

работа по дисциплине **Object-Oriented Programming, C#** на тему:

«Компьютерное моделирование Реальной системы – название Предметной Области»

Методические указания по выполнению работы

1. Предметная Область и соответствующие пять классов выбираются студентами **самостоятельно** (область должна соответствовать специальности) – см. таблицу на с. 2.
2. Последовательность выполнения работы – см. с. 3 и с. 4.
3. Содержание (Оглавление) Отчета – см. с. 5.
4. Требования к оформлению Отчета – см. с. 6.
5. Титульный лист – см. в отдельном файле.

Примечание. К защите представляется надлежащим образом оформленный Отчет в бумажном варианте (см. далее стр. 6) и в электронном варианте. Также на защиту должна быть представлена компьютерная программа, работоспособность которой студент должен продемонстрировать на защите.

Группа ИУБ-_____

работа по дисциплине **Object-Oriented Programming, C#** на тему:

«Компьютерное моделирование Реальной системы – название Предметной Области»

/ Предметная Область и классы выбираются студентами **самостоятельно**; область должна соответствовать специальности/

№ пп	Фамилия, Имя	Предметная область	Базовый класс	1-й произв-й класс	2-й произв-й класс	1-й незав-й класс	2-й незав-й класс
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

Последовательность выполнения работы

«Компьютерное моделирование Реальной Системы – привести название выбранной Предметной Области»

1. Выбрать предметную область для ее компьютерного моделирования.
2. Сформулировать цель моделирования.
3. Задаться **пятью** классами, соответствующими предметной области (см. [таблицу](#))
/ см. УМК, стр. 2, Терминология, часть 2, термин 23, 5/
Базовые классы должны соответствовать тесту “is-a” – **является**.
Независимые классы **могут** соответствовать тесту “has-a” – **содержит** (см. лекцию 3, стр. 7).
4. Изобразить графически диаграммы каждого класса;
предусмотреть (**в первом приближении**) в каждом классе **переменные, свойства, методы и конструкторы** (см. лекцию 2, часть 1, рис. 3 и рис. 4; см. лекцию 12 и 13, а также [Дополнение](#)).
5. Написать код (программу) в соответствии с диаграммами классов, с учетом **инкапсуляции и наследования**
/ см. УМК, стр. 2, Терминология, часть 1, термины 21 и 29/
(см. лекции 2, 3, 4, 5 и 6). Предусмотреть класс с методом **Main()**.
6. Модифицировать код (пункт 5) с учетом понятия **полиморфизма**
/ см. УМК, стр. 2, Терминология, часть 1, термин 60/
(см. лекцию 4).
7. Модифицировать код (пункт 6) с учетом понятия **абстрактного класса**
/ см. УМК, стр. 2, Терминология, часть 1, термин 3; Терминология, часть 2, термин 2 /
(см. Прак. зан. 7).
8. Модифицировать код (пункт 7) с учетом понятия **interface**
/ см. УМК, стр. 2, Терминология, часть 1, термин 32; Терминология, часть 3, термин 35 /
(см. лекции 5 и 6).
9. Модифицировать код (пункт 8) с учетом понятия **dynamic-link library (dll)** (см. Прак. зан. 4).
10. Разработать к коду (пункт 9) графический **Windows-интерфейс (GUI)**
в соответствии с шаблоном **Модель-Вид-Контроллер** (см. лекцию 15);

программа управляется **событиями** (см. лекцию 7 и 8, а также доп. 1 и доп. 2).

11. Выполнить **объектно-ориентированный анализ (ООА)** предметной области

/ см. УМК, стр. 2, Терминология, часть 2, термин 36/

(см. лекцию 12 и 13, а также [Дополнение](#)).

по данным **ООА** разработать **концептуальную модель** предметной области.

12. Выполнить **объектно-ориентированное проектирование (ООП)** предметной области

/ см. УМК, стр. 2, Терминология, часть 2, термин 35/

(см. лекцию 14);

по данным **ООА** и **ООП** разработать **объектную модель** предметной области

/ см. УМК, стр. 2, Терминология, часть 2, термин 33/.

13. В соответствии с **объектной моделью** разработать код в среде **MS VS 2008 .NET** на языке **C#**

/ см. УМК, стр. 2, Терминология, часть 2, термин 34/.

14. Оформить работу и представить ее к защите;

[Содержание](#) работы и [требования к оформлению](#) – см. далее стр. 5 и 6.

[Титульный лист](#) – см. в отдельном файле.

Примечание. См. [примеры разработки компьютерных моделей реальных \(концептуальных\) систем](#) – гиперссылки на [стр. 6 Учебно-методического комплекса](#)

Содержание

	Введение (охарактеризовать парадигму объектно-ориентированного программирования)
1.	Описание предметной области
2.	Цель компьютерного моделирования реальной (концептуальной) системы
3.	Объектно-ориентированный анализ (см. лекцию 12, разделы 2 и 3)
3.1	Использование прецедентов для определения возможного применения системы
3.1.2	Отождествление действующих субъектов
3.1.3	Создание списка прецедентов
3.1.4	Определение последовательности событий для двух прецедентов
3.1.5	Моделирование прецедентов (см. лекцию 12 и 13 и Дополнение к ней)
3.1.5.1	Диаграмма прецедентов
3.1.5.2	Диаграммы взаимодействия
3.1.5.2.1	Диаграммы последовательности событий
3.1.5.2.2	Диаграммы сотрудничества
3.1.5.3	Диаграммы ... /по мере необходимости/ (см. Дополнение к лекции 12 и 13)
3.1.6	Построение концептуальной модели
4.	Объектно-ориентированное проектирование (см. лекцию 14, раздел 1).
4.1	Создание исходного списка объектов
4.2	Определение назначения объектов
4.3	Определение точек взаимодействия
4.4	Детализация отношений между объектами
4.5	Построение объектной модели
5	Разработка кода в соответствии с парадигмой компонентно-ориентированного программирования (см. прак. зан. 4, раздел 5; см. лекцию 12 и 13, раздел 1).
5.1	Реализация программы в формате GUI на основе шаблона «Модель-Вид_Контроллер» (см. лекцию 15) *
5.2	UML-диаграмма взаимодействия классов (VS .NET 2008)
6	Результаты работы программы
	Литература

Примечание. В Графическом интерфейсе первым окном должна быть заставка, содержащая информацию титульного листа проекта. На заставку также поместить картинку, отражающую существо предметной области. Переход ко второму окну (здесь осуществляется взаимодействие с системой) выполнить по нажатию на кнопку. Предусмотреть «Справка» - «О программе», где дать подробную характеристику программы

Требования к оформлению

Шрифт – Times New Roman;

Размер шрифта – 12 пт;

Поля – 2 см. со всех сторон;

Междустрочный интервал – множитель 1,2;

Между абзацами – 6 пт.;

Абзацный отступ – нет;

Печать – с одной стороны листа;

Выравнивание – по ширине;

Дизайн оформления отчета – по своим возможностям.

Скоросшиватель (*простой*) / НЕ ВСТАВЛЯТЬ ЛИСТЫ В ОДИН ФАЙЛ,

а подшить для возможности **перелистывания**/.

Разработал
профессор Е.И. Забудский
Editing on February 01 2012