

А.Е. Недоря, март-апрель 2018

Краткая инструкция пользователя среды разработки программ “Вир-1”

Большое и серьезное предупреждение (Disclaimer)

“Вир-1” (“Вир”) - **экспериментальная среда разработки**, на которой отрабатываются подходы сборочного (!) программирования.

Полагаю, что многие, из тех, кто попробует “Вир”, немедленно наткнутся на то, что в нем почти все непривычно, неудобно, и уж точно не так как, как в других средах (системах) разработки.

Прошу не забывать главного: “Вир” другой, потому что программы в нем строятся существенно по-другому. Это “существенно” по-другому является его неотъемлемой частью и следует из его назначения. Основные идеи, заложенные в “Вир”, кратко описаны здесь: <http://алексейнедоря.рф/?p=269>

Для меня полезность этих идей является доказанной, в том числе и тем, что “Вир”, в котором идеи воплощены, позволил существенно упростить и ускорить изготовление программ.

Экспериментальная проверка показала: на “Вире” можно делать вполне успешные программы, например, Сайткрафт-Студия с 2008 года делается на “Вире”. Сам “Вир” тоже сделан на “Вире”.

Я с удовольствием приму участие в обсуждении **основных идей** и в **обсуждении самого “Вира”**, но это две разные области:

1. Основные идеи первичны. Идеи и подходы были выработаны мной на основе моего почти 40-летнего опыта разработки очень разных программных систем.
2. “Вир” же такой, какой он есть, потому что этого требовали идеи, положенные в его основу. Эти идеи воплощены в “Вир” не идеальным образом по множеству причин. Делая “Вир-2” я пытаюсь воплотить их немного лучше.

Итого, критика “Вира” может быть конструктивной, только в том случае, если “критик” принимает идеи. Если же нет, то просьба начинать с критики идей.

Естественно, что я буду участвовать в обсуждении только в том случае, если оно будет вестись корректно и конструктивно.

Введение

“Вир” - это среда для разработки программ, в которую входят программа “Вир” и репозиторий компонент, который называется “Вирту”.

“Вир” существенно отличается от других инструментов разработки программ тем, что в нем программа “собирается” из компонент, а не “кодируется”.

В “Вире” язык программирования и компилятор играют вспомогательную роль. Они нужны, чтобы изготовить компоненту, которой еще нет или доработать существующую.

Вы используете ознакомительную версию “Вира”, которая отличается от полной:

- 1) Часть возможностей “Вира” отключены, например, возможности, позволяющие делать программы для продажи (защита, регистрация, обновления, и т.д.)
- 2) Соответственно, из “Вирту” убраны компоненты, обеспечивающие эти возможности
- 3) Для ознакомления в “Вирту” включены примеры программ

Примеры программ ознакомительной версии:

1. Ханойские башни
2. Анализатор текстов

Ханойские башни

Программа, которую я использовал для того, чтобы научить сына (10 лет) рекурсивному программированию. Часть инструментов написана сыном, он же выбирал цвета на рабочем столе. Я всего лишь сохранил “авторское” видение этой программы.

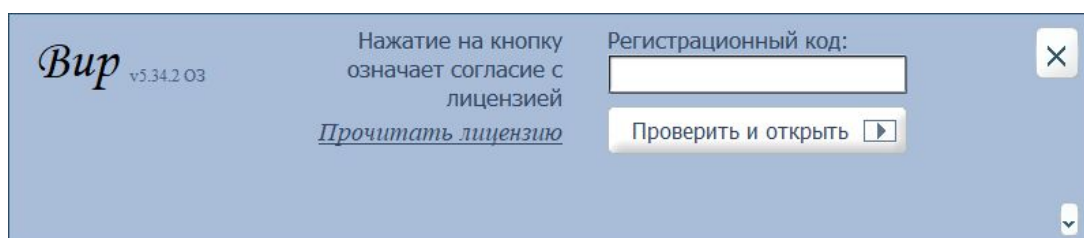
Анализатор текстов

Реальная программа, используемая логопедами для подбора материалов для упражнений для обучения детей, страдающих дислексией. Позволяет подобрать словоформы (слоги, слова) в соответствии с частотой использования в наборе текстов (например, текстов, используемых в начальных классах школы).

Используем примеры программ

1. Открыть программу для редактирования

Запуск “Вира” открывает окно регистрации:



Напечатайте регистрационный код (только в первый раз, дальше программа его запомнит). Нажмите кнопку “Проверить и открыть”. Программа запросит сервер поддержки, и если указан действующий регистрационный код, то откроет окно выбора программы:



В самом левом окошке иконка программы “Ханойские башни”.

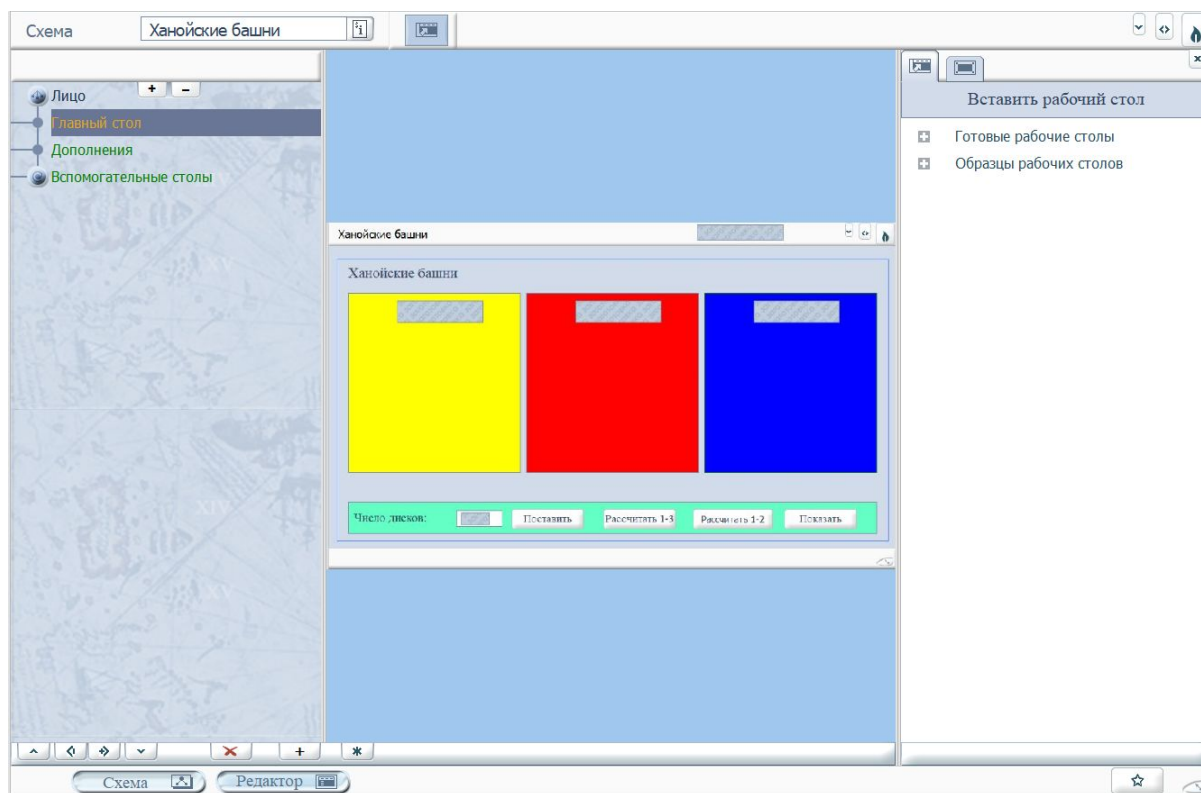
Щелкните по окошку с иконкой и названием программы “Ханойские башни”, чтобы открыть её для редактирования.

При этом откроется один из двух основных рабочих столов “Вира” (тот на котором “Вир” был закрыт в прошлый раз:

- Схема программы
- Редактор рабочего стола

Название рабочего стола вы увидите в левом верхнем углу.

Рабочий стол “Схема программы”:



Слева - дерево рабочих столов программы.

В середине - изображение выбранного рабочего стола

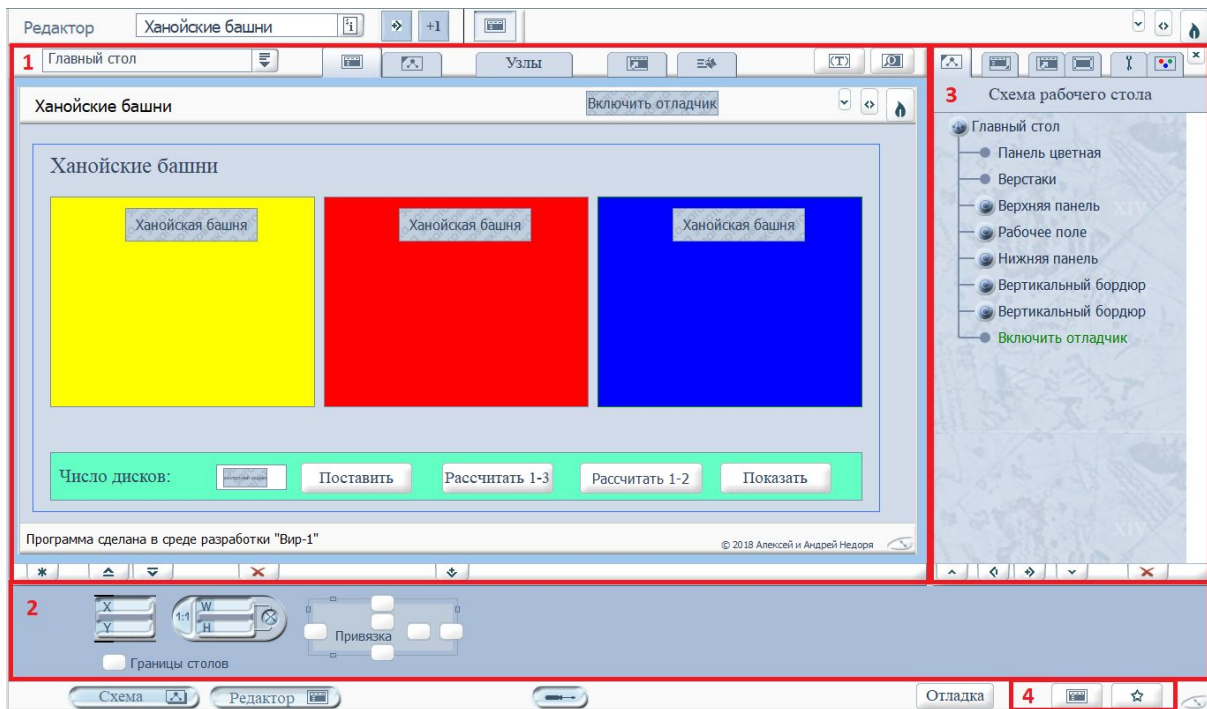
Справа - готовые рабочие столы, которые можно вставить в программу.

Внизу кнопка перехода на "Редактор".

Подробнее о "Схеме программы" расскажем, в главе "Создание новой программы".

Если у вас открылась схема, нажмите на "Редактор".

Основная работа над программой происходит в "Редакторе рабочего стола".



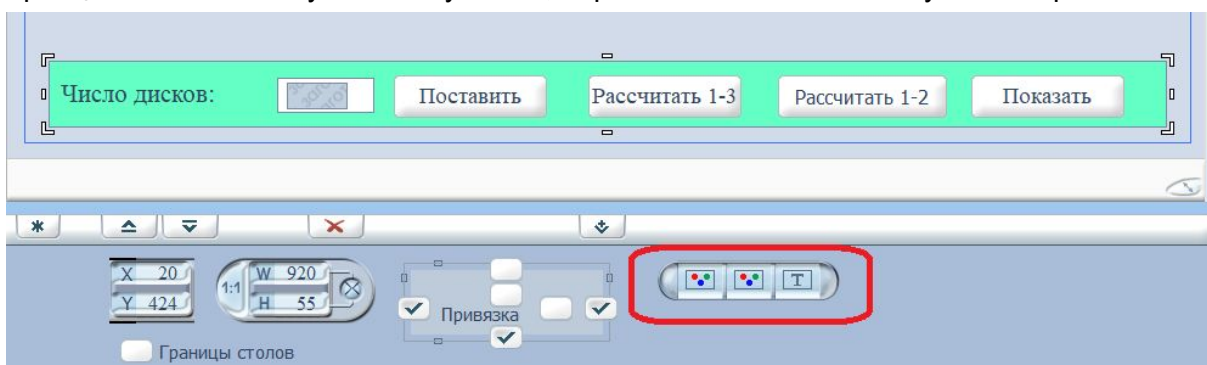
Основные рабочие зоны (см. красные цифры):

- 1) Поле редактора
- 2) Сменный пульт настройки
- 3) Инструменты вставки компонент
- 4) Кнопки запуска программы

2. Редактирование видимого стола

В поле редактора показан рабочий стол, почти в том виде, как он будет выглядеть в рабочей программе.

При щелчке по любому элементу стола, переключается сменный пульт настройки:



На пульте стоят блоки:

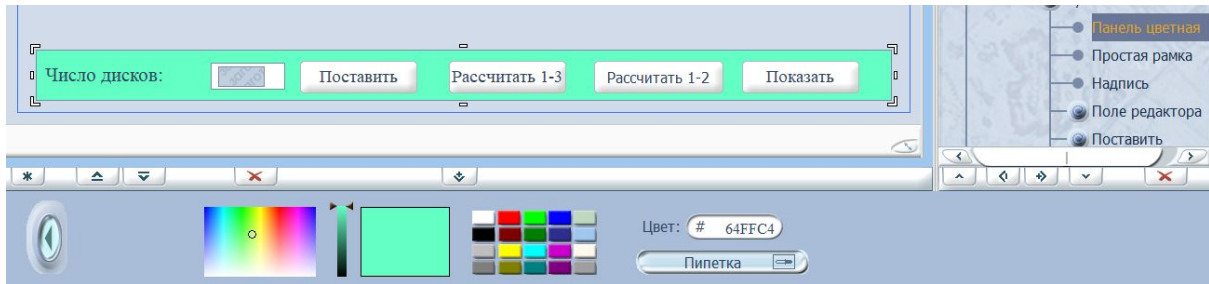
- положение элемента
- размеры элемента
- привязки (позже)

- пульт настройки изображения (в красной рамке)

На выделенном элементе можно настроить

- цвет фона
- цвет рамки
- изображения текста

Соответственно, на пульте настройки стоят три кнопки. Щелчок по первой кнопке цвета открывает пульт выбора цвета:



Обратите внимание, что справа - на схеме выдел инструмент “Панель цветная”. Именно этот инструмент рисования будет настроен через пульт.

Если вернуться на предыдущий пульт (овальная кнопка со стрелкой “назад”) и нажать кнопку “Т”, то откроется пульт настройки надписи, а справа инструмент “Надпись”.

Сделайте любое изменение (цвета или надписи), и нажмите кнопку:



Щелчок по этой кнопке делает сборку программы (build) и запускает ей. Сборка программы может идти несколько секунд.

Проверьте, что изменения появились в работающей программе.

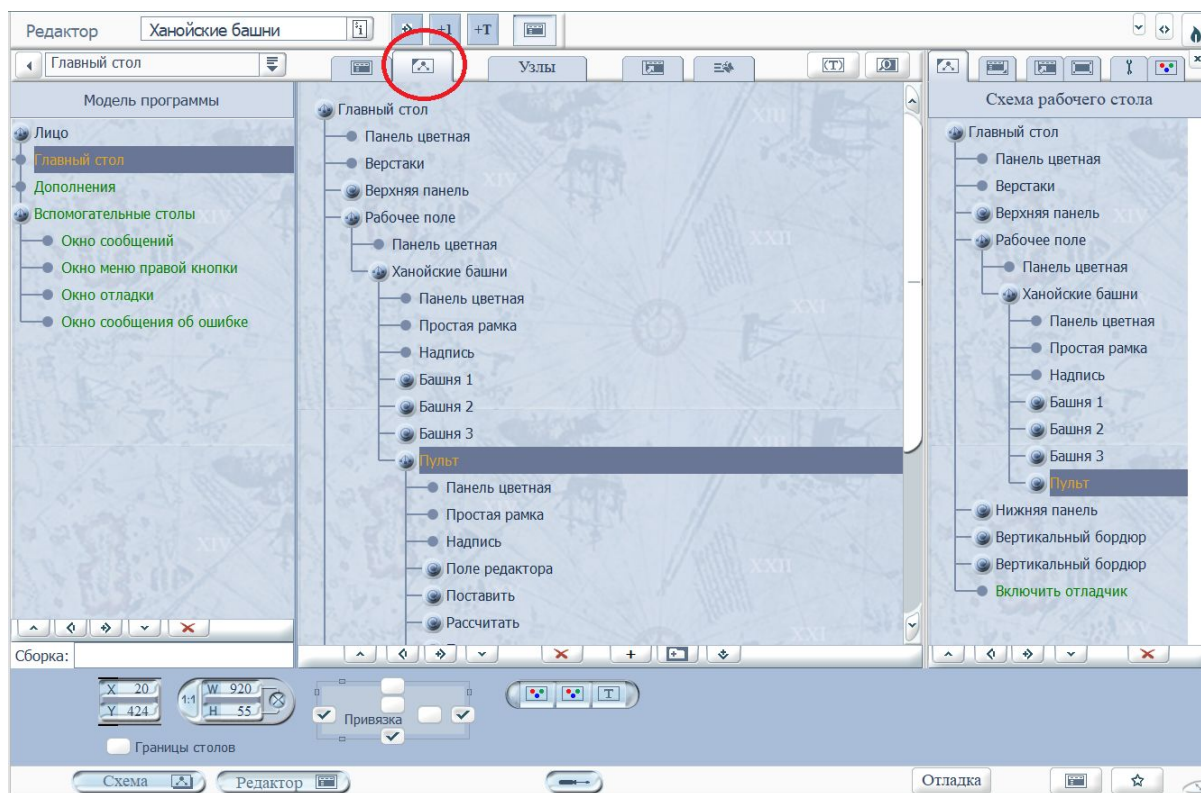
Рядом с кнопкой сборки стоит кнопка запуска:



запускает программу без сборки (без изменений). Полезность этого будет объяснена ниже.

3. Редактирование схемы рабочего стола

Как мы уже видели, схема рабочего стола обычно открыта в правом окне, но для редактирования удобно открыть схему в основном окне, для этого есть кнопка:



В центре открылась схема рабочего стола, а слева схема программы, так же как она была на одном из основных рабочих столов Вира, который мы рассматривали первым, “Схема программы”.

Схема - это код программы “верхнего” уровня. Меняя схему, мы можем получить существенно другую программу из тех же программных компонент.

Схема рабочего стола состоит из “столов”, “инструментов” и “узлов”. Столы создают иерархию, это “строительный элемент”.

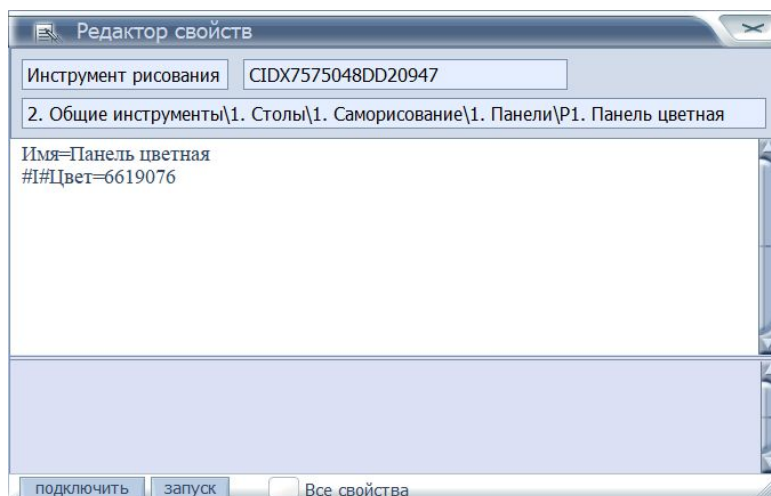
На картинке выше мы видим столы с именами “Рабочий стол”, “Ханойские башни”, “Башня 1” и т.д. Стол - это компонента (объект) с набором команд, все столы в программе одинаковые.

Каждый **инструмент** - также является компонентой (объектом), решающей отдельную задачу. Например, инструмент “Панель цветная” - рисует цветом окно (прямоугольник) на экране.

“Узел” - это компонента более высокого уровня, состоящая из столов и инструментов. На схеме рабочего стола узлы показаны зеленым цветом.

Любая компонента в схеме обладает свойствами, которые можно задать или редактировать.

Двойной щелчок по любому элементу схемы открывает окно свойств. Щелкнем по “Панель цветная”:



Сверху редактор показывает основные свойства (неизменяемые):

- это “инструмент рисования”
- у него есть уникальный номер (code id)
- и указано место инструмента в репозитории (в хранилище инструментов)

Дальше идет список редактируемых свойств:

- Имя (для выполнения программы не имеет значения, важен только code id)
- Цвет. Префикс #I# - указывает, что это Integer (целое число)

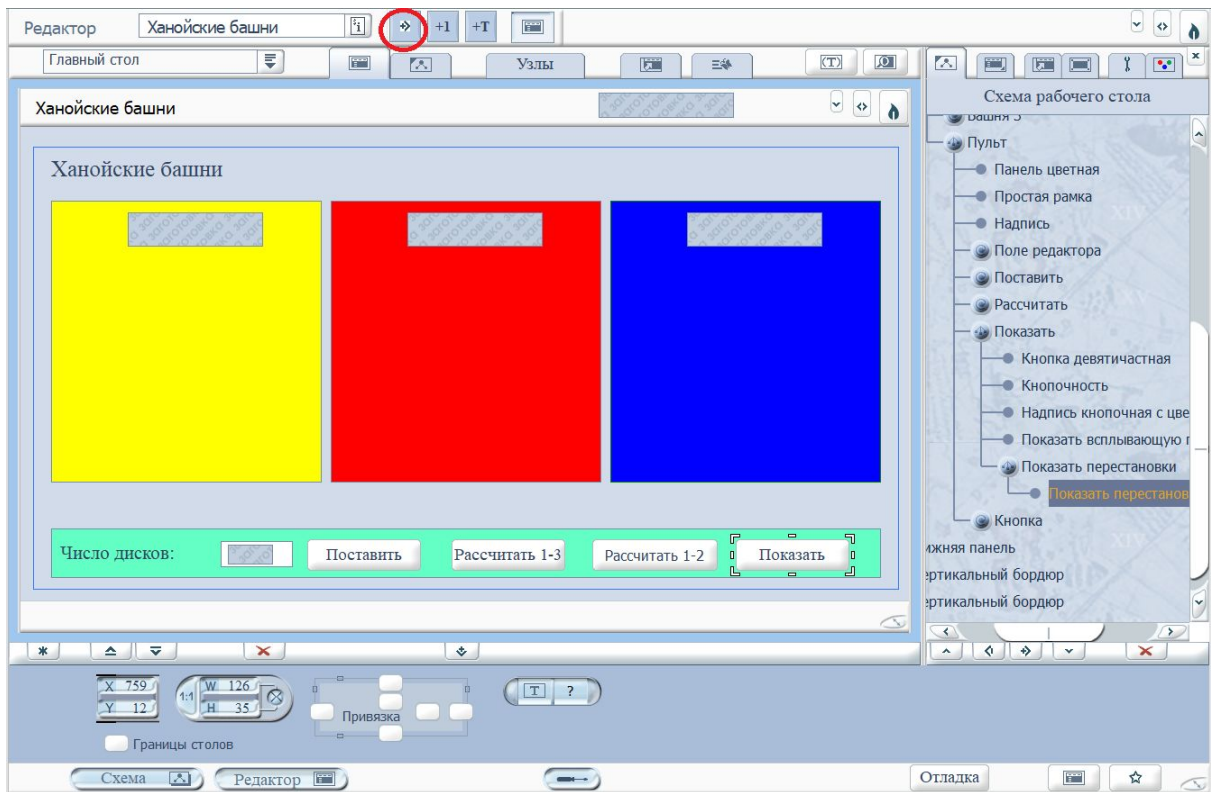
Каждое свойство записывается строкой в редакторе. В основном свойства бывают или числами или цепочками (strings). Для чисел используется префикс #I#, для цепочек использовать префикс не надо. Кодировка цепочек всегда в формате Unicode (UTF-16).

В список свойств можно добавить любое свойство, но есть смысл добавлять только те, что используются инструментом.

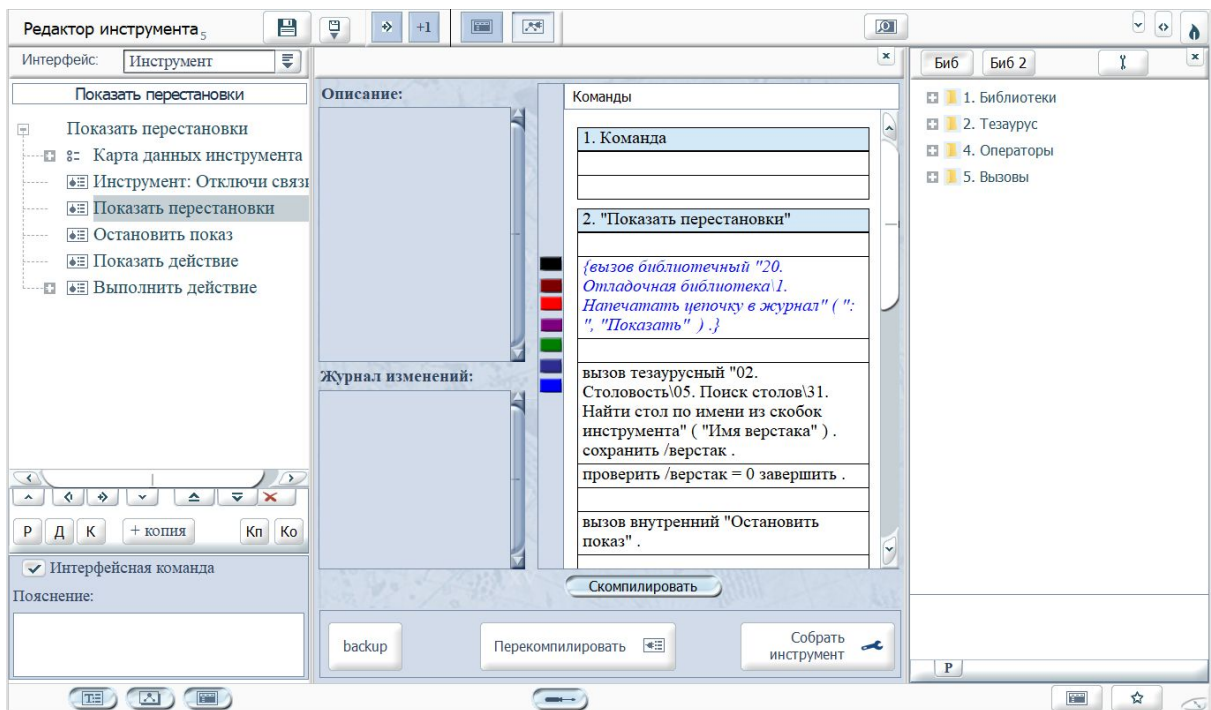
Если вы меняли цвет панели через пульт задания цвета, то здесь можно сделать то же самое, указав цвет, как десятичное число.

4. Редактирование инструмента и статьи

На видимом столе щелкните по кнопке “Показать”. В правом окне откройте схему кнопки, выберите “Показать перестановки” - как на картинке:



Откроется рабочий стол “Редактор инструмента”:



Слева - список команд инструмента

Справа - окно репозитория

Посредине - описание и журнал изменений (тексты) и код выбранной для редактирования команды, который открывается по щелчку слева.

Кратко расскажу про кнопки, не углубляясь в детали:

- “Скомпилировать” - компилирует команду
- “backup” - делает резервную копию инструмента
- “Перекомпилировать” - компилирует все измененные команды
- “Собрать инструмент” - собирает код инструмента из кода команд. Если сделать изменения в инструменте и не собрать его, то при запуске создаваемой программы будет использован предыдущий код инструмента.

После сборки инструмента можно нажать кнопку “Запустить программу” (справа внизу), будет использована свежая версия инструмента (делать сборку программы не обязательно).

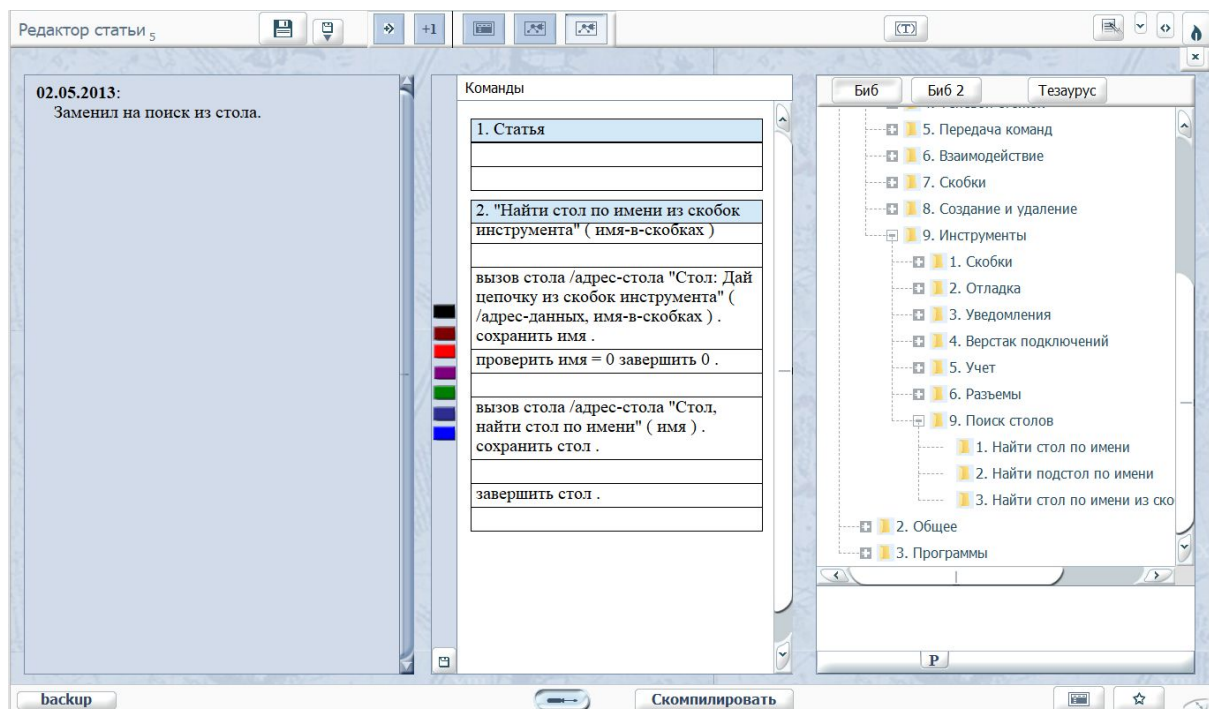
Взглянем на код инструмента:

- в фигурных скобках - комментарий
- следующая строка после комментария начинается с “вызов тезаурусный” - это вызов статьи из тезауруса.

Если щелкнуть по этой строке и нажать кнопку



откроется рабочий стол “Редактор статьи”:



После изменения/компиляции статьи, чтобы запустить программу с изменением, надо собрать инструмент, в котором используется статья, или собрать программу целиком.

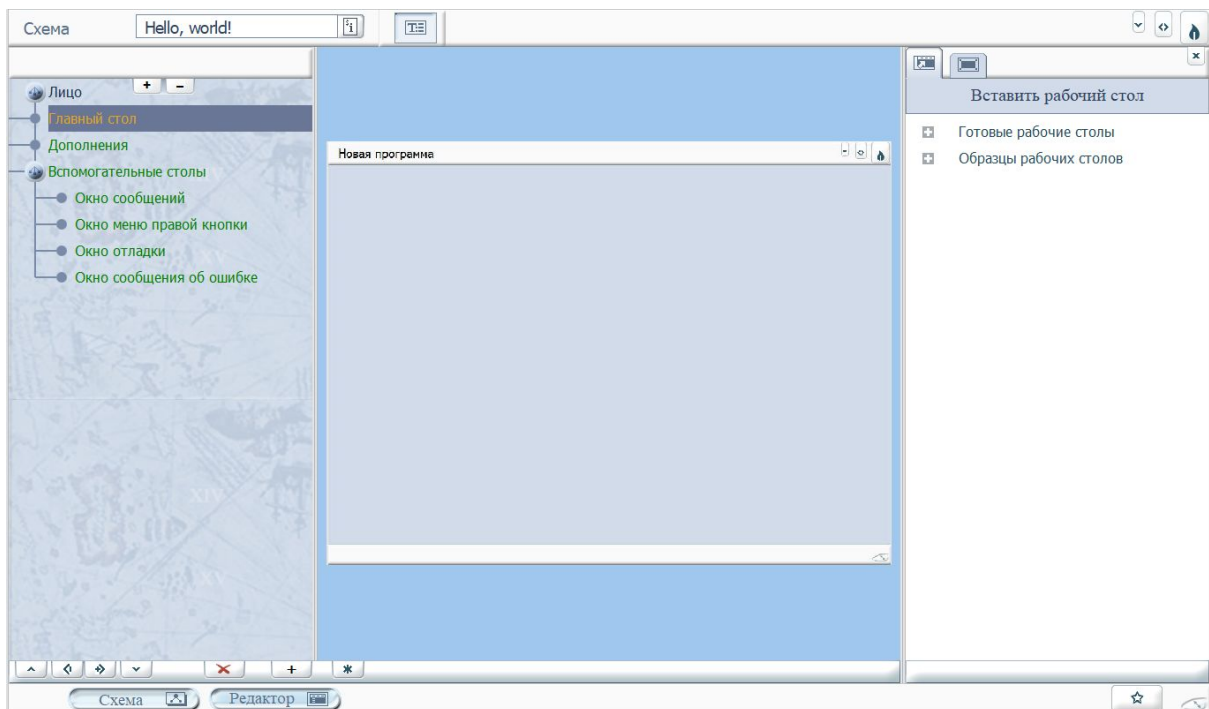
Об используемом языке программирования поговорим немного подробнее в следующем разделе”.

Делаем “Hello, world”

Запустите Вир, напечатайте имя программы:



Нажмите кнопку “создать новую программу” (правее названия). Откроется рабочий стол “Схема программы”:



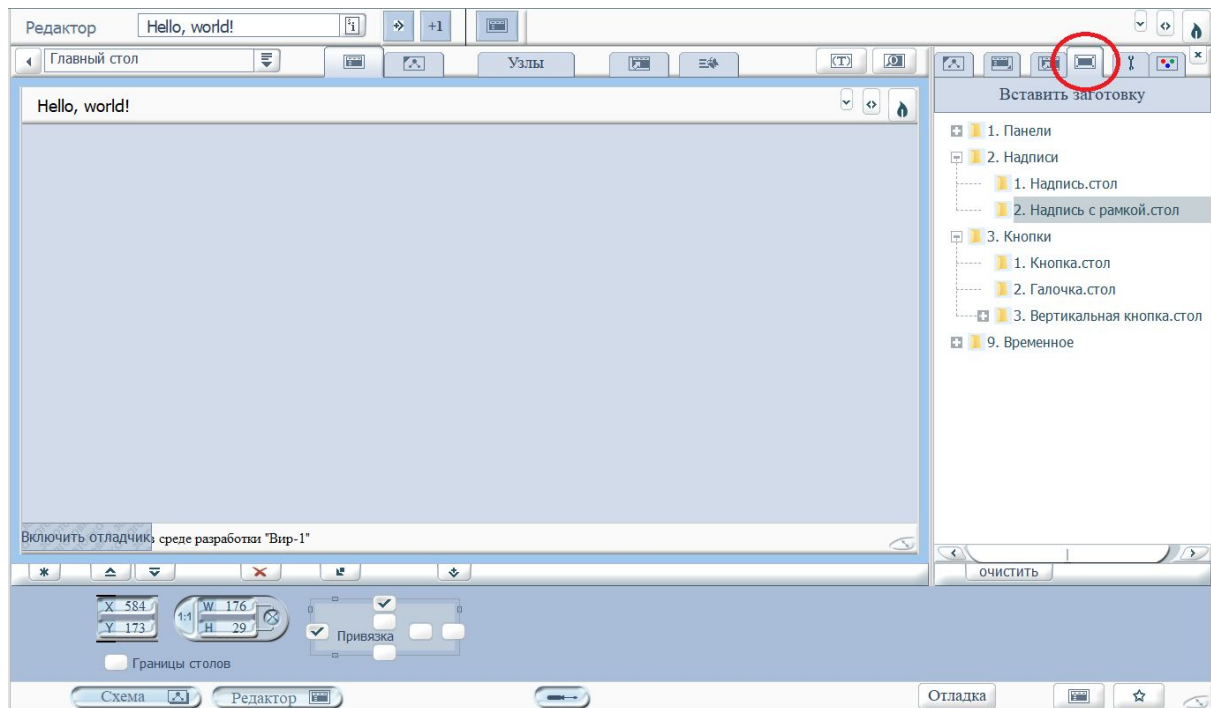
Созданная программа сразу готова к работе. Для проверки нажмите “Собрать и запустить программу” (справа внизу).

По сути мы уже сделали программу, которая показывает надпись “Hello, world!”.

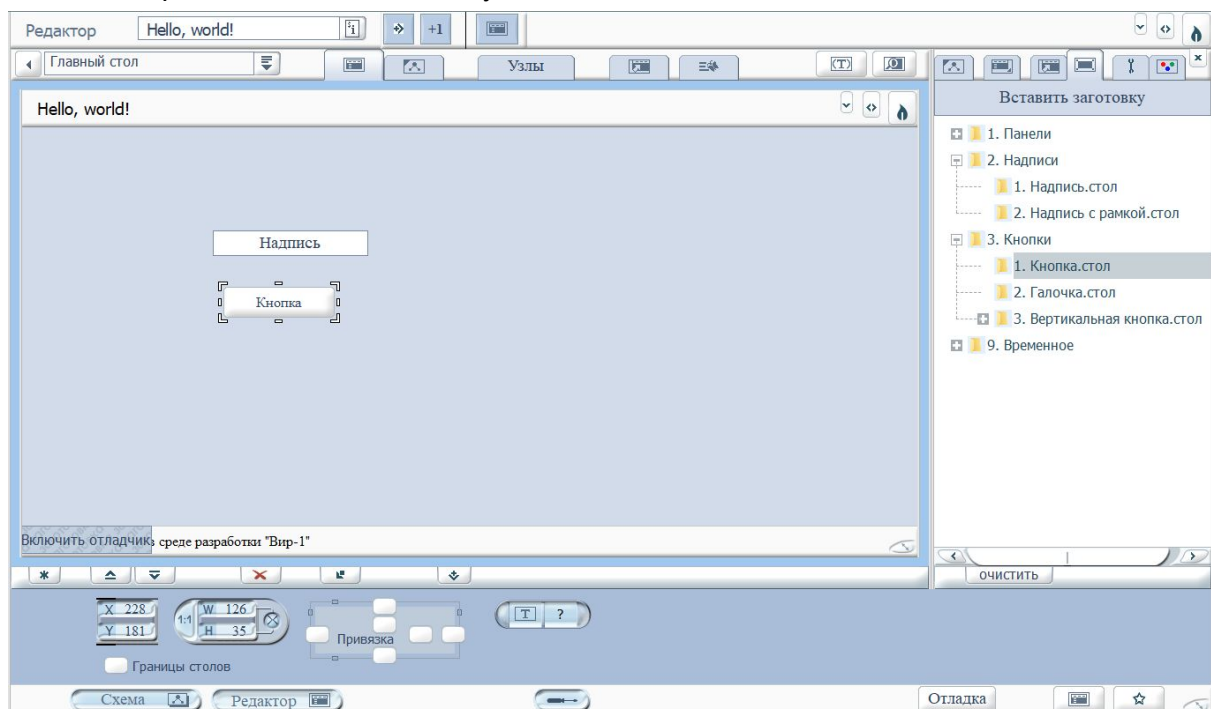
Но мы сделаем чуть более продвинутую версию с кнопкой и полем для ответа.

При нажатии на кнопку “Hello, world”, программа будет отвечать надписью, например, “И тебе привет”.

Так как для простейшей программы не надо ничего менять в схеме программы, нажимаем кнопку “Редактор” и переходим на рабочий стол “Редактор”.



Нажимаем справа кнопку “Вставить заготовку”, переносим мышью на рабочий стол “Надпись с рамкой” и “Кнопка”, получаем:



Дальше делаем:

1. Сотрем слово “Надпись” из поля

2. Изменим надпись на кнопке
3. Подключим к кнопке инструмент, отправляющий сообщение с текстом
4. К надписи подключим прием сообщения с заменой текста

1. Сотрем слово “Надпись” из поля

Двойной щелчок по полю включает на месте редактор надписи. Стираем надпись.

2. Изменить надпись на кнопке

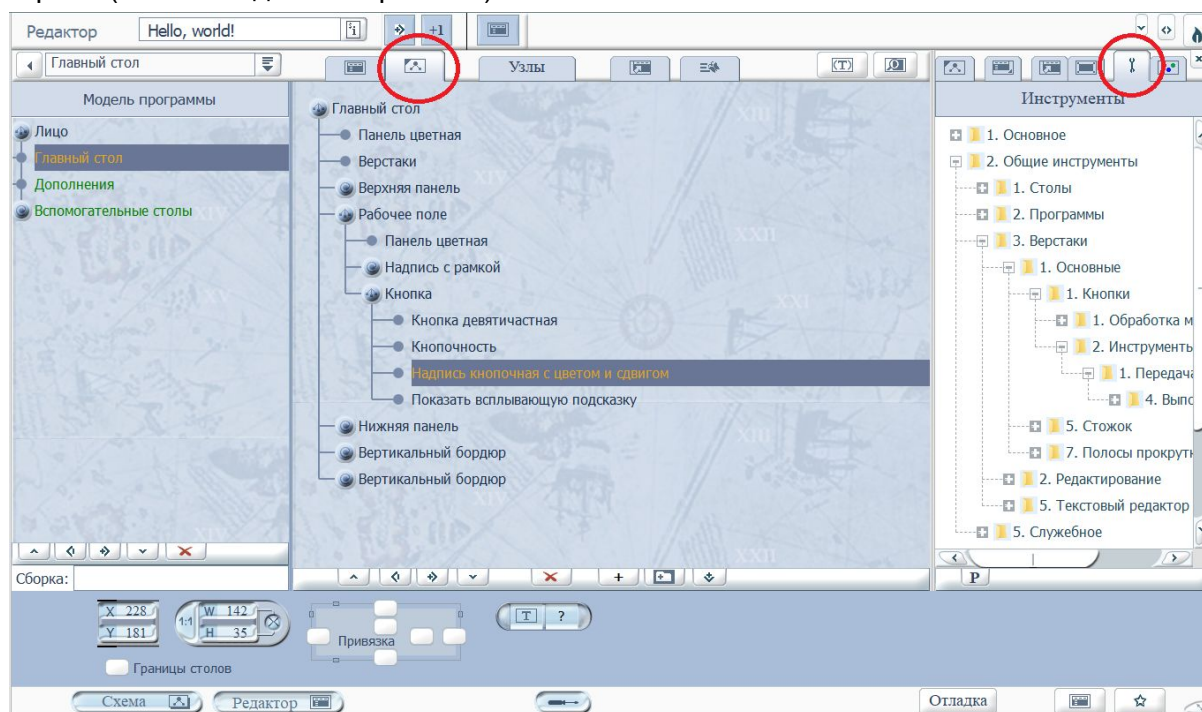
Двойной щелчок по кнопке включает на месте редактор надписи.

Печатаем: “Hello, world!”

Изменить размеры кнопки можно потянув за рамку, переместить - нажав правую кнопку мыши. Изменить параметры надписи (шрифт, цвет, выравнивание, ...) - нажав кнопку T на пульте внизу.

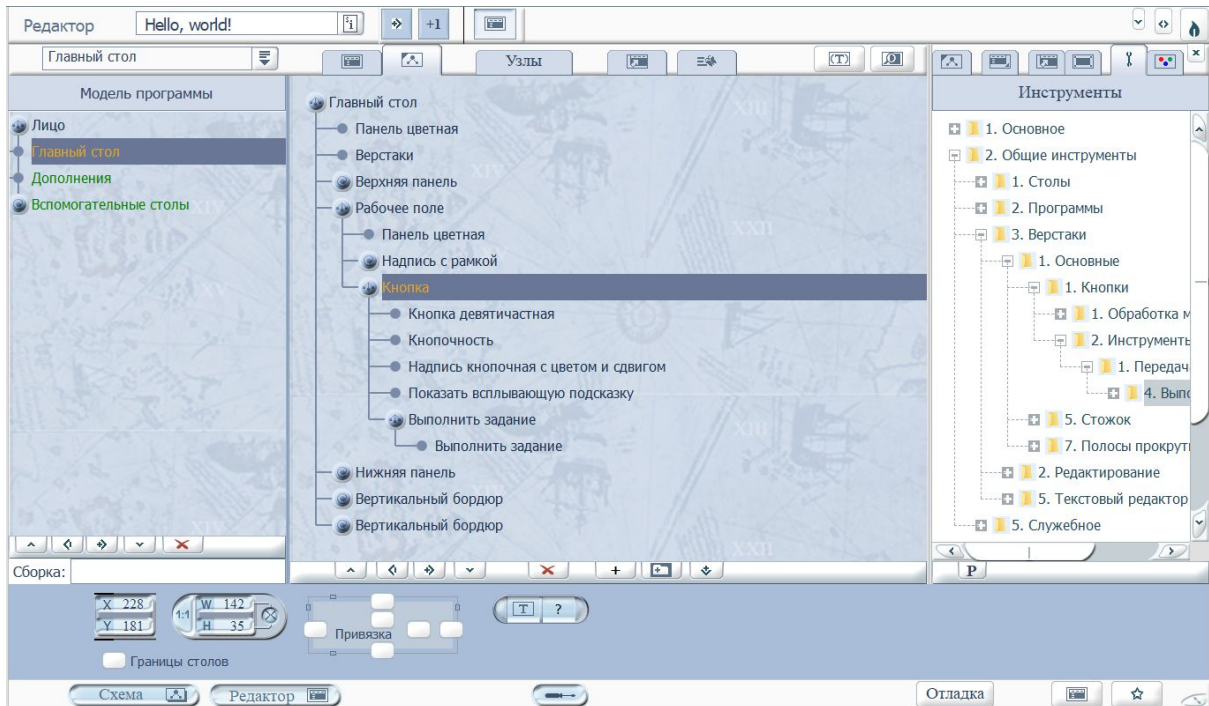
3. Подключить инструмент к кнопке

Включаем схему рабочего стола на средней панели, пульт “Вставить инструмент” - справа (кнопки выделены красным):



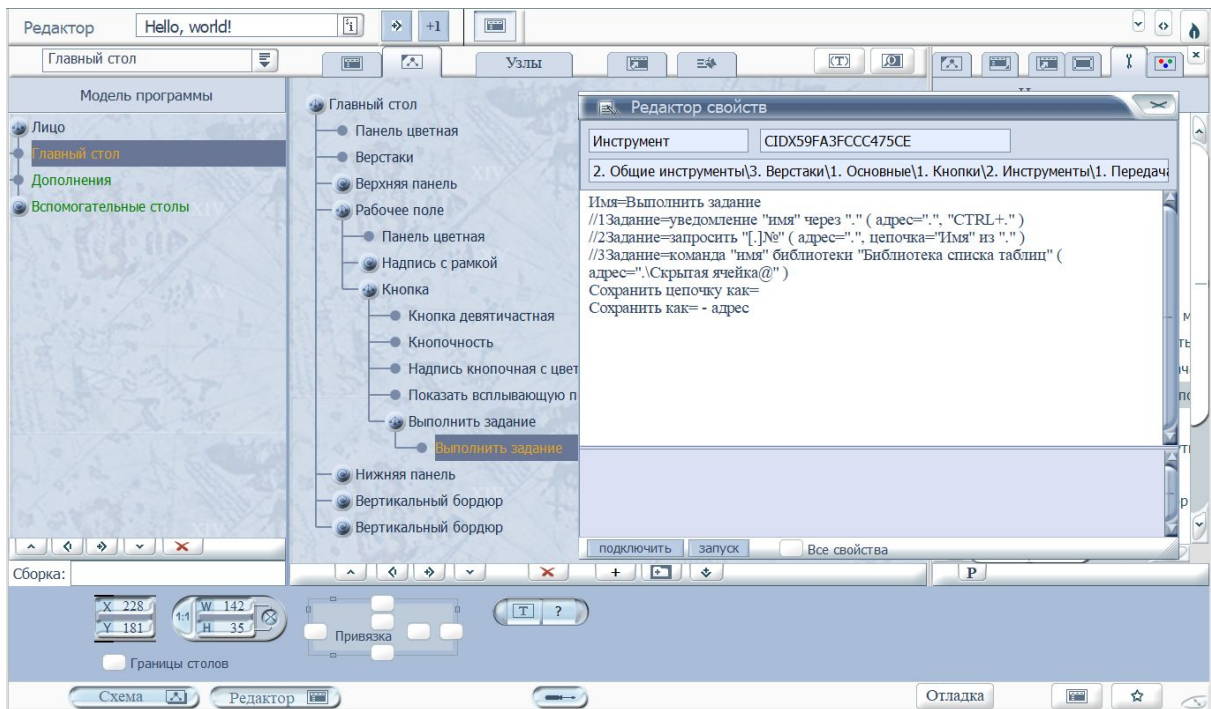
Справа открываем “2. Общие инструменты\3. Верстаки\1. Основные\1. Кнопки\2. Инструменты\1. Передача управления”.

Мышь берем инструмент “4. Выполнить задание”, зажимаем клавишу Ctrl и бросаем в середину на строку “Кнопка”. Должно получиться:



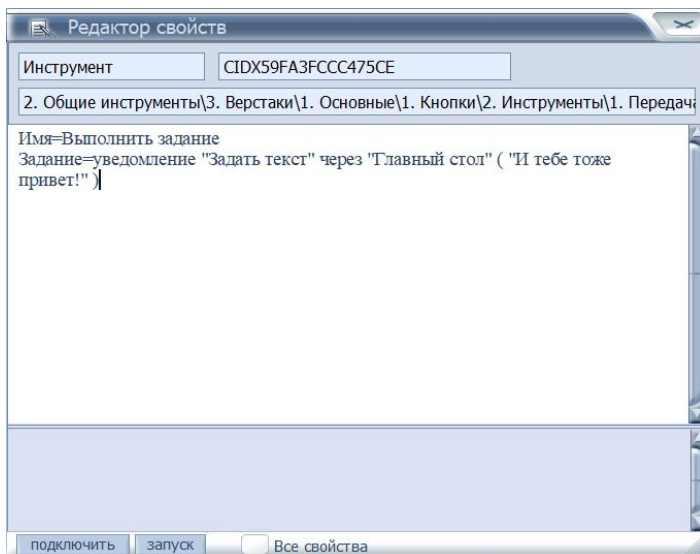
Клавишу “control” надо нажимать, чтобы инструмент вставился вместе с верстаком (столом). На картинке мы видим, что на кнопке теперь стоит стол “Выполнить задание”, а на нем инструмент с тем же именем”. Щелчок по кнопке обходит все столы, стоящие на кнопке и запускает инструмент на каждом столе. Это позволяет собирать составные действия.

Сделаем двойной щелчок на инструменте “Выполнить задание” (это лист дерева), откроется окно свойств:



Инструмент “Выполнить задание”, как и понятно из его название, выполняет задание записанное в скобках. Не будем пока углубляться в детали. Удалим все строки, кроме строки “Имя” и напечатаем:

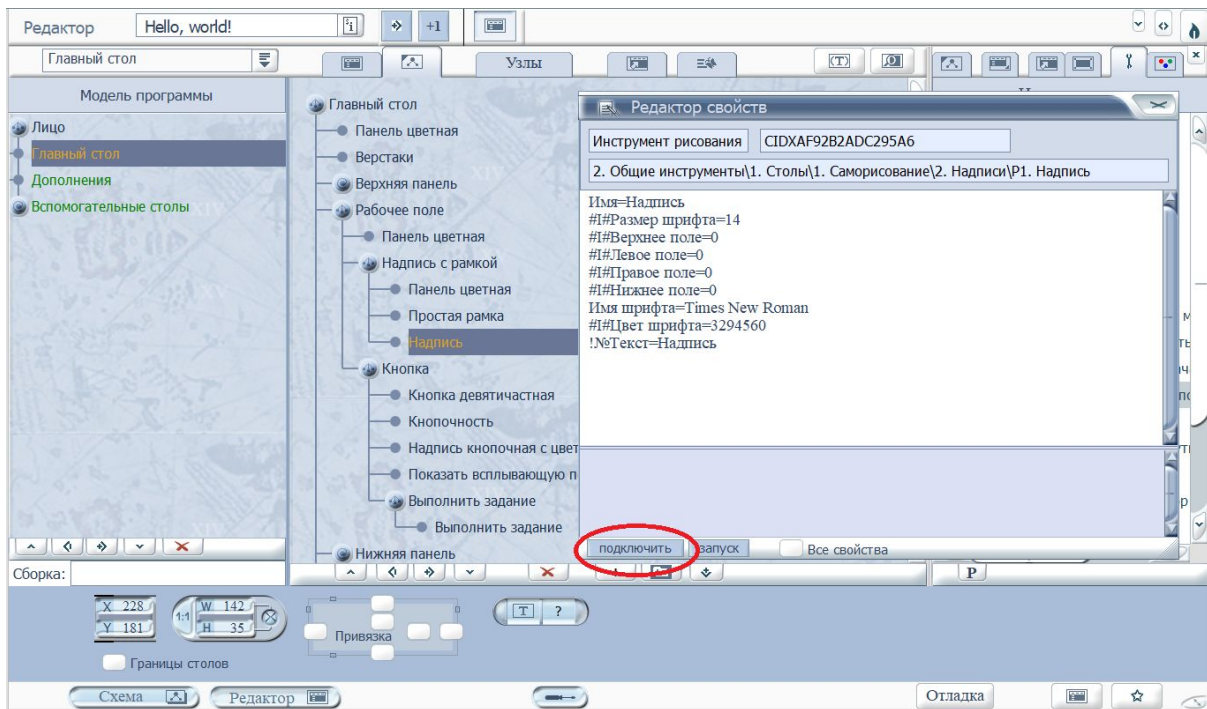
Задание=уведомление "Задать текст" через "Главный стол" ("И тебе тоже привет!")



Мы получили, что при щелчке по кнопке сработает инструмент, который отправит уведомление “Задать текст” через стол с именем “Главный стол”. У уведомления есть параметр - строка “И тебе тоже привет!”.

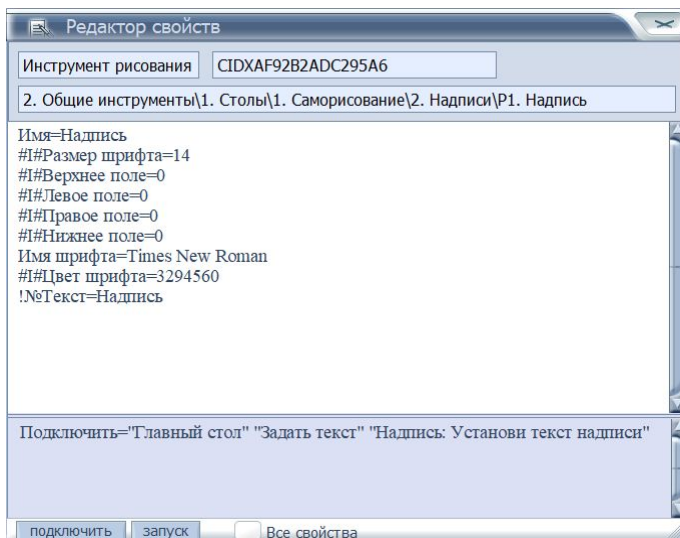
Теперь надо сделать так, чтобы это сообщение было принято.

Щелкаем по инструменту “Надпись” на столе “Надпись с рамкой”



Нажимаем кнопку “Подключить” и вписываем имена в нижней части окна:
 Подключить=“Главный стол” “Задать текст” “Надпись: Установи текст надписи”

чтобы получилось:



Строка “Подключить” задает подключение: инструмент должен реагировать на уведомление “Задать текст”, отправленное через стол с именем “Главный стол”. Реакцией инструмента, который мы редактируем, будет запуск команды “Надпись: Установи текст надписи” со всеми параметрами, которые были заданы при отправке уведомления. Напомню, выше мы писали эти самые параметры: “У уведомления есть параметр - строка “И тебе тоже привет!””.

Запускаем программу, нажав кнопку “Собрать и запустить программу”. Нажимаем кнопку и получаем ответ:



Что дальше?

На самом общем уровне, “Вир” описан. За бортом осталось множество деталей, в том числе описание языка программирования.

Если будут вопросы, я на них отвечу и напишу следующие части инструкции.