__КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Рисунок 11 – Модуль документа

Переходим к настройке последнего документа – «Возврат». Данный документ настраивается аналогично документу «СдачаВАренду», как показано на рисунках 12, 13, 14, 15.

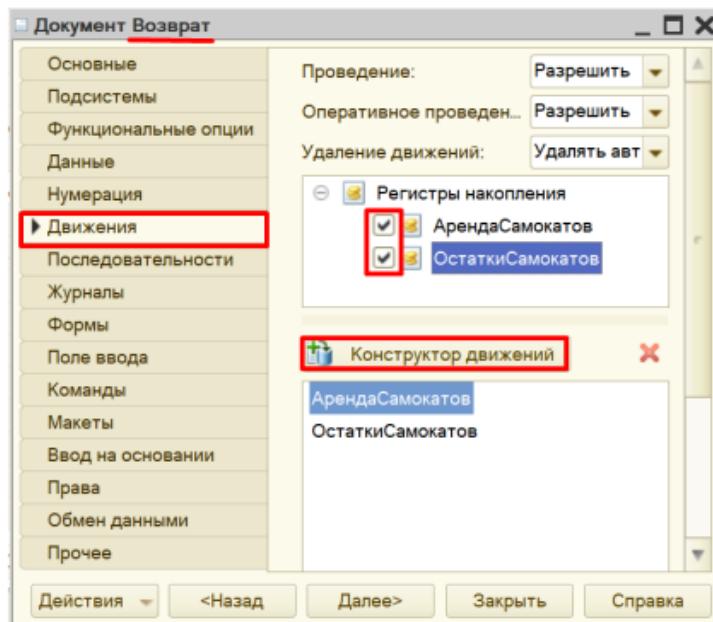


Рисунок 12 – Редактирование документа «Возврат»

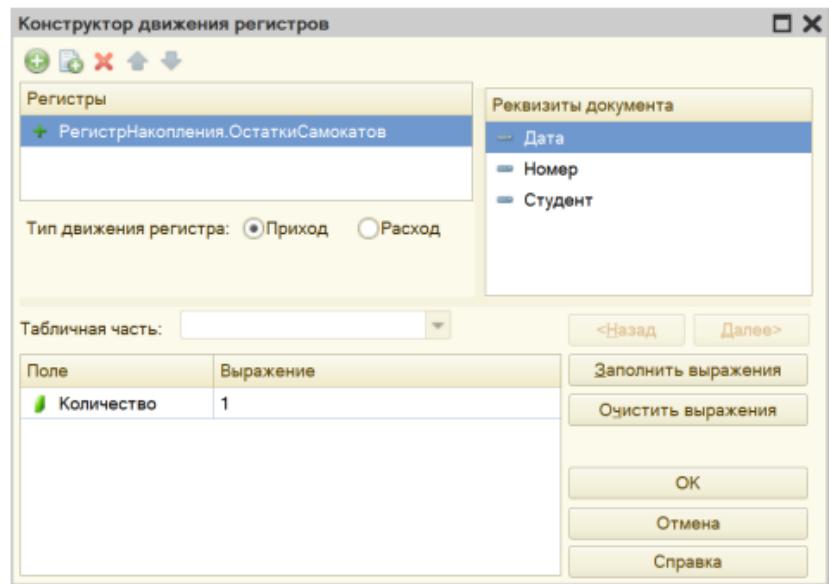


Рисунок 13 – Конструктор движения регистра

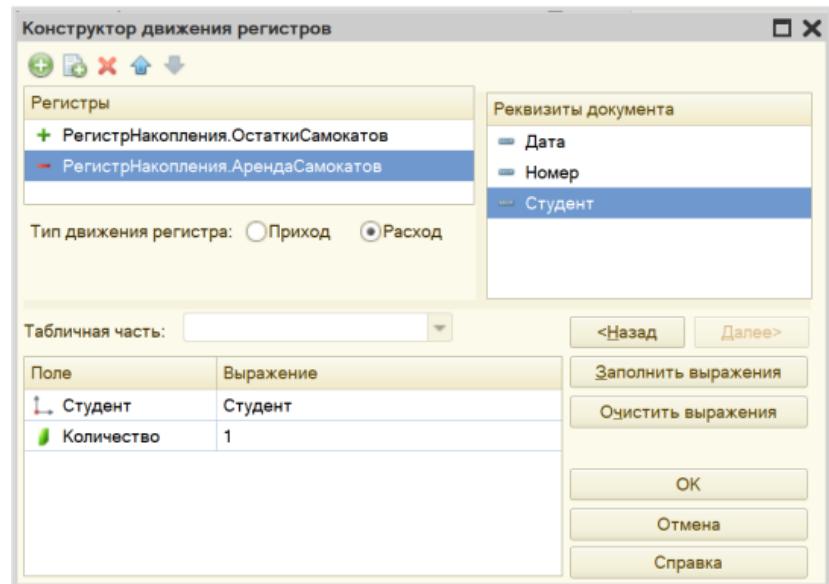


Рисунок 14 – Конструктор движения регистра

```
Процедура ОбработкаПроведения(Отказ, Режим)
// __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
// Данный фрагмент построен конструктором.
// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!
// регистр ОстаткиСамокатов Приход
Движение.ОстаткиСамокатов.Записывать = Истина;
Движение = Движения.ОстаткиСамокатов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
Движение.Период = Дата;
Движение.Количество = 1;

// регистр АрендаСамокатов Расход
Движение.АрендаСамокатов.Записывать = Истина;
Движение = Движения.АрендаСамокатов.Добавить();
Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
Движение.Период = Дата;
Движение.Студент = Студент;
Движение.Количество = 1;

// __КОНСТРУКТОР_ДВИЖЕНИЙ_РЕГИСТРОВ
КонецПроцедуры
```

Рисунок 15 – Модуль документа

Теперь каждый из этих документов копирует данные в регистры накопления, прибавляя или вычитая количество электросамокатов.

Проверим работу регистра в режиме «1С:Предприятие».

Для начала нужно перепровести (провести заново) все созданные ранее документы «Закупка» и «Сдача в аренду», чтобы данные из них попали в регистр накопления. Кроме того, добавим новый документ «Возврат».

Чтобы заглянуть в содержимое регистра накопления, следует прибегнуть к функциям технического специалиста.

Найдем наши регистры, откроем их и посмотрим на движения, как показано на рисунке 16.

The screenshot displays two windows from a 1C application:

**Остатки самокатов (Bicycle Inventory):**

Период	Регистратор	Номер строки	Количество
+ 04.09.2020 19:06:10	Закупка 00000001 от 04.09.2020 19:06:10	1	12
- 04.09.2020 19:06:16	Сдача в аренду 00000001 от 04.09.2020 19:06:16	1	1
- 04.09.2020 19:06:20	Сдача в аренду 00000002 от 04.09.2020 19:06:20	1	1
+ 04.09.2020 19:06:23	Возврат 00000001 от 04.09.2020 19:06:23	1	1
- 04.09.2020 19:17:39	Сдача в аренду 00000003 от 04.09.2020 19:17:39	1	1

**Аренда самокатов (Bicycle Rental):**

Период	Регистратор	Номер строки	Студент	Количество
+ 04.09.2020 19:06:16	Сдача в аренду 00000001 от 04.09.2020 19...	1	Иванов	1
+ 04.09.2020 19:06:20	Сдача в аренду 00000002 от 04.09.2020 19...	1	Сидоров	1
- 04.09.2020 19:06:23	Возврат 00000001 от 04.09.2020 19:06:23	1	Иванов	1
+ 04.09.2020 19:17:39	Сдача в аренду 00000003 от 04.09.2020 19...	1	Петров	1

Рисунок 16 – Аренда и остатки самокатов

В регистры накопления действительно добавляются строки, соответствующие проведенным документам. Для решения задачи осталось построить отчет, который будет отображать список студентов, которые еще не вернули самокаты.

Создадим новый отчет «ОтчетПоАрендаторам». Для наполнения отчета воспользуемся конструктором схемы компоновки данных.

Все созданные нами объекты конфигурации представляют собой таблицы базы данных. В режиме «1С:Предприятие» мы заполняем эти таблицы данными. Чтобы получить эти данные для отображения в отчете, нужно сформировать запрос к базе данных.

Воспользуемся конструктором запроса.

Данные будем брать не из регистра накоплений напрямую, а из виртуальной таблицы, которую создает этот регистр автоматически. Данная виртуальная таблица позволит получить уже просуммированные значения по всем документам. Чтобы перенести данные из одного окна в другое, просто перетащите нужные поля с помощью мыши либо воспользуйтесь стрелочками,

расположенными между окнами. В результате данное окно должно быть заполнено следующим образом, как показано на рисунке 17.

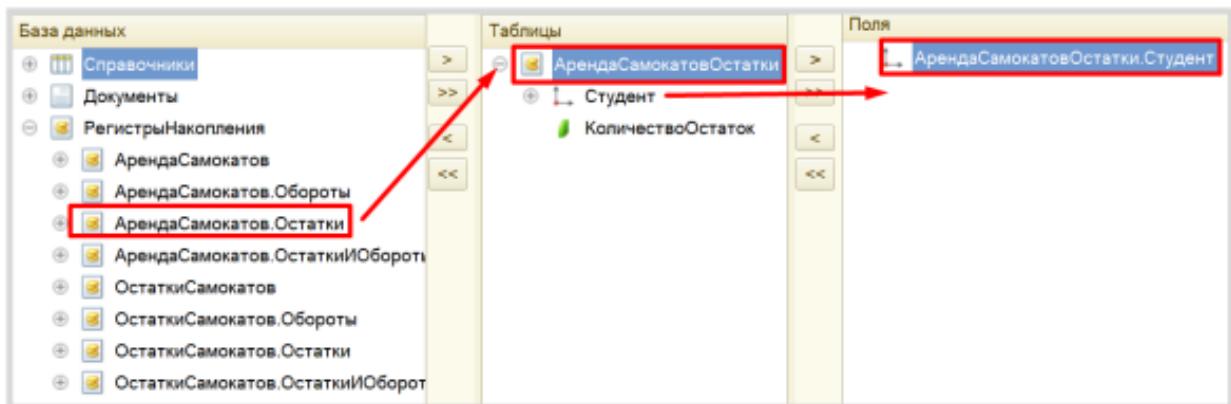


Рисунок 17 – Заполнение конструктора запросов

Запрос к базе данных должен выглядеть следующим образом, как показано на рисунке 18.

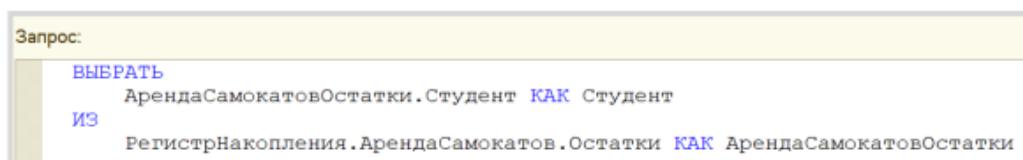


Рисунок 18 – Запрос к БД

Нужно запустить режим «1С:Предприятие» и создать несколько документов, как показано на рисунке 12.

Запрос к базе данных сформирован. Теперь нужно настроить внешний вид отчета. Для этого следует перейти на вкладку «Настройки».

Для настройки внешнего вида отчета воспользуемся конструктором настроек отчета.

Наш отчет будет иметь форму списка.

Далее нужно выбрать поля, которые будут отображены в отчете, как показано, на рисунке 19.

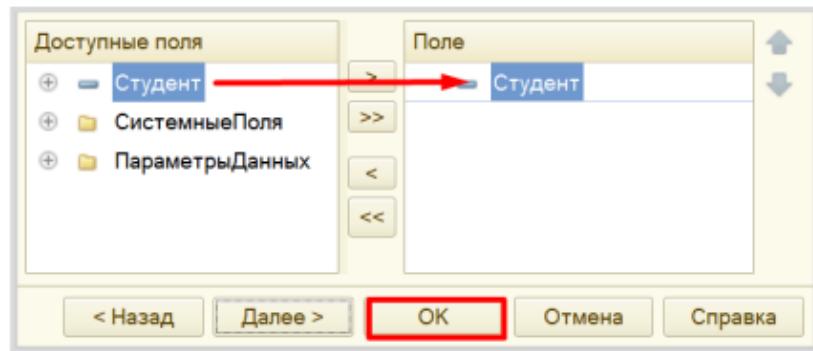


Рисунок 18 – Выбор отображаемых полей в отчете

Осталось лишь убедиться, что отчет работает верно. Для этого посмотрим на отчет в режиме «1С:Предприятие», как показано на рисунке 19.

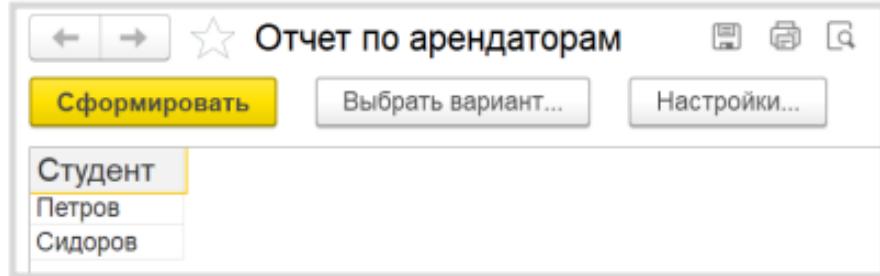


Рисунок 19 – Итоговый отчет

Таким образом, отчет выводит список арендаторов, не вернувших самокат в пункт проката.

Стоит отметить, что получившаяся информационная система не способна отслеживать остатки и работает «на честном слове». Это значит, что система не способна отслеживать, когда количество самокатов на складе станет отрицательным или тот факт, что один студент решит арендовать два самоката (хотя мы ограничили это условием задачи).

## **Самостоятельная работа**

### **1. Задание:**

Заказчик просит создать небольшую информационную систему для регистрации продаж в новом онлайн-магазине цифровых товаров.

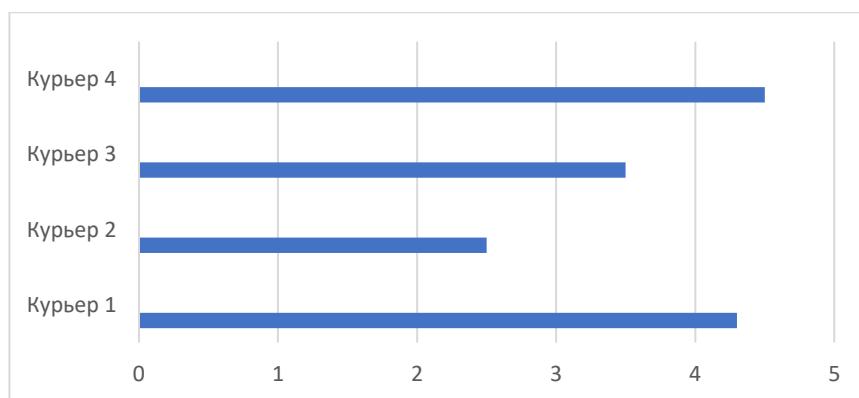
- В магазине продают электронные книги, лицензии на программное обеспечение и онлайн-курсы. Нужно отобразить только факт продажи товаров через сайт.
- В результате выполнения лабораторной работы должен получиться отчет следующего вида:

<b>Товар</b>	<b>Продано</b>
Курс по 1С	12
Photoshop	7
Java. Руководство	5

### **2 Задание:**

Заказчик просит разработать конфигурацию для учета рейтинга курьеров службы доставки. Учет ведется по месяцам.

- В системе необходимо зарегистрировать Ежедневные отчеты. В шапке документа указывается дата, а в табличной части – какие курьеры какой рейтинг клиентов получили за день (по 5-балльной шкале).
- Нужно построить Отчет по среднему рейтингу курьеров за период.



Отчет строится по среднему арифметическому баллу курьера за выбранный период (месяц/квартал).

### **3 Задание:**

Заказчик просит автоматизировать систему читательских билетов в студенческой библиотеке. Нужно фиксировать в информационной системе, какой студент получил или сдал свой читательский билет (например, для замены или продления). Выдача и возврат билетов должны фиксироваться отдельно. Количество билетов учитывать не нужно, поскольку каждый студент может получить (и, соответственно, сдать) только один личный читательский билет. В результате выполнения лабораторной работы должен получиться отчет вида:

Читатели, получившие билеты
Иванов Иван Иванович
Петров Петр Петрович