Содержание

Введение

Во время разработки программного обеспечения не всегда необходима вся полнота функциональности, которая позволяет проектировать приложения высокой сложности и большой гибкости. Часто, достаточно упрощенных обобщающих средств разработки, предоставляющих готовые функции в простом виде. Это сильно упрощает и ускоряет процесс разработки.

Также, подобный подход является удачным для применения в образовательных процессах, при обучении программированию. Т.к. позволяет обучать учащихся с использованием заинтересованности, вызываемой быстрым получением результата поставленной задачи. Делает освоение материала очень наглядным.

На данный момент существует системы, позволяющих упростить процесс разработки специализированных приложений для конкретных задач. Например, пакет Microsoft Access предоставляет упрощенную модель организации приложения для работы с базой данных. Однако применение Access для решения более общих задач становится неудобным или вовсе невозможным. Например, типичных пользовательских действий в операционной системе, таких как копирование файлов, доступ в интернет, запуск приложений, выключения компьютера и т.д.

Разрабатываемая система должна предоставлять широкий набор инструментальных средств, обеспечивающих создание программ на основе сборки из готовых компонент. Компоненты должны в совокупности предоставлять возможность решения основных пользовательских задач в операционной системе

В соответствии с этим, целью дипломного проекта является разработка

среды структурно-визуального программирования обеспечивающей возможность решения основных пользовательских задач в операционной системе по средствам использования готовых компонент.

Исходя из поставленных целей, решаются задачи:

- Создание средств, позволяющих пользователю собирать программы, с помощью готовых компонент;
- Обеспечение интерфейса взаимодействия компонент со средой структурно-визуального программирования;
- Организация упрощенного проектирования на основе алгоритмических примитивов, таких как циклы, условия, массивы данных.

1. Анализ технического задания

Разработать набор инструментальных средств, обеспечивающих создание программ на основе сборки из готовых компонент.

1.1 Технологический процесс

Процесс работы среды структурно-визуального программирования включает в себя следующие действия:

- разработка пользователем графического интерфейса программы;
- объединение компонент по средствам структурно-визуальной генерации промежуточного кода;
- отладка программы, с использованием контрольных точек и пошагового выполнения;
- компиляция проекта в готовое приложение.

1.1.1 Функциональное назначение

Среда структурно-визуального программирования должна предоставлять следующие возможности:

- Создание программ на основе готовых компонент;
- Обучение принципам программирования на алгоритмических примитивах.

1.2 Требования к программному продукту

Среда структурно-визуального программирования должна:

работать с проектами любых размеров;

компилировать проекты в исполняемый файл;

иметь инструменты отладки программ;

предоставлять алгоритмические примитивы для связи компонент;

обеспечить интерфейс для создания новых компонент;

иметь собственный формат хранения проекта.

2. Существующие способы реализации системы структурно-визуального программирования

На сегодня, сформировались несколько схем, по которым может быть реализовано взаимодействие компонент, в среде структурно-визуального программирования. Каждый способ имеет свои достоинства и свои недостатки, влияющие, в конечном счете, на простоту работы в среде и набор предоставляемых пользователю функций.

2.1 Связь компонент с использованием графического объединения линиями

Примером программ с использованием графического объединения компонент линиями связи могут служить программы A-Flow (www.aflow-designer.com), HiAsm (www.hiasm.com). Пример последней среды приведен на Рис.1.

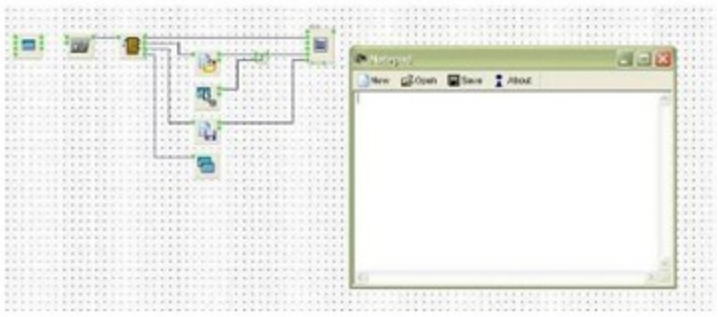

Рис.1. Пример создания блокнота в среде HiAsm

Построение алгоритма программы осуществляется путем соединения программных элементов линиями-связями. Каждая линия связывает событие, вызванное одним объектом с действием над другим компонентом. Например, проведя связь от выхода, обозначающего клик мышки по компоненту, до входа вызывающего показ формы, можно реализовать вызов окна программы.

Данная архитектура имеет следующие преимущества:

- Процесс создания программ наглядный;
- Логика программирования является интуитивно понятной;
- Расширение возможностей программы можно легко проводить увеличением компонент.

Однако данный подход порождает также некоторые проблемы:

- Создание большой программы сильно запутывает графическую схему;
- Сложные математические операции довольно трудно реализуются;
- На экране монитора помещается только простые программы. Более большие разработки могут иметь очень большие размеры;
- Из-за ограниченности размеров компонент при графическом объединении линиями связей, невозможно наделить компонент хорошей управляемостью.

2.2 Связь компонент структурно-визуальной генерации промежуточного кода

Суть этого подхода состоит в том, что компоненты связываются с

помощью специальных инструментов. Инструменты являются программными средствами, позволяющими легко настраивать нужное взаимодействие между компонентами. В результате получается некий блок кода на промежуточном языке, который и является логической частью разрабатываемой программы. Примерами подобных сред разработки являются DevelStudio (www.develstudio.ru), GameMaker (www.gamemaker. nl). Пример последней среды приведен на Рис.2.

Рис.2. Пример создания игры в среде GameMaker

Разработка происходит следующим образом. Для компонентов, путем простого выбора из списка, создаются обработчики событий (Events), таких как клик, нажатие клавиши, столкновение с другими объектами и т.д.

На каждое событие создаются действия (Actions) по средствам специального инструмента. Он предоставляет набор заготовленных шаблонных операций, таких как перемещение объекта, изменения его свойств и т.д. Эти действия настраиваются в процессе создания через предоставляемые инструментальные средства. Например, задается расстояние, на которое необходимо переместить объект.

Достоинства метода:

- Простота создания за счет проработанных инструментов;

- Б*о*льшая гибкость программной логики;
- Есть некий программный код, который можно редактировать вручную;
- Сложность программ менее ограничена.

Недостатки:

- Программа создается чуть менее наглядно;
- Сложная расширяемость логики из-за необходимости изменений инструментов программы.

2.3 Вывод

В связи с приведенными аргументами, был выбран подход, с использованием структурно-визуальной генерации кода. Этот подход позволяет создавать более сложные продукты, при этом процесс разработки остается простым для понимания.

Данных способ также очень приближен к логике программирования, что позволяет выработать необходимые навыки учащимися в образовательных процессах, при обучении программированию. Делает освоение материала очень наглядным.

Отличительными особенностями разрабатываемой системы структурно-визуального программирования являются использование собственных инструментальных средств генерации промежуточного кода, рассчитанных на широкий спектр задач, а не только на разработку, например, игр или баз данных. Логика разработки в среде приближена к логике программирования. Применение проработанных компонент предоставляющих большие возможности при небольшом наборе операций настройки и программирования. Все это делает разработку среды структурно-визуального программирования целесообразной и отличной от существующих систем разработки.

3. Общая модель системы и описание ее компонентов

Среда структурно-визуального программирования - это набор инструментальных средств, обеспечивающих создание программ на основе сборки из готовых компонент. Разрабатываемая система, должна включать в себя следующие необходимые для работы части: интерфейс пользователя, компоненты, инструментальные средства для связи компонент, отладчик, компилятор.

Обобщенная структура среды разработки приведена на Рис.1.

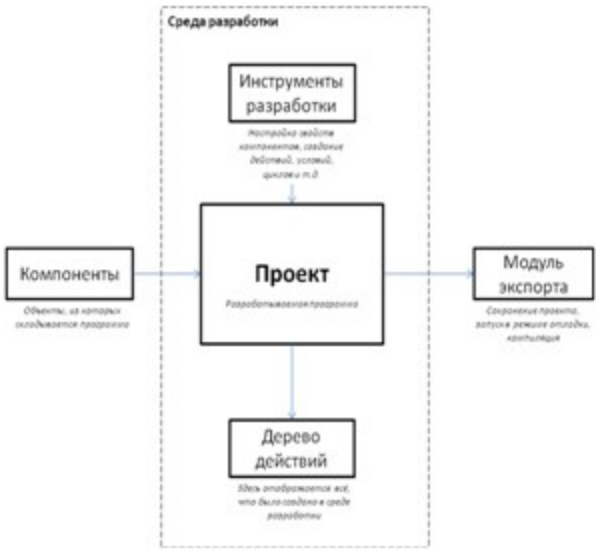

Рис.3. Обобщенная структура системы структурно-визуального программирования

Описание элементов системы:

- *компоненты*, структурные блоки, на основе которых строится программа;
- *инструменты разработки* - средства позволяющие связывать

компоненты в единую программу, посредствам алгоритмических примитивов;

- *дерево действий*, в котором хранится разрабатываемая пользователем программа;
- *проект*, представляющий собой объектное представление в среде программы, которую проектирует пользователь;
- *класс экспорта*, позволяющий сохранять проект в другие форматы представления данных, например в исполняемое приложение.

Рассмотрим подробнее все элементы системы структурно-визуального программирования.

3.1 Пользовательский интерфейс системы

Для реализации обобщенной структуры системы структурно-визуального программирования, был разработан пользовательский интерфейс, обеспечивающий доступ ко всем возможностям среды разработки (Рис.2).

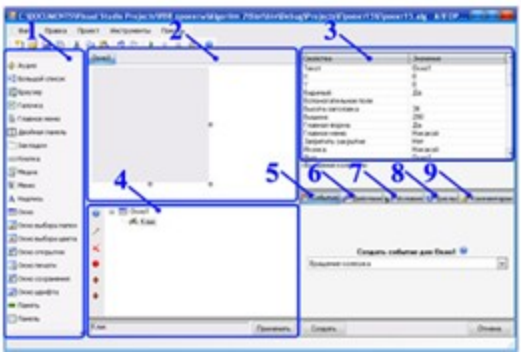

Рис.4. Главное окно среды разработки

Элементы главного окна среды разработки:

1. *Панель объектов* содержит перечень всех компонент, доступных для использования в разрабатываемом приложении;

2. В *разделе окон* разрабатывается графическое оформление создаваемой программы, путем расположения на окне программы компонент и задания их размеров и расположения;

3. *Раздел свойств* предоставляет инструмент просмотра и изменения свойств компонент в режиме проектирования;

. В *дереве действий* происходит отображение программного кода, созданного для обработчиков событий компонентов. Предоставляются инструменты для редактирования элементов дерева, а также отображения процесса отладки и трассировки программы.

5. *Раздел событий*. События позволяют компоненту уведомлять проект о возникновении каких-либо ситуаций или действий пользователя над компонентом (клик, движение курсора, нажатие клавиш). Раздел событий позволяет создавать компонентам обработчики событий, в которых будут размещаться программные выражения, описывающие реакцию программы на событие

6. *Раздел действий* позволяет создавать программные команды управления программой. Процесс создания происходит через специальный инструмент, где необходимо выбрать компонент, затем его свойство, а после значение, на которое на которое это свойство требуется изменить (Рис.3). Спроектированные действия размещаются в обработчике событий, созданном в разделе событий.

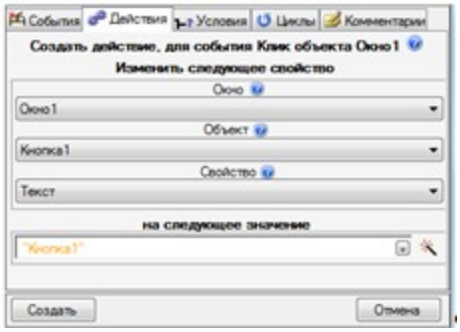

Рис.5. Инструмент создания программных команд

7. Раздел условий предназначен для создания условных конструкций, позволяющих организовать разветвленную логику в программе. Пример создания условия проверки, является ли ширина компонента Кнопка большим, чем 100 точек, приведен на Рис.4.

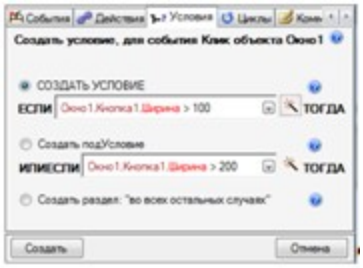

Рис.6. Инструмент создания условных конструкций

8. Раздел циклов предназначен для создания циклических конструкций, позволяющих организовать повторяющийся перебор одних и тех же действий. Пример создания цикла, выполняющегося пока текст компонента Кнопка меньше либо равен 10, приведен на Рис.4.

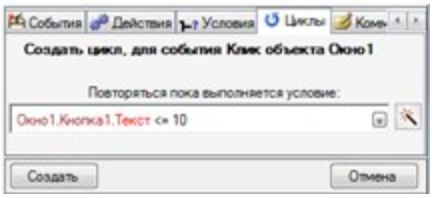

Рис.7. Инструмент создания циклических конструкций

9. Раздел комментариев используется для создания текстовых заметок в программном коде, предназначенных для пояснения участков кода, и не несущих функциональной значимости.

3.2 Разработка приложений в среде программирования

Процесс разработки приложений в среде структурно визуального программирования происходит следующим образом:

.