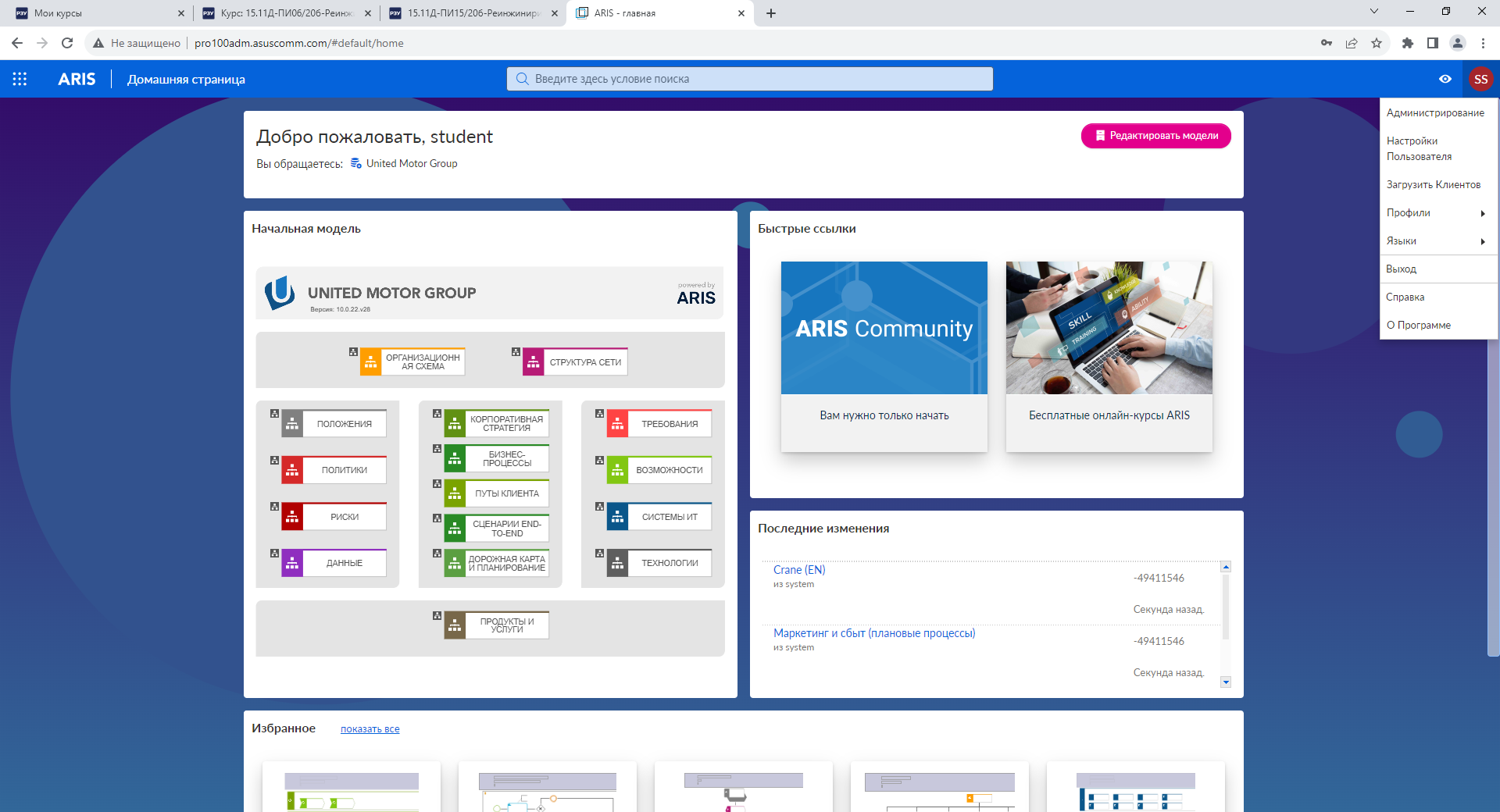
**Описание примера разработки моделей**

**компьютерной фирмы в ARIS Architect**

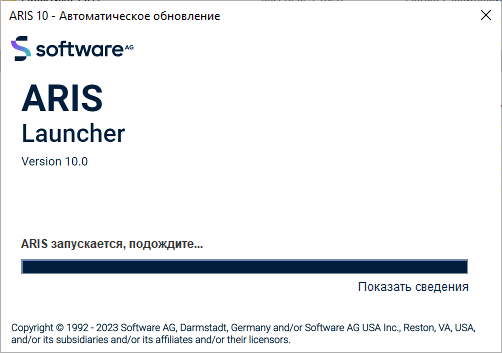
**для лабораторной работы**

1. Для выполнения работы необходимо использовать приложение **ARIS Architect**, доступное на портале ARIS <http://pro100adm.asuscomm.com/>.
2. При входе на портал авторизуйтесь с именем *student* и паролем *student*. Вы попадете на главную страницу **ARIS Portal**. Язык интерфейса ARIS Portal можно настроить на главной странице в меню *SS* в правом верхнем углу страницы: пункт *Languages (Язык)*.
3. В том же меню на главной странице портала выбираем раздел *Download Clients (Загрузить клиентов)*:



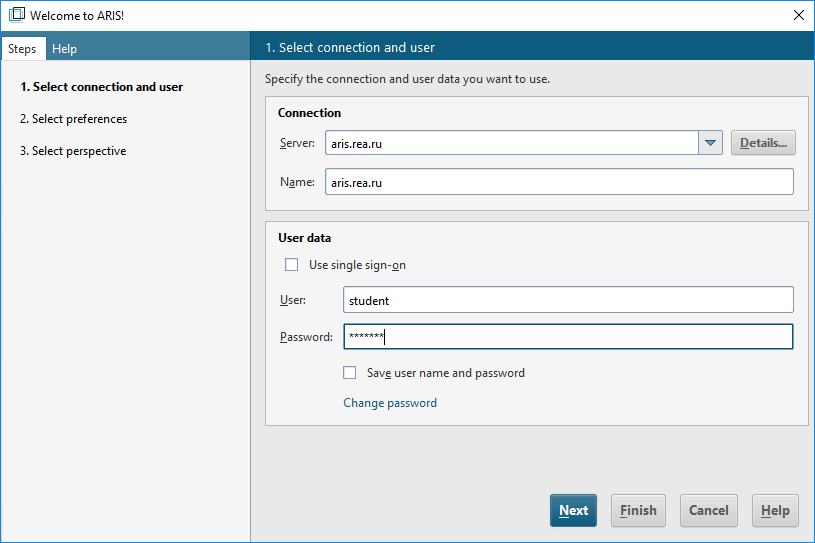
Примечание: Вы можете также работать (строить модели) непосредственно в web-клиенте – для этого выберите раздел *Edit models* *(Редактировать модели)*, но функциональность среды моделирования при этом будет ограничена. Здесь описана работа в полноценном клиенте ***ARIS Architect***.

1. Далее нажимаем кнопку *Start (Пуск)* возле пункта *ARIS Architect/Designer* – загружается файл загрузчика ARIS Architect - ***ARIS\_\*\_default.jar***. Файл является программой Java – поэтому на компьютере необходимо предварительно установить поддержку Java (пакет **JRE)** с сайта Oracle, иначе загруженный файл откроется как обычный архив.
2. Запускаем загруженный файл на выполнение:



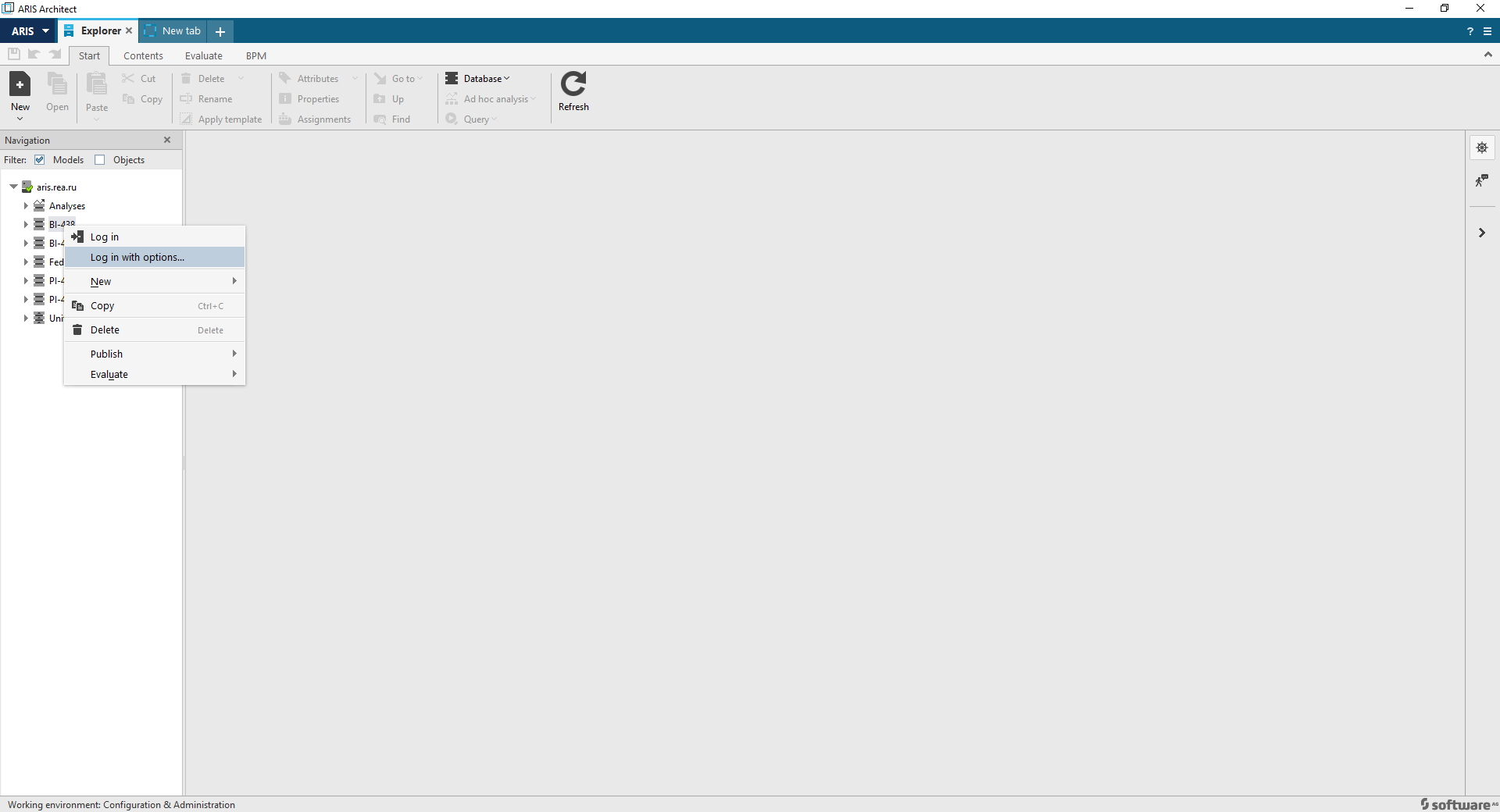
При первом запуске загрузчик автоматически скачает и установит на Вашем компьютере клиентскую часть ARIS Architect. В дальнейшем будет сразу запускать программу.

1. Когда программа запустится, снова авторизуйтесь с теми же именем и паролем:



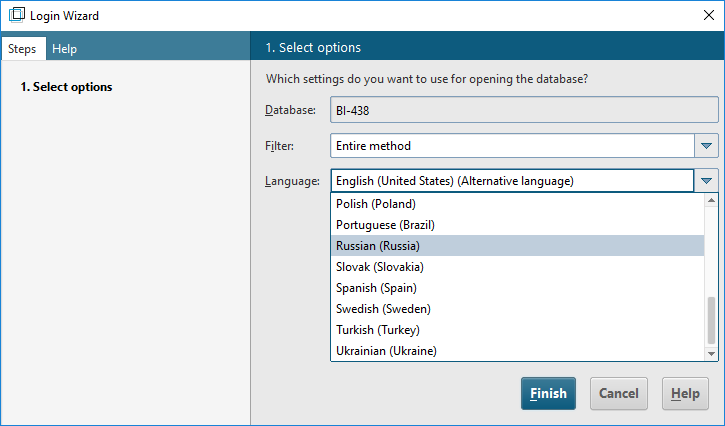
Другие настройки оставьте без изменения и нажмите *Finish (Готово)*. Запустится приложение ARIS Architect.

1. В главном окне ARIS Architect выберите вкладку *Explorer* (*Проводник*) и зайдите в нужную базу данных через пункт контекстного меню *Log in with options..(Вход с параметрами…).*:



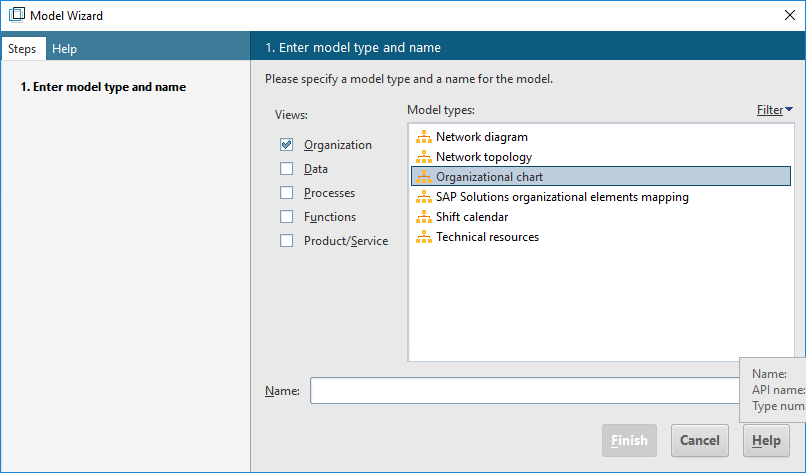
Это позволит Вам указать параметры работы БД (главное – язык БД) при входе.

1. Выберите в качестве языка работы (Language) – Russian (Russia), т.к. контент создаваемых моделей будет на русском языке:



При этом *Filter* должен быть указан *Entire method* (Полный фильтр).

1. Нажмите Finish для входа в БД и найдите/создайте собственную папку (group) для работы в структуре папок БД. Все модели создаются в папках.
2. Перед тем как начать построение процессных моделей нам необходимо будет ***построить модель оргструктуры фирмы*** (*Organizational Chart*). Обратите внимание при создании модели, что она относится к группе организационных моделей:



1. При построении модели используем только символ для обозначения оргединиц (*Organizational unit*) для обозначения сотрудников (должностей) и подразделений.
2. Для обозначения связей используем два вида:
   1. “*имеет в подчинении*” (is superior) – для обозначения подчинения одной оргединицы другой (обозначается сплошной линией) и
   2. “*состоит из*” (is composed of) – для обозначения вхождения одной оргединицы в состав другой (обозначается пунктирной линией).
3. Готовая модель оргструктуры выглядит так:



Для автоматизированного отображения типов связей используйте специализированный шаблон оформления (*Template*) по курсу.

1. Начинаем построение моделей бизнес-процессов с создания схемы ***системы бизнес-процессов фирмы***. Для этого мы используем модель типа ***Диаграмма цепочки добавленного качества (Value-Added Chain Diagram, VAD)***.
2. Сначала мы ***выделяем основные бизнес-процессы***. Это – основа нашей системы бизнес-процессов компьютерной фирмы:



1. На следующем шаге дополняем схему ***внутренними (вспомогательными) процессами*** – *процессами управления и развития* (красные, сверху) и *обеспечивающими процессами* (серые, снизу). Обратите внимание, что символы обеспечивающих и управляющих процессов являются пользовательскими и в стандартную конфигурацию не входят (в крайнем случае можно использовать обычный символ *Value-Added-Chain*, покрасив его в красный или серый цвет):



На этом наша схема системы бизнес-процессов фирмы завершена.

1. В дальнейшем мы ***рассматриваем (идентифицируем) один из основных бизнес-процессов*** – процесс производства компьютеров под заказ. Для этого мы создаем детализацию соответствующего элемента модели (процесса) в виде модели VAD. В новой модели мы строим ***Цепочку добавленной стоимости (ценности)***. Строим цепочку на двух уровнях – сначала «длинную»:



а затем и «короткую» для основного процесса «длинной» цепочки:



1. Продолжаем идентификацию процесса и ***определяем идентификационные характеристики основного процесса «длинной» цепочки*** – *процесса сборки и продажи компьютеров под заказ*. Для этого создаем детализацию соответствующего элемента модели, снова как модель VAD, в которой отражаем входы и выходы процесса, Владельца (связь *accepts*/*принимает*), менеджера (связь *decides on*/п*ринимает решение о*) и основных исполнителей процесса (связь *carries out/выполняет*), а также ключевые показатели результативности процесса. Обратите внимание, что используемые в схемах оргединицы должны быть взяты здесь и далее только из схемы оргструктуры:



1. После завершения идентификации процесса приступаем к ***описанию отдельных частей процесса по «короткой» цепочке процесса***. Для этого создаем детализацию соответствующего элемента цепочки добавленной стоимости (в нашем случае это «Прием заказа от клиента»), как *модель цепочки процесса, управляемой событиями* (Event-driven Process Chain, EPC) и составляем собственно цепочку событий и функций:



1. Затем необходимо ***указать для каждой функции ее окружение*** – ответственность и информационные входы и выходы. ***Ответственность*** обозначается присоединением слева к функции необходимых оргединиц. При этом каждая функция должна иметь одного и только одного ответственного исполнителя со связью *carries out/выполняет* и может иметь связи других типов с другими оргединицами (например, *accepts/принимает* – утверждает или *decides on/принимает решение о* – согласует, принимает решение):



1. ***Информационные входы и выходы*** обозначаются присоединением справа к функции базового значка I*nformation carrier* (информация в неопределенном виде – например в памяти у сотрудника или в его рабочих записях) и его разновидностей – *Document* (документ), *File* (файл, база данных), *Telephone* (телефонный звонок), *Fax* (факсимильное сообщение), *Internet* (информация из Интернета) и т.д. При этом на выходе всегда используем связь *creates output to*:



1. Целиком ***модель приема заказа от клиента*** будет выглядеть так:



1. По подготовленным моделям можно построить отчет «Вывод информации о моделях» для контроля правильности использования объектов в схемах. Для этого необходимо вызвать контекстное меню папки, содержащей Ваши модели (для подпапок делается отдельный отчет) и выбрать в нем пункт *Анализировать/Запустить отчет…* В появившемся окне выбрать отчет «Вывод информации о модели»:

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Нажать *Далее* и выбрать формат вывода «Документ Word (\*.DOCX)». Нажать *Готово*.

1. После запуска скрипта отчета появиться окно настройки, в котором необходимо в группе *Выход объектов модели* выбрать *Объекты* и *С группами*:

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Далее нажимаем *Ok* и получаем файл с отчетом.

1. В полученном отчете можно проконтролировать что все объекты, содержащиеся в ваших схемах, хранятся только в ваших папках, т.е. вы не используете объекты других студентов.