Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Кафедра информационных систем и технологий**

# ОБЪЕКТНОЕ МОДЕЛИРОВАНИЕ В СРЕДЕ DELPHI

**Методические указания к выполнению контрольных работ по одноименной дисциплине для студентов специальности 1-40 01 02-03 «Информационные системы и технологии (издательско-полиграфический комплекс)» заочной формы обучения**

Минск 2013

УДК 004.434(075.4)(0.034)
ББК 32.97я73
О-29

Рассмотрены и рекомендованы к изданию редакционно-издательским советом университета.

Составители:
*С. И. Акунович, Т. П. Брусенцова*

Рецензент
кандидат педагогических наук, доцент кафедры автоматизации производственных процессов и электротехники БГТУ
*Н. П. Коровкина*

По тематическому плану изданий учебно-методической литературы университета на 2013 год. Поз. 173.

Для студентов специальности 1-40 01 02-3 «Информационные системы и технологии (издательско-полиграфический комплекс)» заочной формы обучения.

# ПРЕДИСЛОВИЕ

Настоящее издание предназначено для студентов специальности «Информационные системы и технологии» заочной формы обучения в помощь при изучении ими дисциплины «Объектное моделирование в среде DELPHI». В соответствии с учебным планом дисциплины аудиторные занятия проводятся в форме лекций – 10 ч (в 8-м семестре), и лабораторных занятий – 8 ч. Промежуточный контроль знаний студентов осуществляется по результатам выполнения индивидуальных заданий в форме тестов на компьютере.

Изучению дисциплины должно предшествовать усвоение базовых курсов высшей математики, физики, микропроцессорных и вычислительных устройств, базовых языков программирования, основ унифицированного языка моделирования UML.

В процессе изучения настоящей дисциплины студент должен освоить основы создания компьютерных моделей систем логического управления (СЛУ) технологическими машинами, оборудованием и процессами на базе общих принципов объектно-ориентированного программирования, моделирования и проектирования, а также использование этих моделей для проверки адекватности процессов функционирования СЛУ на логической стадии проектирования (до начала аппаратной реализации).

В результате изучения дисциплины студенты должны знать и уметь применять на практике:

- способы логического описания дискретных систем логического управления оборудованием и процессами;
- способы и средства создания логических моделей СЛУ в среде Excel;
- структурный анализ и моделирование СЛУ в среде Excel;
- объектное визуальное конструирование программных моделей СЛУ в среде Delphi;
- конструирование UML-моделей СЛУ в среде Delphi.

Основной материал представлен в следующем порядке:

- разработка проектов в среде Delphi;
- структурный анализ и моделирование СЛУ в среде Excel;
- разработка UML-модели СЛУ в среде Delphi.

Контрольное задание (тест) охватывает изучаемые вопросы по всем разделам учебного плана. Настоящее пособие призвано помочь студентам в овладении соответствующими знаниями и в подготовке к тесту.

# ВВЕДЕНИЕ

Современные системы управления (СУ) автоматизированным оборудованием и процессами строятся на общих принципах структурно-функциональной организации. Системы управления химико-технологическими процессами, системы управления автоматическими линиями в машиностроении, системы управления полиграфическим оборудованием или системы управления спутниками навигационных систем могут с достаточной степенью адекватности описываться компьютерными моделями, построенными на базе общих принципов объектно-ориентированного моделирования и проектирования.

Одной из ключевых задач в решении сформулированной проблемы является моделирование и проверка процессов функционирования СУ на стадии проектирования, до начала аппаратной реализации системы.

В результате изучения дисциплины студенты должны знать:

- способы логического описания дискретных систем управления оборудованием и процессами;
- создание баз данных систем управления в среде Excel;
- основы программирования на языке VBA;
- объектное визуальное конструирование программных систем в среде Delphi.

После изучения дисциплины студенты должны уметь:

- создавать проекты в среде Delphi;
- создавать логические описания основных классов компонентов;
- создавать базы данных отдельных модулей систем управления в среде Excel;
- программировать на языке VBA задачи обработки баз данных;
- конструировать объектные модели в среде Delphi.

# 1. РАЗРАБОТКА ПРОЕКТОВ В СРЕДЕ DELPHI

## 1.1. Создание нового проекта в среде Delphi

Программы, которые вы писали до этого, состояли обычно из одного файла (изредка из нескольких), содержащего исходный код. При разработке визуального приложения нам недостаточно хранить только исполняемый код. Нам нужно хранить параметры отображаемой формы, компоненты, расположенные на ней, и их параметры, ресурсы (иконки, картинки и т. д.). Поэтому одно приложение содержит большое количество файлов. Совместно они называются проектом и информация о них хранится в одном главном файле – файле проекта, он имеет расширение ***.dpr**.

После открытия Delphi вам необходимо создать новый проект. Это можно сделать, нажав на «New Project…» на странице приветствия (рис. 1)

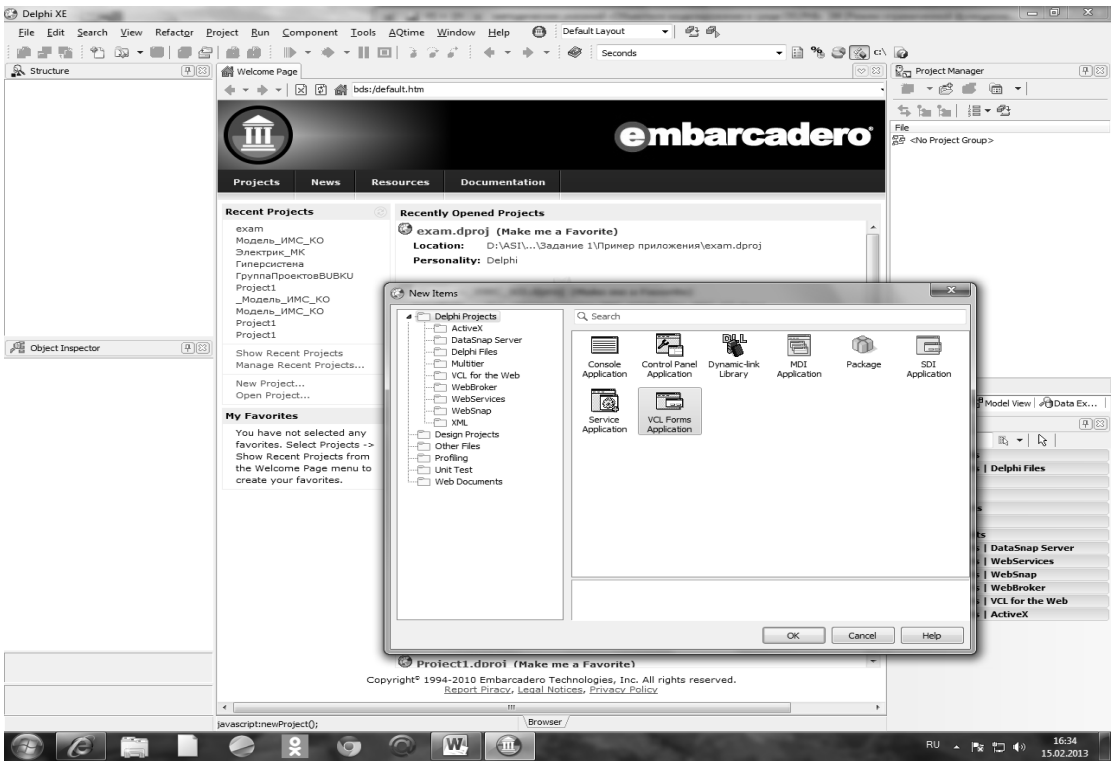



Рис. 1. Создание нового проекта

Выберите в появившемся окне **VCL Forms Application** (рис. 2) выбрав в главном меню **File / New / VCL Forms Application**.

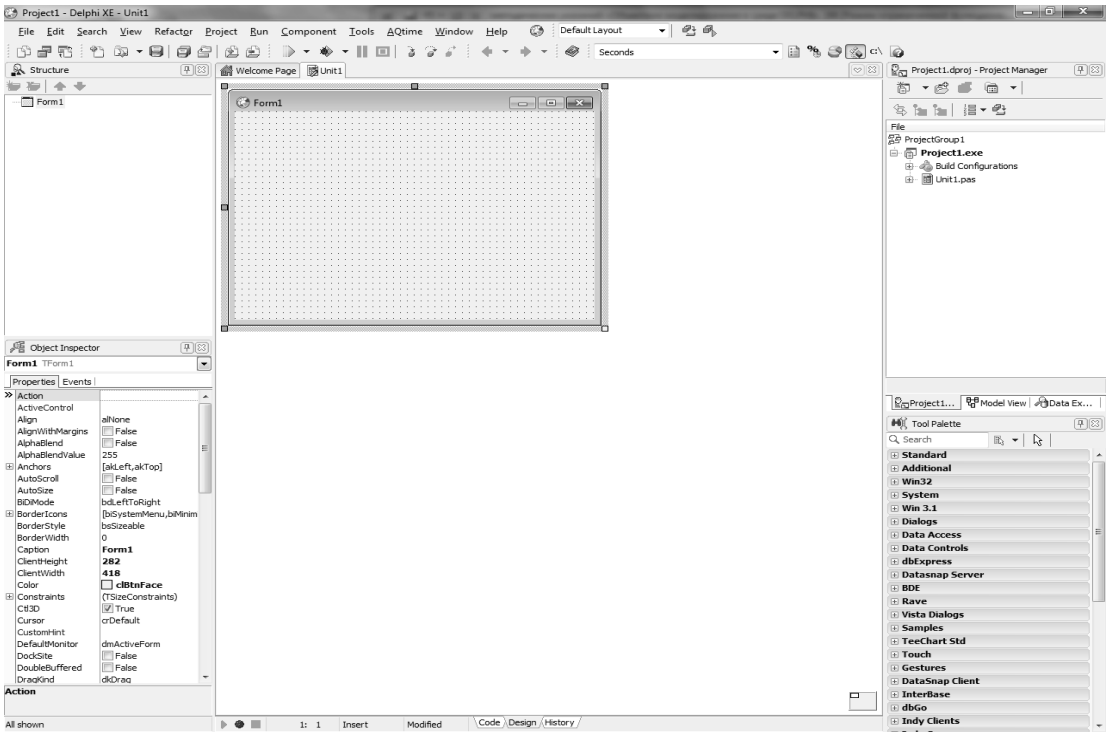



Рис. 2. Вид окна нового проекта

При сохранении проекта очень рекомендуется для каждого проекта создавать отдельную папку, так как количество файлов одного проекта может быть очень большим и возможна путаница. Так же старайтесь не использовать стандартные имена, а придумывать свои. Это облегчит работу с большими проектами.

## 1.2. Основы объектно-ориентированного программирования

### 1.2.1. Объекты и классы

Класс – это тип объекта. Класс характеризует объект вместе с его свойствами и правилами поведения. Объект есть экземпляр класса. В Delphi все объекты динамические. Для выделения памяти под объект используется специальный метод данного класса Constructor Create. Освобождение памяти, выделенной ранее объекту, осуществляется методом Dectructor Destroy. Для объектов и классов сделано одно исключение из общих правил работы на Паскале с динамическими переменными – для них не используется символ ^ (тильда) для указателей и содержимого указываемой памяти. Переменные, описанные

в классе, называют полями. Любая процедура или функция, описанная в классе, является уже методом. При вызове любого метода ему неявным образом первым параметром передается параметр Self, являющийся указателем на объект, который вызвал данный метод. Если процедура описана вне класса, но за ее описанием следуют слова of object, то это тоже будет метод. Классы могут быть описаны или в интерфейсной части модуля Unit, или в самом начале секции реализации Implementation. Не допускается их описание внутри процедур и функций. Если перед описанием метода стоит ключевое слово class, то это классовый метод и его можно вызывать даже в том случае, если объекту еще не была выделена память. Типичный пример класса:

```
Type TmyObject=class(Tobject)
x,y:integer;
Constructor Create;
Destructor Destroy; virtual;
Procedure Show;
End; …
```

В скобках после ключевого слова class указывается наследуемый класс. Объект Tobject является прародителем всех классов и не наследует никаких других классов.

### 1.2.2. Области видимости класса

1. Private – личная, внутренняя область класса. Поля и методы, заданные в этой области, доступны только внутри модуля Unit, где описан данный класс. Для всех других модулей, которые подсоединяют данный модуль и наследуют этот класс, они недоступны.

2. Protected – защищенная область. Поля и методы этой области доступны только внутри классов, наследующих данный класс.

3. Public – общедоступная область. Поля и методы этой области не имеют ограничений на видимость.

4. Published – область публикаций. Поля и методы этой области имеют такую же видимость, как для области public, но они еще видны инспектору объектов на этапе разработки программы. В дочерних классах можно переносить методы и свойства из области Protected в область Published и обратно.

### 1.2.3. Свойства и инкапсуляция

**Объектно-ориентированное программирование (ООП)** основано на трех принципах – инкапсуляция, наследование и полиморфизм.

Классическое ООП утверждает, что чтение и обновление полей должно производиться только специальными методами и не допускается прямое обращение к полям класса. Это правило и называется инкапсуляцией, а такие поля – свойствами. Свойство определяется полем и двумя методами, которые осуществляют чтение и запись заданных значений в поле. Пример определения свойства:

```
Type TmyObject=class(Tobject)
Privete
FmyField:String;
Protected
Procedure SetMyField(Value:String);
Published
Property MyProp:String Read FmyField
Write SetMyField
Default ‘Начальное значение’;
End; …
```

Здесь в классе TmyObject определено свойство MyProp строкового типа. В качестве метода чтения выступает само значение строки, а запись осуществляется методом SetMyField. Само поле FmyField определено в области Privete и поэтому к нему нет прямого доступа из других модулей. Метод чтения этого поля находится в защищенной области, а свойство MyProp – в области публикаций и доступно инспектору объектов во время проектирования программы.

### 1.2.4. Методы и наследование

Наследование означает, что при создании нового класса он наследует все поля, свойства и методы, определенные в родительском классе. В новом классе только добавляются новые поля, методы и свойства. Унаследованные от предка поля и методы доступны в дочернем классе, но с учетом областей видимости. Если имеет место совпадение имен, то говорят, что они перекрываются. В Delphi допускается только последовательное единичное наследование классов. Методы подразделяются на 4 группы:

- статические (Static);
- виртуальные (Virtual);
- динамические (Dynamic);
- абстрактные (Abstract).

Адрес вызова статического метода определяется на этапе трансляции проекта и вызов этих методов осуществляется быстрее всех остальных методов. Такие методы можно перекрывать, при этом име-

ется возможность менять список передаваемых параметров. По умолчанию методы объектов являются статическими.

Адреса виртуальных и динамических методов определяются во время выполнения программы и находятся в специальных таблицах: таблице виртуальных методов (VMT) и таблице динамических методов (DMT). В таблицу VMT включаются адреса всех определенных в данном классе виртуальных методов и всех наследуемых методов. В таблицу DMT включаются адреса динамических методов, определенных только в данном классе. Поэтому виртуальные методы вызываются быстрее динамических, но размеры таблиц VMT существенно больше таблиц DMT. Динамические методы позволяют экономить память, но их вызов осуществляется медленнее всех остальных методов, так как приходится для поиска адреса метода проходить по всем таблицам DMT родительских классов, пока не найдется нужный нам динамический метод.

Для перекрытия виртуальных и динамических методов используется ключевое слово Override. Список параметров перекрываемых виртуальных и динамических методов не должен отличаться от списка параметров этих методов в родительском классе.

Абстрактные методы определяют только интерфейсную часть метода, такие методы нельзя использовать без перекрытия в дочерних классах, где должна находиться и реализация такого метода.

### 1.2.5. События

События в Delphi – это свойства процедурного типа, предназначенные для создания пользовательской реакции на те или иные входные воздействия. Пример объявления события:

```
Property OnMyEvent:TmyEvent Read FOnMyEvent
Write FOnMyEvent; …
```

Присвоить такому свойству значение – это значит указать адрес метода, который будет вызываться в момент наступления события. Такие методы называются обработчиками событий. События имеют разные типы, но общим для всех является параметр Sender – указатель на объект источника события. Самый простой тип события:

```
TnotifyEvent=procedure(Sender:Tobject) of Object; …
```

Здесь «of object» означает, что данный тип определяет именно метод, а не обычную процедуру. Приставка On в имени свойства означает, что данное свойство является событием, хотя не каждое событие

может иметь такую приставку. Для определения события необходимо в разделе Private объявить поле указателя на метод. В разделе Protected нужно объявить методы чтения и записи адреса обработчика события в это поле и объявить событие как свойство процедурного типа. В инспекторе объектов есть две страницы: страница свойств (Properties) и страница событий (Events). Двойной щелчок левой клавишей мыши по событию приводит к появлению обрамления обработчика события в тексте программного модуля Unit.

### *Вопросы для контроля и самоконтроля*

1. Этапы решения задач в Delphi.
2. Основные составляющие среды Delphi и их назначение.
3. Состав и назначение пунктов главного меню системы.
4. Каково назначение Инспектора объектов?
5. Из каких основных файлов состоит проект приложения?
6. Чем класс отличается от объекта?
7. Какими бывают объекты: статическими или динамическими?
8. Для каких целей используется метод Create?
9. Что собой представляет неявно передаваемый в объект параметр Self?
10. Области видимости класса.
11. Что такое свойства объектов?
12. Что обозначает принцип инкапсуляции в ООП?
13. Чем метод отличается от обычной процедуры?
14. Какие вы знаете типы методов?
15. Что означает принцип наследования классов?
16. Что такое полиморфизм в ООП?
17. Что такое событие и чем оно отличается от свойства класса?
18. Чем динамические методы отличаются от виртуальных?
19. Где можно давать определение классу?
20. Структура программы в Delphi.
21. Основные типы данных в Delphi.
22. Структурные типы.
23. Основные операторы.
24. Исключительные ситуации.
25. Процедура.
26. Функция.
27. Локальные и глобальные переменные.
28. Формальные и фактические параметры.

29. Файл. Описание файлов.
30. Классификация файлов.
31. Процедуры и функции для работы с файлами.
32. Потоки данных.
33. Подпрограмма.
34. Модуль.
35. Структура модуля.
36. Разработка модуля.
37. Библиотеки подпрограмм.
38. Визуальные компоненты.
39. Управление приложением с помощью команд.
40. Технология естественного ввода.
41. Управление процессами.
42. Сетевое взаимодействие.
43. Платформа FireMonkey и VCL.
44. Кадровая и траекторная анимация.

# 2. СТРУКТУРНЫЙ АНАЛИЗ И МОДЕЛИРОВАНИЕ СЛУ В СРЕДЕ EXCEL

Моделирование СЛУ выполняется для проверки правильности логических формул, обеспечивающих обработку входных сигналов (например, от датчиков космического аппарата (КА)) и выдачу выходных сигналов на исполнительные механизмы КА.

Для этой цели разработчик переключает в нужной последовательности входные элементы. Excel вычисляет все логические формулы по событийному принципу и отображает новые состояния всех элементов. После их анализа разработчик делает выводы о правильнсти логических формул или необходимости их исправления.

Рассмотрим описанный процесс на примере моделирования переключения функций памяти ZРОБИВК и ZЗТРЕВ в книгах EXCEL ИМС_КО и ЦПМ. Процесс моделирования состоит из следующих этапов:

- В книге EXCEL ИМС_КО определяем структурные связи между элементами (логическими формулами). Для включения ZРОБИВК нужно включить входной контакт X011_17 (рис. 3).

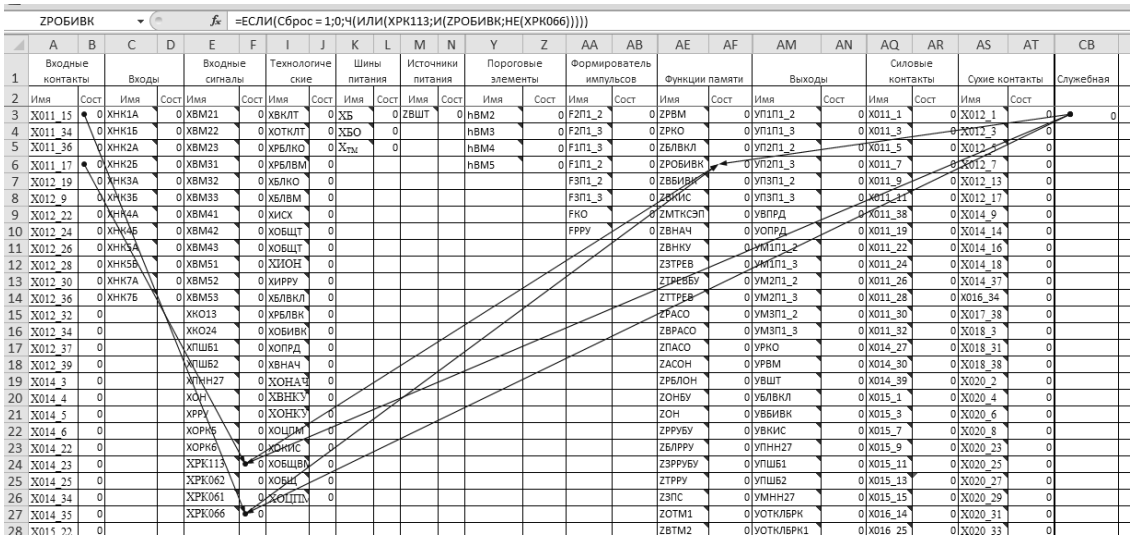
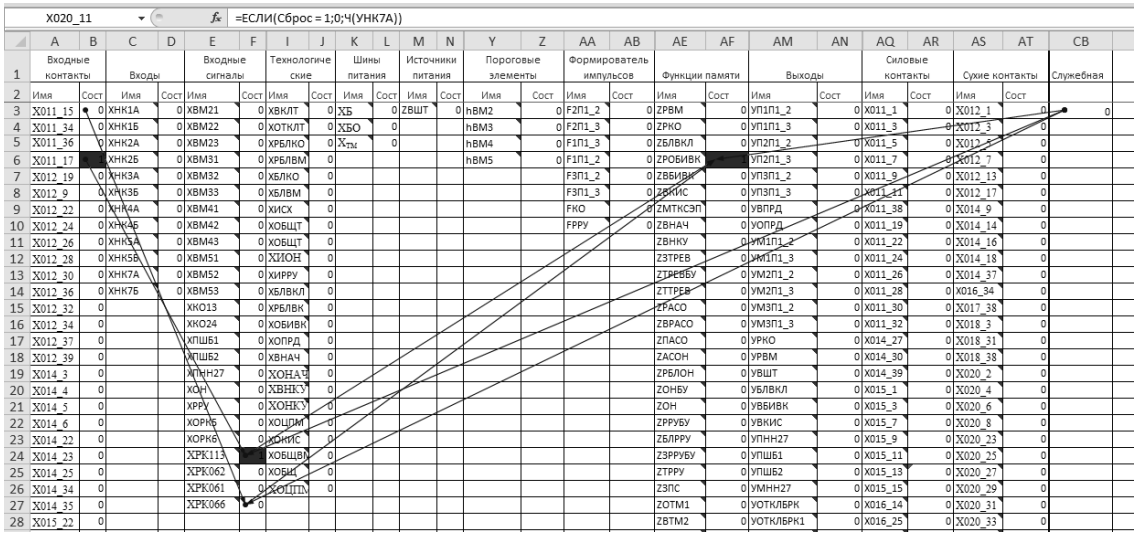



Рис. 3. Определение структурных связей между элементами (логическими формулами)

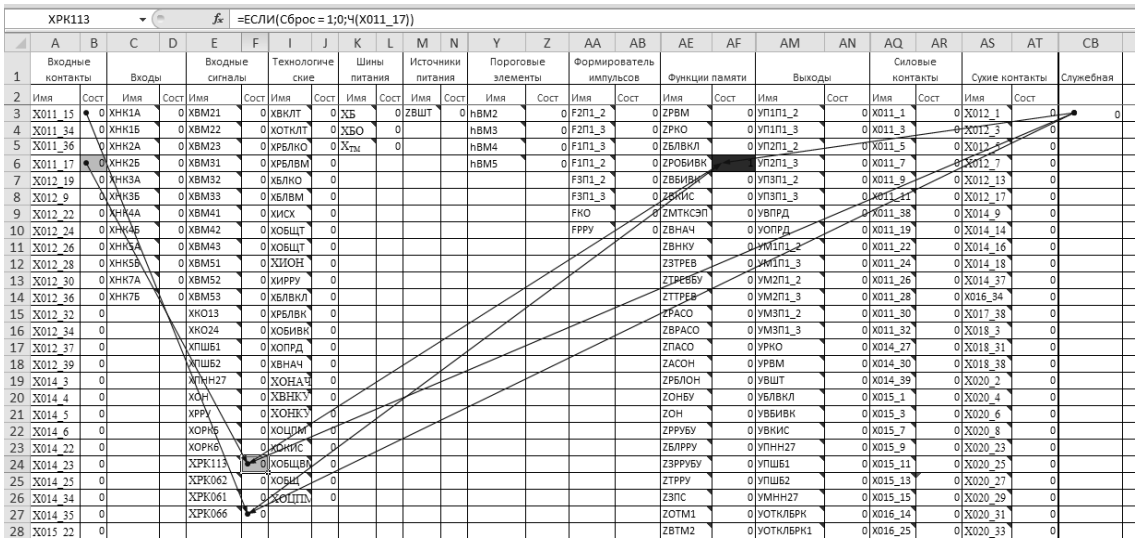
- Включаем X011_17, в результате чего включается ХРК113 и ZРОБИВК. Все переключения выделяем красным цветом (рис. 4).



| | Входные контакты | | Входы | | Входные сигналы | | Технологические | | Шины питания | | Источники питания | | Пороговые элементы | | Формирователь импульсов | | Функции памяти | | Выходы | | Силовые контакты | | Сухие контакты | | Служебная |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | |
| 3 | X011_15 | 0 | XHK1A | 0 | XBM21 | 0 | XBKLT | 0 | XB | 0 | ZBWT | 0 | hBM2 | 0 | F2П1_2 | 0 | ZPBM | 0 | УП1П1_2 | 0 | X011_1 | 0 | X012_1 | 0 | 0 |
| 4 | X011_34 | 0 | XHK1B | 0 | XBM22 | 0 | XOTKLT | 0 | XBO | 0 | | | hBM3 | 0 | F2П1_3 | 0 | ZPKO | 0 | УП1П1_3 | 0 | X011_3 | 0 | X012_3 | 0 | |
| 5 | X011_36 | 0 | XHK2A | 0 | XBM23 | 0 | XPBLKO | 0 | $X_{TM}$ | 0 | | | hBM4 | 0 | F1П1_3 | 0 | ZBLVKL | 0 | УП2П1_2 | 0 | X011_5 | 0 | X012_5 | 0 | |
| 6 | X011_17 | 0 | XHK2B | 0 | XBM31 | 0 | XPBLBM | 0 | | | | | hBM5 | 0 | F1П1_2 | 0 | ZPOBIVK | 0 | УП2П1_3 | 0 | X011_7 | 0 | X012_7 | 0 | |
| 7 | X012_19 | 0 | XHK3A | 0 | XBM32 | 0 | XBLKO | 0 | | | | | | | F3П1_2 | 0 | ZBBIVK | 0 | УП3П1_2 | 0 | X011_9 | 0 | X012_13 | 0 | |
| 8 | X012_9 | 0 | XHK3B | 0 | XBM33 | 0 | XBLVM | 0 | | | | | | | F3П1_3 | 0 | ZBKIS | 0 | УП3П1_3 | 0 | X011_11 | 0 | X012_17 | 0 | |
| 9 | X012_22 | 0 | XHK4A | 0 | XBM41 | 0 | XISX | 0 | | | | | | | FKO | 0 | ZMTKSEP | 0 | УВПРД | 0 | X011_38 | 0 | X014_9 | 0 | |
| 10 | X012_24 | 0 | XHK4B | 0 | XBM42 | 0 | XOBSHT | 0 | | | | | | | FPPU | 0 | ZBNACH | 0 | УОПРД | 0 | X011_19 | 0 | X014_14 | 0 | |
| 11 | X012_26 | 0 | XHK5A | 0 | XBM43 | 0 | XOBSHT | 0 | | | | | | | | | ZBNKU | 0 | УМ1П1_2 | 0 | X011_22 | 0 | X014_16 | 0 | |
| 12 | X012_28 | 0 | XHK5B | 0 | XBM51 | 0 | XION | 0 | | | | | | | | | Z3TREB | 0 | УМ1П1_3 | 0 | X011_24 | 0 | X014_18 | 0 | |
| 13 | X012_30 | 0 | XHK7A | 0 | XBM52 | 0 | XIRPU | 0 | | | | | | | | | ZTREBBU | 0 | УМ2П1_2 | 0 | X011_26 | 0 | X014_37 | 0 | |
| 14 | X012_36 | 0 | XHK7B | 0 | XBM53 | 0 | XBLVKL | 0 | | | | | | | | | ZTTREB | 0 | УМ2П1_3 | 0 | X011_28 | 0 | X016_34 | 0 | |
| 15 | X012_32 | 0 | | | XKO13 | 0 | XPBLVK | 0 | | | | | | | | | ZPACO | 0 | УМ3П1_2 | 0 | X011_30 | 0 | X017_38 | 0 | |
| 16 | X012_34 | 0 | | | XKO24 | 0 | XOBIVK | 0 | | | | | | | | | ZBPACO | 0 | УМ3П1_3 | 0 | X011_32 | 0 | X018_3 | 0 | |
| 17 | X012_37 | 0 | | | XPShB1 | 0 | XOPPD | 0 | | | | | | | | | ZPACO | 0 | УРКО | 0 | X014_27 | 0 | X018_31 | 0 | |
| 18 | X012_39 | 0 | | | XPShB2 | 0 | XBNACH | 0 | | | | | | | | | ZACOH | 0 | УРВМ | 0 | X014_30 | 0 | X018_38 | 0 | |
| 19 | X014_3 | 0 | | | XMNN27 | 0 | XOHAC | 0 | | | | | | | | | ZPBLON | 0 | УВШТ | 0 | X014_39 | 0 | X020_2 | 0 | |
| 20 | X014_4 | 0 | | | XOH | 0 | XBNKU | 0 | | | | | | | | | ZOHBU | 0 | УБЛВКЛ | 0 | X015_1 | 0 | X020_4 | 0 | |
| 21 | X014_5 | 0 | | | XPPU | 0 | XONKU | 0 | | | | | | | | | ZOH | 0 | УВБИВК | 0 | X015_3 | 0 | X020_6 | 0 | |
| 22 | X014_6 | 0 | | | XOPK5 | 0 | XOCPM | 0 | | | | | | | | | ZPPUBU | 0 | УВКИС | 0 | X015_7 | 0 | X020_8 | 0 | |
| 23 | X014_22 | 0 | | | XOPK6 | 0 | XOKIS | 0 | | | | | | | | | ZBLPPU | 0 | УПНН27 | 0 | X015_9 | 0 | X020_23 | 0 | |
| 24 | X014_23 | 0 | | | XPK113 | 0 | XOBSHB | 0 | | | | | | | | | Z3PPUBU | 0 | УПШБ1 | 0 | X015_11 | 0 | X020_25 | 0 | |
| 25 | X014_25 | 0 | | | XPK062 | 0 | XOBSH | 0 | | | | | | | | | ZTPPU | 0 | УПШБ2 | 0 | X015_13 | 0 | X020_27 | 0 | |
| 26 | X014_34 | 0 | | | XPK061 | 0 | XOCPM | 0 | | | | | | | | | Z3PC | 0 | УМНН27 | 0 | X015_15 | 0 | X020_29 | 0 | |
| 27 | X014_35 | 0 | | | XPK066 | 0 | | | | | | | | | | | ZOTM1 | 0 | УОТКЛБРК | 0 | X016_14 | 0 | X020_31 | 0 | |
| 28 | X015_22 | 0 | | | | | | | | | | | | | | | ZBTM2 | 0 | УОТКЛБРК1 | 0 | X016_25 | 0 | X020_33 | 0 | |

Рис. 4. Вид окна при включенном контакте X011_17

• Выключаем входной контакт X011_17, но ZPOБИВК остается включенным (рис. 5).



| | Входные контакты | | Входы | | Входные сигналы | | Технологические | | Шины питания | | Источники питания | | Пороговые элементы | | Формирователь импульсов | | Функции памяти | | Выходы | | Силовые контакты | | Сухие контакты | | Служебная |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | Имя | Сост | |
| 3 | X011_15 | 0 | XHK1A | 0 | XBM21 | 0 | XBKLT | 0 | XB | 0 | ZBWT | 0 | hBM2 | 0 | F2П1_2 | 0 | ZPBM | 0 | УП1П1_2 | 0 | X011_1 | 0 | X012_1 | 0 | 0 |
| 4 | X011_34 | 0 | XHK1B | 0 | XBM22 | 0 | XOTKLT | 0 | XBO | 0 | | | hBM3 | 0 | F2П1_3 | 0 | ZPKO | 0 | УП1П1_3 | 0 | X011_3 | 0 | X012_3 | 0 | |
| 5 | X011_36 | 0 | XHK2A | 0 | XBM23 | 0 | XPBLKO | 0 | $X_{TM}$ | 0 | | | hBM4 | 0 | F1П1_3 | 0 | ZBLVKL | 0 | УП2П1_2 | 0 | X011_5 | 0 | X012_5 | 0 | |
| 6 | X011_17 | 0 | XHK2B | 0 | XBM31 | 0 | XPBLBM | 0 | | | | | hBM5 | 0 | F1П1_2 | 0 | ZPOBIVK | 0 | УП2П1_3 | 0 | X011_7 | 0 | X012_7 | 0 | |
| 7 | X012_19 | 0 | XHK3A | 0 | XBM32 | 0 | XBLKO | 0 | | | | | | | F3П1_2 | 0 | ZBBIVK | 0 | УП3П1_2 | 0 | X011_9 | 0 | X012_13 | 0 | |
| 8 | X012_9 | 0 | XHK3B | 0 | XBM33 | 0 | XBLVM | 0 | | | | | | | F3П1_3 | 0 | ZBKIS | 0 | УП3П1_3 | 0 | X011_11 | 0 | X012_17 | 0 | |
| 9 | X012_22 | 0 | XHK4A | 0 | XBM41 | 0 | XISX | 0 | | | | | | | FKO | 0 | ZMTKSEP | 0 | УВПРД | 0 | X011_38 | 0 | X014_9 | 0 | |
| 10 | X012_24 | 0 | XHK4B | 0 | XBM42 | 0 | XOBSHT | 0 | | | | | | | FPPU | 0 | ZBNACH | 0 | УОПРД | 0 | X011_19 | 0 | X014_14 | 0 | |
| 11 | X012_26 | 0 | XHK5A | 0 | XBM43 | 0 | XOBSHT | 0 | | | | | | | | | ZBNKU | 0 | УМ1П1_2 | 0 | X011_22 | 0 | X014_16 | 0 | |
| 12 | X012_28 | 0 | XHK5B | 0 | XBM51 | 0 | XION | 0 | | | | | | | | | Z3TREB | 0 | УМ1П1_3 | 0 | X011_24 | 0 | X014_18 | 0 | |
| 13 | X012_30 | 0 | XHK7A | 0 | XBM52 | 0 | XIRPU | 0 | | | | | | | | | ZTREBBU | 0 | УМ2П1_2 | 0 | X011_26 | 0 | X014_37 | 0 | |
| 14 | X012_36 | 0 | XHK7B | 0 | XBM53 | 0 | XBLVKL | 0 | | | | | | | | | ZTTREB | 0 | УМ2П1_3 | 0 | X011_28 | 0 | X016_34 | 0 | |
| 15 | X012_32 | 0 | | | XKO13 | 0 | XPBLVK | 0 | | | | | | | | | ZPACO | 0 | УМ3П1_2 | 0 | X011_30 | 0 | X017_38 | 0 | |
| 16 | X012_34 | 0 | | | XKO24 | 0 | XOBIVK | 0 | | | | | | | | | ZBPACO | 0 | УМ3П1_3 | 0 | X011_32 | 0 | X018_3 | 0 | |
| 17 | X012_37 | 0 | | | XPShB1 | 0 | XOPPD | 0 | | | | | | | | | ZPACO | 0 | УРКО | 0 | X014_27 | 0 | X018_31 | 0 | |
| 18 | X012_39 | 0 | | | XPShB2 | 0 | XBNACH | 0 | | | | | | | | | ZACOH | 0 | УРВМ | 0 | X014_30 | 0 | X018_38 | 0 | |
| 19 | X014_3 | 0 | | | XMNN27 | 0 | XOHAC | 0 | | | | | | | | | ZPBLON | 0 | УВШТ | 0 | X014_39 | 0 | X020_2 | 0 | |
| 20 | X014_4 | 0 | | | XOH | 0 | XBNKU | 0 | | | | | | | | | ZOHBU | 0 | УБЛВКЛ | 0 | X015_1 | 0 | X020_4 | 0 | |
| 21 | X014_5 | 0 | | | XPPU | 0 | XONKU | 0 | | | | | | | | | ZOH | 0 | УВБИВК | 0 | X015_3 | 0 | X020_6 | 0 | |
| 22 | X014_6 | 0 | | | XOPK5 | 0 | XOCPM | 0 | | | | | | | | | ZPPUBU | 0 | УВКИС | 0 | X015_7 | 0 | X020_8 | 0 | |
| 23 | X014_22 | 0 | | | XOPK6 | 0 | XOKIS | 0 | | | | | | | | | ZBLPPU | 0 | УПНН27 | 0 | X015_9 | 0 | X020_23 | 0 | |
| 24 | X014_23 | 0 | | | XPK113 | 0 | XOBSHB | 0 | | | | | | | | | Z3PPUBU | 0 | УПШБ1 | 0 | X015_11 | 0 | X020_25 | 0 | |
| 25 | X014_25 | 0 | | | XPK062 | 0 | XOBSH | 0 | | | | | | | | | ZTPPU | 0 | УПШБ2 | 0 | X015_13 | 0 | X020_27 | 0 | |
| 26 | X014_34 | 0 | | | XPK061 | 0 | XOCPM | 0 | | | | | | | | | Z3PC | 0 | УМНН27 | 0 | X015_15 | 0 | X020_29 | 0 | |
| 27 | X014_35 | 0 | | | XPK066 | 0 | | | | | | | | | | | ZOTM1 | 0 | УОТКЛБРК | 0 | X016_14 | 0 | X020_31 | 0 | |
| 28 | X015_22 | 0 | | | | | | | | | | | | | | | ZBTM2 | 0 | УОТКЛБРК1 | 0 | X016_25 | 0 | X020_33 | 0 | |

Рис. 5. Вид окна после выключения контакта X011_17

## *Вопросы для контроля и самоконтроля*

1. Что такое Microsoft Excel?
2. Что представляет собой книга в Excel?
3. Что такое относительная ссылка?
4. Что такое абсолютная ссылка?
5. Изменяются или нет относительные адреса при копировании формул?

6. Изменяются или нет абсолютные адреса при копировании формул?
7. Каких числовых форматов нет в Excel?
8. Каких типов функций нет в Excel?
9. Какие ссылки используются в формулах Excel?
10. Какие функции относят к основным логическим функциям в Microsoft Excel?
11. Как создать и встроить в Excel пользовательскую логическую функцию?
12. Как могут быть озаглавлены столбцы в Excel?
13. Что может быть содержимым ячейки в Excel?
14. Что означает ошибка #ЗНАЧ! ?
15. Что означает ошибка #ЧИСЛО!?
16. Что означает ошибка #ССЫЛКА?
17. Как выполняется сжатие таблицы для отображения избранных диапазонов?
18. Как использовать имена ячеек вместо адресов в строке формул?
19. Какие данные можно изменять с помощью пользовательских функций?
20. Как создаются и обрабатываются реляционные таблицы в Excel?
21. Чем отличается операторное представление формул алгебры логики от функционального представления в строке формул Excel?
22. Как реализуется событийная модель вычислений в Excel?
23. Как определяются влияющие ячейки?
24. Как определяются зависимые ячейки?
25. Как представляются значения времени в Excel?
26. Какие средства для обработки значений времени имеются в Excel?
27. Как создать модель программного таймера в Excel?
28. Как можно проверить семантику логических формул в Excel?
29. Как можно визуализировать процесс логических вычислений?

# 3. РАЗРАБОТКА UML-МОДЕЛИ В СРЕДЕ DELPHI

Разработка UML-модели в среде Delphi выполняется на базе модели системы логического управления (СЛУ), созданной в Excel.

Модель разрабатывается по варианту. Вариант определяется номером строки в книге EXCEL ИМС_КО по формуле:

Номер строки = (последняя цифра номера зачетки × 5) + 3.

Состав классов модели соответствует книге EXCEL ИМС_КО, поэтому для всех вариантов *диаграмма классов* одинакова. Зависимости на диаграмме классов задаются в соответствии с вариантом.

Кроме диаграммы классов необходимо создать фрагменты диаграммы кооперации и диаграммы последовательностей.

Диаграмма *Пакеты UML-модели* (рис. 6) создается автоматически по мере разработки других диагрмм.

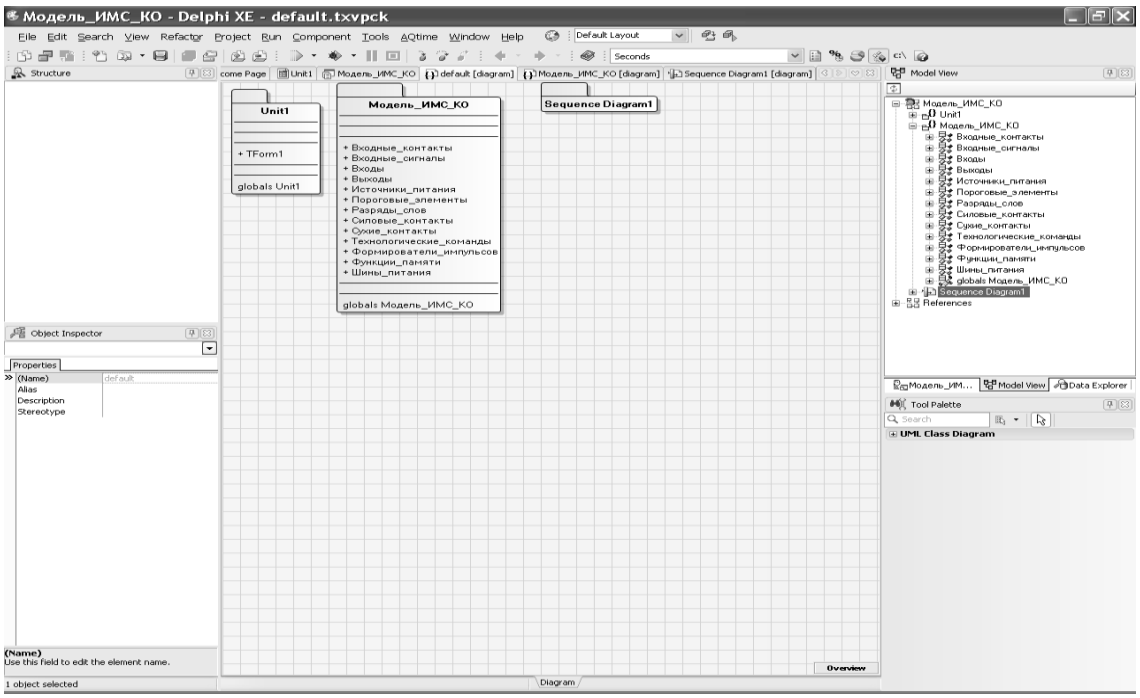



Рис. 6. Диаграмма *Пакеты UML-модели*

На рис. 7 представлен пример диаграммы классов для одного из вариантов.

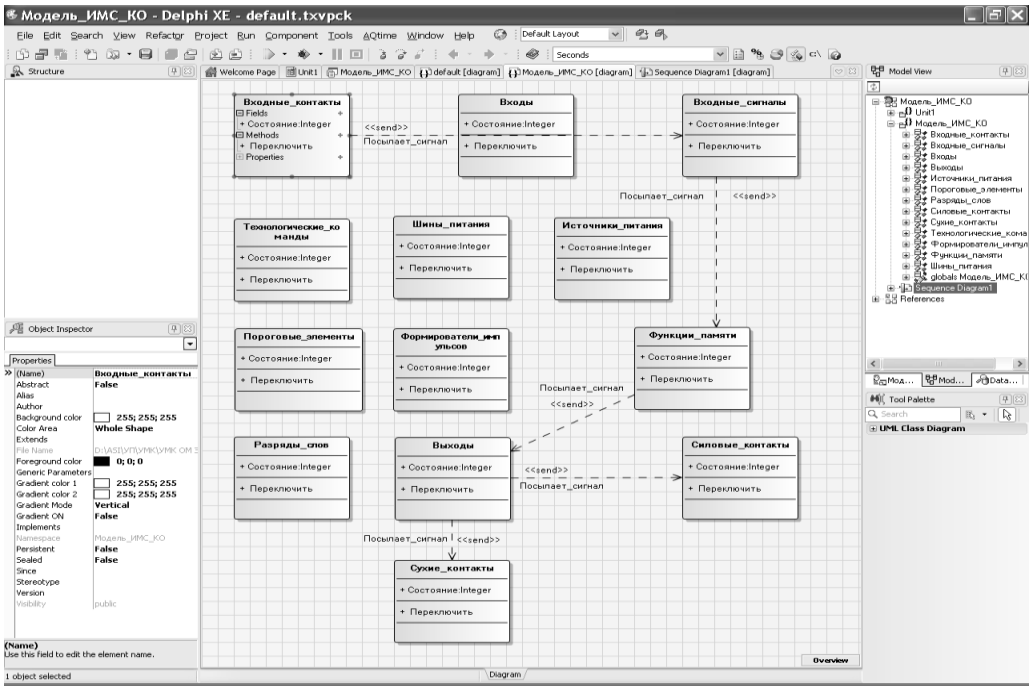



Рис. 7. Пример диаграммы классов

По диаграмме классов можно создать программу на языке Delphi (рис. 8).

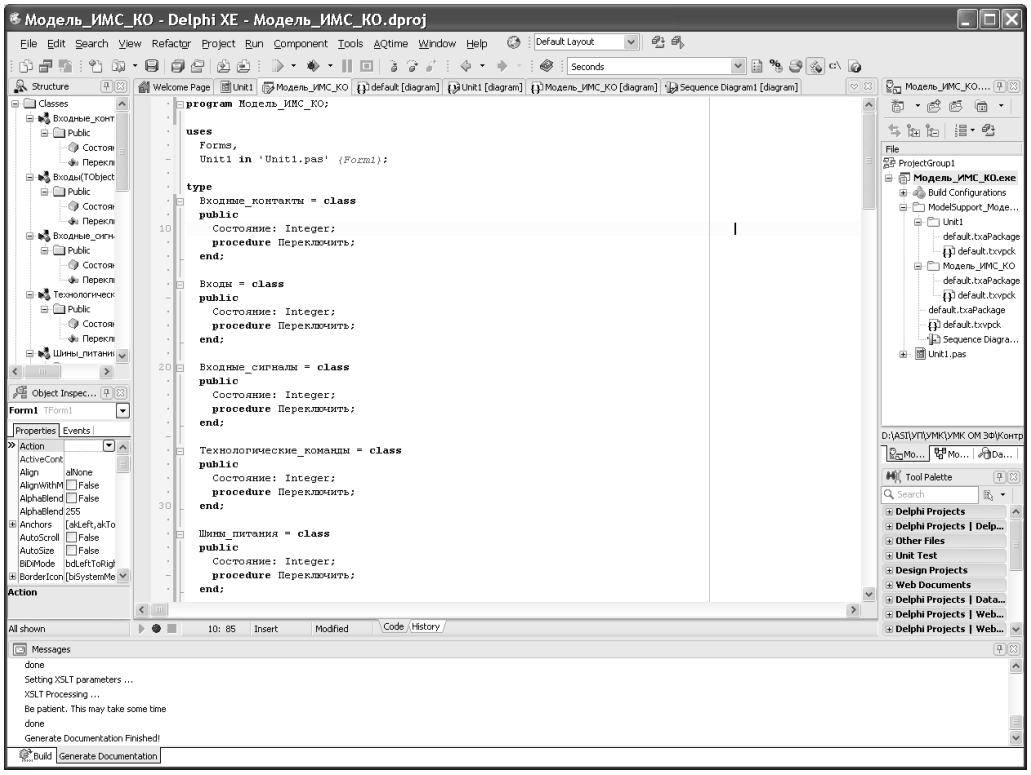



Рис. 8. Скелетный код программы, сгенерированный по диаграмме классов

На диаграмме последовательности (рис. 9) создается только по одному объекту для классов, между которыми установлены зависимости на диаграмме классов.

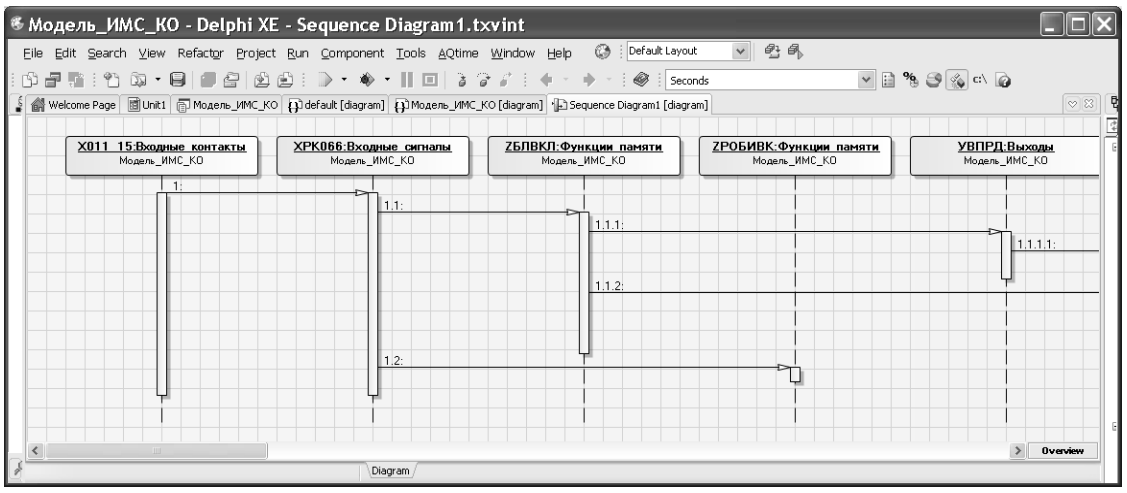



Рис. 9. Диаграмма последовательности

Диаграмма кооперации (рис. 10) создается автоматически по диаграмме последовательности.

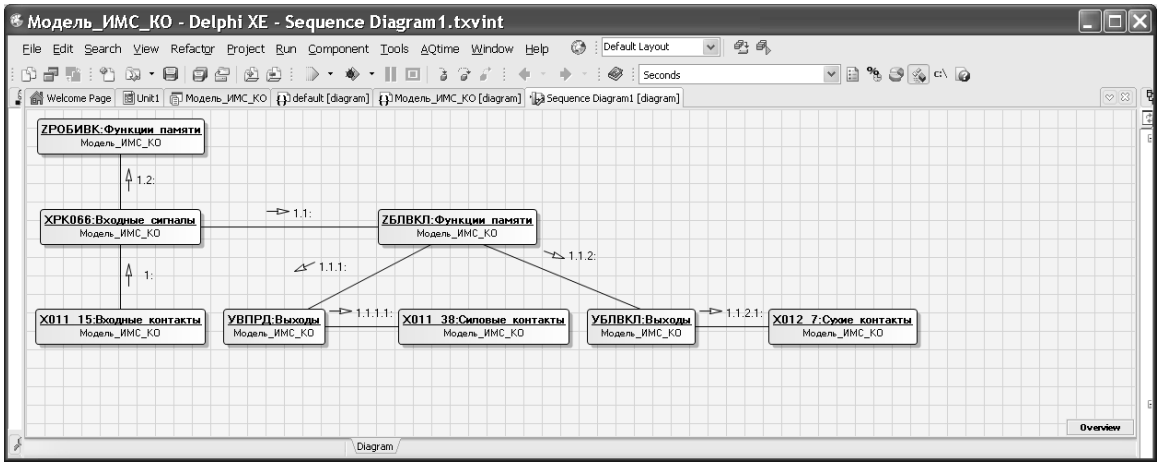



Рис. 10. Диаграмма кооперации

## *Вопросы для контроля и самоконтроля*

1. Что обозначает аббревиатура UML?
2. Кому и зачем нужен UML?
3. Что послужило причиной возникновения UML?
4. Для чего используют UML на практике?
5. Как определен UML?

6. Из чего состоит UML?
7. Что такое модель UML?
8. Из каких элементов состоит модель?
9. Как комбинируются элементы модели?
10. Какова общая структура модели?
11. Что такое моделирование использования и зачем оно нужно?
12. Какие средства применяются при моделировании использования?
13. Как идентифицировать варианты использования и действующих лиц?
14. Как реализуются варианты использования?
15. Что такое моделирование структуры?
16. В чем заключаются особенности объектно-ориентированного моделирования структуры?
17. Какие элементы применяются для моделирования структуры?
18. Как и для чего создаются диаграммы классов?
19. Какие сущности используют на диаграмме классов?
20. Какие структурные взаимосвязи можно описать с помощью отношений на диаграмме классов?
21. В каких случаях применяются диаграммы компонентов и размещения?
22. Что такое диаграмма внутренней структуры и для чего она используется?
23. Что такое моделирование поведения?
24. Как может статическая модель описывать динамическое поведение?
25. Что такое конечный автомат?
26. Какие элементы применяются для моделирования поведения?
27. Для чего применяются диаграммы автомата?
28. Что общего и в чем различие между диаграммами автомата и деятельности?
29. Какова область применения диаграмм взаимодействия?
30. Как моделируются параллельные процессы?
31. Какие диаграммы UML реализованы в Delphi?
32. Как создается визуальная модель проекта?
33. Что содержит скелетный код диаграммы классов?
34. Чем отличаются диаграмма последовательности и диаграмма кооперации?
35. Как преобразуются диаграмма последовательности и диаграмма кооперации?

# СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ


1. Объектно-ориентированный анализ и проектирование с примерами приложений / Г. Буч [и др.] – 3-е изд. – М.: Вильямс, 2008. – 718 с.

2. Новиков, Ф. А. Моделирование на UML / Ф. А. Новиков, Д. Ю. Иванов. – СПб.: Наука и техника, 2010. – 635 с.

3. Акунович, С. И. Композиционное конструирование моделей систем дискретного управления в среде DELPHI / С. И. Акунович. – Минск: БГТУ, 2009. – 75 с.

4. Уокенбах, Дж. Microsoft Office Excel 2007: профессиональное программирование на VBA / Дж. Уокенбах. – М.: Вильямс, 2008. – 918 с.

5. Акунович, С. И. Дискретные системы логического управления технологических машин / С. И. Акунович, А. А. Гончаров, Ю. Н. Петренко. – Минск: Юнипак, 2006. – 334 с.

6. Акунович С. И. Моделирование и анализ проектов систем дискретного управления / С. И. Акунович // Основы построения комплексной САПР систем дискретного управления технологическим оборудованием и процессами: учеб. пособие: в 2 ч. / С. И. Акунович, А. А. Гончаров, А. А Дятко. – Минск: БГТУ, 2002. – Ч. 2. – 38 с.

7. Осипов, Д. Delphi XE2 / Д. Осипов. – СПб.: БХВ-Петербург, 2012. – 892 с.

8. Бобровский, С. Технологии DELPHI / С. Бобровский. – СПб.: Питер, 2007. – 719 с.

9. Культин, Н. Основы программирования в TURBO DELPHI / Н. Культин. – СПб.: БХВ-Петербург, 2007. – 384 с.

10. Колосов, С. В. Объектно-ориентированное программирование в вреде Delphi. Лабораторный практикум для студентов всех специальностей / С. В. Колосов. – Минск: БГУИР, 2001. – 47 с.

11. Леоненков, А. В. Самоучитель UML / А. В. Леоненков. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2004. – 427 с.

# ОБЪЕКТНОЕ МОДЕЛИРОВАНИЕ В СРЕДЕ DELPHI

Методические указания

Составители: **Акунович** Станислав Иванович
**Брусенцова** Татьяна Палладьевна

Редактор *О. П. Приходько*

Компьютерная верстка *Я. Ч. Болбот*

Корректор *О. П. Приходько*

Издатель:
УО «Белорусский государственный технологический университет».
ЛИ № 02330/0549423 от 08.04.2009.
Ул. Свердлова, 13а, 220006, г. Минск