

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ДИСЦИПЛИНЫ "ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C#"

Евгений Забудский

Аннотация: Разработан учебно-методический комплекс дисциплины "Объектно-ориентированный анализ и программирование на языке C#", который включает программу, нормативно-справочные материалы, тексты лекций и практических занятий, компьютерные модели реальных и концептуальных систем и др. Комплекс размещен в Интернете.

Ключевые слова: объект, анализ, проектирование, программирование, C#, .NET, комплекс.

ACM Classification Keywords: D.1.5 Object-oriented Programming

Conference: The paper is selected from Third International Conference "Modern (e-) Learning" MeL 2008, Varna, Bulgaria, June-July 2008

Введение

Объектно-ориентированная парадигма получает признание во многих областях программирования: от создания графического интерфейса пользователя (Graphic User Interface) до сетевого программирования и разработки компьютерных моделей реальных и концептуальных бизнес-систем. Платформа .NET Framework, разработанная Microsoft, поддерживает все объектно-ориентированные концепции, включая: классификацию, инкапсуляцию, наследование и полиморфизм. Новинкой платформы .NET Framework, созданной на основе объектно-ориентированного языка C#, является исключение языковых барьеров, что способствует дальнейшему распространению объектно-ориентированной парадигмы. В докладе представлен учебно-методический комплекс дисциплины с общепринятым названием – Object-Oriented Programming; комплекс размещен на Web-странице автора. Использование комплекса в учебном процессе преподавателями и студентами повышает эффективность изложения, усвоения и понимания объектно-ориентированного анализа, проектирования и программирования.

Учебно-методический комплекс

Основная цель, которую необходимо достичь в результате обучения дисциплине "Объектно-ориентированный анализ и программирование" – научить студентов разрабатывать компьютерные модели реальных и концептуальных систем соответствующих направлению Бизнес-информатика¹.

В процессе изучения дисциплины рассматриваются следующие вопросы: объектно-ориентированный анализ (OOA), объектно-ориентированное проектирование (OOD), объектно-ориентированное программирование (OOP), шаблоны проектирования, унифицированный язык моделирования UML (Unified Modeling Language), объектно-ориентированный язык программирования C_Sharp (C#) и другие аспекты парадигмы объектно-ориентированного программирования.

В основе всех этих вопросов лежит один и тот же фундамент – способность и необходимость мыслить категориями объектов реального мира, так как специалисту-программисту необходимо разрабатывать Windows-приложения, эмулирующие те или иные системы реального мира. Поэтому изучение концепции

¹ Упомянуто данное направление, так как курс читается студентам факультета Бизнес-Информатика Государственного Университета – Высшая Школа Экономики (г. Москва)

объектного подхода не заканчивается изучением отдельно взятого метода или набора средств разработки. Иными словами, объектный подход является образом *объектно-ориентированного мышления*, которому также необходимо обучить студентов.

Важно понимать, что между изучением концепции объектов и использованием основанных на ней методов и средств существует значительная разница. Необходимо искать такие варианты обучения и обучающие материалы, в которых навыки мышления категориями объектов ставятся выше, чем навыки использования основанных на ней методов и средств.

Таким образом, осваивая объектно-ориентированные методы разработки приложений, важно овладевать фундаментальными принципами объектно-ориентированного подхода и уделять внимание освоению объектно-ориентированного мышления. Поэтому не только на первых лекциях и практических занятиях, но и на всех последующих этому уделяется первостепенное значение.

Переходить на новый способ мышления всегда непросто, поэтому вербальный метод обучения сопровождается активным привлечением компьютерных и информационных технологий. Это позволяет сопровождать рассуждения о концепциях объектов демонстрацией и анализом соответствующих фрагментов программного кода, а также иллюстративной графики.

На лекционных занятиях лектор пользуется компьютером, включенным в Internet. Практические занятия проводятся в компьютерных классах. На всех видах занятий используется мультимедийный проектор. Основными средами, в которых работают и преподаватель и студенты являются: Microsoft Visual Studio .NET 2005 и Rational Rose. Используются следующие Internet-ресурсы: новые книги раздела C# – <http://books.dore.ru/bs/f6sid16.html>; C# и .NET по шагам – <http://www.firststeps.ru>; язык графического моделирования UML – <http://www.uml.org> и др.

Сложность обучения дисциплине также заключается в дефиците выделенного аудиторного времени. Поэтому *особое внимание уделяется организации самостоятельной работы студентов и ее методическому обеспечению.*

С этой целью автором доклада разработан Учебно-методический комплекс дисциплины "Объектно-ориентированный анализ и программирование", который помещен на сайте ГУ-ВШЭ [1..4]. Все материалы комплекса доступны студентам и преподавателям и используются ими в учебном процессе.

Разделы Учебно-методического комплекса (<http://new.hse.ru/C7/C17/zabudskiy-e-i/default.aspx>):

1. Доклад Е.И. Забудского на 1-й Международной конференции по бизнес-информатике, 9-11 октября 2007 г., г. Звенигород.
2. Программа дисциплины и нормативно-справочные материалы
3. Материалы к лекциям
4. Материалы к практическим занятиям
5. Курсовая работа / реферат: темы и рекомендации
6. Компьютерные модели реальных и концептуальных систем, разработанные в соответствии с парадигмой ООП

В материалах к лекциям и практическим занятиям приводятся: тема и примерное количество аудиторных часов, отводимых на ее изучение; содержание; теория и листинги программ; резюме; контрольные вопросы и задания (упражнения) по объектно-ориентированному программированию; список литературы; ссылки на Internet-ресурсы и др.

На практических занятиях осуществляется компьютерное моделирование реальных и концептуальных систем. Для иллюстрации далее в таблицах приводится содержание раздела 2 и раздела 6 Учебно-методического комплекса.

2. Программа дисциплины и нормативно-справочные материалы

Наименование	Содержание	Объем файла, КБ
<u>Программа дисциплины</u>	Объектно-ориентированный анализ и программирование	437
<u>Терминология дисциплины</u>	Определения основных терминов платформы .NET Framework	282
<u>Список литературы</u>	Книги, статьи, гиперссылки на Internet -ресурсы	197
<u>Standard C#</u>	C# Language Specification: 4th Edition / June 2006. Стандарт утвержден в 2000 году ассоциацией European Computer Manufacturer's Association (ECMA)	2560
<u>Язык C#</u>	Объектно-ориентированный язык программирования C# : история создания и специфика	151
<u>Язык C#</u>	Ключевые слова C# и их семантика	227
<u>Нанотехнология</u>	Будущее компьютера: спинтроника	140

....

6. Компьютерные модели реальных и концептуальных систем, разработанные в соответствии с парадигмой ООП

Наименование моделируемой предметной области	Объем файла, КБ
Система <u>Домовладелец (LandLord)</u>	758
Система <u>Высотные Лифты Здания (Elevator)</u>	373
<u>Компьютерная Игра (Blackjack)</u> /см. предварительно Дополнение 1/	677
<u>Дополнение 1 к проекту Компьютерная игра (Blackjack)</u>	602
Система <u>Ипподромные Состязания (Derby)</u>	291
Система <u>Расчет Оценок Студента (см. Практическое занятие 8)</u>	1151

....

Далее приводится часть исходного код программы Моделирование работы банка.

Листинг 5.2. Программа моделирования работы банка (файл **BankSimulation.cs**)

```

001: using System; // Программа реализует доступ и управление несколькими счетами
002:
003: namespace ConsoleApplication_ BankSimulation
004: {
005: //////////////////////////////////////////////////////////////////// начало класса Account ////////////////////////////////////////////////////////////////////
006: class Account // Account – это пользовательский класс
007: { // Объект Account (счет) должен хранить следующие переменные экземпляра:
008: private decimal balance; // 1) свой баланс,
009: private decimal currentInterestRate; // 2) свою текущую процентную ставку,
010: private decimal totalInterestPaid; // 3) сумму начисленных процентов
011:
012: public Account() // Инициализация переменных экземпляра класса Account
013: {
014: balance = 0;

```



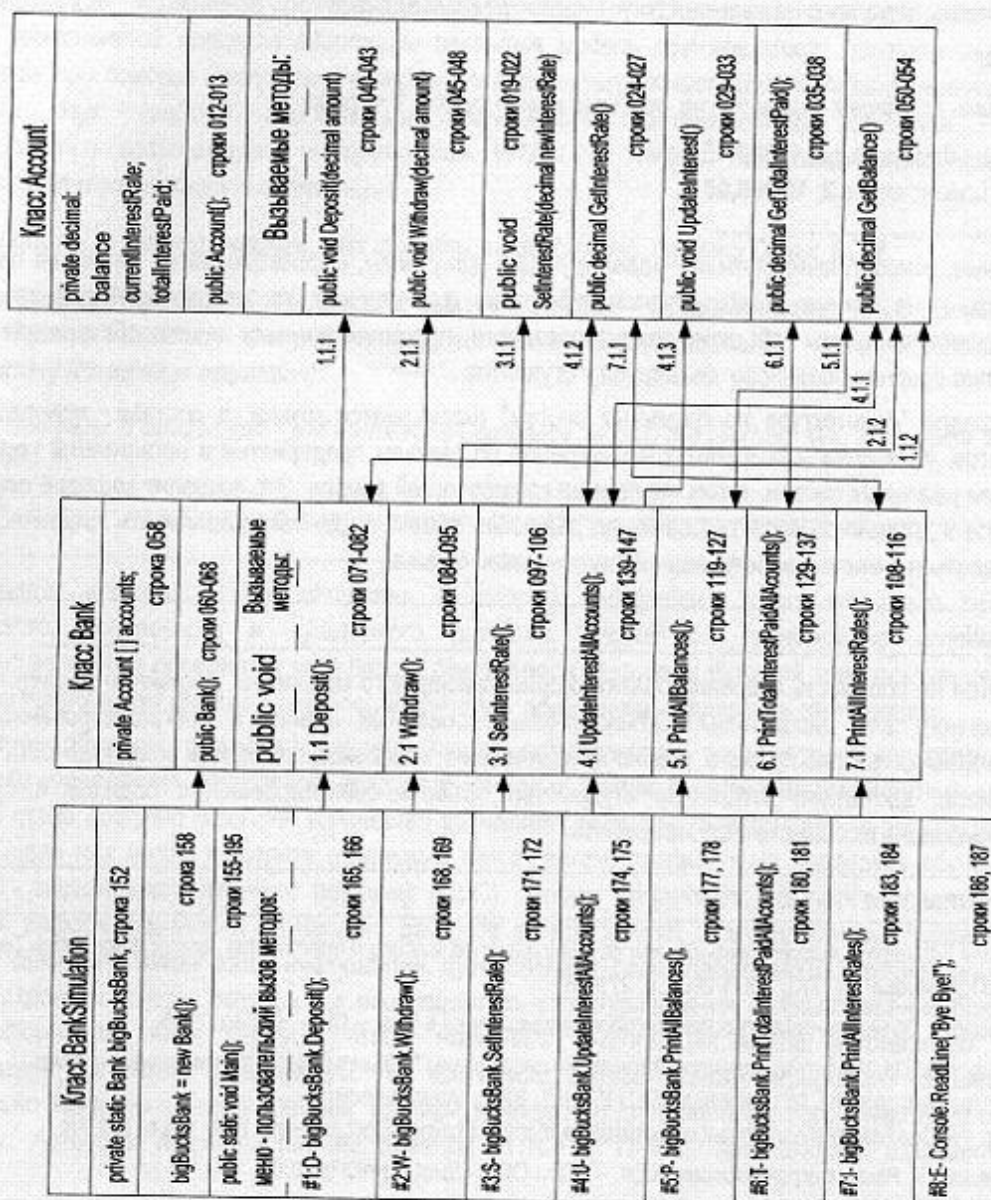
```

015: currentInterestRate = 0;
016: totalInterestPaid = 0;
017: }
018:
019: public void SetInterestRate(decimal newInterestRate) // Установить процентную ставку # 3.1.1
020: {
021:     currentInterestRate = newInterestRate;
022: }
023:
024: public decimal GetInterestRate() // Получить процентную ставку # 4.1.2
025: { // # 7.1.1
026:     return currentInterestRate;
027: }
028: // # 4.1.3
029: public void UpdateInterest() // 1. Определение суммы начисленных процентов
030: { // 2. Определение баланса с учетом суммы начисленных процентов
031:     totalInterestPaid += balance * currentInterestRate;
032:     balance += balance * currentInterestRate;
033: }
034: // # 6.1.1
035: public decimal GetTotalInterestPaid() // Получить значение суммы начисленных процентов
036: {
037:     return totalInterestPaid;
038: }
039: // # 1.1.1
040: public void Deposit(decimal amount) // Определение баланса (положить деньги на счет)
041: {
042:     balance += amount;
043: }
.....
163: switch(command) // Ниже указаны: 1) номера строк, выполняемых при
164: { // нажатии на клавишу с буквой "D", "W" ..., "I" или "E"; 2) номера
165:     case "D": // 1, 2, ..., 8 соответствующие пунктам меню (строки 200-208)
166:         bigBucksBank.Deposit(); // 071 – 082 , # 1
167:         break;
168:     case "W":
169:         bigBucksBank.Withdraw(); // 084 – 095, # 2
170:         break;
.....
199: Console.WriteLine("\nЧто Вы желаете сделать?\n" + // Пункты меню:
200: "D) – Положить деньги на счет\n" + // # 1
201: "W) – Снять средства со счета\n" + // # 2
202: "S) – Установить процентную ставку счета\n" + // # 3
203: "U) – Добавить проценты ко всем счетам\n" + // # 4
204: "P) – Вывести балансы всех счетов\n" + // # 5
205: "T) – Вывести сумму процентов, начисленных по каждому счету\n" + // # 6
206: "I) – Вывести процентную ставку по каждому счету\n" + // # 7
207: "E) – Завершить моделирование\n" + // # 8
208: "Примечание: первому счету соответствует индекс равный единице");
209: }
210: } ////////////////////////////////////////////////// конец класса BankSimulation //////////////////////////////////////
211: }

```

Программа имеет 211 строк, то есть является достаточно большой для рассмотрения на практическом занятии. Для удобства выполнения анализа строки пронумерованы (номера строк и знак «:» в программный код не входят). Кроме того, для удобства объяснения и понимания составлена Диаграмма взаимодействия классов (см. ниже). На диаграмме указаны номера строк и приведены номера возле стрелок, которые также проставлены в тексте программы. Эти номера, и комментарии к ним, выделены в тексте программы заливкой серого цвета. Нумерация, принятая в диаграмме, и соответствующая нумерация в тексте программы позволяют преподавателю с минимальной затратой времени разъяснить логику работы C#-программы "Моделирование работы банка".

Диаграмма взаимодействия классов программы "Моделирование работы банка"



Далее приводятся **Результаты работы программы** (выделено курсивом), при моделировании выполнения одной банковской операции "**Положить деньги на счет**".

Поздравляем! Вы создали новый банк.

Пожалуйста, введите количество счетов в банке: 5 <Enter>

Что Вы желаете сделать?

- D) – Положить средства на указанный счет
- W) – Снять средства с указанного счета
- S) – Установить процентную ставку указанного счета
- U) – Добавить проценты ко всем счетам
- P) – Вывести балансы всех счетов
- T) – Вывести сумму процентов, начисленных по каждому счету
- I) – Вывести процентную ставку по каждому счету
- E) – Завершить моделирование

Примечание: первому счету соответствует индекс равный единице

D <Enter>

Положить средства. Пожалуйста, введите номер счета: 2 <Enter>

Введите объем вклада: **10000** <Enter>

Новый баланс счета 2: **10 000,00** р.

.....

Подобные, относительно большие моделирующие программы, рассматриваются только на практических занятиях, а на лекциях анализируются небольшие фрагменты кода, иллюстрирующие теоретические положения парадигмы объектно-ориентированного программирования и способствующие развитию *объектно-ориентированного мышления* у студентов.

На кафедре "Архитектура программных систем" формируется группа, в составе преподавателей и студентов, ее задача – разработка и внедрение по заказам предприятий и организаций компьютерных моделей реальных систем, в том числе и на коммерческой основе. Это позволит кафедре осуществлять курсовое и дипломное проектирование по реальным темам, студентам накапливать практический опыт и определиться с местом работы еще на студенческой скамье.

Выводы

Несмотря на сложность освоения объектно-ориентированного мышления, и крайнем дефиците времени, выделенного для дисциплины "Объектно-ориентированный анализ и программирование", за счет использования разработанного учебно-методического комплекса повышена эффективность изложения материала, достигнуто *понимание* студентами объектно-ориентированного подхода и существенно активизирована их самостоятельная работа.

Библиография

1. Evgeny Zabudsky. Lecturer's web-site and its role in distance learning // International Journal "Information Technologies and Knowledge" (IJ ITK), Vol. 1/2007, p. 279-281.
2. Лафоре Р. Объектно-ориентированное программирование в С++. – СПб. : Питер, 2005.
3. Забудский Е.И. Учебно-методический комплекс дисциплины "Объектно-ориентированный анализ и программирование". М.: Кафедра АПС ГУ-ВШЭ, 2008, Internet-ресурс – http://vorona.hse.ru/sites/infospace/podrazd/facul/facul_bi/koiippo/DocLib3/New_Web_Page_ZEI.doc
4. Микелсен К. Язык программирования С#. – СПб.: ООО «ДиаСофтЮП», 2002.

Информация об авторе

Евгений Забудский – профессор кафедры "Архитектура программных систем" Государственного Университета – Высшая школа экономики; ул. Кирпичная, 33/5, Москва-105187, Россия;
e-mail: zei@inbox.ru; web-site <http://zei.narod.ru>