

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
АВТОМОБІЛЬНО-ДОРОЖНІЙ ІНСТИТУТ

Кафедра «Інформаційні системи в економіці»

**МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ
З ДИСЦИПЛІНИ
«ОБ'ЄКТНИЙ АНАЛІЗ І МОДЕЛЮВАННЯ СИСТЕМ»
(ДЛЯ СТУДЕНТІВ НАПРЯМУ
6.0305 «ЕКОНОМІЧНА КІБЕРНЕТИКА»)**

07/53 – 2012-06

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
АВТОМОБІЛЬНО-ДОРОЖНИЙ ІНСТИТУТ

«ЗАВЕРДЖУЮ»
Директор АДІ ДВНЗ «ДонНТУ»
М.М. Чальцев

Кафедра «Інформаційні системи в економіці»

**МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ
З ДИСЦИПЛІНИ
«ОБ'ЄКТНИЙ АНАЛІЗ І МОДЕЛЮВАННЯ СИСТЕМ»
(ДЛЯ СТУДЕНТІВ НАПРЯМУ
6.0305 «ЕКОНОМІЧНА КІБЕРНЕТИКА»)**

07/53 – 2012-06

«РЕКОМЕНДОВАНО»
Навчально-методична комісія
факультету
«Економіка та управління»
Протокол № 3 від 21.12.2011 р.

«РЕКОМЕНДОВАНО»
Кафедра
«Інформаційні системи в економіці»
Протокол № 4 від 25.11.2011 р.

УДК 681.32 (07)

Методичні вказівки до курсової роботи «Об'єктний аналіз і моделювання систем» (для студентів напряму 6.0305 «Економічна кібернетика») [Електронний ресурс] / укладачі.: В. Л. Ніколаєнко, Д. В. Ніколаєнко. – Електрон. дані. – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012. – 1 електрон. опт. диск (CD-R); 12 см. – Систем. вимоги: Pentium; 32 MB RAM; WINDOWS 98/2000/NT/XP; MS Word 2000. – Назва з титул. екрану.

Наведені вимоги до оформлення звіту про виконані роботи. Порядок виконання робіт. Також наведені необхідні теоретичні відомості об'єктного аналізу, моделювання й проектування систем. Основні етапи об'єктного аналізу, моделювання й проектування проілюстровані спеціально підібраними прикладами. Методичні вказівки націлені на практичне засвоєння теоретичних положень об'єктного моделювання систем.

Укладачі:

Ніколаєнко В. Л., к.т.н., доц.
Ніколаєнко Д. В., к.т.н., доц.

Відповідальний за випуск:

Ніколаєнко В. Л., к.т.н., доц.

Рецензент:

Вовк Л. П., д.т.н., проф.
каф. «Вища математика»

© Державний вищий навчальний заклад
«Донецький національний технічний університет»
Автомобільно-дорожній інститут, 2012

ЕЛЕКТРОННЕ НАУКОВО-МЕТОДИЧНЕ ВИДАННЯ

Ніколаєнко Володимир Леонідович
Ніколаєнко Денис Володимирович

МЕТОДИЧНІ ВКАЗІВКИ ДО КУРСОВОЇ РОБОТИ
З ДИСЦИПЛІНИ
«ОБ'ЄКТНИЙ АНАЛІЗ І МОДЕЛЮВАННЯ СИСТЕМ»
(ДЛЯ СТУДЕНТІВ НАПРЯМУ
6.0305 «ЕКОНОМІЧНА КІБЕРНЕТИКА»)

Підписано до випуску 2012 р. Гарнітура Times New.
Умов. друк. арк. Зам. № .

Державний вищий навчальний заклад
«Донецький національний технічний університет»
Автомобільно-дорожній інститут
84646, м. Горлівка, вул. Кірова, 51
E-mail: druknf@rambler.ru

Редакційно-видавничий відділ

Свідоцтво про внесення до Державного реєстру видавців, виготовників
і розповсюджувачів видавничої продукції ДК № 2982 від 21.09. 2007 р.

ЗМІСТ

ВСТУП	5
1 ВИМОГИ ЩОДО ОФОРМЛЕННЯ ЗВІТУ	6
1.1 Структурні елементи звіту	6
1.2 Рубрикація файлу звіту	7
2 ПОРЯДОК ВИКОНАННЯ РОБОТИ.....	9
2.1 Кроки виконання роботи.....	9
2.2 Порядок захисту роботи.....	10
3 ВКАЗІВКИ ДО ВИКОНАННЯ РОБОТИ	11
3.1 Вказівки до підготовки Web доповіді	11
3.2 Вказівки до підготовки додатків	12
4 НЕОБХІДНІ ВІДОМОСТІ	15
5 ПРИКЛАД ОБ'ЄКТНОГО АНАЛІЗУ ТА МОДЕЛЮВАННЯ	22
5.1 Аналіз прецедентів (User Case Analysis)	23
5.2 Аналіз класів (Class Analysis).....	24
5.3 Аналіз послідовностей подій (Sequence Analysis)	28
5.4 Аналіз потоку подій (Flow Events Analysis)	30
5.4.1 Потік подій	30
5.4.2 Часовий потік подій.....	30
5.5 Аналіз станів об'єктів (State Analysis)	32
5.6 Аналіз проекції системи в простір станів	33
5.7 Аналіз кооперації (Collaboration Analysis).....	34
5.8 Аналіз діяльності (Activity Analysis)	35
6 ПРИКЛАД ОБ'ЄКТНОГО ПРЕКТУВАННЯ	36
6.1 Аналіз модулів (Component Analysis).....	36
6.2 Визначення специфікацій класу (Class Specifications)	40
6.3 Тестування програмної моделі (ProModel Test)	41

6.3.1 Тестування потоку подій (Flow Events Test).....	41
6.3.2 Тестування діяльності об'єктів (Activity Test)	41
6.4 Аналіз розміщення (Deployment Analysis)	42
СПИСОК РЕКОМЕНДОВАНОЇ ЛИТЕРАТУРИ.....	43

ВСТУП

Методичні вказівки до курсової роботи покликані ознайомити студентів з теоретичними основами об'єктного аналізу, моделювання й проектування програмних моделей економічних систем, допомогти в придбанні практичних навичок аналізу й розробки об'єктної моделі системи, проектуванні її програмної моделі, керованої виявленим в результаті об'єктного аналізу потоком подій.

З метою закріплення теоретичних положень дисципліни «Об'єктний аналіз і моделювання систем» і придбання практичних навичок застосування методології об'єктного аналізу, моделювання й проектування систем студенти виконують індивідуальні завдання курсової роботи.

На прикладі з наочної області студенти отримують перші навички об'єктного аналізу систем – побудови абстракцій класів, визначення властивостей, методів і подій об'єктів.

В процесі виконання завдань студенти засвоюють методи відладки алгоритмів, технології тестування систем, прийоми візуального проектування додатків.

В результаті виконання курсової роботи студенти повинні

знати:

- технологію та методику об'єктного аналізу, моделювання й побудови об'єктної моделі економічної системи;
- технологію та методику об'єктного проектування програмної моделі системи;

вміти:

- виконувати об'єктний аналіз і будувати об'єктну модель;
- виконувати об'єктне проектування;
- використовувати засоби об'єктно-орієнтованого кодування програмної моделі системи й алгоритмів поведінки об'єктів;
- будувати програмну оболонку системи.

Отримані навички при виконанні завдань курсової роботи надалі можуть бути використані в дипломному проектуванні при моделюванні економічних систем.

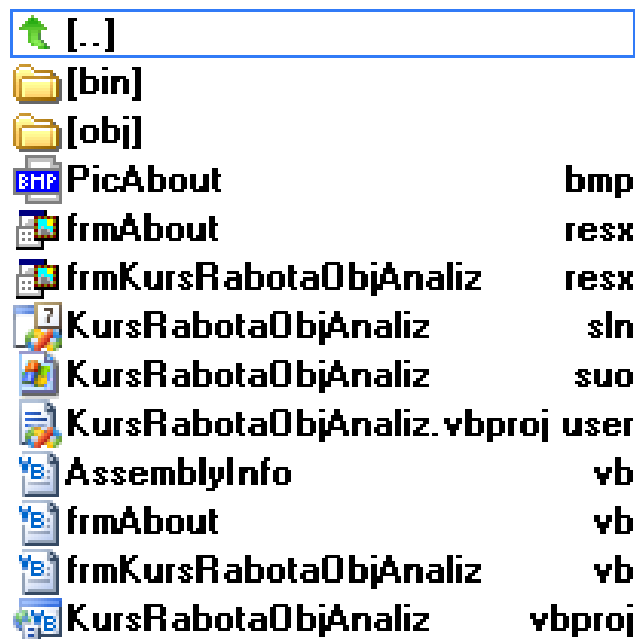
1 ВИМОГИ ЩОДО ОФОРМЛЕННЯ ЗВІТУ

1.1 Структурні елементи звіту

Звіт про виконані роботи складається з:

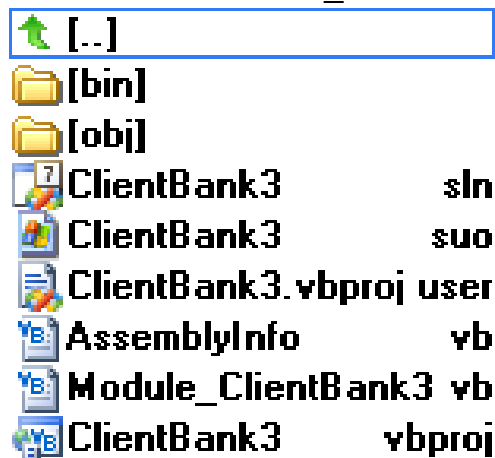
- 1) текстового файлу звіту;
ObjModel KursRabota Ek_00a Sidorov I.P. Report.doc;
- 2) проекту оболонки;

ObjModel KursRabota Ek_00a Sidorov I.P. Shell



- 3) проекту системи;

ObjModel KursRabota Ek_00a Sidorov I.P. SystemXXX



4) веб доповіді

ObjModel KursRabota Ek_00a Sidorov I.P. WebReport

↑ [..]	
[CSS]	
[IMG]	
Empty	htm
Index	htm
List	htm
Top	htm
Плакат 1 Постановка задач	mht
Плакат 2 Анализ прецедентов	mht
Плакат 3 Анализ классов	mht
Плакат 4 Анализ последовательностей событий	mht
Плакат 5 Модель потока событий	mht
Плакат 6 Анализ состояний объектов	mht
Плакат 7 Анализ кооперации	mht
Плакат 8 Анализ деятельности	mht
Плакат 9 Анализ состояний системы	mht
Плакат 10 Спецификации классов	mht
Плакат 11 Анализ модулей	mht
Плакат 12 Анализ топологии	mht
Плакат 13 Тестирование программной модели	mht
Плакат 14 Выводы	mht
Плакат 15 Список использованных источников	mht
Плакат 16 Приложение А ЗаконПоле	mht
Плакат 17 Приложение Б ИсходныйКод	mht
Плакат 18 Приложение В ТолковыйСловарь	mht
Плакат 19 Приложение В БанкЦитат	mht

Жорстка копія текстового файлу звіту складається з роздрукованих на принтері титульного аркуша та аркуша завдань.

1.2 Рубрикація файлу звіту

ВСТУП

1 ОГЛЯД МЕТОДІВ АНАЛІЗУ ТА МОДЕЛЮВАННЯ СИСТЕМ

1.1 Актуальність моделювання систем

1.2 Критика існуючих методів аналізу та моделювання

1.3 Недоліки існуючих методів аналізу й моделювання

1.4 Обґрунтування об'єктного аналізу і моделювання систем

2 ОБ'ЄКТНИЙ АНАЛІЗ І МОДЕЛЮВАННЯ СИСТЕМ

2.1 Аналіз прецедентів

2.2 Аналіз класів

2.3 Аналіз послідовностей подій

2.4 Аналіз потоку подій

2.4 Аналіз станів об'єктів

2.4 Аналіз проєкцій системи в простір станів

2.4 Аналіз кооперації

2.4 Аналіз діяльності

3 ОБ'ЄКТНЕ ПЕРЕКЛУЧАННЯ Й РОЗРОБКА ПРОГРАМНОЇ МОДЕЛІ СИСТЕМИ

3.1 Аналіз модулів

3.2 Аналіз топології

3.3 Визначення специфікацій класів

3.4 Тестування програмної моделі

3.4.1 Тестування потоку подій

3.4.2 Тестування діяльності об'єктів

ВИСНОВОК

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А Законодавче поле системи

ДОДАТОК Б Сенсорне поле системи

ДОДАТОК В Код програмної моделі системи

ДОДАТОК Г Словник термінів

ДОДАТОК Д Банк цитат

2 ПОРЯДОК ВИКОНАННЯ РОБОТИ

2.1 Кроки виконання роботи

Крок 1 «Отримання завдання».

Індивідуальні завдання відповідно до варіанту слід отримати у викладача.

Крок 2 «Побудова фрагмента дерева каталогів».

Всі файли слід розміщувати в каталозі Obmodel. Нижче наведений фрагмент дерева каталогів.

E:

```
|_Student
  |_Ek_00a
    |_Sidorov I.P.
      |_ObModel
        |_ObModel Ek_00_a Sidorov P.N. Report.doc
      |_ObjModel KursRabota Ek_00a Sidorov I.P. Shell
        |_ ...
      |_ObjModel KursRabota Ek_00a Sidorov I.P. SystemXXX
        |_ ...
      |_ObjModel KursRabota Ek_00a Sidorov I.P. WebReport
        |_ ...
      |_ObjModel KursRabota Ek_00a Sidorov I.P.
Report.doc
```

Крок 03 «Підготовка шаблону файлу звіту».

ObModel KursRabota Ek_00_a Sidorov P.N. Report.doc

Крок 04 «Аналіз прецедентів».

Крок 05 «Аналіз класів».

Крок 06 «Аналіз послідовностей подій».

Крок 07 «Аналіз потоку подій».

Крок 08 «Аналіз станів об'єктів».

Крок 09 «Проекція системи в простір станів».

Крок 10 «Аналіз кооперації».

Крок 11 «Аналіз діяльності».

Крок 12 «Аналіз модулів».

Крок 13 «Аналіз топології».

Крок 14 «Визначення специфікацій класів».

Крок 15 «Розробка програмної моделі».

Крок 16 «Тестування потоку подій».

Крок 17 «Тестування діяльності об'єктів».

Крок 18 «Розробка веб-сервера сторінки звіту».

Крок 19 «Оформлення файлу-звіту»

ObModel KursRabota Ek_00_a Sidorov P.N. Report.doc.

2.2 Порядок захисту роботи

Для захисту роботи студент надає викладачеві титульний лист і лист завдань, звіт про виконання курсової роботи, веб-сервер сторінки доповіді, проект системи.

Захист роботи включає:

- доповідь студента з використанням його веб-сервера сторінки;
- перевірку викладачем працездатності проектів;
- обговорення елементів об'єктної моделі системи;
- відповіді на питання.

3 ВКАЗІВКИ ДО ВИКОНАННЯ РОБОТИ

3.1 Вказівки до підготовки Web доповіді

Web доповідь – це сукупність веб-сервера файлів, що визначають структуру веб-сервера доповіді:

Index.htm

Top.htm

List.htm

Empty.htm

і файлів «Плакати доповіді»:

Плакат 1 Постанова завдань.mht

Плакат 2 Аналіз прецедентів.mht

Плакат 3 Аналіз класів.mht

Плакат 4 Аналіз послідовностей подій.mht

Плакат 5 Модель потоку подій.mht

Плакат 6 Аналіз станів об'єктів.mht

Плакат 7 Аналіз кооперації.mht

Плакат 8 Аналіз діяльності.mht

Плакат 9 Аналіз станів системи.mht

Плакат 10 Специфікацій класів.mht

Плакат 11 Аналіз модулів.mht

Плакат 12 Аналіз топології.mht

Плакат 13 Тестування програмної моделі.mht

Плакат 14 Висноаки.mht

Плакат 15 Список використаних джерел.mht

Плакат 16 Додаток А ЗаконПоле.mht

Плакат 17 Додаток Би ПочатковийКод.mht

Плакат 18 Додаток В ТолковийСловарь.mht

Плакат 19 Додаток В БанкЦитат.mht

Зауваження

При використанні вбудованого в MS Word конвертора mht файл можна побудувати за схемою – Кн_F12 – Длг_СохранДокумента – Tab в Стр_ИмяФайла – Вбр_ВебСтраницаВОдномФайле при активній Кн_Сохранить – Кн_Enter.

3.2 Вказівки до підготовки додатків

Додаток А

Початкові тексти законів, що регламентують поведінку економічної системи в законодавчому полі.

Зауваження

Початкові тексти законів – це список вигляду:

Номер [знак табуляції] [знак м'якого переносу]
 Цитата [знак табуляції] [знак м'якого переносу]
 Источник [знак кінця строки]

Наприклад:

1

«Акт (справка) документальної виїздної перевірки, визначеної статтями 77 і 78 нинішнього Кодексу, складений в двох екземплярах, підписується посадовими особами органу податкової служби, що проводила перевірку, і реєструється в органі державної податкової служби в термін п'яти днів з дня, наступного за днем закінчення встановленого дня проведення перевірки строку ...»

[Податковий кодекс України / уклад. В. Кузнецов. – Х.: Фактор, 2011. – 496 с., с 119]

Додаток Б

Лістинг початкового коду програмної моделі

Важливо представити його моноширинним шрифтом Couriernew і «драбинкою» для підкреслення логіки алгоритму.

Наприклад:

```
Sub Main()
    Dim myDay As New Date(2002, 1, 1), myD As String
    Dim i As Byte
    For i = 1 To 12
        myD = myDay.ToShortDateString
        MsgBox("Начался день - " & myD)
```

```

        oA.MainA(myD)

        MsgBox("Закончился день - " & myD)

        myDay = myDay.AddDays(1)

    Next i

End Sub

Public Class D

    Dim Runev1fromDforC As Boolean = False

    Public Event ev1fromDforC()

    Public Sub MainD(ByVal myD As String)

        If "10.01.02" <= myD And myD <= "11.01.02" Then

            If Not Runev1fromDforC Then

                AddHandler Me.ev1fromDforC, AddressOf oC.pev1fromDforC

                RaiseEvent ev1fromDforC()

                RemoveHandler Me.ev1fromDforC, AddressOf oC.pev1fromDforC

                Runev1fromDforC = True

            ...

        End Sub

        Public Sub pev1fromEforD()

        ...

    End Sub

End Class

```

Додаток В

Тематичний словник термінів наочної області, зібраний студентом в результаті виконання роботи.

Зауваження

Тематичний словник – це список вигляду:

Номер	[знак табуляції]	[знак м'якого переноса]
Термін	[знак табуляції]	[знак м'якого переноса]

Определение [знак табуляції] [знак м'якого переноса]
 Источник [знак конца строки]

Наприклад:

1

Об'єктна модель, object model

«Совокупность основополагающих принципов, лежащих в основе объектно-ориентированного проектирования; парадигма программирования, основанная на принципах абстрагирования, инкапсуляции, модульности, иерархичности, типизации, параллелизма и устойчивости»

[Г Буч Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 2000. – 569 с., ил.]

Додаток Г

«Банк цитат», тобто список цитат, що були використані в записці пояснення звіту.

Зауваження

«Банк цитат» – це список вигляду:

Номер [знак табуляції] [знак м'якого переноса]
 Цитата [знак табуляції] [знак м'якого переноса]
 Источник [знак конца строки]

Наприклад:

3

«... объектная декомпозиция существенно снижает риск при создании сложной программной системы, так как она развивается из меньших систем, в которых мы уже уверены»

[Г Буч Объектно-ориентированный анализ и проектирование с примерами приложений на С++, 2-е изд./Пер. с англ. – М.: «Издательство Бином», СПб.: «Невский диалект», 2000. – 569 с., ил., с. 35]

Зауваження

Запис списку – це дані, розділені службовим символом «Табуляція» (Tab), і що закінчується службовим символом – «Кінець рядка» (Enter).

Знак «м'якого перенесення» (Ctrl+Enter) використовується для переходу на новий рядок без завершення рядка.

4 НЕОБХІДНІ ВІДОМОСТІ

При побудові програмних моделей систем використовують алгоритмічне або об'єктне моделювання наочної області. В алгоритмічному підході будівельним елементом виступає процедура й основні зусилля направлені на декомпозицію складних алгоритмів, проте, отримані моделі важко адаптуються, що легко долається в об'єктному моделюванні, де будівельним елементом виступає об'єкт, що володіє станом і поведінкою.

Процес об'єктно-орієнтованого аналізу й проектування

Макропроцес

Мікропроцес

Макропроцес

Концептуалізація

Аналіз

Проектування

Еволюція

Супровід

Концептуалізація

Формулювання вимог до системи:

- для якого рівня підготовки користувача розрахована система;
- яким ергономічним критеріям повинна задовольняти;
- які завдання повинна дозволяти вирішувати;
- якими технічним засобами має бути забезпечена.

Аналіз

Визначення моделі поведінки системи:

- визначення складу класів;
- визначення взаємодій класів;
- визначення потоку подій.

Проектування

Побудова архітектури системи:

- визначення складу модулів;
- визначення процесів.

Еволюція
Побудова плану розвитку системи.

Супровід
Побудова плану управління еволюцією системи.

Мікропроцес

- Ідентифікація класів і об'єктів
- Ідентифікація семантики класів і об'єктів
- Ідентифікація стосунків між класами і об'єктами
- Специфікація інтерфейсу
- Реалізація класів і об'єктів
- Визначення класів і об'єктів

Ідентифікація класів і об'єктів

Ідентифікація класів і об'єктів – це визначення класів і об'єктів системи.

Ідентифікація семантики класів та об'єктів

Ідентифікація семантики класів та об'єктів – це визначення структури класів і об'єктів.

Ідентифікація стосунків між класами і об'єктами

Ідентифікація стосунків між класами і об'єктами – це визначення моделі потоку подій системи.

Специфікація інтерфейсу

Специфікація інтерфейсу – це визначення характеристик класів – відповідальності класів.

Реалізація класів і об'єктів

Реалізація класів і об'єктів – це кодування класів і об'єктів.

Визначення класів і об'єктів

Визначення класів і об'єктів – це одне з найскладніших завдань об'єктно-орієнтованого проектування.

В основі роботи за визначенням об'єктів і класів лежить класифікація.

Метою класифікації «є знаходження загальних властивостей об'єктів. Класифікуючи, ми об'єднуємо в одну групу об'єкти, що мають однако-ву будову або однакову поведінку» [1, с. 148].

Методи класифікації

1 Класична категоризація

2 Концептуальна кластеризація

3 Теорія прототипів

1 Класична категоризація

В одну категорію (клас) входять об'єкти з схожими властивостями.

Наприклад:

Стільці. Високі люди – і в першому і в другому випадку є довгі «ноги».

Наприклад:

Червоні автомобілі. Ця властивість значуща при побудові класу в завданні обліку продукції заводу й не має сенсу в завданні автоматизації роботи світлофора.

2 Концептуальна кластеризація

«При такому підході спочатку формуються концептуальні описи класів (кластерів об'єктів), а потім ми класифікуємо суть відповідно до цього опису» [1, с. 154].

Наприклад:

Любовна пісня. Це не властивість тому, що вона не вимірюється. Але це є поняття і, маючи це поняття, ми можемо відносити пісні в цю категорію (клас).

Цей підхід перекликається з теорією нечітких множин, де об'єкт може належати до різних множин з різним ступенем точності.

3 Теорія прототипів

Об'єкт відносять до деякого класу не тому, що у нього така ж властивість, але тому, що він якось схожий на прототип класу.

Наприклад:

Клас «дитячі ігри». У ігор немає загального – вони всі різні. Проте, нову гру ми вважатимемо за елемент класу «Гри», оскільки вона чимось нагадує вже відому гру – має прототип.

Суть об'єктно-орієнтованого аналізу

«В процесі аналізу ми *моделюємо* проблему, *виявляючи* класи і об'єкти, які складають словник проблемної області» [1, с. 156].

Результат моделювання й його «матеріальний образ» – віртуальна модель системи – не обов'язково яка не повинна повторювати реальне явище.

Суть об'єктно-орієнтованого проектування

«При об'єктно-орієнтованому проектуванні ми *винаходимо* абстракції та механізми, що забезпечують поведінку, потрібну моделлю» [1, с. 156]. «В процесі проектування розробник повинен придумати не тільки начинку класів, але й те, як об'єкти класів взаємодіятимуть один з одним» [1, с. 164].

Прийоми об'єктно-орієнтованого аналізу:

- I Класичний
- II Аналіз поведінки
- III Аналіз наочної області
- IV Аналіз варіантів
- V Неформальний опис
- VI Структурний аналіз

I Класичний

Виконуючи об'єктно-орієнтований аналіз, дивляться на відчутні елементи наочної області як на джерело класів.

Слід ввести клас і об'єкт (по Меллору) якщо:

Відчутні предмети.

Приклад:

Автомобілі, сенсори системи ...

Ролі

Приклад:

Вчитель, батько, політик.

Події

Приклади:

Посадка літака, запит ...

Взаємодія

Приклади:

Зустріч, кредитування ...

Слід ввести клас і об'єкт (по Росу) якщо:

Люди

«виконуючі функції».

Приклади:

Менеджер...

Місця

«де бувають люди, лежать предмети».

Приклади:

Склад...

Предмети

«відчутні речі та їх набори».

Приклади:

Клієнти банку.

Організації

«сукупність речей, що має мету й існування якої не залежить від окремої речі цієї сукупності».

Приклади:

Підприємства...

Концепції

«це принципи, ідеї. Вони не відчутні, але вимагають спостереження, оцінки».

Приклади:

Демократичність суспільства, злочинність в країні...

Події

«щось трапляється з чимось в заданий час або послідовно».

Приклади:

Датчики управління ...

Слід ввести клас і об'єкт (по Йордану) якщо:

Структури

«Стосунки «целое-часть», «общее-частное».

Інші системи

«то, з чим доводиться взаємодіяти».

Пристрої

«пристрої, з якими взаємодіє додаток».

Події

«події, які треба запам'ятати».

Ролі

«ролі виконують користувачі, що працюють з додатком».

Місця

«будівлі, офіси ..., істотні для роботи застосування».

Організаційні одиниці

«групи, до яких належать користувачі».

II Аналіз поведінки

Виконуючи об'єктно-орієнтований аналіз, дивляться на поведінку елементів наочної області як на джерело класів.

Тут вводиться поняття «Відповідальність об'єкту», під якою розуміють «... його знання й уміння. Відповідальність – це спосіб виразити мету об'єкту і його місце в системі. Відповідальність об'єкту – це є сукупність всіх послуг, які він може надати по всіх його контрактах» [1, с. 157].

«тобто, ми об'єднуємо разом ті об'єкти, які мають схожі відповідальності і будуємо ієрархію класів у якій кожен підклас, виконуючи зобов'язання суперкласу, привносить свої додаткові послуги» [1, с. 157].

«ініціатори й учасники, що грають істотні ролі, пізнаються як об'єкти і робляться відповідальними за ці ролі» [1, с. 157].

III Аналіз наочної області

«спроба виділити ті об'єкти, операції та зв'язки, які експерти наочної області вважають за найбільш важливі» [1, с. 158].

Експерт – це «часто просто користувачі системи, наприклад, інженер або диспетчер. Він не обов'язково має бути програмістом, але має бути близько знайомий з досліджуваною проблемою й розмовляти мовою цієї проблеми» [1, с. 158].

«зазвичай для початкового з'ясування проблеми достатньо короткої зустрічі експерта та розробників. Дивно, як мало інформації потрібно для

продуктивної роботи розробника. Проте, ми вважаємо надзвичайно корисними такі зустрічі протягом всієї розробки. Аналіз наочної області краще вести крок за кроком – трохи проаналізувати, потім трохи пропроекувати й так далі. Об'єктно-орієнтований аналіз – це ітеративний процес» [1, с. 158].

IV Аналіз варіантів

Цей підхід добре поєднується з класичним, поведінковим, аналізом наочної області. Його можна починати разом з аналізом вимог до системи.

«... експерти й розробники перераховують сценарії, найбільш істотні для роботи системи (поки не заглиблюючись в деталі). Потім вони ретельно опрацьовують сценарії, розкладаючи їх по кадрах. При цьому вони встановлюють, які об'єкти беруть участь в сценарії які обов'язки кожного об'єкту і як вони взаємодіють в термінах операцій» [1, с. 159].

«далі набір сценаріїв розширюється, щоб врахувати виняткові ситуації й вторинну поведінку» [1, с. 159].

Зауваження

В об'єктному аналізі корисне використання CRC карток.

(Class – Responsibilities – Collaborators Клас – Відповідальність – Учасники)

«Йдете» за сценарієм і заводите картку для нового «персонажа». Аналізуючи сценарій, уточнюється набір карток. Розкладаючи на столі картки по динаміці сценарію, бачимо потоки повідомлень. З погляду статички бачимо ієрархію класів.

V Неформальний опис

Ідея полягає в описі завдання на простій мові. А потім, підкресливши іменники – бачимо класи, підкресливши дієслова – бачимо методи.

VI Структурний аналіз

Виконавши його, можна виділити класи, але завжди слід пам'ятати, що діаграми потоків даних структурного аналізу не є модель наочної області, а є лише підставою для подальшої розробки механізмів поведінки системи.

5 ПРИКЛАД ОБ'ЄКТНОГО АНАЛІЗУ ТА МОДЕЛЮВАННЯ

Розглянемо систему «Приватний підприємець» («ChaP») – юридична особа, мале приватне підприємство, що працює по спрощеній системі оподаткування з ставкою 10 %, здійснює комерційну діяльність – комп'ютерне навчання. Припустимо, що кількість працівників дорівнює одному, форма податкової звітності є «Єдиний податок», орендується приміщення. Є у наявності договір з «Енергонаглядом» на користування електроенергією. Маємо словесні описи сценаріїв прецеденту «Подача відомостей про витрату електроенергії».

Основний потік подій варіанту використання «Подача відомостей про витрату електроенергії»:

- 1 «Визначення ліміту витрати на наступний період»;
- 2 «Перевірка боргу по оплаті за попередній період»;
- 3 «Визначення суми рахунку оплати електроенергії»;
- 4 «Оплата в банку спожитої електроенергії».

1 Альтернативний потік подій «Перевищення ліміту споживання електроенергії»:

- 1 «Визначення ліміту витрати електроенергії»;
- 2 «Визначення ліміту витрати при його перевищенні»;
- 3 «Перевірка боргу по оплаті електроенергії»;
- 4 «Визначення суми рахунку оплати електроенергії»;
- 5 «Оплата в банку спожитої електроенергії».

2 Альтернативний потік подій «Погашення боргу оплати електроенергії»:

- 1 «Визначення ліміту витрати на наступний період»;
- 2 «Перевірка боргу по оплаті за попередній період»;
- 3 «Оплата рахунку боргу по оплаті електроенергії»;
- 4 «Визначення суми рахунку оплати електроенергії»;
- 5 «Оплата в банку спожитої електроенергії».

Завдання 1

Використовуючи методологію об'єктного аналізу, побудувати об'єктну модель системи.

Завдання 2

Використовуючи методологію об'єктного проектування побудувати програмну модель системи.

5.1 Аналіз прецедентів (User Case Analysis)

Розуміючи прецедент або варіант використання як сервіс, який система повинна надавати зовнішнім акторам, відзначимо наступні прецеденти системи:

- 1) «Відкриття діяльності»;
- 2) «Оренда приміщення»;
- 3) «Реклама діяльності»;
- 4) «Наочна діяльність»;
- 5) «Здача виручки в банк»;
- 6) «Отримання зарплати в банці»;
- 7) «Звітність в міській службі статистики»;
- 8) «Звітність в міській податковій службі»;
- 9) «Звітність в центрі зайнятості»;
- 10) «Подача відомостей про витрату електроенергії»;
- 11) «Звітність в міській службі соціального страхування смерті й нещасних випадків»;
- 12) «Звітність в міській службі соціального страхування втрати працездатності»;
- 13) «Закриття діяльності».

Відповідна діаграма прецедентів системи «Приватний підприємець» представлена на рисунку 5.1, де як зовнішній актор виступають інші системи, що ініціюють прецеденти даної системи «Приватний підприємець».

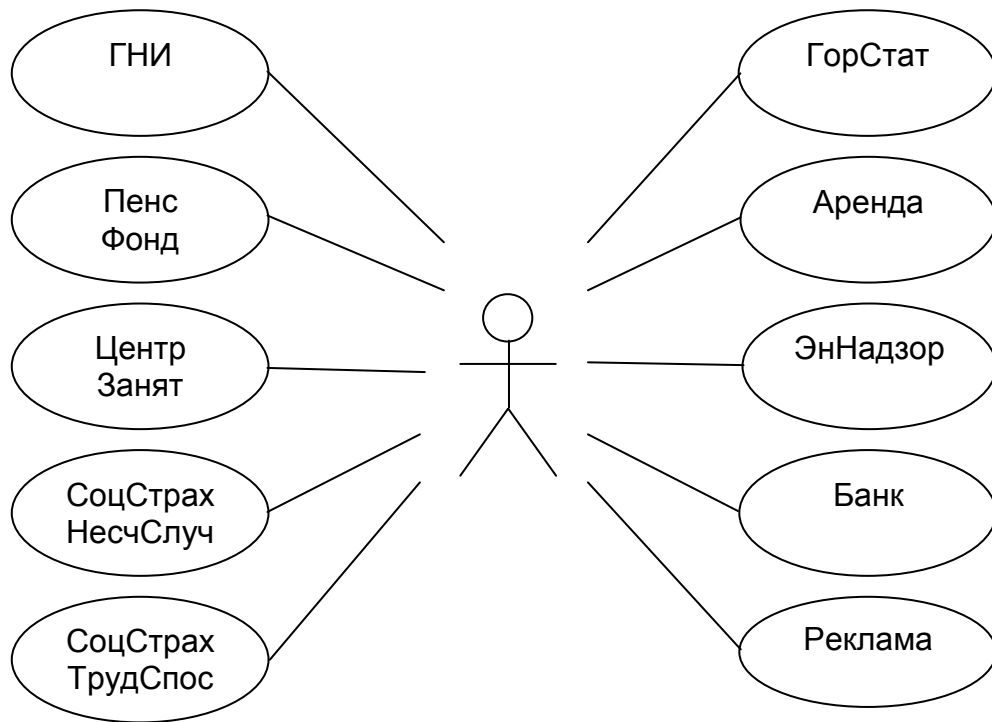


Рисунок 5.1 Діаграма прецедентів системи «Приватний підприємець»

Розглянемо детальніше прецедент «Подача відомостей про витрату електроенергії», оскільки цей прецедент є одним з тих, в якому представляє інтерес функціональність деяких об'єктів, що реалізують нетривіальну з обчислювальної точки зору діяльність, пов'язану з отриманням кількісних характеристик елементів потоку управління.

5.2 Аналіз класів (Class Analysis)

З метою виявлення класів системи застосуємо прийом об'єктно-орієнтованого аналізу – «Неформальний опис», ідея якого полягає в описі завдання на простій мові. А потім, «підкресливши іменники – бачимо класи, підкресливши дієслова – бачимо методи».

Типова (основна) поведінка об'єктів системи, даного прецеденту наступне.

«ChaP» подає «Номер договору» з «Енергонаглядом» і «Витрата електроенергії за звітний період» у «Службу визначення й перевірки «Ліміту» витрати електроенергії».

«Служба визначення й перевірки «Ліміту» витрати електроенергії», прийнявши «Номер договору» і «Витрата електроенергії», визначає чи

«Не перевищений Ліміт?» і якщо «ні», то визначає «Ліміт» на наступний звітний період і «ChaP» переходить в «Службу перевірки довга по оплаті за попередній період» для перевірки відсутності боргу.

Якщо «Ліміт» перевищений, то «Номер договору», «Витрата» і «Відсоток перевищення ліміту» подаються «Директорові» для ухвалення рішення. «Директор», ухваливши рішення, встановлює «Ліміт» для «ChaP».

Якщо є борг по оплаті за електроенергію, то «Номер договору», «Витрата» й «Величина боргу» подаються «Директорові» для ухвалення рішення. «Директор», ухваливши рішення, встановлює порядок погашення боргу. Далі «ChaP» звертається до «Інспектора» для визначення суми оплати за електроенергію.

Далі «ChaP» готує платіжне доручення й оплачує в банку споживання електроенергії.

Бачимо:

– іменник	«ChaP»	клас;
– дієслово	«Подає ...»	метод.

Бачимо:

– іменник	«Служба визначення й перевірки Ліміту»	клас;
– дієслово	«Визначає й перевіряє ліміт»	метод.

Бачимо:

– іменник	«Директор»	клас;
– дієслово	«Ухвалює рішення про Ліміт»	метод.

Бачимо:

– іменник	«Директор»	клас;
– дієслово	«Встановлює Ліміт»	метод.

Замечание

Зараз на попередньому етапі аналізу бачиться метод «Встановлює Ліміт», але не обов'язково, що це стане ще одним методом.

Бачимо:

– іменник	«Служба перевірки боргу»	клас;
– дієслово	«Перевіряє відсутність боргу»	метод.

Бачимо:

– іменник	«Директор»	клас;
– дієслово	«Встановлює порядок погашення боргу»	метод

Бачимо:

- | | | |
|------------|------------------------|-------|
| – іменник | «Інспектор» | клас; |
| – дієслово | «Визначає суму оплати» | метод |

Бачимо:

- | | | |
|------------|----------------------------|-------|
| – іменник | «ChaP» | клас; |
| – дієслово | «Готує платіжне доручення» | метод |

Бачимо:

- | | | |
|------------|-------------------|-------|
| – іменник | «ChaP» | клас; |
| – дієслово | «Оплачує в банку» | метод |

Бачимо:

- | | | |
|------------|------------------------------|-------|
| – іменник | «Банк» | клас; |
| – дієслово | «Приймає платіжне доручення» | метод |

Зауваження

Очевидно, що витрата електроенергії, номер договору, номер лічильника «ChaP» були заздалегідь приготовані, отже.

Бачимо:

- | | | |
|------------|--------------------------------------|--------|
| – іменник | «ChaP» | клас; |
| – дієслово | «Знімає показники електролічильника» | метод. |

Таким чином, визначені класи: «ЕлЛічильник», «ChaP», «Ліміт», «Борг», «Директор», «Інспектор», «Банк».

Бачимо стосунки:

– «ЕлЛічильник» – «ChaP» – агрегація,
оскільки електролічильник може виконувати свої функції поза належністю підприємцеві;

– «Ліміт», «Директор» – композиція,
оскільки клас «Ліміт» не може виконувати свої функції поза підприємством «ЕнергоНагляд»;

– «Долг», «Директор» – композиція,
оскільки клас «Борг» не може виконувати свої функції поза підприємством «ЕнергоНагляд»;

– «Інспектор», «Директор» – композиція,
оскільки клас «Інспектор» не може виконувати свої функції поза підприємством «ЕнергоНагляд»;

– «ChaP», «Банк», «ЕнергоНагляд» – асоціація, оскільки асоційовані однією ідеєю (завданням).

Таким чином, проекція системи в простір класів матиме вид наступної діаграми класів (рис. 5.2).

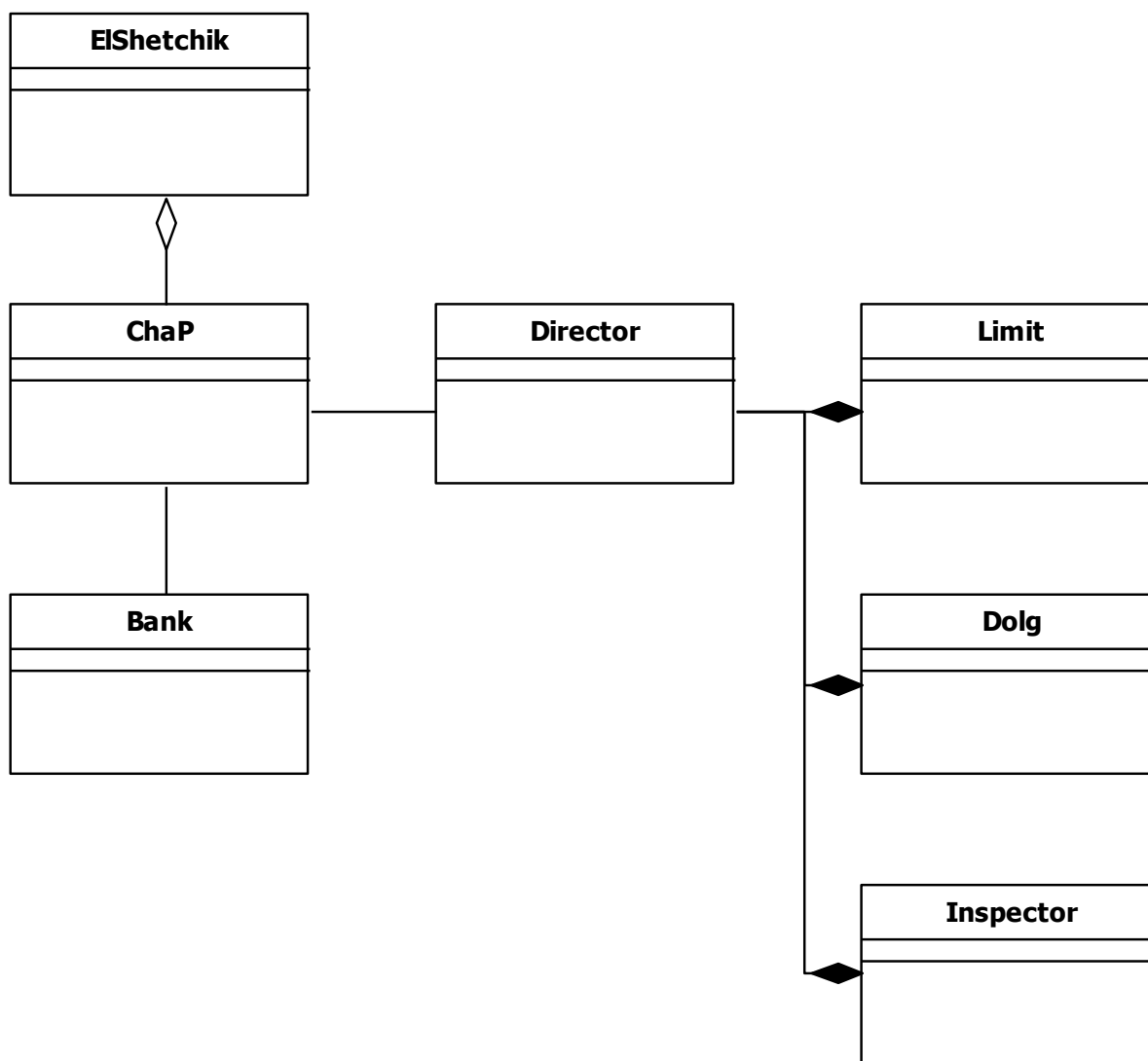


Рисунок 5.2 Діаграма класів реального сценарію

Зауваження

Робити акцент на атрибутах класів на цьому етапі побудови об'єктної моделі немає необхідності. Тому достатньо подати графічні примітиви, що представляють класи без вказівки властивостей, методів і подій.

Зауваження

Особливістю віртуального представлення системи є те, що структурна й функціональна модель системи не тотожна її реальному уявленню. Інакше переваги віртуалізації явищ і суті зводяться до нуля. Тоді проекція системи в простір класів має наступний вигляд (рис. 5.3).

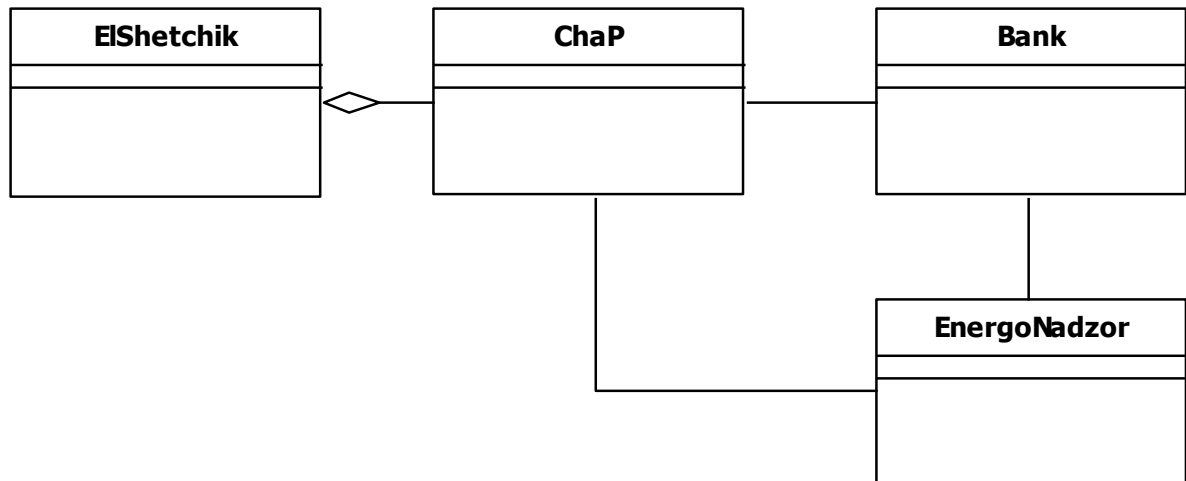


Рисунок 5.3 Діаграма класів віртуального сценарію

Аналіз діаграм показує, що у разі віртуального сценарію немає необхідності фізичного представлення таких об'єктів як «Ліміт», «Борг», «Інспектор», «Директор», а достатньо їх функціональність представити одним об'єктом «ЕнергоНагляд», що приводить до спрощення структури системи, до підвищення її надійності й стійкості в роботі, а також до подешевіння системи, за рахунок виключення деяких суб'єктів управління.

5.3 Аналіз послідовностей подій (Sequence Analysis)

Графічне представлення результату аналізу послідовностей подій – діаграма послідовності подій (Sequence Diagram) робиться для кожного сценарію прецеденту. Діаграми дають уявлення про послідовність реалізації подій у часі (правда, без їх масштабування – прив'язки до дат «народження» й «смерті» подій) і дозволяють виявити логічну послідовність подій в сценарії (рис. 5.4, 5.5).

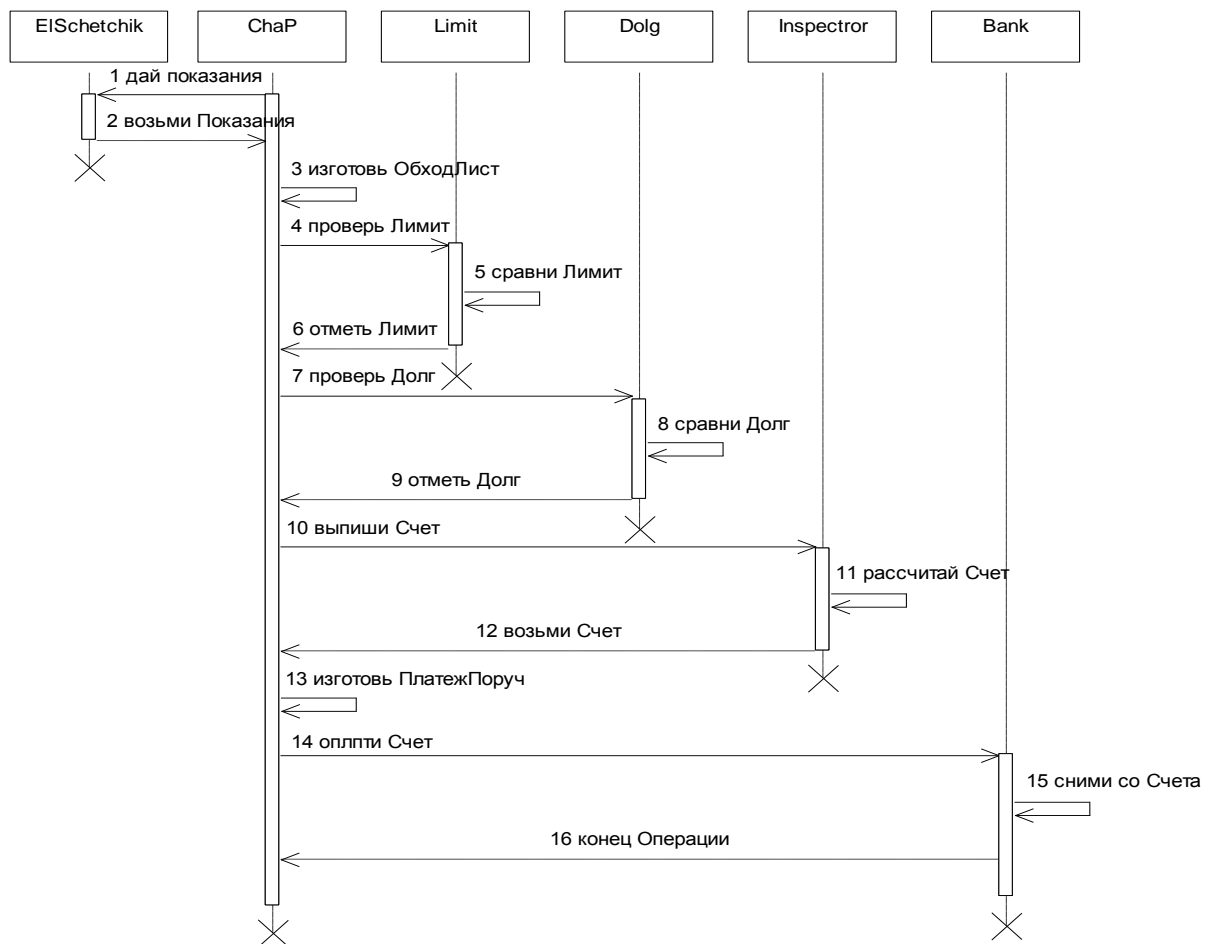


Рисунок 5.4 Последовательность событий реального сценария

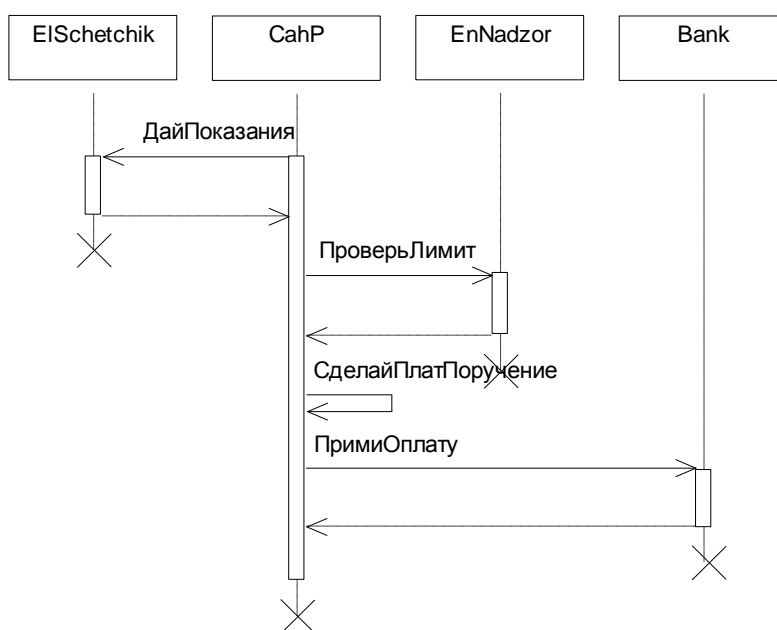


Рисунок 5.5 Последовательность событий виртуального сценария

5.4 Аналіз потоку подій (Flow Events Analysis)

Аналіз послідовності подій доповнює аналіз потоку подій, графічним представленням якого є «Діаграма потоку подій» і «Тимчасова діаграма потоку подій».

Діаграма потоку подій відображає потік подій, який лежить в основі взаємодії суті системи, але на відміну від діаграми послідовності подій не переобтяжена графічними примітивами й наочніше представляє потік подій сценарію.

5.4.1 Потік подій

Для даного прикладу діаграма потоку подій реального сценарію матиме вигляд приведений нижче (рис. 5.6).

5.4.2 Часовий потік подій

Тимчасова діаграма потоку подій використовується для тимчасового масштабування потоку подій, доповнює аналіз потоку подій масштабуванням подій в часі, дозволяє, зокрема, обчислити вартісну характеристику сценарію прецеденту.

Для даного прикладу діаграма потоку подій реального сценарію матиме вигляд наведений нижче (рис. 5.6).

Аналіз діаграм дозволяє визначити активність об'єктів реального сценарію:

– ElChetchik	10 хв.;
– ChaP	10 хв. + 20 хв. = 30 хв.;
– Limit	15 хв.;
– Dolg	15 хв.;
– Inspektor	20 хв.;
– Bank	25 хв..

У разі віртуального сценарію активність об'єктів складе:

– ElChetchik	5 тік.;
– ChaP	10 тік.;
– EnNadzor	5 тік.;
– Bank	5 тік..

Сумарний час активності об'єктів реального сценарію складе

$$10 \text{ хв.} + 30 \text{ хв.} + 15 \text{ хв.} + 20 \text{ хв.} + 25 \text{ хв.} = 100 \text{ хв.}$$

Сумарний час активності об'єктів віртуального сценарію складе

$$5 \text{ тік.} + 10 \text{ тік.} + 5 \text{ тік.} + 5 \text{ тік.} = 25 \text{ тік.}$$

Зауваження

У розглянутому прикладі час тік прийнято умовно. Точне визначення часу тік припускає проведення комп'ютерних експериментів за визначенням активності об'єктів.

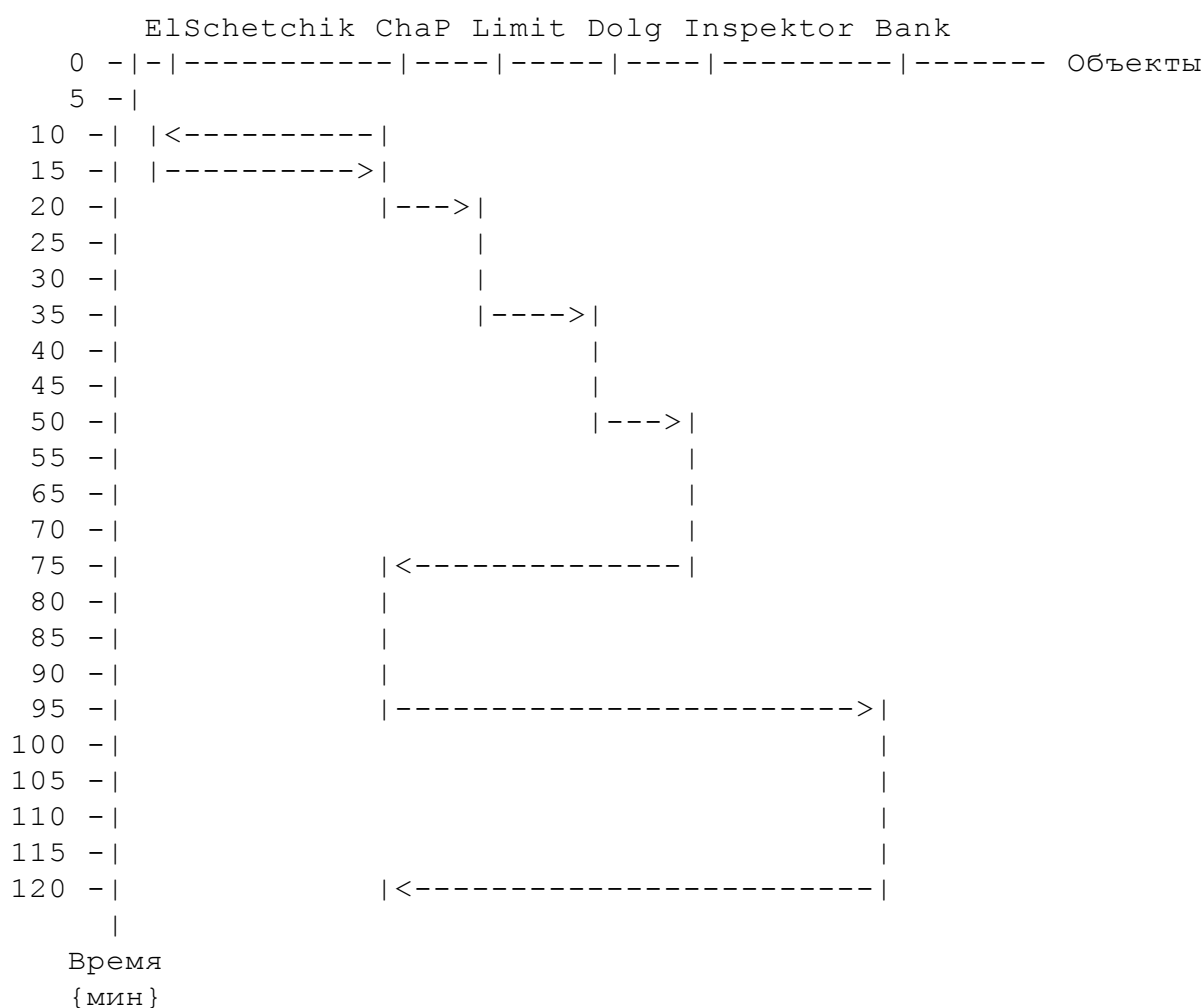


Рисунок 5.6 Тимчасова діаграма потоку подій реального сценарію

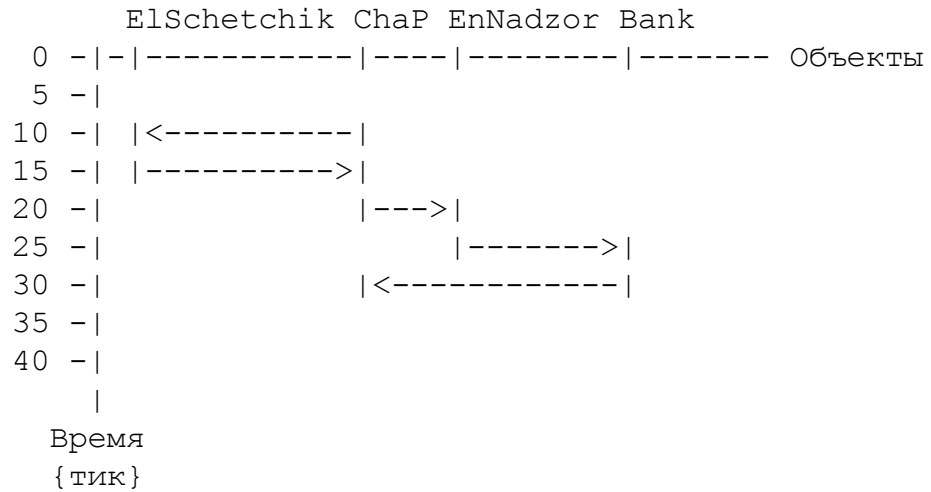


Рисунок 5.7 Тимчасова діаграма потоку подій віртуального сценарію

5.5 Аналіз станів об'єктів (State Analysis)

Для тих об'єктів, поведінка яких відрізняється великою кількістю станів, роблять аналіз станів об'єктів з метою зрозуміти динаміку класу й реалізувати виявлену логіку при генеруванні коду моделі (рис. 5.8).

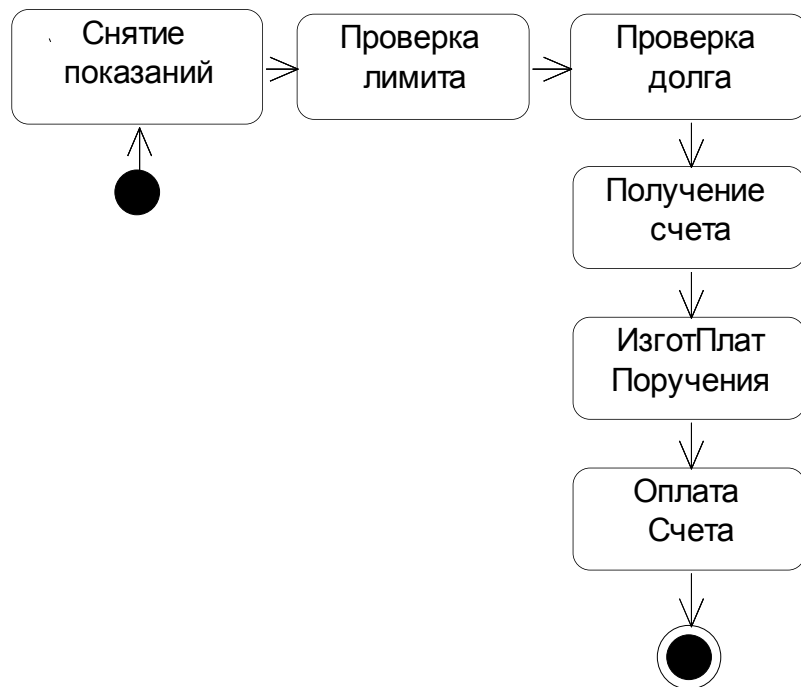


Рисунок 5.8 Станів об'єкту «Приватний підприємець»

5.6 Аналіз проєкції системи в простір станів

На завершення об'єктного аналізу системи слід виконати аналіз станів системи як станів об'єктів системи при виконанні сценарію, що дозволяє перейти до формалізації завдання побудови математичних моделей потоку управління системи у вигляді рівнянь алгебри кінцевих предикатів. На рисунку 5.9 наведена діаграма проєкцій системи в простір станів для основного сценарію.

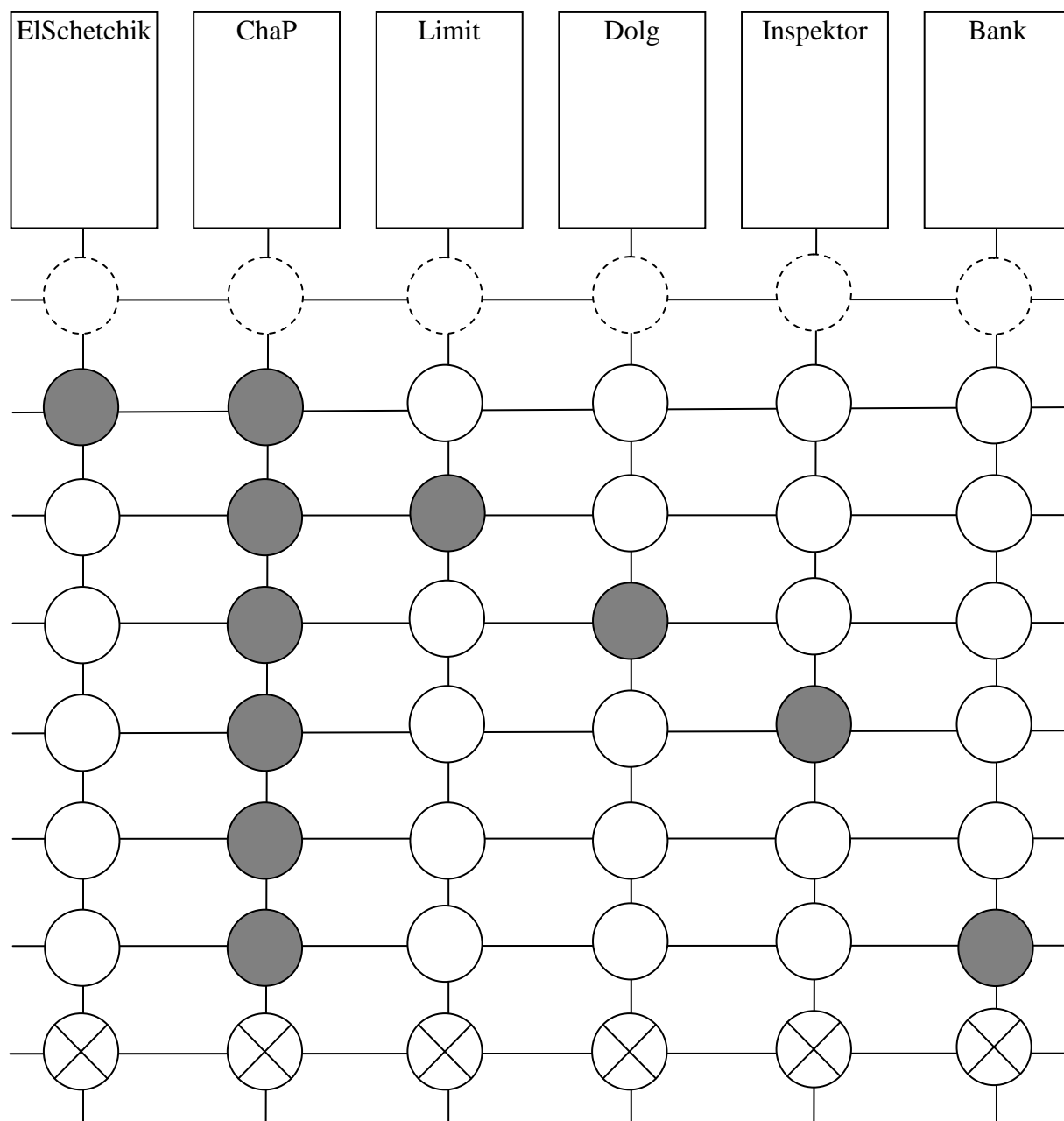


Рисунок 5.9 Діаграма проєкцій системи в простір станів

5.7 Аналіз кооперації (Collaboration Analysis)

На діаграмах кооперації (collaboration diagram) відображається та ж інформація, що і на діаграмах послідовностей подій. Відмінність полягає в цілях відображення. Якщо у разі діаграм послідовностей подій акцент робиться на тимчасовій послідовності подій, то у разі діаграм кооперації, акцент робиться на відображенні об'єктів і подій що їх зв'язують незалежно від їх тимчасової послідовності, але з метою відображення факту їх існування, що дозволяє архітекторові системи легко виявити переобтяжені обробкою подій об'єкти й перепланувати розподіл процесів системи. Кооперацію розглядають двох видів – кооперації рівня специфікації (використовуються рідко) й кооперації рівня екземплярів (використовуються часто). Аналіз кооперації рівня екземплярів (раніше називалася діаграма об'єктів) переслідує мета показати об'єкти, зв'язки й повідомлення між ними у взаємодії, відображаючи міжоб'єктний потік подій сценарію прецеденту (рис. 5.10).



Рисунок 5.10 Кооперація об'єктів сценарію «Подача відомостей в енергонагляд»

5.8 Аналіз діяльності (Activity Analysis)

Діаграми діяльності корисні в описі поведінки, що включає велику кількість паралельних процесів. Гідністю діаграм діяльності є підтримка паралелізму. Завдяки цьому вони є могутнім засобом моделювання потоків робіт і, по суті, паралельного програмування. Нижче показана діяльність об'єкту «Приватний підприємець» в прецеденті «Квартальний звіт в податковій інспекції» (рис. 5.11).

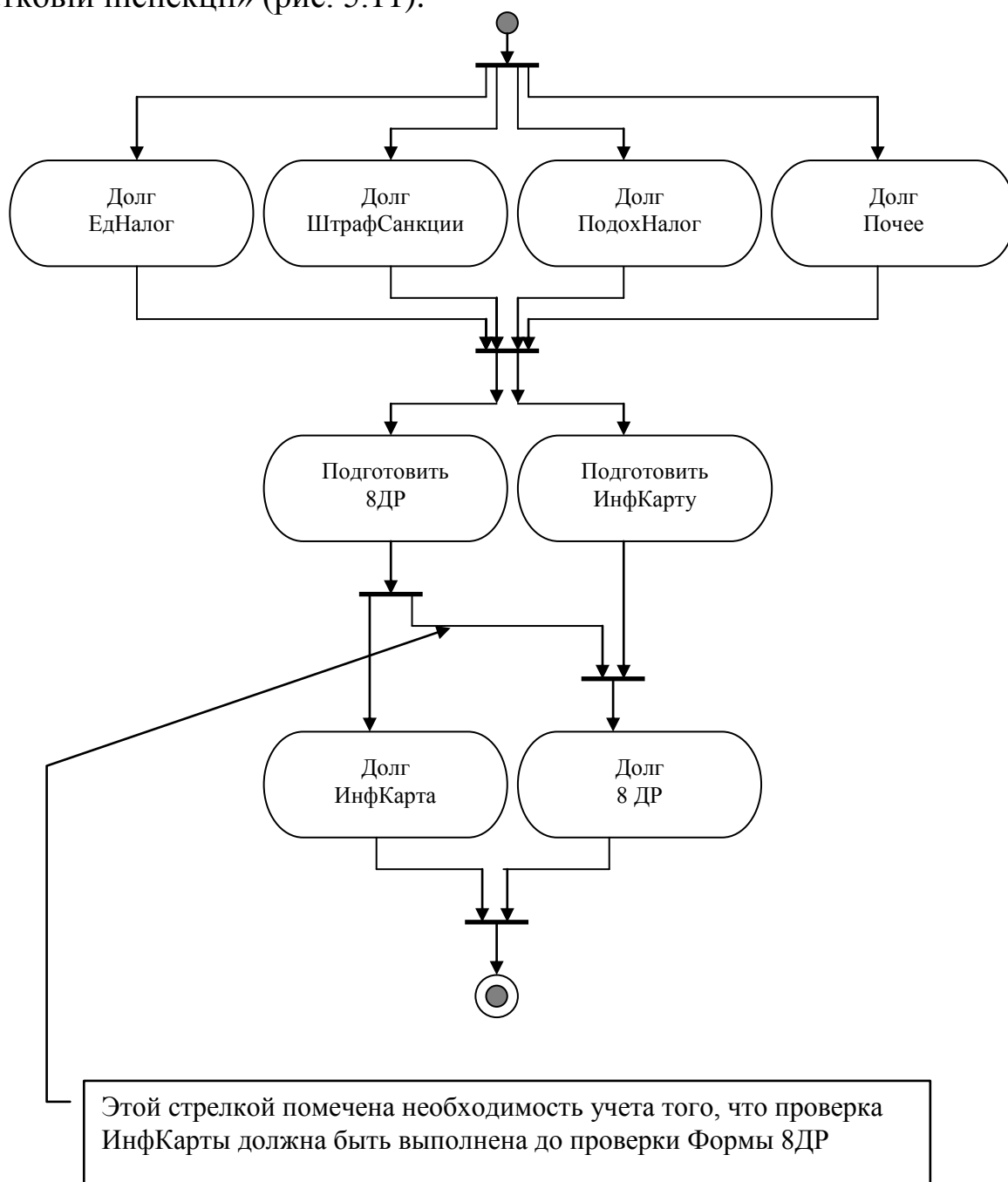
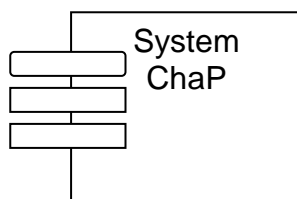


Рисунок 5.11 Діяльність об'єкту «ChaP»

6 ПРИКЛАД ОБ'ЄКТНОГО ПРЕРКТУВАННЯ

6.1 Аналіз модулів (Component Analysis)

Припустимо, що всі класи об'єктної моделі системи розташовуються на одному листі модуля `ModuleSystemChaP`. Припустимо в проекті системи відсутні файли вбудованої системи допомоги, бібліотеки коду і т.д. Тоді має місце однокомпонентна система наступного вигляду (рис. 5.12).



Якщо припустити, що кожному класу відповідає окремий компонент (модуль), що в достатньо великих проектах є природним, тоді діаграма модулів набере вигляду.

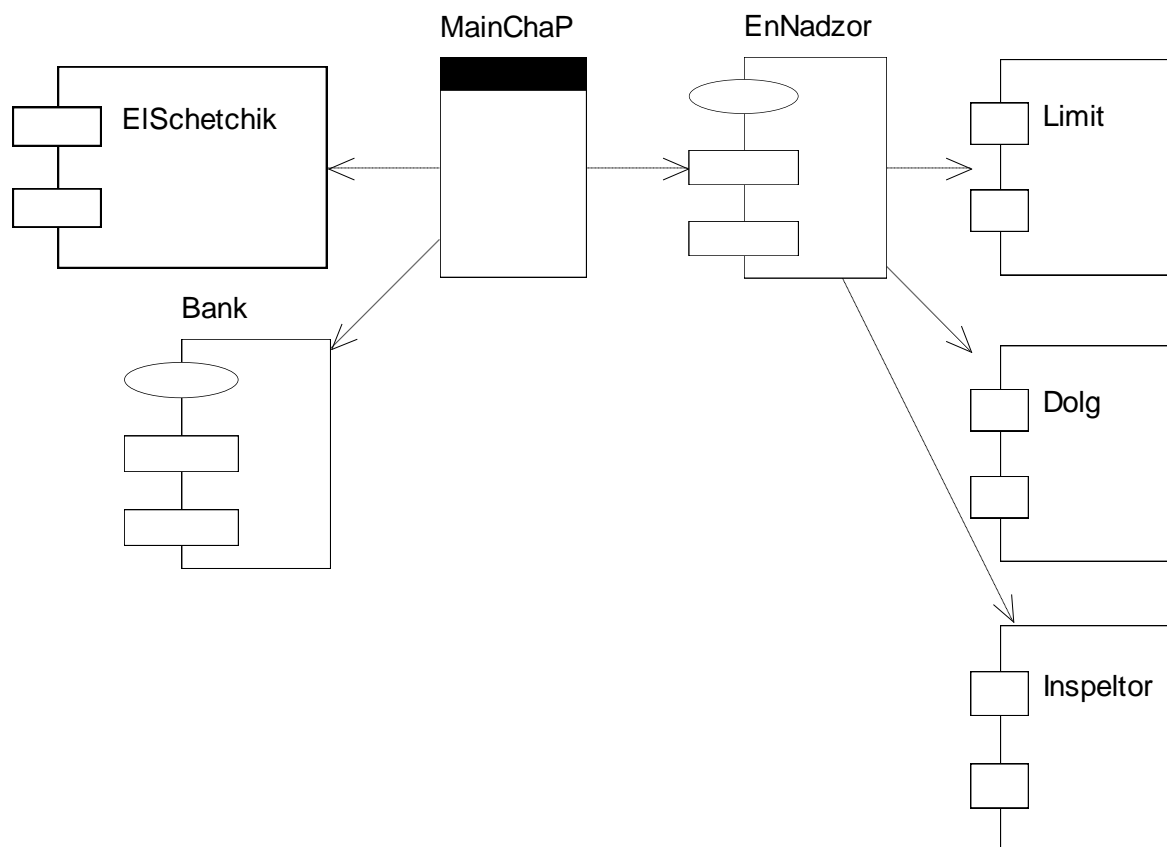


Рисунок 512 Модулі системи «Приватний підприємець»

Всі компоненти є здійсненими файлами. Всі стосунки є стосунками залежності, оскільки зміни, що відбуваються в компоненті «ChaP», впливають на роботу інших компонентів.

Розглянемо словесний опис послідовностей основного й альтернативних сценаріїв прецеденту «звітність в службі енергонагляду».

Типова (основна) поведінка об'єктів системи, даного прецеденту наступне – «ChaP» подає «Номер договору» із енергонаглядом і «Витратою електроенергії» за звітний період в «Службу визначення й перевірки «Ліміту» витрати електроенергії».

«Служба визначення й перевірки «Ліміту» витрати електроенергії», прийнявши «Номер договору» і «Витрату електроенергії» визначає чи «Не перевищений «Ліміт?» і якщо «ні», то визначає «Ліміт» на наступний звітний період і «ChaP» переходить до визначення «Боргу сплати електроенергії».

Якщо «Ліміт» перевищений, то «Номер договору», «Витрата» й «Відсоток перевищення ліміту» подаються «Директорові» для ухвалення рішення.

«Директор», ухваливши рішення, встановлює «Ліміт» для «ChaP».

«ChaP», отримавши «Ліміт» від «Директора», знову звертається в «Службу визначення і перевірки ліміту витрати електроенергії» де фіксується «Ліміт» і «ChaP» переходить до визначення «Боргу по оплаті електроенергії».

Припустимо, що «Номер договору», «Витрата» й «Ліміт» будуть властивостями об'єкту «ChaP».

Припустимо, що «Служба визначення й перевірки ліміту витрати електроенергії» має метод «Визначення перевищення «Ліміту» і «Визначення «Ліміту» на новий звітний період».

Припустимо, що «Директор» має метод «Визначення «Ліміту» на новий звітний період у разі, коли перевищений Ліміт за звітний період».

Припустимо, що клас «ChaP», аналізуючи час, відповідний для виконання звіту в службі енергонагляду («Чи прийшов час?») і факт невиконання події («Чи не оброблялася подія раніше?»), генерує подію «Перевірка «Ліміту» для об'єкту «Ліміт».

Об'єкт «Ліміт», перехопивши подію «Перевірка ліміту», своїм методом обробки події встановлює нове значення властивості «Ліміт» об'єкту «ChaP».

У разі перевищення «Ліміту», генерує подію «Перевищення ліміту» для об'єкту «Директор», який, перехопивши цю подію, своїм методом обробки події визначає новий «Ліміт» і повертає управління процедурі обробки події «Перевірка ліміту», де об'єкт «Ліміт» встановлює нове значення властивості «Ліміт» об'єкту «ChaP».

Далі, аналогічно, об'єкт «ChaP» перевіряє чи «Прийшов Час?» і чи «Не оброблялася подія раніше?» для події «Перевірка Довга».

Можна сказати, що в системі реалізуються наступні подіїв примітив:

```
ChaP  ENLimit
----->
```

```
ChaP  ENLimit  Director
----->
```

Припустимо, що об'єкт «ChaP», в процедурі з ім'ям MainИмяКласса, що приймає час, як аргумент, перевіряють чи «Не прийшов час обробляти подію?» і чи «Не оброблялася подія раніше?».

```
Public Sub MainChaP(ByVal D As String)
End Sub
```

Код перевірки умов настання події.

```
If "15.03.2001"<= D AND D <= "18.03.2001" Then
  If NOT RANevLimitfromChaPforLimit Then

    RaiseEvent evLimitfromChaPforLimit()
    RANevLimitfromChaPforLimit = True

  End If
End If
```


У розділі **General Declaration** листа модуля оголошуємо логічну змінну

```
Dim RANevLimitfromChaPforLimit As Boolean = False
```

Припустимо, що цикл `for – next` кожним своїм кроком (циклом) представляє відрізок часу, після закінчення якого система переходить в новий свій стан.

Припустимо, що цей цикл знаходиться в **Main** процедурі модуля, і відповідний фрагмент коду має вигляд

```
Dim i As Integer
For i = 1 To LifeDays
    D = Day.ToShortDateString
    MsgBox("День -- " & D & " -- почався!")
    \-----
    oChaP.MainChaP(D)
    ...
    \-----
    MsgBox("День -- " & D & " -- закончився!")
    Day = Day.AddDays(1)
Next i
```

В **General Declaration** модуля треба також зробити наступні оголошення

```
Dim Day As New Date(2001, 3, 14)
Dim oOChaP as ChaP
ChaP = new ChaP("Name ChaP")
або
Dim oChaP as = new ChaP("Name ChaP")
```

Таким чином, на початку етапу проектування, ми точніше формулюємо вимоги до атрибутів класів.

6.2 Визначення специфікацій класу (Class Specifications)

Визначення специфікацій класів – це процес документування атрибутів класів, метою якого є підготовка до кодування програмної моделі системи.

У прикладі «ChaP» специфікація класу «ChaP» матиме вигляд:

Клас ChaP

Поля

mName

Закрите поле властивості Name.

String типа.

Властивості.

Name

Ім'я класу

ReadOnly. String типа.

ENLimit

Кількість електроенергії, яку можна було витратити в звітному періоді.

ReadWrite. Single типа.

ENDolg

Кількість грошей, які треба було заплатити за електроенергію у минулому звітному періоді. ReadWrite. Decimal типа.

ENCount

Кількість грошей, які треба заплатити за електроенергію в поточному звітному періоді.

ReadWrite. Decimal типа.

Методи

MainChaP((ByVal D As String)

Використовується для перевірки умов генерації події.

D – вхідний параметр

Події

evLimitfromChaPforLimit

Викликає метод визначення «Ліміту» об'єкту ENLimit

Аналогічно визначаються специфікації інших класів.

Після визначення специфікацій класів будується перша реалізація програмної моделі. При необхідності, здійснюється друга ітерація об'єктного аналізу, проектування й кодування програмної моделі системи.

6.3 Тестування програмної моделі (ProModel Test)

6.3.1 Тестування потоку подій (Flow Events Test)

Після того, як виписаний код, що забезпечує генерацію подій відповідно до діаграмній послідовності подій сценаріїв і виписаний код процедур, що реагують на події системи ще не виписаний код що представляє функціональність об'єктів, виконується тест потоку подій. При цьому слід добиватися появи всіх передбачених повідомлень.

6.3.2 Тестування діяльності об'єктів (Activity Test)

Тестувати має сенс лише ті методи об'єктів, які з обчислювальної точки зору є складними тобто реалізують розрахунки по складних математичних залежностях, що представляють моделі деяких явищ.

Розглянемо об'єкт oENLimit, який перевіряє перевищення ліміту споживання електроенергії за звітний період. Суть його діяльності полягає в порівнянні знов поданих (поточних) свідчень електролічильника зі свідченнями за попередній звітний період, наявними в базі даних. З обчислювальної точки зору об'єкт реалізує не складну діяльність, примітивну функціональність, тривіальний інтелект, що полягає в реалізації обчислювальної структури типу «якщо А, то Б, інакше С». Проте, структура тесту, що в цьому простому випадку, що в найскладнішому буде однаковою.

Припустимо, що свідчення електролічильника за попередній звітний період є 214 квт.г., а подані свідчення електролічильника є 154 квт.г. Тоді тест однієї та іншої гілки обчислювального процесу матиме вигляд:

Тест 1

Якщо

$$\text{Raschod} = 154$$

То очікується відповідь:

«Ліміт не перевищений»

Дійсно:

«Ліміт не перевищений»

Тест 2

Якщо

Raschod = 334

То очікується відповідь:

«Ліміт перевищений»

Дійсно:

«Ліміт перевищений»

Зауваження

При тестуванні діяльності об'єкту, як деякого алгоритму, слід підбрати тестові набори даних для граничних і «типового» випадків, пам'ятаючи, що тестування не має яких-небудь вичерпних рекомендацій.

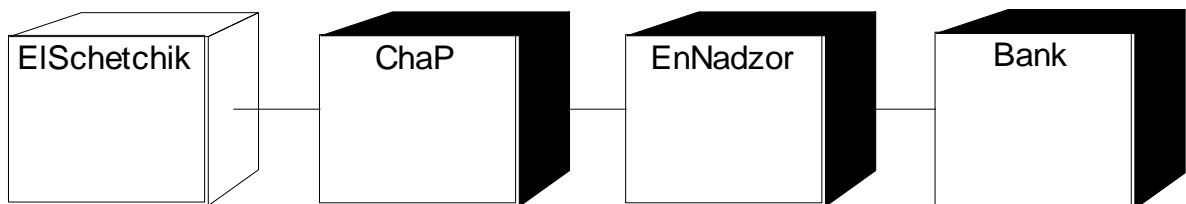
6.4 Аналіз розміщення (Deployment Analysis)

Діаграма розміщення використовується менеджером проекту, користувачами, архітектором системи і експлуатаційним персоналом, щоб зрозуміти фізичне розміщення системи і розташування її окремих підсистем.

Кожен вузол на діаграмі розміщення є деяким типом обчислювального пристрою – в більшості випадків частина апаратури. Ця апаратура може бути простим пристроєм, датчиком, процесором.

Діаграма розміщення показує фізичне розташування мережі (її топологію) й місцезнаходження в ній різних компонентів (локалізацію елементів).

Діаграма розміщення відображає фізичні взаємозв'язки між програмними й апаратними компонентами системи. Показує процеси та зв'язки між пристроями системи.



СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Г. Буч. – 2-е изд. пер. с англ. – М.: Издательство Бином, СПб.: Невский диалект, 2000. – 560 с.
2. Буч Г. Язык UML. Руководство пользователя. 2-е изд.: пер. с англ. Мухин Н. / Г. Буч, Д. Рамбо, И. Якобсон. – М.: ДМК Пресс, 2006. – 496 с.
3. Леоненков А. В. Самоучитель UML / А. В. Леоненков. – 2-е изд., перераб. и доп. – СПб.: БХВ – Петербург, 2004. – 432 с.
4. Эдвард Й. Объектно-ориентированный анализ и проектирование систем / Й. Эдвард, А. К. Аргила. – М.: Издательство Лори, 2007. – 284 с.
5. Ларман К. Применение UML 2.0 и шаблонов проектирования / К. Ларман. – М.: Издательский дом Вильямс, 2007. – 736 с.
6. Трофимов С. А. CASE-технологии. Практическая работа в Rational Rose / С. А. Трофимов. – М.: Издательство БИНОМ, 2002. – 288 с.
7. CASE-технологии. Практикум. / Д. Э. Федотова, Ю. Д. Семенов, К. Н. Чижик. – М.: Издательство БИНОМ, 2005. – 160 с.
8. Фаулер М. UML. Основы. Краткое руководство по унифицированному языку моделирования / М. Фаулер, С. Кендалл. – М.: Символ-Плюс, 2002. – 192 с.
9. Максимчук Р. А. UML для простых смертных / Р. А. Максимчук, Э. Д. Нейбург. – М.: Издательство Лори, 2008. – 304 с.
10. Путилин А. Б. Компонентное моделирование и программирование на языке UML: практическое руководство по проектированию информационно-измерительных систем / А. Б. Путилин, Е. А. Юрагов. – М.: Издательство ИТ Пресс, 2005. – 664 с.
11. Киммел П. UML. Основы визуального анализа и проектирования / П. Киммел. – Издательская группа АСТ, 2008. – 272 с.
12. Боггс У. UML и Rational Rose 2002 / У. Боггс, М. Боггс. – М.: Издательство Лори, 2004. – 509 с.