Клиенту важно, чтобы продукт, который он использует, был прост в употреблении и при этом решал как можно больше сложных задач. Продукт может соответствовать этим ожиданиям при условии, что его разработкой будут заниматься профессиональные исследователи и разработчики. При этом с одной стороны, жизненный цикл функционирования программных продуктов постоянно сокращается, что автоматически приводит к ускорению процессов разработки и рыночного продвижения новинок. С другой стороны, усложнение решаемых задач, сопряженное с увеличением количества операций и необходимостью выстраивания сложных связей между ними, приводит к усложнению техники. Эта изменяющаяся ситуация требует принципиально новых стратегий выстраивания процесса создания программного продукта, где, с одной стороны, необходимым условием является высокий уровень профессионализма как исследователей, так и разработчиков, а с другой стороны, их способность к продуктивному взаимодействию.

Одной из возможных линий взаимодействия является «технологическая интеграция». Данный вид взаимодействия подразумевает процесс выбора и настройки технологий, используемых компаниями для разработки программных продуктов, осуществляемой в два этапа. Первый, когда небольшое количество высококлассных исследователей решают стратегически и тактически поставленную практикой задачу. Второй, когда разработчики реализуют данное решение, фактически становясь интеграторами технологий. Результатом их деятельности является конкретный программный продукт, выпускаемый на рынок.

Следует отметить, данная интеграционная стратегия предполагает, что исследователи в основном ориентированы на разработку новых технологий и несут ответственность за отдельные компоненты общего проекта. Разработчики обеспечивают выпуск конечной продукции, они ориентированы на функциональные возможности готовой продукции, и при этом отвечают за весь процесс в целом. Все это делает программные продукты, выводимые на рынок, более компактными и менее дорогими по сравнению с решениями, реализованными с использованием иных стратегий взаимодействия участников проекта.

Таким образом, стратегия «технологической интеграции» предполагает слаженность и согласованность между разными организационными задачами, включая поддержку экспериментальной составляющей в разработке проекта и привлечение сторонних специалистов для решения конкретной задачи. Это является важным звеном в процессе решения поставленной задачи в целом.

Следуя этой стратегии, компанией НПФ «И.В.А.» была создана платформа «LogicProgram». Данная платформа представляет среду программирования и эксплуатации программных продуктов, которая позволяет включить в процесс разработки персонализируемых приложений большое количество людей. Технологическая интеграция лучших алгоритмов с одновременным снижением требований к уровню владения языками программирования позволяет получать высококачественные программы в более короткие сроки и с требуемым уровнем функциональности. Благодаря наличию единого стандарта оформления

компонента данной платформы сторонние разработчики могут дополнять её новыми компонентами, обеспечивающими решение поставленных задач.

Таким образом, специфическими особенностями платформы LogicProgram является следующее:

Во-первых, эта платформа позволяет любому человеку, не владеющему особыми навыками в области программирования, создавать приложения. Для программистов же со стажем эта программа может быть интересна тем, что она позволяет повысить скорость разработки приложений.

Во-вторых, существует много систем, выполняющих сходную с LP задачу. Но их принцип работы сводится к тому, что после составления блок схемы или матрицы связей, формируется «код», который компилируется (переносится) в готовую, исполняемую программу. Основное отличие программирования в LP состоит в том, что она позволяет миновать процесс «написания кода».

При проектировании LP-программы разработчик включает в проект готовый компонент (неизменяемый по внутренней структуре), который может быть настроен для работы в определенном для него режиме. Для формирования логики поведения программы между компонентами устанавливается связь. Набор этих связей - и есть программа (*уже готовая к работе*). Внешне способ программирования в LP - это установка связей между компонентами приложения либо через ссылки (как в HTML), либо методом «копирования» ссылок между компонентами (как при работе с файлами).

За счет изложенного принципа функционирования размер LP-программы, оформленной в виде специального файла в формате XML, получается небольшим. Но для её исполнения нужна LP-машина, в которой содержится исполнительная логика компонентов платформы LogicProgram.

Для исполнения программы в каждой операционной системе существует своя LP-машина. При этом программа будет работать без проблем и переделок везде, если программист использовал одинаковые компоненты для каждой ОС. Если использовались специфичные компоненты для операционной системы Android, например, работа с SMS на телефоне, то на компьютере с операционной системы Windows, часть программы, использующая такие компоненты работать не будет, при этом остальная логика программы будет функционировать полностью.

Платформа LogicProgram является результатом предшествующих исследований, разработок и методик, апробированных в системах Miracle, MiraclePlus с 1995г.

Свойства LP-платформы:

- **Простота интерфейса программирования** (ориентация на предметную часть решения, а не на специфику синтаксиса языка программирования, операционной системы, аппаратной платформы и т.п.);
- **Быстрота разработки**, особенно при не четком требовании к конечному результату; разработка в режиме постепенной формализации результата;
- **Простота поддержки** и модификации (в том числе и при модификации «чужого кода»);
- **Простота** разработки **при росте сложности** программы;
- **Исключение** из обзора программиста **специфичных языковых конструкций программирования**, при этом предоставление возможности сосредоточиться на «предметной» логике решения.

LP-платформа - это язык программирования LP (*есть формальная запись программы в формате XML*), LP-Studio - среда разработки программ, LP-машины - среда исполнения программ.

Аналоги:

- Google App Inventor (наиболее схожа с точки зрения идеи)
- HIAsm (внешнее сходство)
- DyCham от компании JetBrains
- Lab View (оболочка для работы со специализированным оборудованием);
- A-Flow (позволяющая создавать Win32 программы на основе сетчатой схемы и внешних списков с данными)

Принципы программирования на LP

1. **Короткий цикл разработки**, контроль полученного результата (развивающийся, эволюционный подход к проектированию программ). Предпосылки: требования, предъявляемые к конечному результату, не всегда являются полными. Понимание требования по мере получения результата. Текущий результат как метод рефлексии. Готовность к изменениям с незначительными затратами. Поддержка регулярных переделок.
2. **Планирование требований к** текущей **итерации** (циклу) программы, а не ко всему результату в целом.
3. **Кодирование в глубину**. В рамках цикла – получение полностью работоспособного решения, закрывающего требования к данной итерации проекта.
4. **Высокая скорость проектирования**. Высокая скорость реализации за счет «упрощенного» интерфейса программирования, за счет полной базы верифицированных компонентов платформы. Графические методы представления структурной логики программы. Использование средств адресного комментария (в теле компонентов). Наличие нескольких средств диагностики исполнения программы: «поточный» (пошаговое исполнение связей компонентов), «диаграмма» (цепочка исполнения связей компонентов).
5. **Итерация**. Способ развития логики программы. Позволяет фиксировать текущий уровень приближения функционала программы к ожидаемому результату. Планирование новой итерации.
6. **Поддержка**. Простота восприятия структурной логики программы. Простота включения в «чужую» программу. Возможность продолжения прерванных действий, за счет графических средств представления структуры связей между компонентами.
7. **Формирование новой системы ценностей**. Вовлечение в процесс проектирования специалистов предметной области при помощи графического представления функционального решения в программе. Проведение качественной оценки результата в процессе разработки. Итерация как метод контроля функциональной полноты исполнения.

Операционные системы, на которых работают программы созданные в LP

- Android
- QNX
- Windows
- Java
- Linux

Примеры LP-Studio

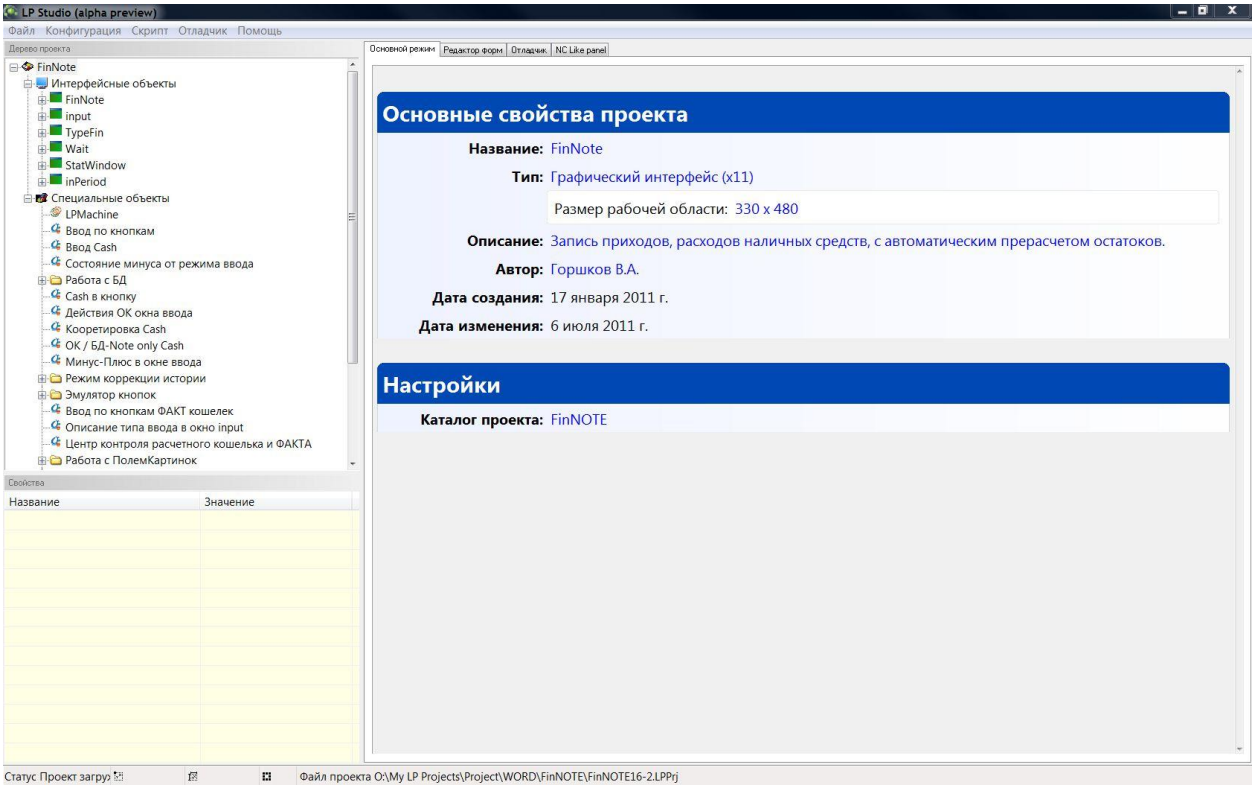


Рисунок 1 Режим работы LP-Studio, среда программирования. Описание проекта

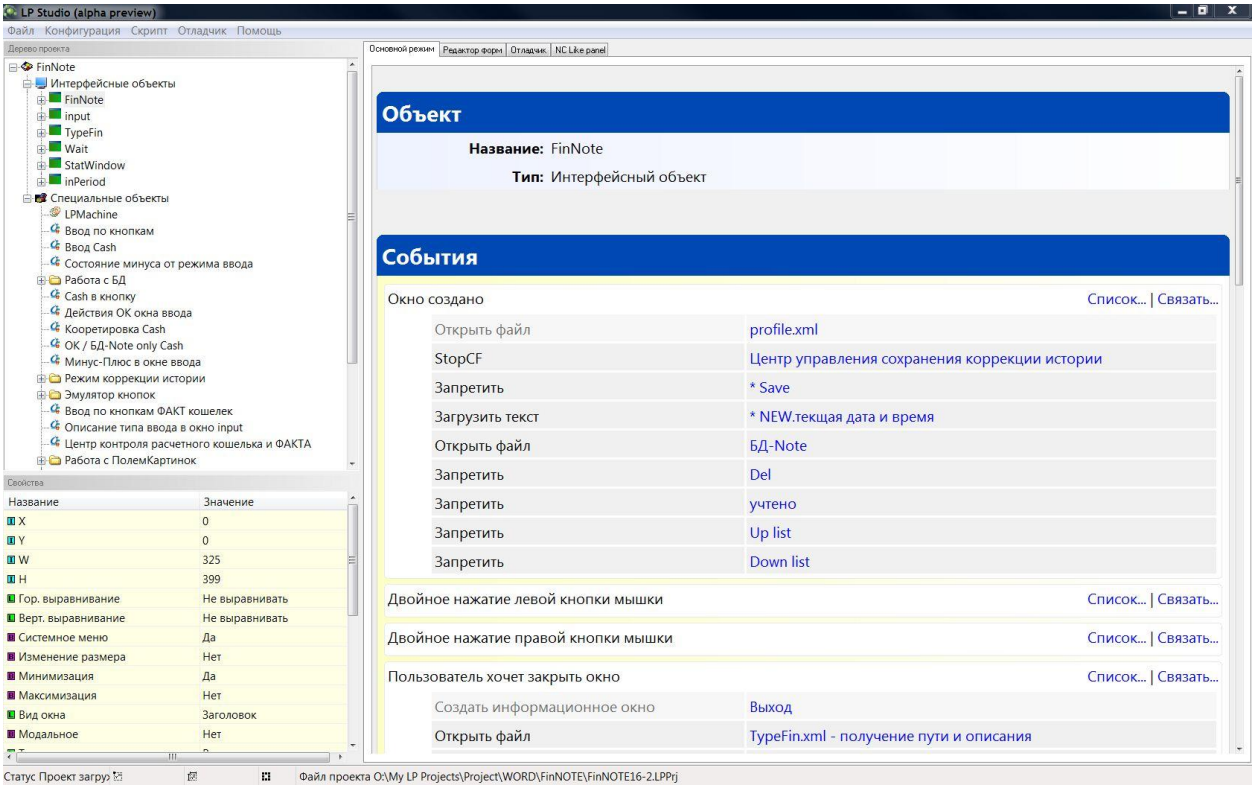


Рисунок 2 Режим работы LP-Studio, среда программирования. Режим программирования связей

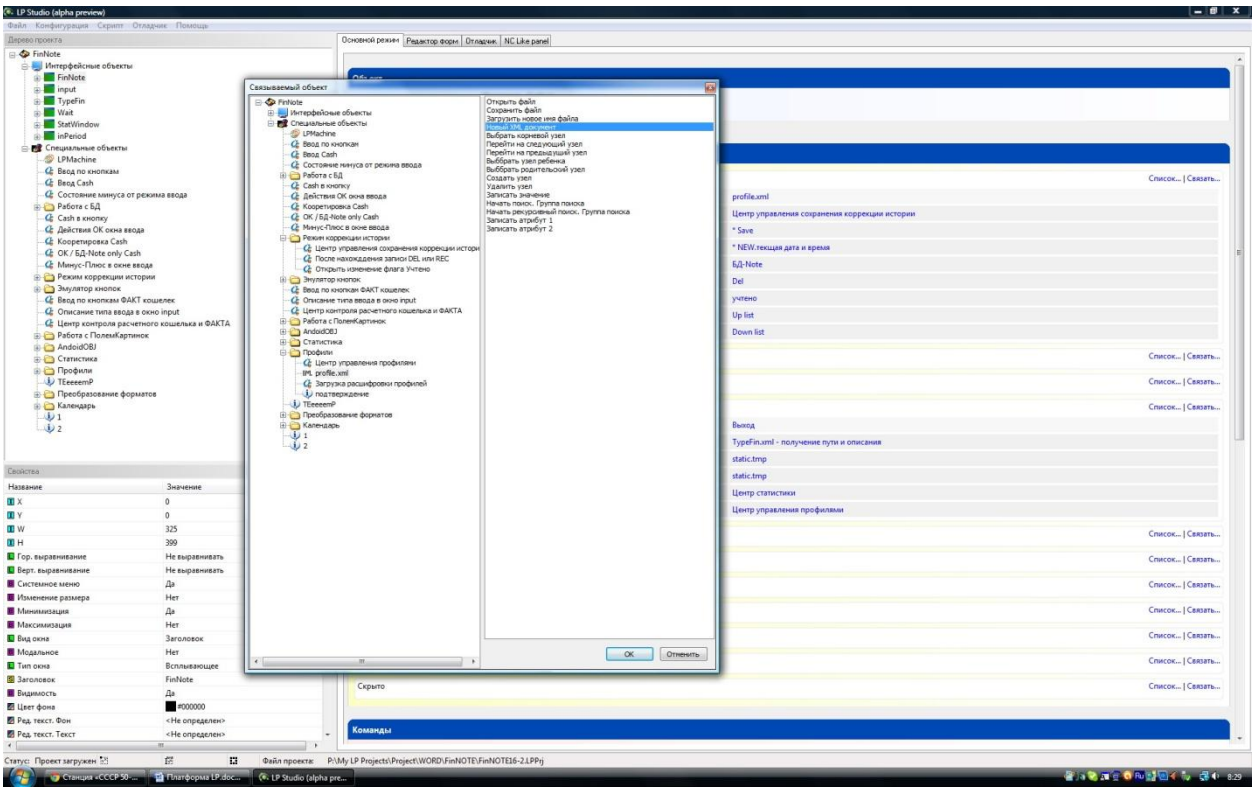


Рисунок 3 Режим работы LP-Studio, среда программирования. Режим программирования связей между компонентами

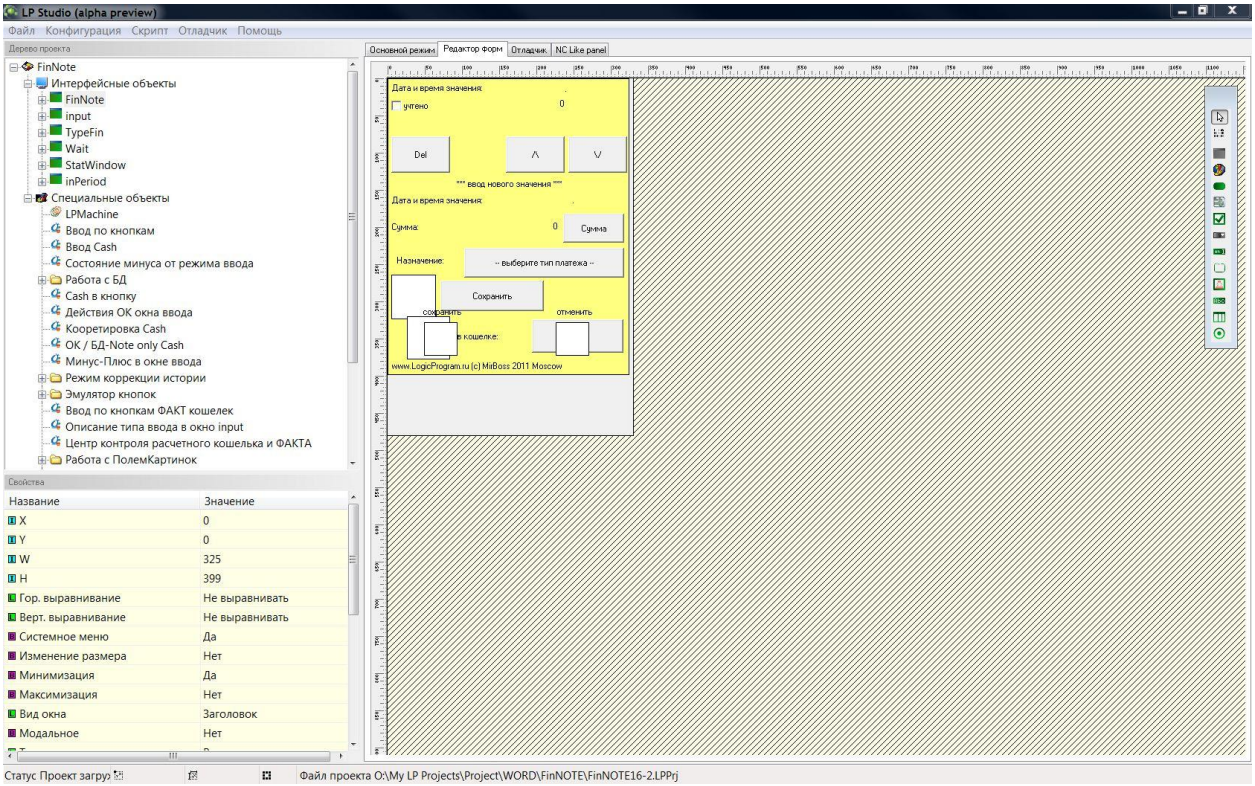


Рисунок 4 Режим работы LP-Studio, среда программирования. Режим проектирования интерфейса пользователя

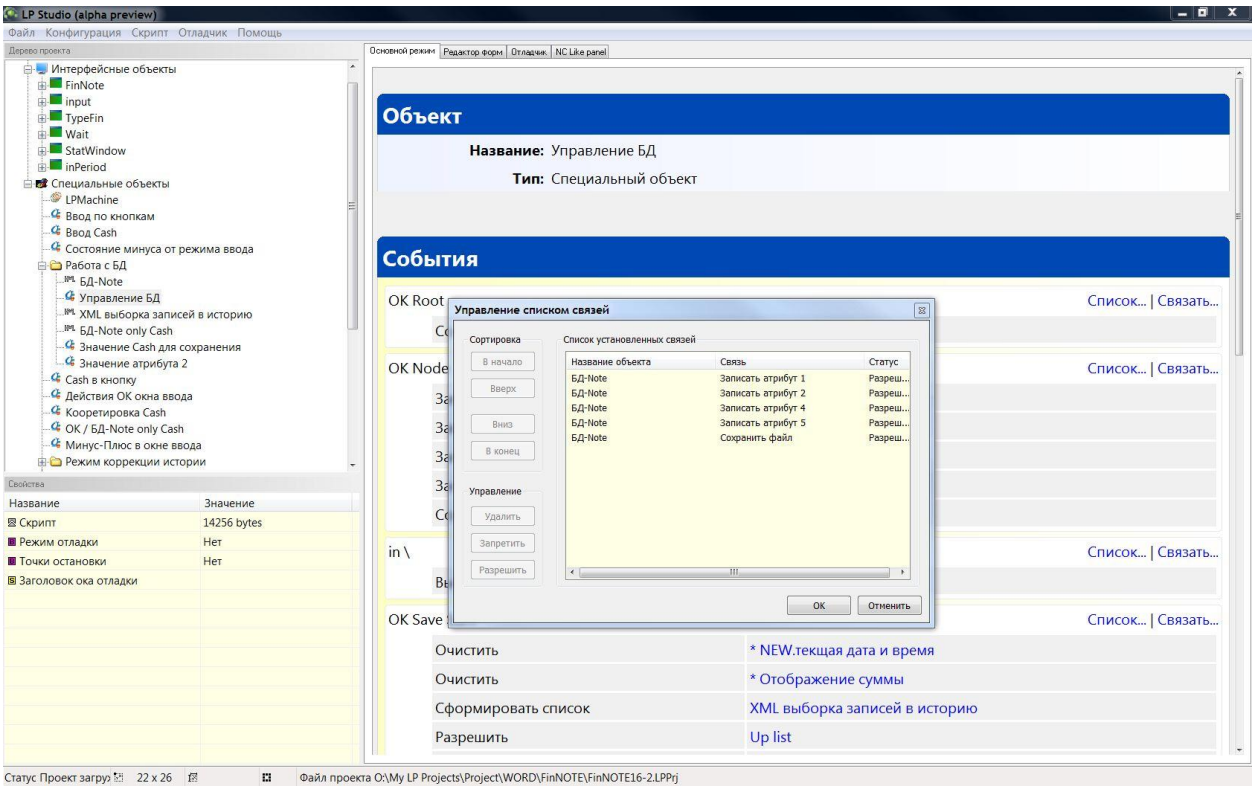


Рисунок 5 Режим работы LP-Studio, среда программирования. Режим управления очередью исполнения

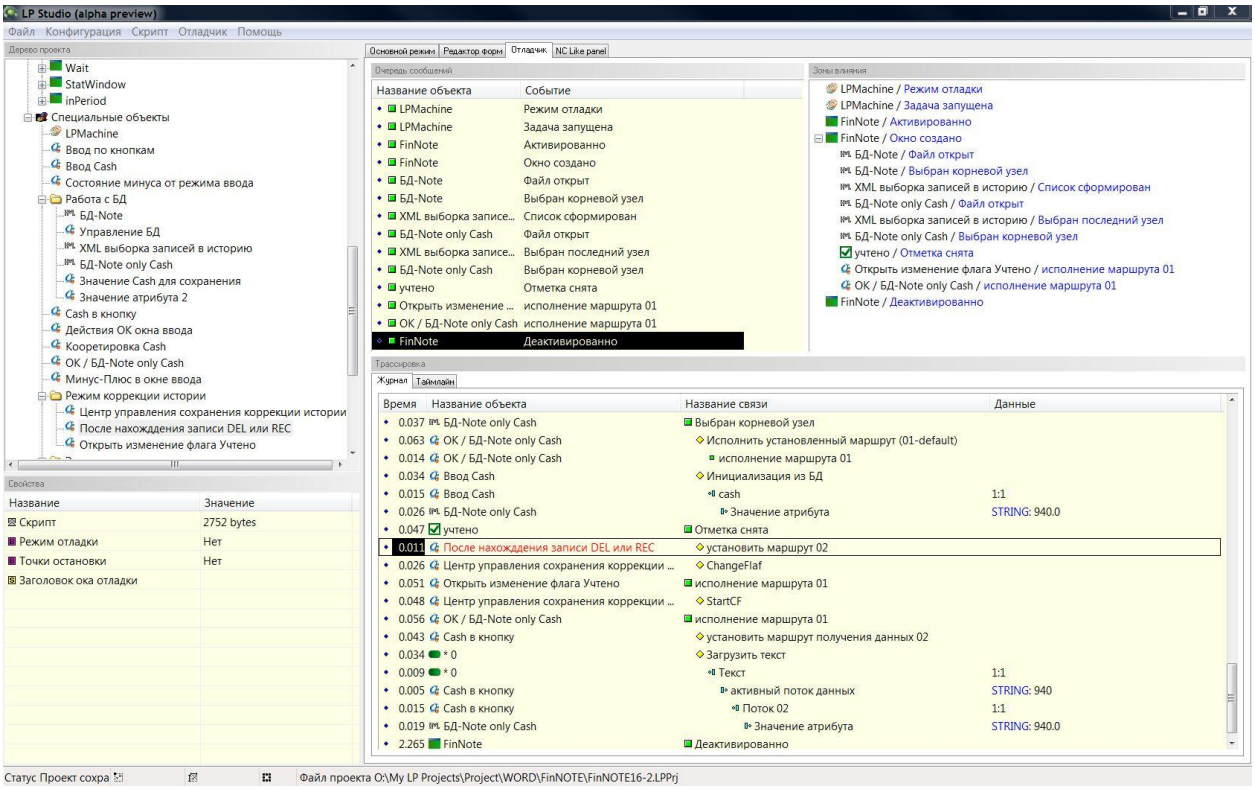


Рисунок 6 Режим работы LP-Studio, среда программирования. Режим отладки работы программы

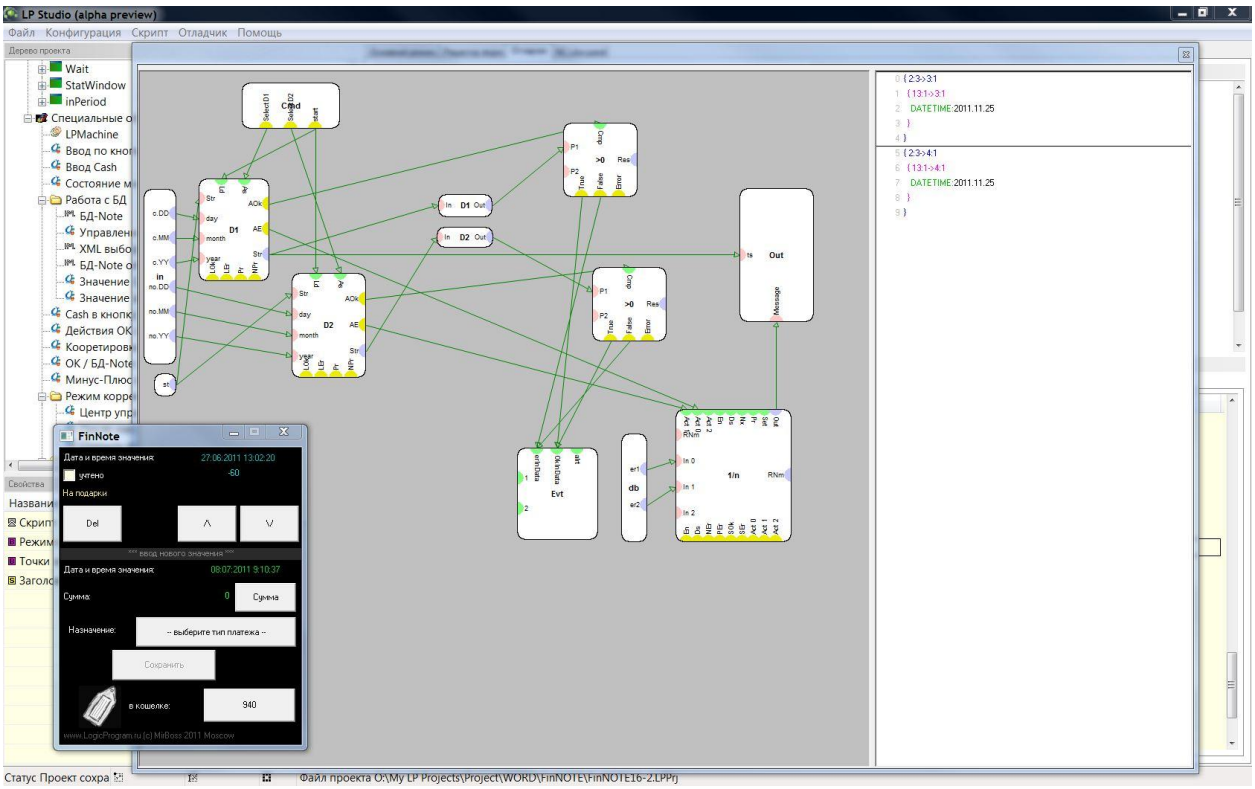


Рисунок 7 Режим работы LP-Studio, среда программирования. Режим отладки работы компонента, спроектированного при помощи LP-Studio

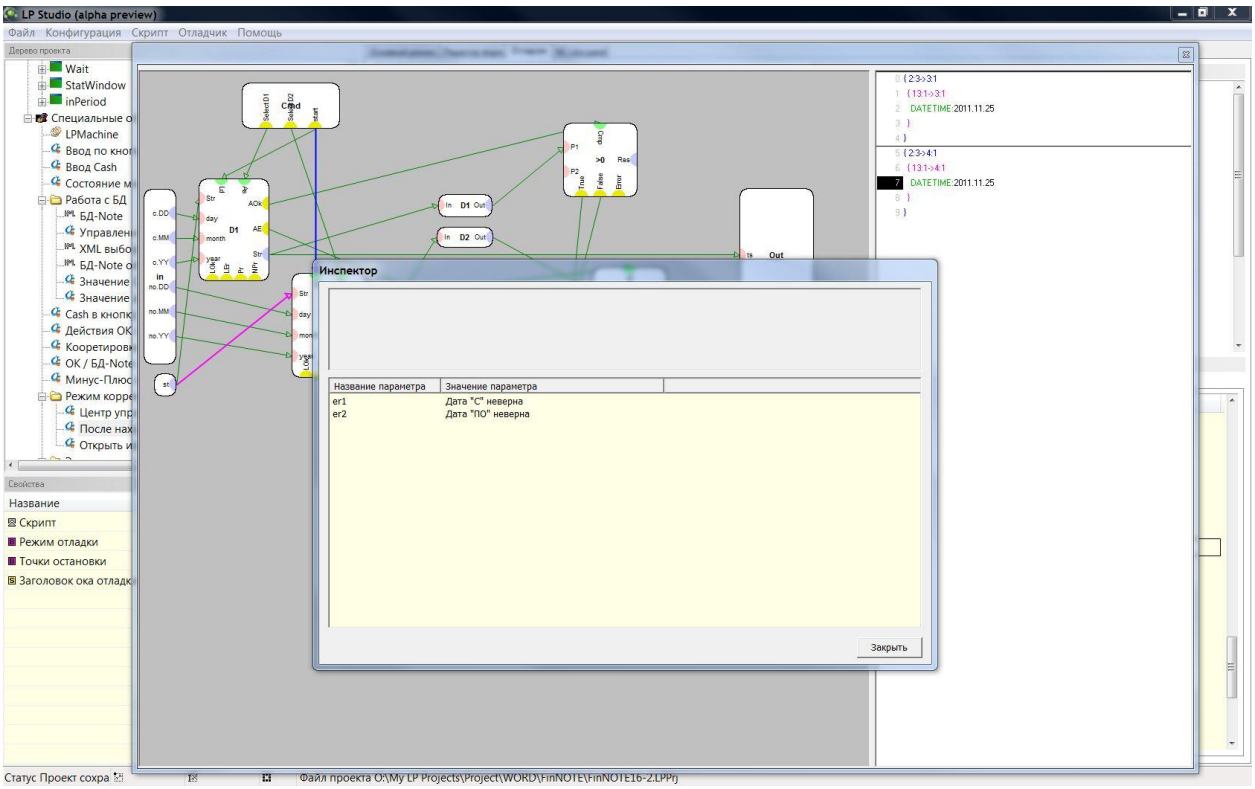


Рисунок 8 Режим работы LP-Studio, среда программирования. Режим отладки компонента, созданного при помощи LP-Studio; отображение содержимого переменных

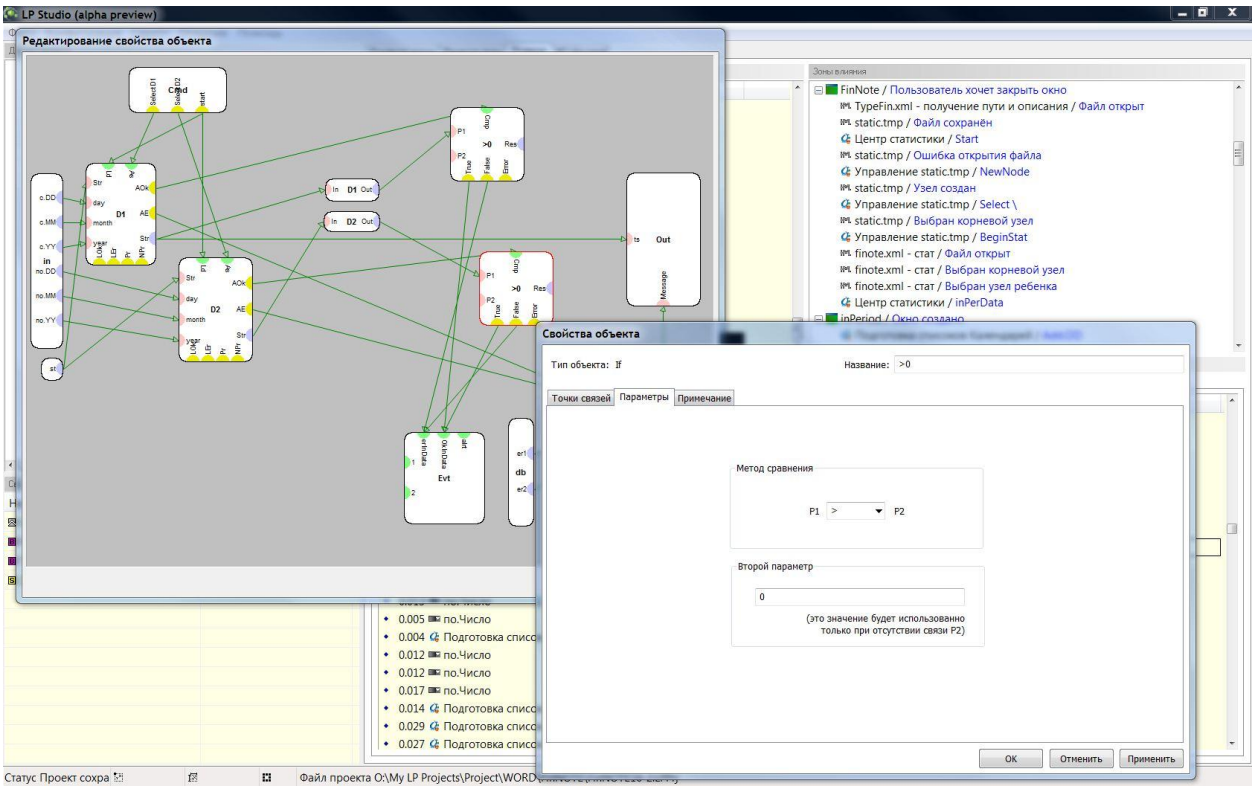


Рисунок 9 Режим работы LP-Studio, среда программирования. Режим программирования структуры нового компонента, установка условия срабатывания одного из элементов компонента

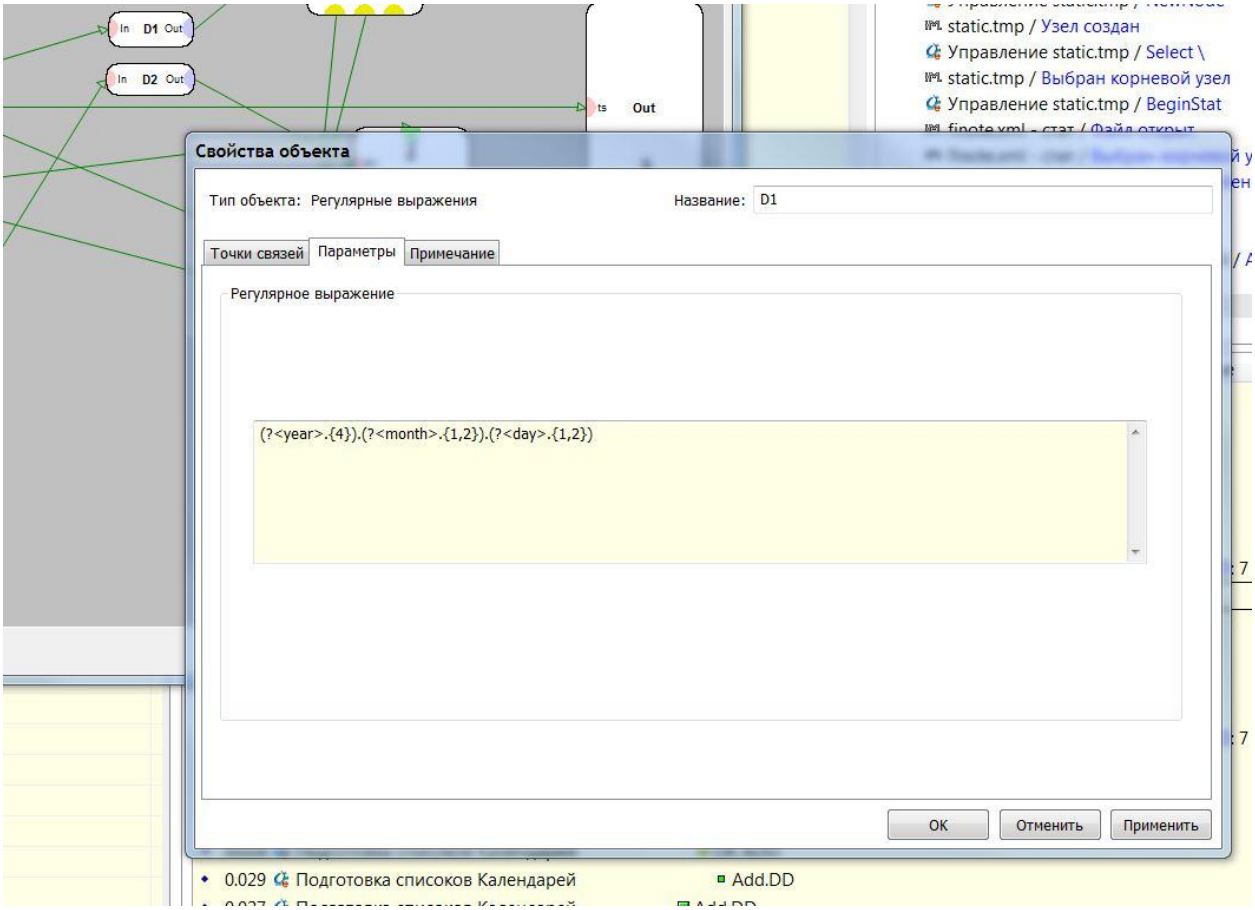


Рисунок 10 Режим работы LP-Studio, среда программирования. Режим программирования структуры нового компонента, настройка элемента компонента

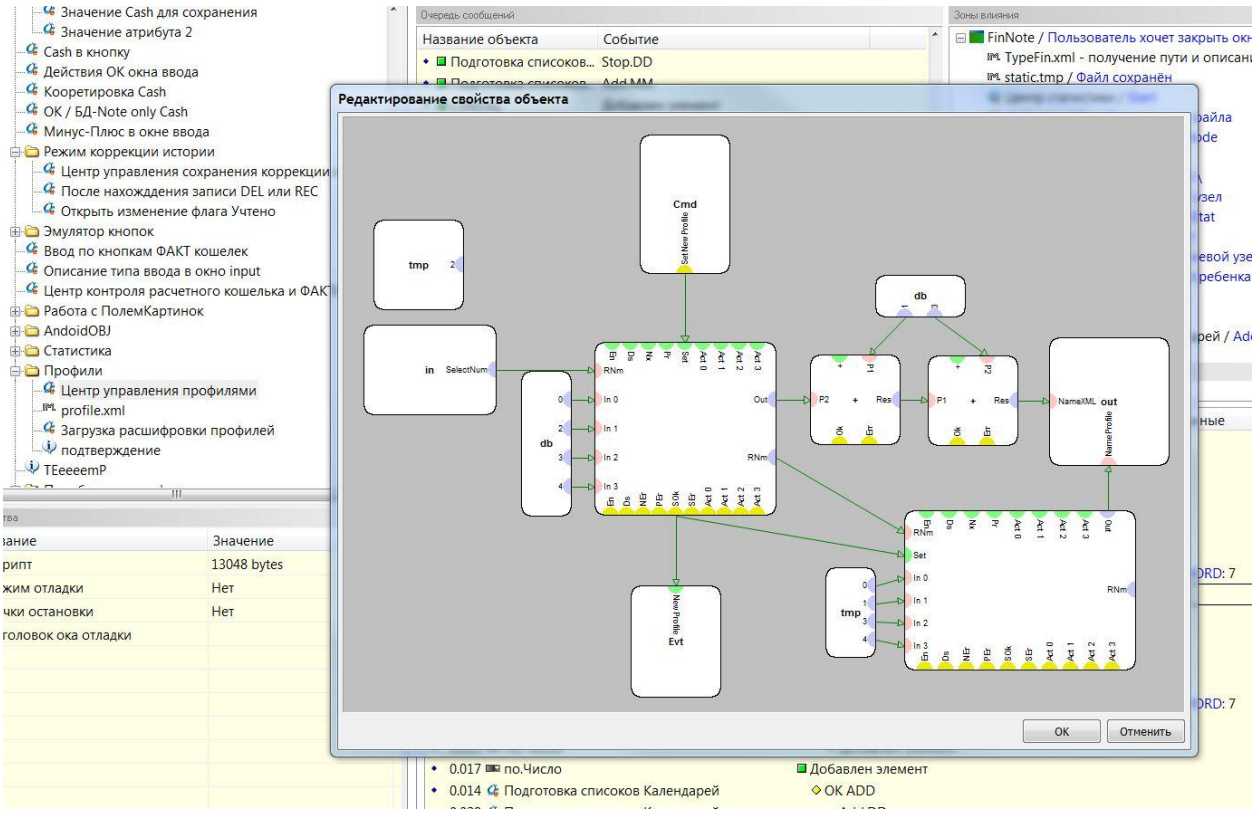


Рисунок 11 Режим работы LP-Studio, среда программирования. Среда программирования структуры нового компонента

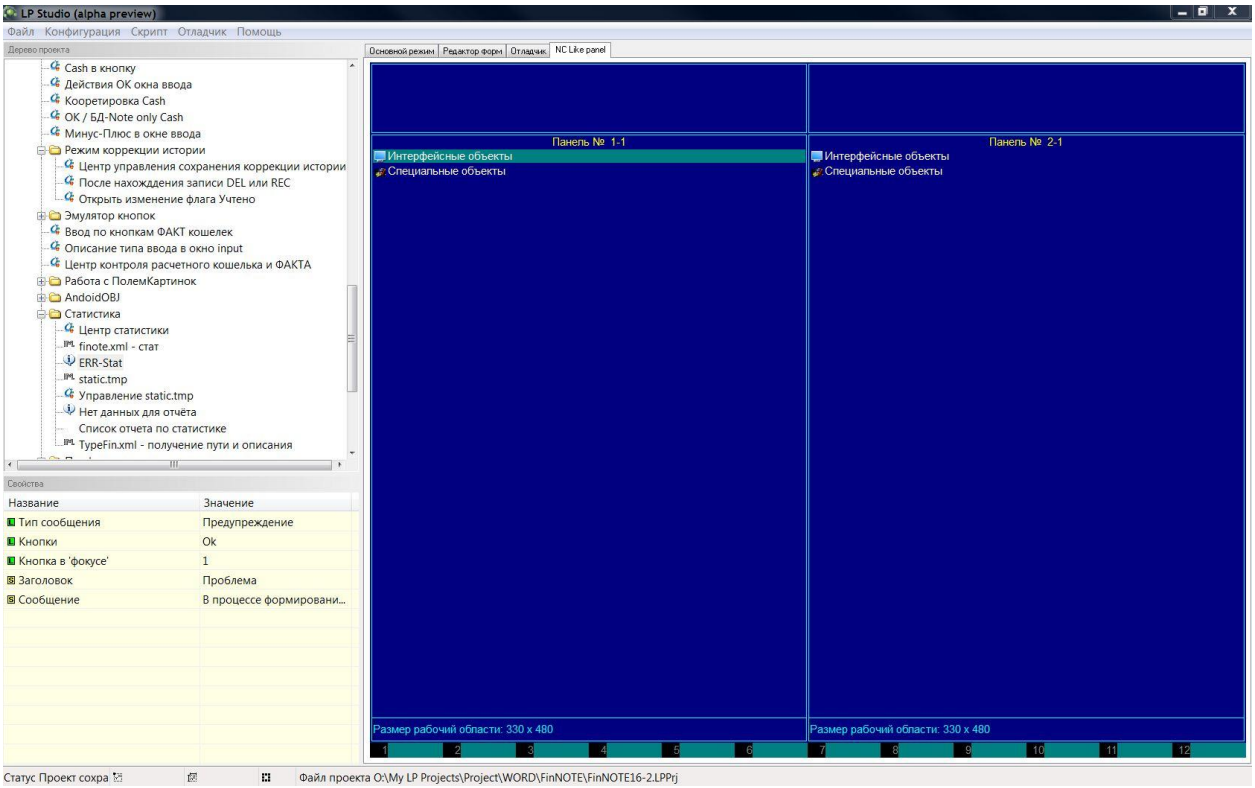


Рисунок 12 Режим работы LP-Studio, среда программирования. Режим «быстрого» программирования

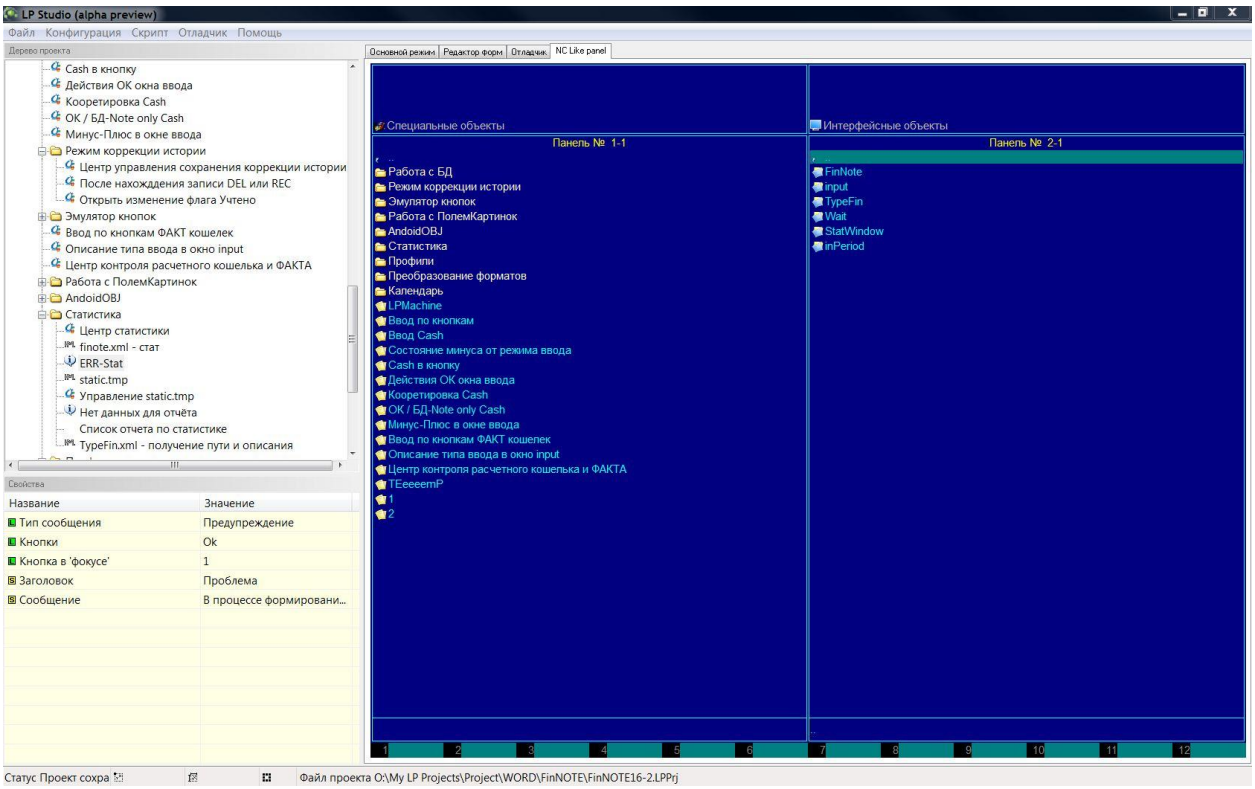


Рисунок 13 Режим работы LP-Studio, среда программирования. Режим «быстрого» программирования, отображение списка компонентов

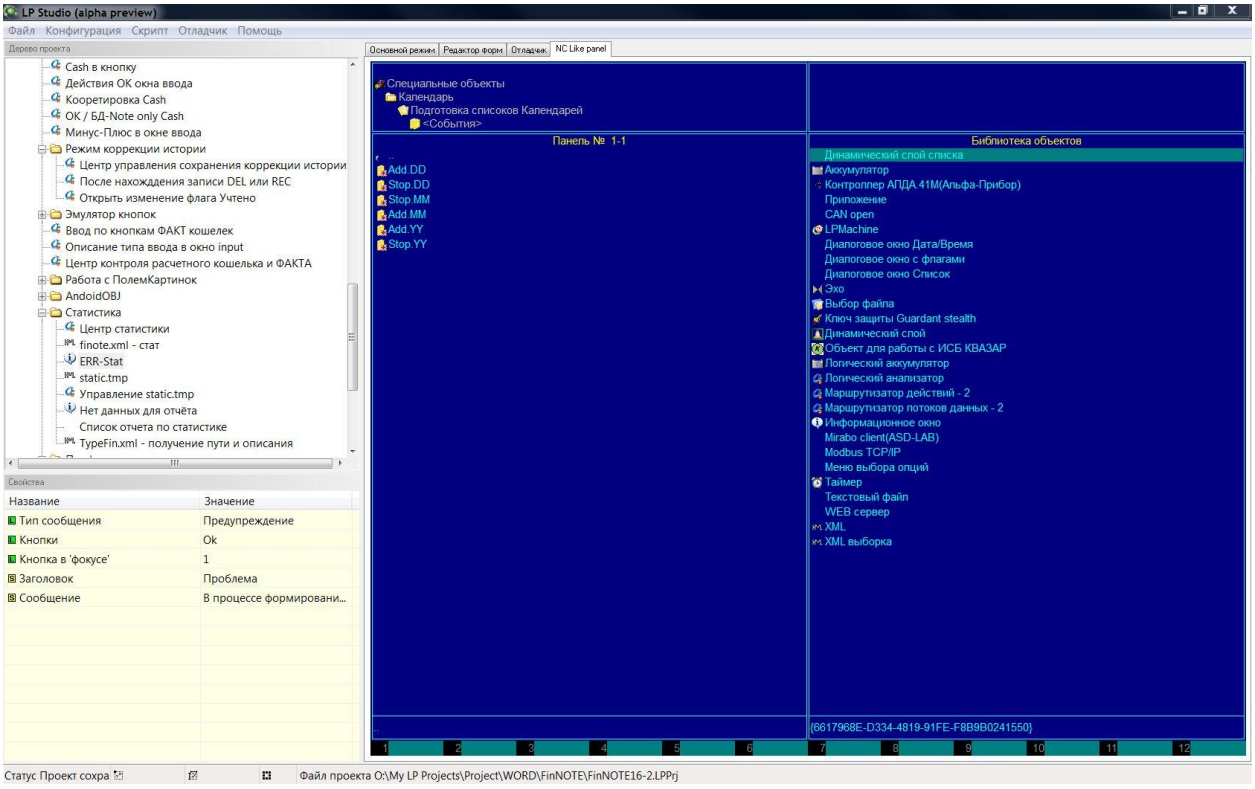


Рисунок 14 Режим работы LP-Studio, среда программирования. Режим «быстрого» программирования, отображение библиотеки компонентов

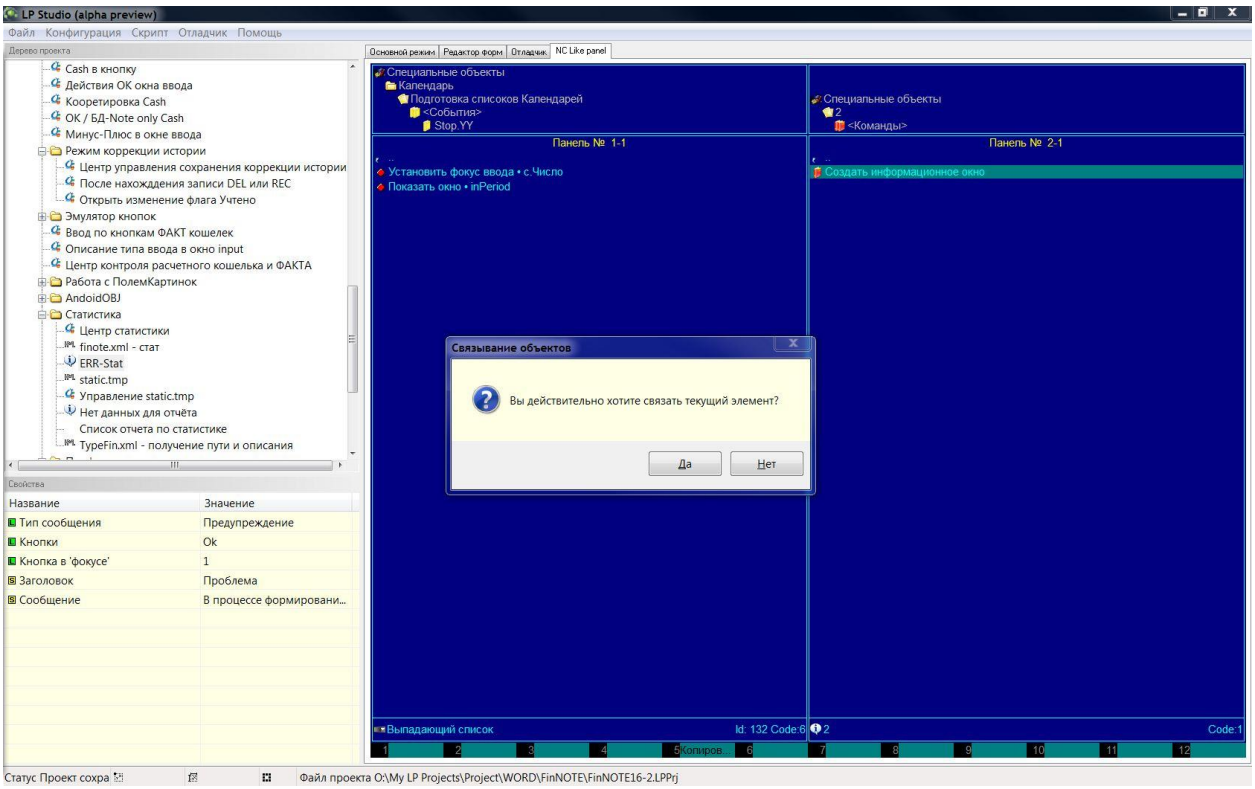


Рисунок 15 Режим работы LP-Studio, среда программирования. Режим «быстрого» программирования, режим создания связей между компонентами

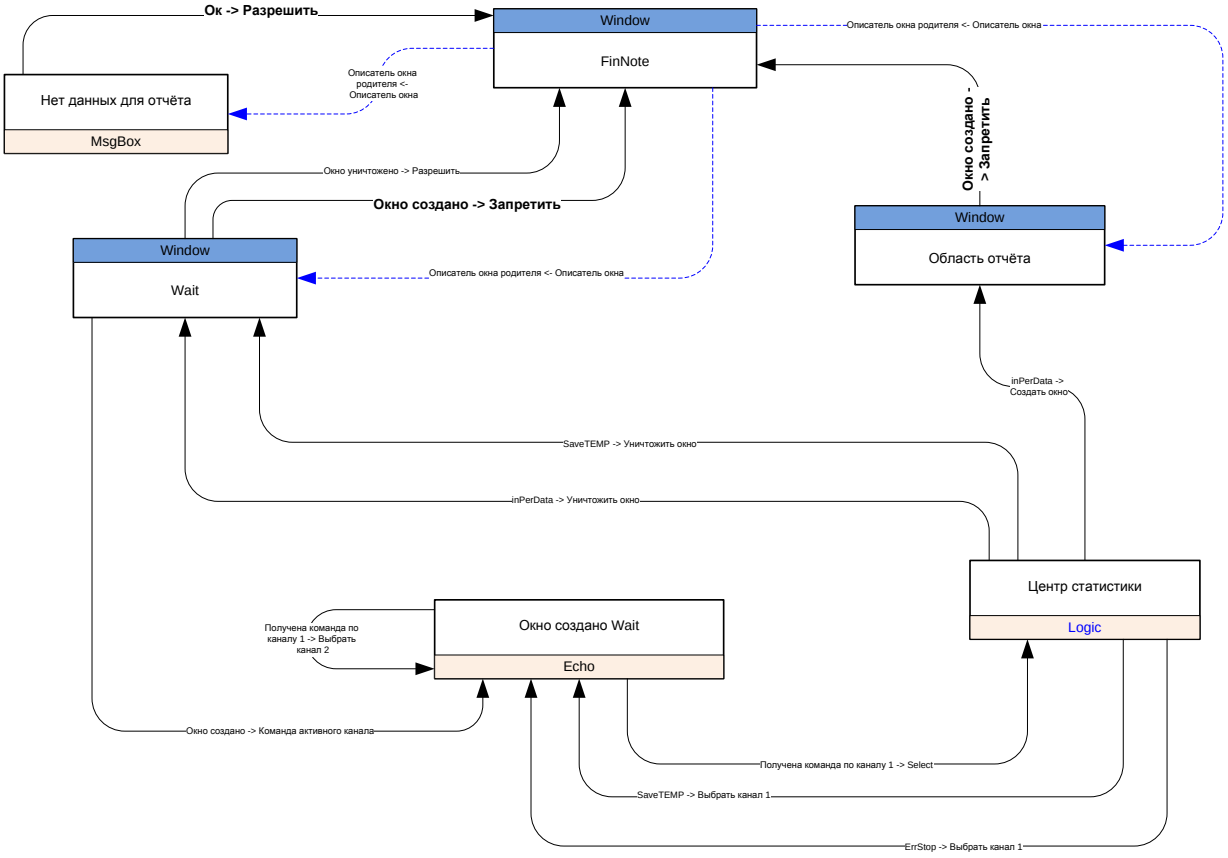


Рисунок 16 Пример отчёта схемы связей компонентами проектируемой программы

Пример интерфейсных решений LP-программ

Windows

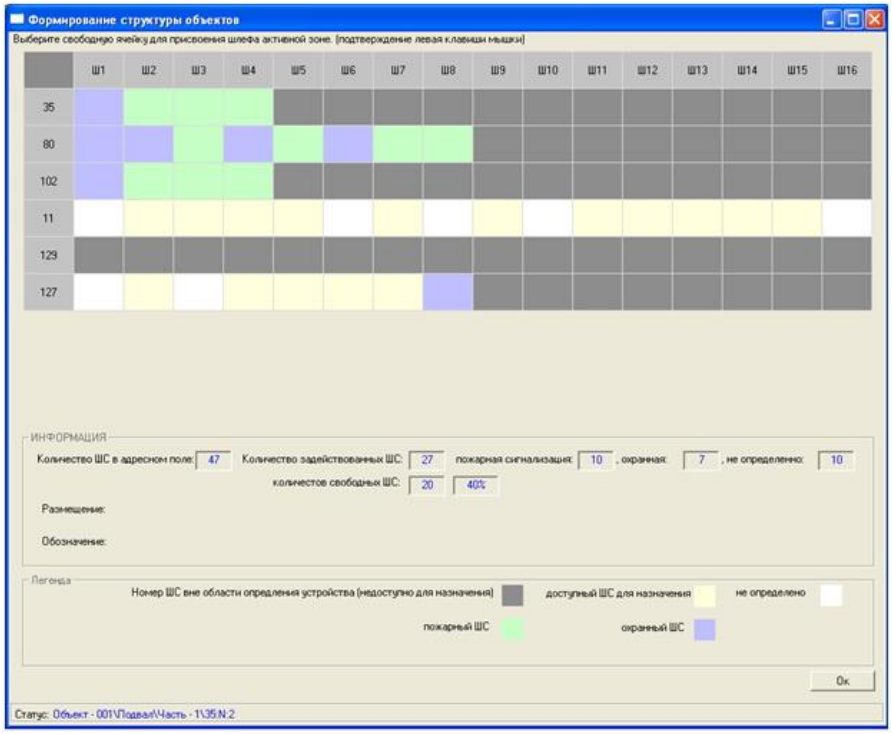


Рисунок 17 Пример работы программы. Представление таблицы

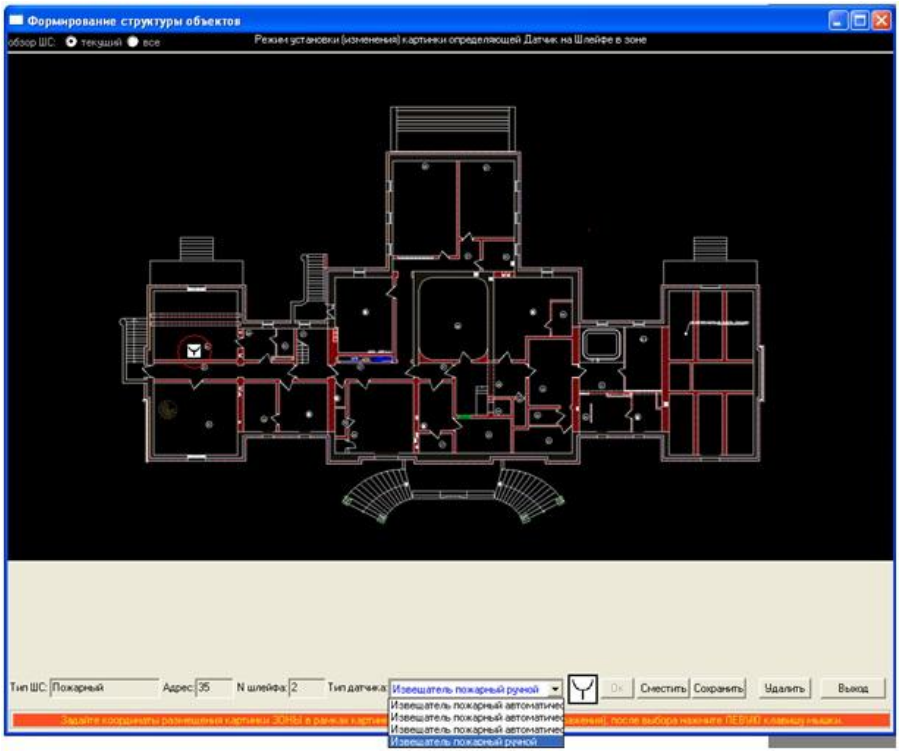
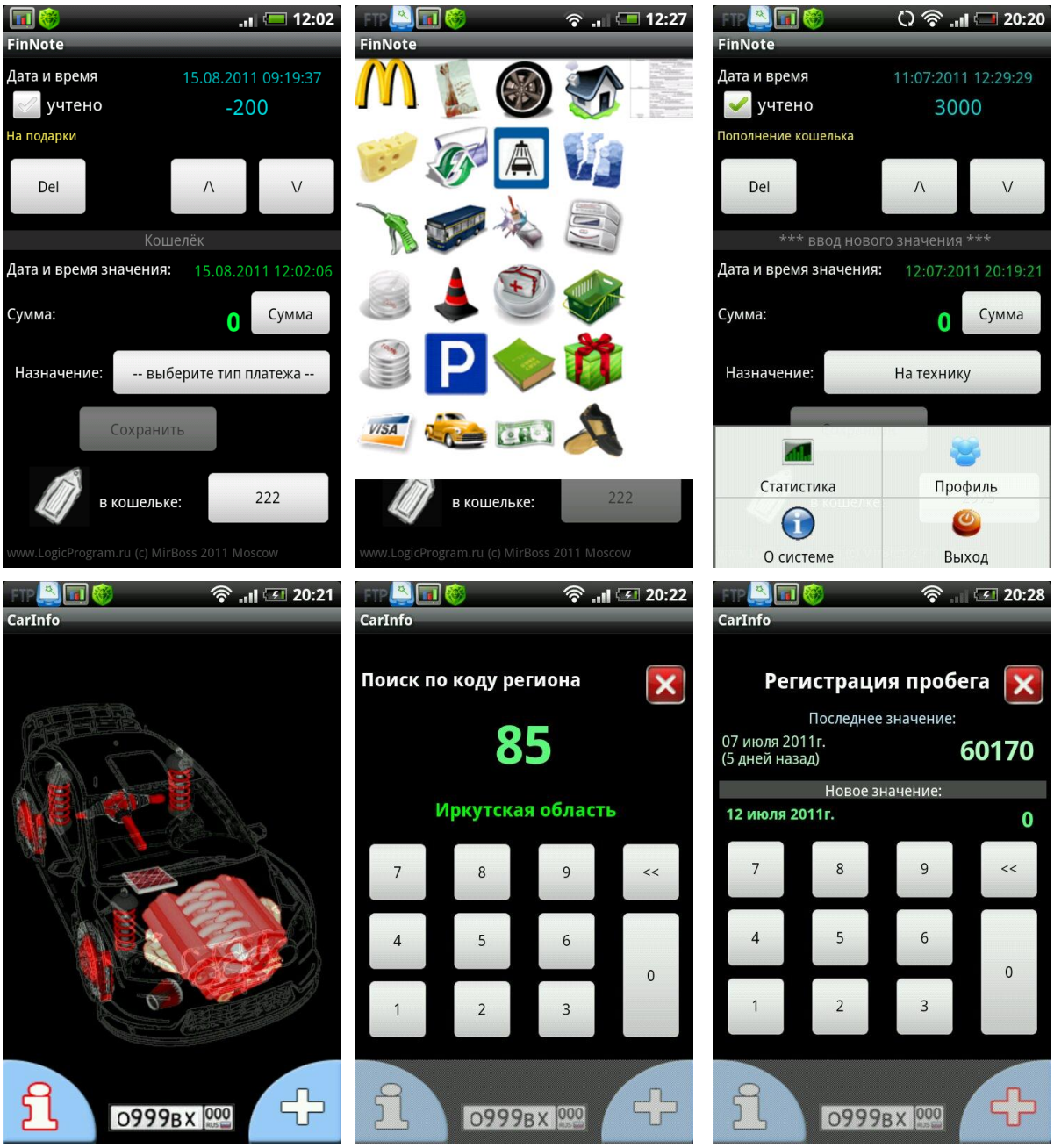


Рисунок 18 Пример работы LP-программы. Работа с изображением

Android



ЗАО НПФ «И.В.А.»

Телефон: +7 (499) 246-7446
Skype: mirboss_w
Почтовый адрес: 119034 г. Москва, ул.Пречистенка, д.40/2 строение 3, офис 212
Email: pr@logicprogram.ru
www.logicprogram.ru