1. Министерство образования и науки РФ
2. Волгоградский государственный технический университет
3. Кафедра «Программное обеспечение автоматизированных систем»
4. Семестровая работа
5. по курсу «Технология программирования»
6. по теме «Редактор диаграмм IDEF0 - Diagram Editor V2.0»
7. Листов 44

|  |  |  |
| --- | --- | --- |
|  |  | Научный руководитель:  доц. каф. ПОАС\_\_\_\_\_\_Жукова И.Г.  ас.каф. ПОАС\_\_\_\_\_\_\_\_Ужва А.Ю.  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2012 |
| Нормоконтролёр:  доц. каф. ПОАС\_\_\_\_\_Жукова И.Г.  ас. каф. ПОАС\_\_\_\_\_\_\_Ужва А.Ю.  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2012 |  | Исполнители:  студенты группы ИВТ-461  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Етеревский Г.П.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Климов Н.В.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Щербаков П.Д.  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2012 |

1. 2012

Данный документ представляет собой документацию к программному продукту под названием «Редактор диаграмм стандарта IDEF0 – «Diagram Editor V2.0».

В первой части документа приводится описание целей и задач, поставленных перед разработчиками, описывает область деятельности и объект автоматизации, требования и ограничения к проекту, уточняются детали, которых не будет реализовывать программа. Чтобы иметь представление о результате, который должен быть получен программой, имеется диаграмма-образец, на которую ровняются программисты при описании поведения объектов рабочей области. Программистами приводится макеты интерфейса программного продукта, производится документация выбора программной технологии.

Во второй части приводятся проектные решения, достигнутые программистами. Доработанный интерфейс программы. UseCase диаграмма, диаграмма классов на концептуальном и физическом уровне. Описывается сценарий работы программы, и тестирование, производимое по этому сценарию, а так же недочёты, допущенные программистами в реализации программного кода, выявленные в ходе тестирования, которые в дальнейшем были полностью исправлены. Так же приводится фрагмент кода программы, с целью показать качество написания продукта.

В завершении говорится о внедрении программы.

Всего листов в документе 44, рисунков 18, таблиц 8, приложений 6.

Содержание

1 Цели создания проекта, основные задачи 4

2 Функциональные требования 4

3 Нефункциональные требования 5

4 Ограничения проекта 5

5 Проектные решения 6

6 Тестирование 8

7 Общий сценарий работы программы 11

8 Недочёты, найденные в ходе отладки программы и их исправления 14

9 Внедрение 16

10 Интерфейс разработанной программы 16 11 Структура сохраняемого файла 18

12 Спецификация классов 19

Приложение А. Руководство системного администратора 21

Приложение Б. Руководство пользователя 22

Приложение В. Фрагмент кода программы 26

Приложение Г. Лист исправлений по 1 релизу 29

Приложение Д. Паспорта классов и методов 29

Приложение Е. Спецификация кода программы 29

1 Цель создания проекта, основные задачи

1.1 Цели разработки:

Сократить время и повысить удобство разработки диаграмм IDEF0 за счет удобного интерфейса и соответствующего набора инструментов. Результатом разработки является графический редактор.

1.2 Задачи

1. Доработка программного продукта в соответствии с требованиями первого релиза, включая необходимые функции, нефункциональные требования, требования к инсталлятору.
2. Реализация функций предоставленных заказчиком на второй релиз.
3. Тестирование программного продукта.
4. Документирование программного продукта.
5. Внедрение программного продукта.

2 Функциональные требования

1. Главная функция:

* Построение IDEF0-диаграммы.

1. Основные функции на второй релиз:

* Реализовать “Укороченные стрелки” к блокам.
* Реализовать отрисовку и заполнение IDEF рамки диаграммы (в соответствии со стандартом) с заполнением всей информации в рамке.
* Реализовать автоматическое выкладывание картинки диаграммы на сервер (в любой файлообменник) и предоставление ссылки пользователю.
* Реализовать множественное выделение элементов, дублирование элементов при перетаскивании с зажатой клавишей Ctrl.
* Добавить копирование, вырезание и вставку выделенных элементов, включая копирование-вставку между несколькими экземплярами запущенной программы.

1. Вспомогательные функции:

* Масштабирование рабочей области программы

3 Нефункциональные требования

1. Программа должна поддерживать русский язык.
2. Должен быть реализован установщик, позволяющий запуск программы на «чистой» ОС.
3. Редактор должен работать под управлением ОС Windows XP и выше.
4. Интерфейс программы должен состоять из тулбара, панели инструментов и рабочей области. На тулбаре должны находятся контекстные меню «Файл», «Редактирование», «Помощь». Меню «Файл»: создать диаграмму, открыть, сохранить, сохранить как, экспорт в формат .png, экспорт на файлообменник. Меню «Редактирование»: перечень всех режимов работы програмы. Меню «Помощь»: справка, информация о программе. Панель инструментов состоять из режимов работы программы: добавление блока, добавление связи, добавление укороченной стрелки, удаление, перемещение объектов, редактирования информации о рамке, а так же масштабирование. Построение диаграммы происходит на рабочей области, которая содержит IDEF рамку.
5. Изображение диаграммы должно загружаться на сервис «Яндекс Фотки». Чтобы отправить изображение на сервер, необходимо выбрать в меню «Файл» подменю «Экспорт на YA». После выполнения операции пользователю будет выведено сообщение с ссылкой на файл. Для управления базой изображений предоставляется профиль с логином bornpavel, и паролем 350353752.

4 Ограничения проекта

* + - 1. Стандарт IDEF0-диаграммы, чтобы придерживаться стандарта
      2. Выданное задание на второй релиз по предмету «Технология программирования», чтобы придерживаться пожеланиям заказчика
      3. Рекомендованное количество стрелок входящих в одну грань – 8, для упрощения визуального восприятия

5 Проектные решения

5.1 Диаграмма прецедентов

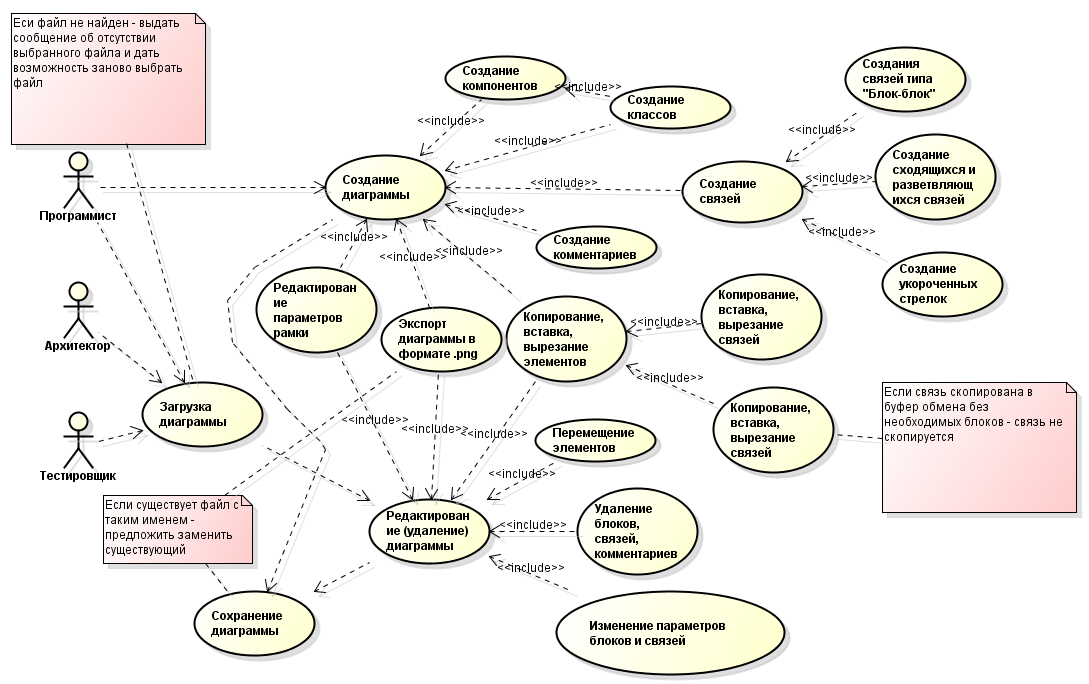


Рисунок 1. Диаграмма прецедентов для программы Diagram Editor

5.2 Диаграмма классов на концептуальном уровне

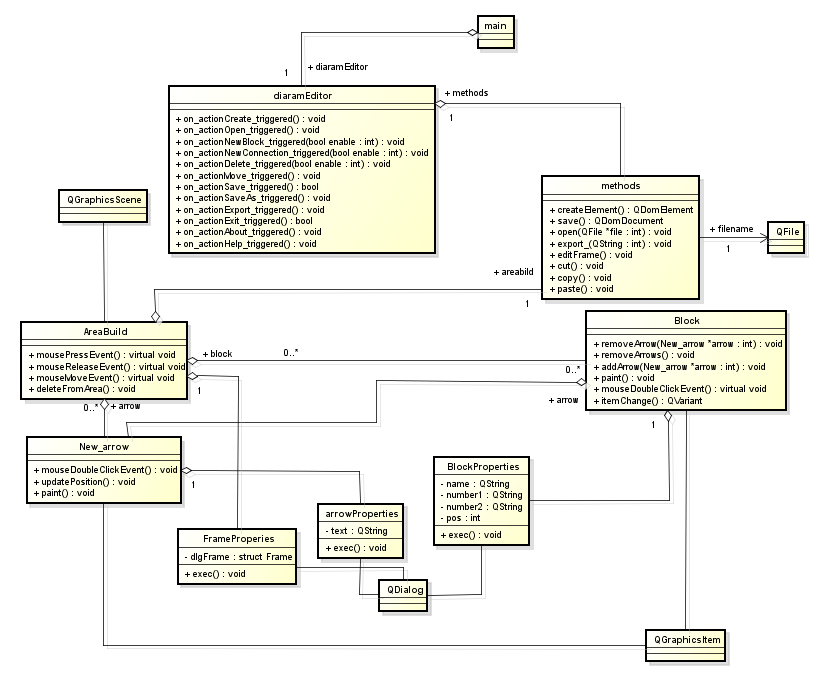


Рисунок 2. Диаграмма классов на концептуальном уровне для программы Diagram Editor

5.3 Диаграмма классов на физическом уровне

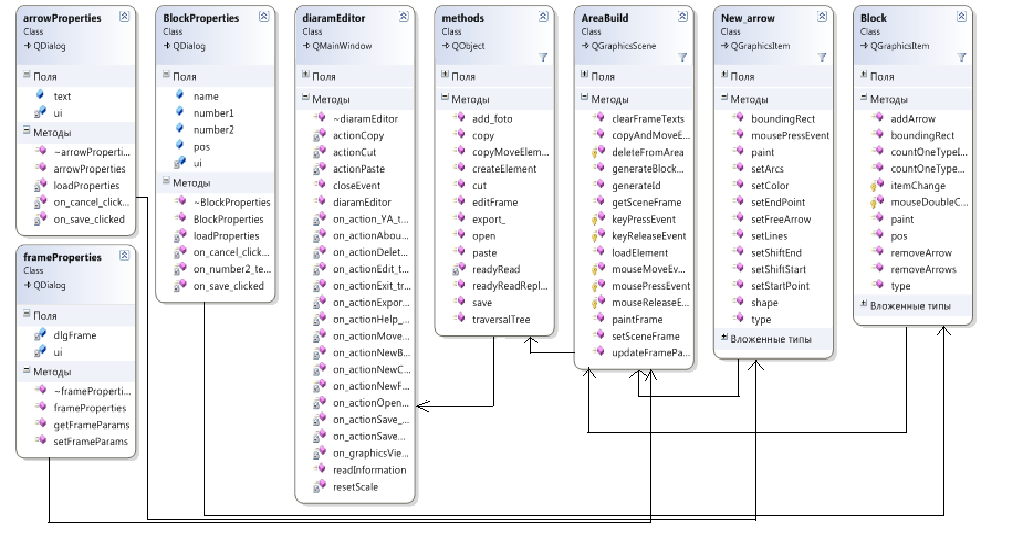


Рисунок 3. Диаграмма классов на физическом уровне для программы Diagram Editor

6 Тестирование

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Краткое Описание | Ожидаемое поведение программы | Вход | Выход | Статус |
| 1 | Перемещение функционального блока за рабочую область | Блок и связи, соединённые с блоками, остаются в пределах рабочей области | Выбрать блок со связями, перетащить его за рабочую область | Блок и связи возвращается в рабочую область, стрелки не выходят на IDEF рамку | Работает |
| 2 | Тестирование укороченных стрелок | Укороченная стрелка добавляется, не выходит за грани блока, корректно загружается при открытии диаграммы, корректно копируется | Построить блок, выбрать режим построения укороченных стрелок, кликнуть на грань где хотим построить укороченную стрелку | Стрелки строятся, не выходят за пределы грани, при копировании и открытии диаграммы остаются на своих позициях | Работает |
| 3 | Отрисовка и заполнение IDEF рамки | После создания новой диаграммы открывается окно, в котором можно заполнить поля рамки. Нажав «ОК» , сверху и снизу рабочей области отображается рамка с введённым в окно параметрами, которые можно редактировать по нажатию кнопки в тулбаре | Создать диаграмму, заполнить необходимые поля, нажать «ОК» | На рабочей области отображается IDEF рамка с введёнными значениями, которые можно редактировать по нажатию кнопки в тулбаре | Работает |
| 4 | Выкладывание картинки на файлообменник | После выбора действия «Экспорт на YA» делается скриншот рабочей области и отправляется на «Яндекс Фотки», после чего возвращается ссылка на загруженное изображение | Выбрать действие «Экспорт на YA» | Отправка изображения на «Яндекс фотки», возвращение ссылки на файл | Работает |
| 5 | Множественное выделение объектов | После выделения объекта резинкой или левой кнопкой мыши блок окрашивается в синий цвет, на связях появляются индикаторы, выбранные элементы можно перемещать | Клик левой кнопкой мыши или растягивание резинки | Выделенные объекты можно перемещать по рабочей области группой, блоки окрашены в синий цвет, на связях имеются индикаторы | Работает |
| 6 | Дублирование элементов с зажатой клавишей CTRL | Выделенные объекты дублируются | Выделение объектов, нажатие клавиши CTRL, перетаскивание мышкой | Образовались копии выделенных объектов | Работает |
| 7 | Копирование и вставка элементов в одной программе | Выделенный объект копируется в буфер и вставится на рабочую область | Выделить объект, клик левой кнопкой мыши, выбор действия копировать, затем вставить | Выделенный элемент скопирован, а затем вставлен в новую позицию на рабочей области | Работает |
| 8 | Вырезание и вставка элементов в одной программе | Выделенный объект копируется в буфер, при этом удаляется с рабочей области и вставляется в новую позицию | Выделить объект, клик левой кнопкой мыши, выбор действия вырезать, затем вставить | Выделенный элемент удалён с рабочей области и вставлен в новой позиции | Работает |
| 9 | Копирование, вырезание и вставка между несколькими экземплярами программы | Выделенные объекты копируются в буфер (в случае вырезания удаляется с рабочей области), при открытии нового экземпляра программы возможно провести операцию вставки этих данных | Выделить элементы, выбрать действие, открыть новый экземпляр программы, вставить данные | Данные скопированы (вырезаны) из одной программы и вставлены в другой экземпляр программы | Работает |

Таблица 1. Тестирование программы Diagram Editor

7 Общий сценарий работы программы

* + - 1. При запуске программы сцена и панель инструментов недоступны.

1. Выбор в меню пункта «Создать диаграмму» делает доступными пустую сцену и панель инструментов.
2. По выбору пункта меню “Открыть...” программа выдаст сообщение с предложением сохранить диаграмму в новый файл(если имя не задано) или в уже имеющийся. Иначе программа вызывает диалог для выбора имени файла, после чего загружает выбранную диаграмму в редактор и делает доступными сцену и панель инструментов. Если файла не существует или его структура не соответствует виду открываемого файла, программа выдаст соответствующее предупреждение и вернется в исходное состояние
3. По выбору пункта меню “Сохранить” программа либо сохраняет диаграмму в старый файл(если диаграмма уже редактировалась), либо вызывает диалог выбора имени файла сохранения(если диаграмма новая). По выбору пункта меню «Сохранить как» вызывается диалог для выбора имени сохраняемого файла. Сохранение производится в файл формата \*xml.
4. По выбору пункта меня «Экспорт в .png» вызывается диалоговое окно, предлагающее задать имя файла, и по нажатию на кнопку «Сохранить» создает изображение области диаграммы в формате .png.
5. При создании новой диаграммы пользователю предлагается ввести параметры рамки. В самой программе предусмотрено добавление\перемещение\удаление ICOM-блока, добавление\изменение\удаление аннотации к блоку, добавление\удаление связи между пустой точкой и блоком, пустой точкой и связью (вариант сходящейся стрелки),  блоком и блоком, блоком и пустой точкой, блоком и связью(вариант сходящейся стрелки),  связью и пустой точкой\блоком\связью(вариант расходящейся стрелки). Предусмотрено создание\редактирование\удаление аннотации к стрелке.
   * + 1. Для создания блока необходимо выбрать на панели инструментов “Блок” и кликнуть левой клавишей мыши на область сцены.
6. Для создания связи необходимо выбрать на панели инструментов “Связь” и соединить, не отпуская левую кнопку мыши, два нужных объекта.
7. Для создания укороченной стрелки необходимо выбрать инструмент «Укороченная стрелка» и кликнуть ЛКМ на блоке в одну из областей, в зависимости от необходимого типа укороченной стрелки
8. Для добавления\удаления\редактирования нотаций к блоку или к связи двойным щелчком левой кнопкой мыши вызывается диалоговое окно для редактирования нотаций.
9. При добавлении нотации к блоку длиной более 30 символов в строку и более 4 строк, нотация выводится в виде следующего текста «текст с длиной строки в 30 символов и с количеством строк более 4…»; при наведении на блок с такой нотацией всплывает подсказка с полным текстом. Можно также вводить номер блока в иерархии. Блок не будет добавлен на сцену, пока поля «Описание» и «Порядковый номер» не будут заполнены
10. При добавлении нотации к связи длиной более 10 символов в начале стрелки выводится текст вида «Начало текста…»; при наведении на связь с такой нотацией всплывает подсказка с полным текстом.
11. При добавлении нотации к разветвляющейся связи нотация добавляется также ко всем связям, имеющим с данной одно начало
12. Для перемещения блока необходимо навести мышь на нужный блок и перетащить в нужную область, удерживая левую кнопку мыши.
13. При перемещении блока за пределы сцены он возвращается на прежнее положение
14. Для удаления элементов  необходимо выбрать на панели инструментов “Удалить элемент” и кликнуть на удаляемый элемент.
15. При удалении блока удаляются также все ассоциированные с ним связи
16. Выделить объекты на сцене можно 2 способами: заключив в резиновую рамку или по клику ЛКМ. С выделенными элементами (в том числе и с множеством выделенных элементов) можно производить следующие действия:
17. дублирование с зажатой клавишей Ctrl;
18. копирование, вырезание и вставка (в том числе и между двумя экземплярами программы);

Если при копировании связь выделена без необходимых ей блоков – копирование связи не происходит

1. Перед выходом программа, если была изменена текущая диаграмма, выдаст сообщение с предложением сохранить диаграмму – в новый файл (если имя не задано) или в уже имеющийся.

8 Недочёты, найденные в ходе отладки программы и их исправления

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Краткая формулировка | Что ожидалось | Как была найдена | Скриншот | Решение проблемы |
| 1 | Выход стрелки за рабочую область и на область рамки | Блоки и стрелки останутся в пределах рабочей области | На рабочей области создаём два блока, соединяем связями вход-выход, перетаскиваем блоки к краю рабочей области | C:\Users\Леся\Pictures\Безымянный4.png | Улучшен алгоритм контроля выхода элементов за пределы рабочей области |
| 2 | Дублирование выделенных элементов при перетаскивании с зажатой клавишей CTRL | Два блока скопируются связанными между собой связью | Два блока, соединённых связью. Выделяем оба, зажимаем CTRL, перетаскиваем в новую область – блоки скопированы, связь нет | C:\Users\Леся\Pictures\Снимок5.PNG | Улучшен алгоритм копирования |
| 3 | Укороченные стрелки при большом количестве выходят за грань блока | Укороченные стрелки остаются в пределах грани блока | Добалено несколько укороченных стрелкок в боковую грань | C:\Users\Леся\Pictures\EvBOO4_xv30.jpg | Установка контроля количества стрелок для каждой грани |
| 4 | Сохранение нецелых координат блока и стрелки, некорректная обработка при загрузке | Загрузка элементов с теми координатами, с которыми они были сохранены | После загрузки диаграммы элементы находились не на своих местах | - | Использование в программе только целых координат |
| 5 | При маленьком масштабе блок можно поставить за рабочую область | Размещение блока за сценой невозможно | При минимальном масштабе был перенесён блок за рабочую область | C:\Users\Леся\Pictures\Безымянный3.jpg | Установлен контроль координат добавляемого блока, увеличен минимальный масштаб |
| 6 | При копировании стрелки только с одним из ее блоков без второго программа вылетает при вставке | Стрелка без связанных с ней блоков копироваться не должна | Два блока соединены связью, выделяем один блок и стрелку, копируем и затем вставляем | C:\Users\Леся\Pictures\ыа.PNG | Установлен контроль на наличие всех необходимых для стрелки блоков |

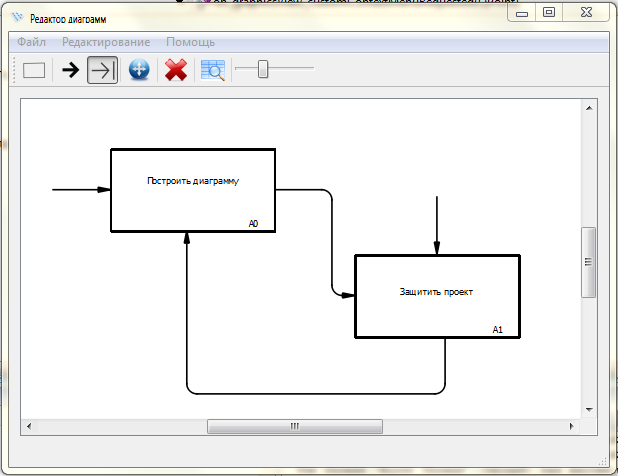
Таблица 2. Ведомость дефектов

Найденная тестировщиком ошибка с подробным описанием отсылалась разработчику, занимающимся написанием функции, в ходе выполнения которой была обнаружена ошибка. После устранения ошибки программа отправлялась на повторное тестирование.

9 Внедрение

1. Для реализации программного продукта предусмотрен инсталлятор, копирующий накомпьютер все необходимые для работы программы компоненты.
2. Подготовка презентации на prezi.com.

10 Интерфейс разработанной программы

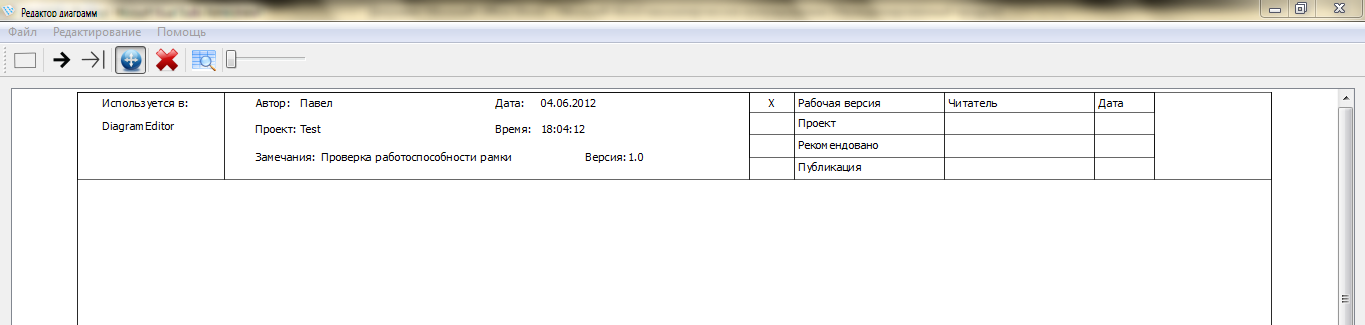
Интерфейс спроектирован согласно нефункциональным требованиям, соответствующие элементы отображены и подписаны на скриншотах: 

Тулбар

Панель инструментов

Рабочая область

Рисунок 4. Интерфейс главного окна программы



Рамка IDEF0

Рабочая область

Рисунок 5. Часть экранной формы главного окна с рамкой IDEF

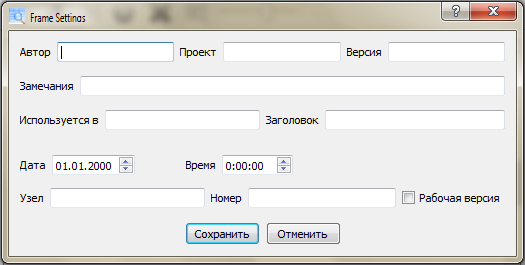
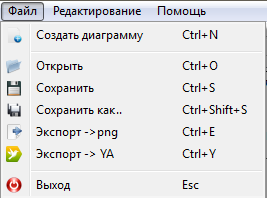
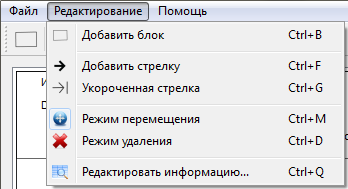


Рисунок 6. Форма для заполнения полей рамки IDEF

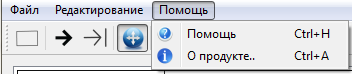


Рисунок 7. Экранные формы меню «Файл», «Редактирование» и «Помощь»

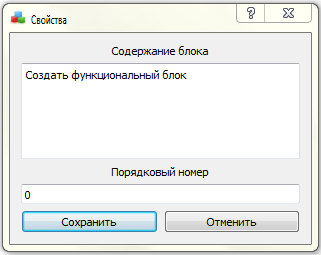
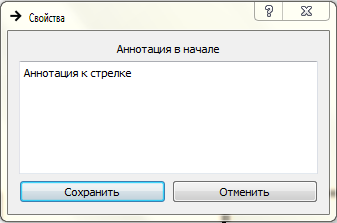
 

Рисунок 8. Экранные формы окон, вызываемых при редактировании блока и добавлении аннотации к стрелке

11 Структура сохраняемого файла

Входные данные: имеющиеся на сцене блоки, связи, укороченные стрелки и формуляр

Выходные данные: XML-документ со следующей структурой:

<!DOCTYPE diagram>

<diagram>

<Block>

<id> </id>

<name> </name>

<number> </number>

<PosX> </PosX>

<PosY> </PosY>

<needless> </needless>

</Block>

<Block>… </Block>

<Arrow>

<startItem> </startItem>

<endItem> </endItem>

<startPosX> </startPosX>

<startPosY> </startPosY>

<endPosX> </endPosX>

<endPosY> </endPosY>

<inputType> </inputType>

<outputType> </outputType>

<annotation> </annotation>

<annotAtStart> </annotAtStart>

<isFlying> </isFlying>

</Arrow>

<Arrow>…</Arrow>

<Frame>

<author> </author>

<project> </project>

<date> </date>

<time> </time>

<version> </version>

<comments> </comments>

<usingAt> </usingAt>

<node> </node>

<name> </name>

<number> </number>

<isWorking> </isWorking>

</Frame>

</diagram>

12 Спецификация классов

|  |  |
| --- | --- |
| Название класса: | Опсиание: |
| arrowProperties | Диалоговое окно  о программе, используется для редактирования нотации к связи |
| blockProperties | Диалоговое окно  о программе, используется для редактирования нотации и номера ICOM-блока |
| frameProperties | Диалоговое окно  о программе, используется для редактирования полей формуляра |
| diagrammEditor | Главный класс программы. Содержит главное окно и задает поведение всех инструментов |
| methods | Вспомогательный класс. Содержит методы, вызываемые из класса diagramEditor для работы с файлам и файлообменником |
| areabuild | Сцена программы. Содержит параметры рамки, текущие выбранные элементы, идентификаторы элементов |
| block | Класс ICOM-блока. Содержит id, нотацию, номер и контейнер стрелок, ассоциированных с блоком |
| new\_arrow | Класс связи. Содержит ассоциированные стрелкой блоки, тип связи. |
| main | Класс для инициализации главного окна программы |

Таблица 3. Спецификация классов в программе Diagram Editor

Приложение А. Руководство системного администратора

1) Общие сведения о программе

Программа предназначена для построения IDEF0-диаграмм. Программа представляет собой графический редактор для создания диаграмм по стандарту IDEF0

Для успешной работы программы необходим компьютер минимальной

конфигурации:

- процессор Intel Pentium 4 2,5 ГГц;

- 512 Mb Ram;

- операционная система Windows XP.

Помимо этого на компьютере должны быть установлены необходимые библиотеки C Run-Time.

2) Настройка программы

Для установки программы нужно запустить инсталлятор и проследовать его инструкциям. Инсталлятор помимо самой программы устанавливает пакет Visual C++ Redistributable Package. Если пакет не установился, то необходимо сделать это вручную. Для этого необходимо скачать Visual C++ Redistributable Package 2008 с сайта http://www.microsoft.com и установить его на компьютер, после чего программа будет готова к работе.

3) Проверка программы

Для проверки работоспособности программы можно открыть один из имеющихся в ней примеров и начать с ним работать

4) Дополнительные возможности

В качестве дополнительных возможностей возможно перемещение тулбара и изменение масштаба текущей диаграммы

5) Сообщения системному администратору

При выполнении работы программа выдает сообщения только пользователю в момент конфликтов имен сохраняемых и загружаемых файлов

Приложение Б. Руководство пользователя

1) При запуске программы сцена и панель инструментов недоступны. Чтобы она активизировалась, нужно создать или открыть диаграмму, выбрав нужный пункт меню «Файл»:

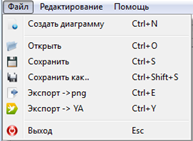


Рисунок 9. Меню «Файл»

1. Для того, чтобы открыть существующую диаграмму, нужно в открывшемся окне диалога выбрать имя файла диаграммы:

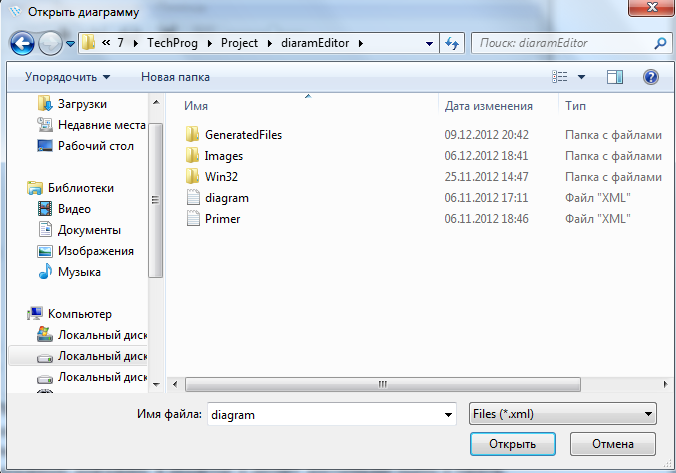


Рисунок 10. Диалог выбора файла для открытия диаграммы

1. Чтобы создать новый блок, необходимо выбрать нужный инструмент и нажать на месте сцены, куда необходимо поставить новый блок. При этом создается диалоговое окно для ввода нотации и номера блока(номер генерируется, исходя из текущего максимального номера блока на сцене, возможно его изменение; кнопка «Сохранить» в диалоге недоступна до тех пор, пока не введена нотация):

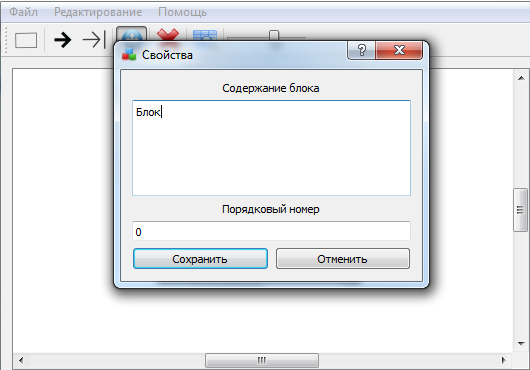


Рисунок 11. Редактирование свойств блока

1. Чтобы создать новую связь, необходимо выбрать нужный инструмент, нажать ЛКМ на блоке или стрелке, от которой необходимо провести связь, подвести мышь к блоку или стрелке, к которой необходимо провести связь, и отпустить ЛКМ:

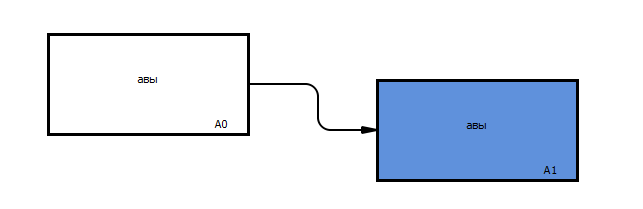


Рисунок 12. Два блока соединены связью

1. Чтобы создать новую укороченную стрелку, необходимо выбрать нужный инструмент, нажать ЛКМ на блоке на область, в зависимости от которой построится свой тип укороченной стрелки: вход, управление, механизм, выход:

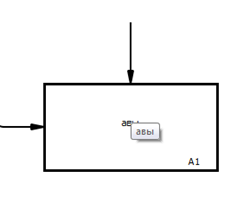


Рисунок 13. Добавление укороченной стрелки

1. Чтобы удалить элемент, необходимо выбрать нужный инструмент и щелкнуть ЛКМ на удаляемом элементе; при этом щелчок ЛКМ на корне разветвляющихся стрелок или на конце сходящихся удалит все разветвляющиеся\сходящиеся стрелки одного типа из этого блока
2. Чтобы выделить элементы, необходимо зажать ЛКМ в свободном месте сцены и заключить нужные объекты в рамку. Причем выделенные элементы меняются: блок меняет цвет, связь выделяется маркерами:

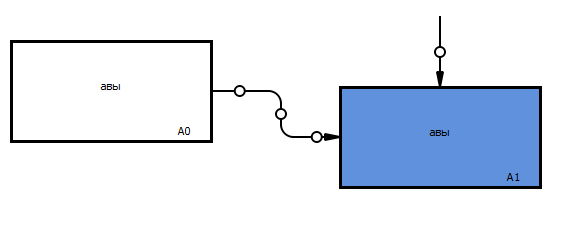


Рисунок 14. Выделение элементов

1. Чтобы переместить выделенные элементы, нужно в режиме перемещения зажать ЛКМ и перетащить нужные элементы
2. Чтобы изменить масштаб сцены, необходимо передвигать слайдер в панели инструментов до достижения нужного масштаба
3. Чтобы редактировать параметры рамки диаграммы, необходимо выбрать нужный инструмент; при этом появится диалоговое окно для редактирования всех параметров рамки
4. Чтобы скопировать или вырезать элемент или группу элементов, необходимо их выделить и нажать ПКМ на одном из выделенных блоков; при этом появляется контекстное меню, из которого нужно выбрать нужное действие:

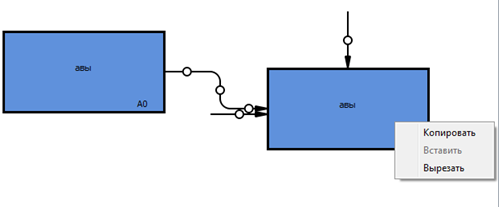


Рисунок 15. Копирование, вырезание, вставка объектов

1. Чтобы вставить элементы из буфера обмена, нужно щелкнуть ПКМ на области сцены и выбрать необходимый пункт из контекстного меню. Если в буфере обмена ничего нет, пункт «Вставить» будет недоступен.
2. Чтобы изменить нотацию связи или параметры блока, необходимо дважды щелкнуть ЛКМ по редактируемому элементу и ввести нужные значения
3. Чтобы сохранить открытый документ, необходимо выбрать в меню «Файл» пункт «Сохранить»
4. Чтобы сохранить новый документ, необходимо выбрать в меню меню «Файл» пункт «Сохранить» или «Сохранить как..». При этом появится диалог для выбора или ввода имени сохраняемого файла:

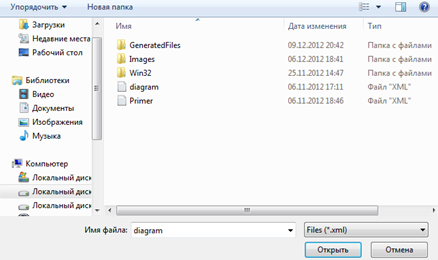


Рисунок 16. Сохранение диаграммы

1. Чтобы экспортировать диаграмму в формате .png, необходимо выбрать нужный инструмент в меню «Файл»
2. Чтобы экспортировать диаграмму на файлообменник, необходимо выбрать нужный инструмент в меню «Файл»

Приложение В. Фрагмент кода программы

Класс block, файл block.cpp

Block::Block(int num)

{

needless=false;

id=0;

number2.setNum(num);

BlockProperties properties;

properties.name=name;

properties.number2=number2;

if(properties.exec()==1)

{

name="";

//number2="";

name=properties.name;

number2=properties.number2;

}

else

needless=true;

setCursor(Qt::OpenHandCursor);

//setAcceptedMouseButtons(Qt::LeftButton);

setFlag(QGraphicsItem::ItemIsMovable, true);

setFlag(QGraphicsItem::ItemIsSelectable, true);

setFlag(QGraphicsItem::ItemSendsGeometryChanges, true);

}

//Конструктор при загрузке

Block::Block(int id\_,QString &name\_,QString &number2\_,int posX\_,int posY\_,bool needless\_)

{

id=id\_;

name=name\_;

number2=number2\_;

posX=posX\_;

posY=posY\_;

needless=needless\_;

setCursor(Qt::OpenHandCursor);

//setAcceptedMouseButtons(Qt::LeftButton);

setFlag(QGraphicsItem::ItemIsMovable, true);

setFlag(QGraphicsItem::ItemIsSelectable, true);

setFlag(QGraphicsItem::ItemSendsGeometryChanges, true);

}

QRectF Block::boundingRect() const

{

return QRectF(0,0,200, 100);

}

void Block::paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget)

{

Q\_UNUSED(option);

Q\_UNUSED(widget);

setToolTip(name);

if(!isSelected())

painter->setBrush(QBrush(Qt::white,Qt::SolidPattern));

else

{

QColor color;

color.setNamedColor("#5F91DC");

painter->setBrush(QBrush(color,Qt::SolidPattern));

}

QRectF rect(0,0,200,100);

painter->drawRect(rect);

painter->setPen(QPen(Qt::black,3,Qt::SolidLine,Qt::SquareCap,Qt::RoundJoin));

painter->drawRect(0,0,200,100);

if(name.size()>120)

{

QString tmp=name;

tmp.replace(120,tmp.size()-120,"...");

painter->drawText(QRectF(10,10,180,70),Qt::AlignCenter,tmp);

}

else

painter->drawText(QRectF(10,10,180,70),Qt::AlignCenter,name);

painter->drawText(QRectF(145,85,58,10),Qt::AlignCenter,"A"+number2);

}

//Вызов диалога для изменения параметров

void Block::mouseDoubleClickEvent(QGraphicsSceneMouseEvent \*mouseEvent)

{

BlockProperties properties;

properties.name=name;

properties.number2=number2;

properties.exec();

name=properties.name;

number2=properties.number2;

}

//Удаление стрелки из внутреннего контейнера блока

void Block::removeArrow(New\_arrow \*arrow)

{

int index = arrows.indexOf(arrow);

if (index != -1)

arrows.removeAt(index);

}

//Удаление всех ассоциированных стрелок

void Block::removeArrows()

{

foreach (New\_arrow \*arrow, arrows) {

if(arrow->outputType!=New\_arrow::freeOut)

arrow->getStartItem()->removeArrow(arrow);

if(arrow->inputType!=New\_arrow::freeIn && arrow->getStartItem()!=arrow->getEndItem())

arrow->getEndItem()->removeArrow(arrow);

scene()->removeItem(arrow);

delete arrow;

}

}

//Добавление новой стрелки

void Block::addArrow(New\_arrow \*arrow)

{

arrows.append(arrow);

}

//Обновление стрелок при перемещении блока

QVariant Block::itemChange(GraphicsItemChange change, const QVariant &value)

{

if (change == QGraphicsItem::ItemPositionChange || AreaBuild::posChanged == true)

{

foreach (New\_arrow \*arrow, arrows) {

arrow->updatePosition(arrow->isSelected());

}

diaramEditor::isChange=true;

}

return value;

}

//Текущая позиция блока

QPoint Block::pos()

{

QPoint point;

point.setX(QGraphicsItem::pos().x()+100);

point.setY(QGraphicsItem::pos().y()+50);

return point;

}

//Возвращает позицию для новой стрелки

int Block::countOneTypeInArrows(DWORD type)

{

int j=0;

QVector <int> tmp; //матрица пустых позиций, доступных для помещения новой стрелки

//j=arrows.length();

foreach (New\_arrow \*arrow, arrows) {

tmp.insert(j,0);

foreach (New\_arrow \*arrow, arrows) {

if (arrow->inputType==type)

{

//если позиция уже занята - отмечаем это

if(arrow->getShiftEnd()==j){

tmp.insert(j,1);

}

}

}

//если после проверки всего массива позиция осталась незанятой - возвращаем ее

if(tmp[j]==0)

return j;

j++;

}

//все позиции заняты - занимаем новую

return arrows.length();

}

//Возвращает позицию для новой стрелки

int Block::countOneTypeOutArrows(DWORD type)

{

int j=0;

QVector <int> tmp; //матрица пустых позиций, доступных для помещения новой стрелки

foreach (New\_arrow \*arrow, arrows) {

tmp.insert(j,0);

foreach (New\_arrow \*arrow, arrows) {

if (arrow->outputType==type)

{

//если позиция уже занята - отмечаем это

if(arrow->getShiftStart()==j){

tmp.insert(j,1);

}

}

}

//если после проверки всего массива позиция осталась незанятой - возвращаем ее

if(tmp[j]==0)

return j;

j++;

}

//все позиции заняты - занимаем новую

return arrows.length();

}

Block::~Block()

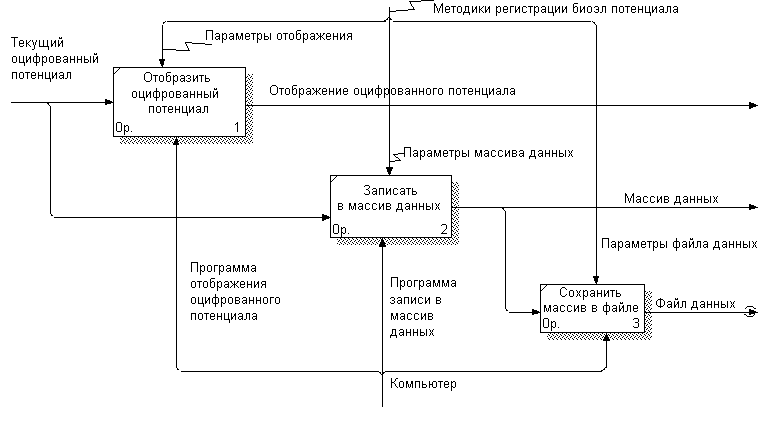
{

}

Приложение Г. Лист исправлений по первому релизу

|  |  |  |
| --- | --- | --- |
| № | Комментарий | Исправление |
| 1 | Документация должна быть оформлена по ЕСПД + исправить титульный лист | Исправления: В тексте не присутствует выделение жирным шрифтом, после номера пункта не ставится точка. Титульный лист см. дополнение 1 |
| 2 | Исправить п.1.1 – «за счёт..»; формулировки «интуитивно понятный интерфейс, типовые ошибки» | Сократить время и повысить удобство разработки диаграмм IDEF0 за счет удобного интерфейса и соответствующего набора инструментов. Результатом разработки является графический редактор. |
| 3 | П.2.2 неверно - исправить | Объект проектирования |
| 4 | П ? Объект проектирования? Состав? Структура? | Объектом проектирования является IDEF0-диаграмма. Построение производится в соответствии заданию, выданному заказчиком. Основные элементы, которые должны быть реализованы в проекте:  функциональные блоки и переходы между ними. Переходы (стрелки) должны быть проаннотированы; должны быть реализованы разветвляющиеся и объединяющиеся стрелки. |
| 5 | В п. 3 исправить функциональную структуру программы, используя понятие FF=(D,G,H) | * Реализовать нижний и боковой скроллы чтобы просматривать всю область отображения диаграммы * Реализовать диалог, отслеживающий выход из программы с проверкой наличия изменений в текущей диаграмме * Наличие справочной информации. * Реализовать экспорт диаграммы в формат .png * Объекты диаграммы не должны выходить за поля области отображения. |
| 6 | В П.4 везде должны быть формулировки «должна быть» | * Программа должна быть русскоязычной; * Программа должна стабильно работать, не вызывая каких-либо ошибок, приводящих к нарушению функционирования или завершению программы; * Должна быть предусмотрена реакция на перетаскивание отдельных блоков, включающая перестроение ассоциированных с этим элементом связей. * Для выделения связей должна быть предусмотрена активная область не менее 7px шириной, а не только область самой стрелки. * Должен быть реализован установщик, позволяющий запуск программы на «чистой» ОС. * Редактор должен работать под управлением ОС Windows XP и выше |
| 7 | Ограничение … могут быть представлены в виде (D,G,H)= FF. Ограничения должны быть обоснованы, т.е. отвечать на вопрос «почему?» | * Стандарт IDEF0-диаграммы, чтобы придерживаться стандарта * Выданное задание по предмету «Технология программирования», чтобы придерживаться пожеланиям заказчика * Ограничения на программный продукт… для упрощения визуального восприятия |
| 8 | Рисунки должны быть подписаны | Все рисунки и таблицы в тексте пронумерованы и подписаны |
| 9 | На диаграмме, которая не может быть построена, какие элементы не могут быть построены? Показать на диаграмме | См. Рис.17  1 – Выносная аннотация  2 – Стрелки, помещённые в «туннель» |
| 10 | Макет интерфейса построен в соответствии с нашими требованиями, где они описаны? | Интерфейс программы должен состоять из тулбара, панели инструментов и рабочей области. На тулбаре должны находятся контекстные меню «Файл», «Редактирование», «Помощь». Меню «Файл»: создать диаграмму, открыть, сохранить, сохранить как, экспорт в формат .png. Меню «Редактирование»: перечень всех режимов работы програмы. Меню «Помощь»: справка, информация о программе. Панель инструментов состоять из режимов работы программы: добавление блока, добавление связи, добавление укороченной стрелки, удаление, перемещение объектов, редактирования информации о рамке, а так же масштабирование. |
| 11 | Раздел 8.2 Добавить объектную модель, добавить физическую диаграмму классов + спецификацию классов в виде тастп???? | Физическая диаграмма классов см. рис 18 |
| 12 | Раздел 9 Отсутствуют «ожидания поведения программы | Добавить колонку «ожидание поведения программы» |
| 13 | П 11Таблица ведомости дефектов. До вставить полный отчёт 1-2 шт. | Таблица состоит из полей: №, Краткая формулировка, Что ожидалось, Как была найдена, Скриншот, Решение проблемы |
| 14 | П 13 Разработанный интерфейс программы соответсвует заявленным действиям? | Добавить диаграмму на область, интерфейс соответствует требованиям (см. пункт 10) |
| 15 | Комментарии. Оформленный код. Спецификация классов. | Учтено на второй релиз |

Таблица 4. Лист исправлений по первому релизу



2

1

Рисунок 17. Пример диаграммы, которую не будет строить программа

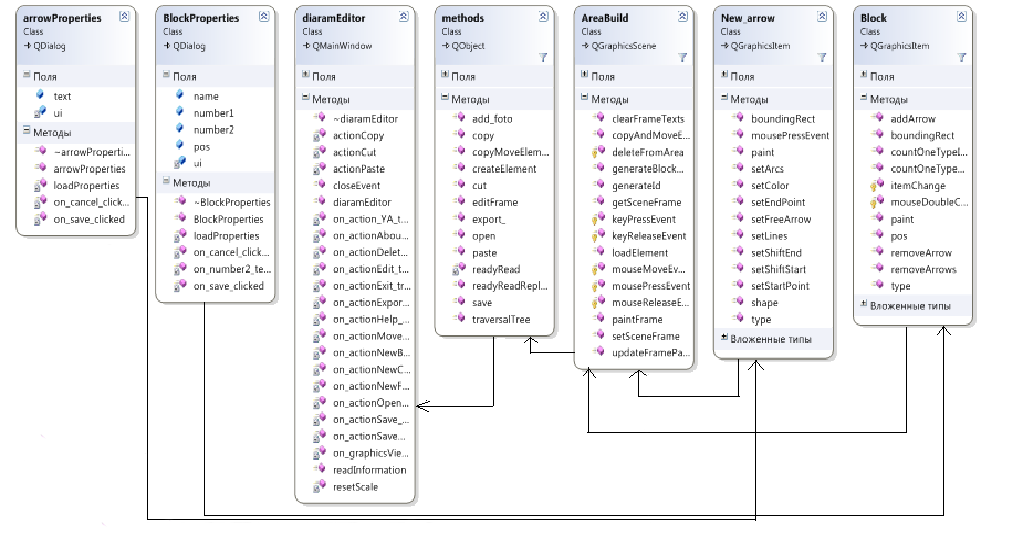


Рисунок 18. Физическая диаграмма классов

Приложение Д. Паспорта классов и методов  
  
1.Щербаков П. Класс arrowproperties

|  |  |  |
| --- | --- | --- |
| **№** | **Пункт карточки** | **Описание** |
| 1 | Имя класса | arrowproperties |
| 2 | Назначение | Диалог для создания\изменения\удаления нотации к связи |
| 3 | Язык | C++ |
| 4 | Суперкласс | QDialog |
| 5 | Подкласс | Отсутствует |
| 6 | Взаимодействующие классы + тип взаимодействия | New\_arrow(однонаправленная ассоциация) |
| 7 | Свойства + уровень доступа + библиотечный/ разрабатываемый | //нотация к связи QString text - public (разрабатываемый) //окно диалога Ui::arrowProperties ui - private (библиотечный) |
| 8 | Методы + уровень доступа + библиотечный/ разрабатываемый | 1. void on\_text\_textChanged() : private (разрабатываемый)  2. void on\_save\_clicked() : private (разрабатываемый)  3. void loadProperties() : private (разрабатываемый) |
| 9 | Описание методов(паспорт метода) | void on\_text\_textChanged() : private (разрабатываемый)    Прототип: Отсутствует    Входные данные:  Текст нотации    Выходные данные: Валидный текст нотации    Назначение: Контроль размеров, занимаемых текстом    Алгоритмы: при изменении текста проверить, стала ли текущая строка больше 9 символов; если да - автоматически поставить перевод строки |
| 10 | Тесты | 1.  Описание: После двойного клика по стрелке, в появившемся окне ввести подряд 15 символов без пробела, 15 символов с пробелом и другими разделителями. Ожидаемое поведение программы: Введёные символы поделены на строки, каждая их которых содержит не более 9 символов.  2. Описание: Напечатать пару строк нотации, копировать часть текста, вставить в конец первой строки. Ожидаемое поведение программы: Исходный текст, претерпев изменения, остаётся валидным. |
| 11 | Литература | Макс Шлее - Qt 4.2. Профессиональное программирование на C++ (2007) |

2 Климов Н. Класс areabuild

|  |  |  |
| --- | --- | --- |
| **№** | **Пункт карточки** | **Описание** |
| 1 | Имя класса | areabuild |
| 2 | Назначение | Класс для работы со сценой и событиями сцены |
| 3 | Язык | C++ |
| 4 | Суперкласс | QGraphicsScene |
| 5 | Подкласс | Отсутствует |
| 6 | Взаимодействующие классы + тип взаимодействия | New\_arrow(агрегация) Block(агрегация) methods(ассоциация) frameProperties(зависимость) |
| 7 | Свойства + уровень доступа + библиотечный/ разрабатываемый | //Начальная точка при построении связи - разрабатываемое   QPointF startPoint; - private //Нажата ли клавиша клавиатуры - разрабатываемое   bool keyPressed; - public //Выставляется в случае, когда объект выбран - разрабатываемое   bool selectItem;  - public                   //Выставляется в случае, когда объект изменил свое положение - разрабатываемое   bool static posChanged; - public //Предыдущие координаты(исп.для возврата элемента)  - разрабатываемое    QPointF lastPos; - public  //Выбранный блок  - разрабатываемое    Block \*currentItem; - public //Выбранная связь  - разрабатываемое   New\_arrow \*currentArrow; - public //Временная линия при построении связи - разрабатываемое   QGraphicsLineItem \*line; - private //Структура формуляра - разрабатываемое    Frame sceneFrame; - private  //Контейнер с текстами формуляра - разрабатываемое    QList <QGraphicsTextItem\*> frameTexts; - private |
| 8 | Методы + уровень доступа + библиотечный/ разрабатываемый | public: //Добавляет ICOM-блок на сцену - разрабатываемый void loadElement(Block \*); //Конструктор сцены - измененный библиотечный AreaBuild(QObject \*parent); //Проверка на наличие у блока входящих\выходящих стрелок - разрабатываемый bool blockHasArrowIn(Block \*,int); bool blockHasArrowOut(Block \*,int); //Генерация идентификатора ICOM-блока - разрабатываемый  int generateId();  //Генерация порядкового номера ICOM-блока - разрабатываемый  int generateBlockNumber(); //Отрисовка макета формуляра - разрабатываемый  void paintFrame(); //Отрисовка полей значения формуляра - разрабатываемый  void updateFrameParams(); //Получить поля структуры формуляра - разрабатываемый  Frame getSceneFrame()  //Установить значения структуры формуляра -  разрабатываемый  void setSceneFrame(Frame frame) //Очистить рамку формуляра - разрабатываемый  void clearFrameTexts()   protected: //Обработка событий мыши - измененный библиотечный  virtual void mousePressEvent(QGraphicsSceneMouseEvent  \*event); virtual void mouseReleaseEvent(QGraphicsSceneMouseEvent  \*event); virtual void mouseMoveEvent(QGraphicsSceneMouseEvent  \*event); virtual void keyPressEvent(QKeyEvent \*event); virtual void keyReleaseEvent(QKeyEvent \*event); //Удаление связи со сцены - разрабатываемый  void deleteFromArea(New\_arrow \*arrow); |
| 9 | Описание методов(паспорт метода) | //Обработка нажатия ЛКМ - измененный библиотечный  virtual void mousePressEvent(QGraphicsSceneMouseEvent  \*event);    Прототип: Отсутствует    Входные данные:  Событие нажатия ЛКМ    Выходные данные: Отсутствуют    Назначение: в зависимости от режима работы - выделение элемента на сцене, удаление, начало построения стрелки, начало копирования элементов при зажатой клавише Ctrl, создание нового блока    Алгоритмы: 1. Смотрим активную команду 1.1Если команда - “Новый блок”, сработала ЛКМ и координата добавления не выходит за рамки сцены - создаем новый блок, генерируем его id и добавляем на сцену 1.2 Если команда - “Новая связь” и  сработала ЛКМ  - строим линию между начальной и текущей точкой курсора 1.3 Если команда - “Удалить” и сработала ЛКМ - создаем контейнер элементов под курсором     1.3.1 Если контейнер не пустой -         1.3.1.1 Если первый элемент в контейнере - блок - удаляем его и все ассоциированные с ним связи         1.3.1.2 Если первый элемент - связь, то удаляем ее и все связи, из которых расходится\к которой сходится первая связь 1.4 Если команда - “Висячая стрелка” и сработала ЛКМ - просматриваем контейнер элементов под курсором. Если первый элемент - блок, то в зависимости от места на блоке, куда была нажата ЛКМ, строим укороченную связь одного из 4 типов (вход, управление, механизм, выход) 1.5 Если не выбрана никакая команда и зажата клавиша Ctrl - испускаем сигнал о копировании элементов. Смотрим контейнер элементов под курсором. Если он не пустой и первый элемент - блок, устанавливаем его в качестве выделенного блока, а также устанавливаем его текущие координаты |
| 10 | Тесты | 1. Описание: Кликнуть ЛКМ за пределами рабочей области в каждом режиме. Ожидаемое поведение программы: Отсутсвие каких-либо изменений в программе.  2. Описание: Перейти в режим добавления нового блока. Кликнуть ЛКМ в рабочей области. Ожидаемое поведение программы: Создание нового блока, генерация его id и добавление на рабочую область.  3. Описание: Добавить два блока на рабочую область. Перейти в режим добавления связи. Поочерёдно кликнуть ЛКМ по двум блокам. Ожидаемое поведение программы: Строим стрелку между начальной и конечной точкой курсора.  4. Описание: Построить схему, состояющую из нескольких блоков, связанных стрелками. Перейти в режим удаления. ЛКМ по блоку или стрелке. Ожидаемое поведение программы: Если удаляемый объект блок, удаляем его и все ассоциированные с ним связи. Если это связь - удаляем ее и все связи, из которых расходится\к которой сходится первая связь.  5. Описание: На рабочей области есть блок. Переходим в режим построения висячей стрелки. Кликаем на блок в районе каждой грани. Ожидаемое поведение программы: В зависимости от положения курсора строятся висячие стрелки у каждой грани одного из 4 типов (вход, управление, механизм, выход) 6. Описание: На рабочей области построена диаграмма. Активен режим редактирования. Если кликнуть ЛКМ по связи, блоку, при этом удерживая клавишу CTRL. Ожидаемое поведение программы: Блоки и стрелки после клика становятся выделенными. |
| 11 | Литература | Макс Шлее - Qt 4.2. Профессиональное программирование на C++ (2007) |

3. Етеревский Г.П. Класс Block

|  |  |  |
| --- | --- | --- |
| **№** | **Пункт карточки** | **Описание** |
| 1 | Имя класса | Block |
| 2 | Назначение | Объекты данного класса выполняют роль “Блоков” в соответвии с нотацией IDEF0 |
| 3 | Язык | C++ |
| 4 | Суперкласс | QGraphicsItem |
| 5 | Подкласс | ---- |
| 6 | Взаимодействующие классы + тип взаимодействия | AreaBuild(агрегация) Block\_properties(ассоциация) |
| 7 | Свойства + уровень доступа + библиотечный/ разрабатываемый | //Уникальный тип класса    enum {Type = UserType + 1};  - public //Идентификатор    int id; - public //Имя     QString name; - public //Порядковый номер     QString number2; - public //Координаты Х и У блока на сцене    int posX,posY; - public //Контейнер, содержащий входящие\исходящие стрелки    QList<New\_arrow \*> arrows; - public |
| 8 | Методы + уровень доступа + библиотечный/ разрабатываемый | public: //Функция отрисовки блока    void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);  //Возвращает тип    int type () const { return Type; } //Функция,определяющая область блока     QRectF boundingRect() const; //Функция,возвращающая центр блока     QPoint pos();  //Функции,возвращающие количество входящих\выходящих стрелок      int countOneTypeInArrows(DWORD type);      int countOneTypeOutArrows(DWORD type); //Удаление стрелки       void removeArrow(New\_arrow \*arrow); //Удаление стрелок       void removeArrows(); //Добавление стрелкт       void addArrow(New\_arrow \*arrow);  protected: //----------------------Обработчик событий---------------------------------- //Реакция на двойной клик.Открывается диалоговое окно"Свойства блока"     virtual void douseDoubleClickEvent(QGraphicsSceneMouseEvent \*mouseEvent);  //Реакция на изменение//Реакция на изменение состояния блока     QVariant itemChange(GraphicsItemChange change, const QVariant &value); |
| 9 | Описание методов(паспорт метода) | //Отрисовка блока  void paint(QPainter \*painter, const QStyleOptionGraphicsItem \*option, QWidget \*widget);    Прототип: Отсутствует    Входные данные:QPainter \*painter, QStyleOptionGraphicsItem \*option, QWidget \*widget    Выходные данные: Отсутствуют    Назначение: Отрисовка блока в соответсвии с нотацией IDEF0 (границы блока, наименование, номер)    Алгоритмы: Устанавливает всплывающую подсказку, которая отображается при наведении курсора мыши на “Блок”. 1 Определение  выделенности “блока”.  1.1 Если “Блок” не выделен (на нем не был произведен клик ЛКМ), то устанавливается стандартный цвет “блока” - белый.  1.2 Иначе “блок” обозначается как “выделен” подсветкой светло-синего цвета 2. Отрисовка оконтовки блока на его границах 3. Отрисовка наименование блока   3.1 Если текст превышает допустимую длину, то часть текста скрывается, выставляется обозначение того , что текст не отображен не полностью (...).   3.2 Иначе в блоке отрисовывается текст наименования “блока” целиком 4. Отрисовка номера блока |
| 10 | Тесты | 1. Описание: Изменение координат блока на сцене. Ожидаемое поведение программы: Если “Блок” имеет стрелки, то при перемешении стрелки перерисовываются. 2. Описание: Удаление блока со сцены. Ожидаемое поведение программы: Если “Блок” имеет стрелки, то при удалении все стрелки также удаляются 3. Описание: Добавление стрелки к  “Блоку”. Ожидаемое поведение программы: К “Блоку” добавляется новая стрелка 4. Описание: Наведение курсора мыши на “Блок” Ожидаемое поведение программы: Открывается всплывающая подсказка с полным наименование “блока” |
| 11 | Литература | Жасмин Бланшет, Марк Саммерфилд. Qt4: Программирование GUI на С++ |

Приложение Е. Спецификация кода

|  |  |  |  |
| --- | --- | --- | --- |
| Файл | Назначение | Размер, строк кода | Расположение |
| arrowproperties.ui | GUI диалога для редактирования нотации связи | 76 | /Form files |
| blockproperties.ui | GUI диалога для редактирования параметров блока | 100 | /Form files |
| diarameditor.ui | GUI главного окна программы | 360 | /Form files |
| frameproperties.ui | GUI диалога для редактирования полей формуляра | 240 | /Form files |
| areabuild.h | Заголовочный файл класса сцены | 76 | /Header Files |
| arrowproperties.h | Заголовочный файл диалога связи | 30 | /Header Files |
| block.h | Заголовочный файл класса блока | 52 | /Header Files |
| blockproperties.h | Заголовочный файл диалога блока | 28 | /Header Files |
| diarameditor.h | Заголовочный файл главного класса программы | 65 | /Header Files |
| frameproperties.h | Заголовочный файл диалога формуляра | 44 | /Header Files |
| methods.h | Заголовочный файл класса вспомогательных методов | 103 | /Header Files |
| new\_arrow.h | Заголовочный файл класса стрелка | 110 | /Header Files |
| areabuild.cpp | Содержит методы для работы с объектами и событиями сцены | 650 | /Source Files |
| arrowproperties.cpp | Содержит методы для изменения нотации связи | 138 | /Source Files |
| block.cpp | Содержит параметры и методы работы с блоком | 181 | /Source Files |
| blockproperties.cpp | Содержит методы для изменения и валидации нотации и номера блока | 55 | /Source Files |
| diarameditor.cpp | Содержит обработчики событий интерфейса главного окна | 369 | /Source Files |
| frameproperties.cpp | Содержит методы для изменения параметров рамки | 56 | /Source Files |
| methods.cpp | Содержит методы для сохранения\заргузки диаграммы и работы с файлообменником | 630 | /Source Files |
| new\_arrow.cpp | Содержит параметры и методы работы со связью | 833 | /Source Files |
| main.cpp | Файл инициализации работы программы | 19 | /Source Files |