

Домашнее задание по дисциплине

" - "

Тема

Моделирование системы лифтов здания **(или) Моделирование работы банка**

(или – см. [Программа дисциплины](#) – “Объектно-ориентированное программирование”)

Выполнил: студент группы -301

Иванов Л.А.

(подпись, дата)

e-mail: _____

Руководитель:

профессор Забудский Е.И.

* 2014

Рекомендации – обязательные для выполнения

1	Титульный лист – образец оформления (см. выше)	1
2	Основные этапы компьютерного моделирования реальных и концептуальных систем – последовательность выполнения домашнего задания (см. также пп. 3.1 и 3.2)	3
3	Перечень предлагаемых тем и направлений курсовой работы	4
3.1	Моделирование системы лифтов здания *	4
3.2	Моделирование работы банка *	5
3.3	Разработать чертежную программу	6
3.4	Разработать компьютерную игру	6
3.5	Разработать базу данных сотрудников	6
3.6	Моделирование работы четырех светофоров на перекрестке	6
3.7	Программу «Моделирование работы банка» наполнить другим содержанием, а именно:	6
3.7.1	Моделирование работы склада	6
3.7.2	Моделирование работы отдела кадров	5
3.7.3	Моделирование работы Дома Мод	6
3.7.4	Моделирование Салона Красоты	6
3.7.5	etc. , etc...	6
4	Авторские темы – предлагаются студентами — NB	6

NB* | см. обязательное Содержание Домашнего задания на с. 4 и с. 5

NB	<p>В Графическом интерфейсе первым окном должна быть заставка, содержащая информацию титульного листа проекта (см. выше с. 1). На заставку также поместить картинку, отражающую существо предметной области.</p> <p>Переход ко второму окну (здесь осуществляется взаимодействие с системой) выполнить по нажатию на кнопку.</p> <p>Предусмотреть в GUI: «Справка» > «О программе», где дать подробную характеристику программы</p>
-----------	--

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;

Титульный лист – по данному образцу (см. выше, с. 1);

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/.

2. Основные этапы компьютерного моделирования реальных и концептуальных систем

(см. также пп. 3.1 и 3.2 на с. 4, 5)

1. Сформулировать целевую функцию моделирования (например, собрать статистическую информацию о качестве функционирования системы >> см. материалы к Практич. Зан. №3 с. 11).
2. Выполнить анализ реальной системы: учесть её важные действия и атрибуты и отбросить вторичные (реализация концепции ООА – абстракция >> см. материалы к Практич. Зан. №3, с. 4).
3. Выделить классы объектов, входящих в реальную систему >> см. материалы к Практич. Зан. №1, с. 9.
4. Установить связь между классами; выстроить иерархию классов и проанализировать целесообразность повторного использования их кода (реализация концепции ООП - наследование >> см. материалы к Практич. Зан. №6 с. 4)
5. Определить объекты классов – экземпляры классов >> см. материалы к Практич. Зан. №1, с. 2.
6. Сформулировать переменные экземпляров, которые отражают атрибуты объектов.
7. Установить какие действия должны выполнять объекты, то есть определить методы объектов и их функциональность.
8. Обдумать степень открытости методов и переменных экземпляров (реализация концепции ООП - инкапсуляция >> см. материалы к Практич. Зан. №3 с. 4).
9. Выполнить внутреннее проектирование методов; обдумать целесообразность обращения к различным реализациям методов одним и тем же вызовом (реализация концепции ООП – полиморфизм > механизм динамического связывания >> см. материалы к Практич. Зан. №7 с. 8)
10. Приступить к разработке исходного кода. На заключительном этапе реализовать парадигму компонентно-ориентированного программирования >> см. материалы к Практич. Зан. №4 с. 21.
11. Разработать графический интерфейс (GUI) на основе шаблона «Модель-Вид-Контроллер» (см. Лекция 15)
12. etc.

Примечание.

Указанная последовательность является примерной. По мере осуществления последующих этапов возникает необходимость уточнять реализацию предыдущих (принята итеративная технология).

См. Содержание Домашнего задания в пп. 3.1 и 3.2 на с. 4 и с. 5

Примеры.

1. Моделирование системы лифтов здания

См. материалы:

- Пр_зан_№ 1, с. 7, 8 .
- Пр_зан_№ 3, с. 4, 5, 6, 7, 8, 11 – 27, листинг 3.1.
- Пр_зан_№ 4, с. 4, 5, 6, 21, 22, 31-39, листинги 4.1, 4.2, 4.7-4.10.
- Пр_зан_№ 5, с. 4-7, 15, 16, 18-21, листинги 5.1, 5.5, 5.6.

2. Моделирование работы банка

См. материалы:

- Пр_зан_№ 5, с. 7-15, 25-32, 37, листинги 5.2, 5.7.

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

3. Темы Домашнего Задания (см. также Программу дисциплины)

3.1. Моделирование системы лифтов здания

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, раздел 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 КАЖДОГО прецедента
3.1.5	Моделирование прецедентов (см. лекцию 12 и дополнение к ней)
3.1.5.1	Диаграммы 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/
3.6	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, 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 2005)
6	Результаты работы программы
	Литература
Спец. вопрос. В системе несколько лифтов (2...10) и соответственно несколько пассажиров. Каждый пассажир пользуется одни и тем же лифтом, то есть: 1-й пассажир + 1-й лифт, 2-й пассажир + 2-й лифт и т.д.	

* **Примечание.** В Графическом интерфейсе первым окном должна быть заставка, содержащая информацию титульного листа проекта (см. выше с. 1). 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

См. материалы:

- П_3_№ 1, стр. 7, 8 .
- П_3_№ 3, стр.: 4, 5, 6, 7, 8, 11 – 27, листинг 3.1, **31 - самостоятельно** .
- П_3_№ 4, стр. : 4, 5, 6, 21, 22, 31-39, **42 - самостоятельно**, листинги 4.1, 4.2, 4.7-4.10.
- П_3_№ 5, стр.: 4-7, 15, 16, 18-21, листинги 5.1, 5.5, 5.6.
- **6. Компьютерные модели реальных и концептуальных систем, разработанные в соответствии с парадигмой ООП (с. 6)**

3.2. Моделирование работы банка

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, раздел 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 КАЖДОГО прецедента
3.1.5	Моделирование прецедентов (см. лекцию 12 и дополнение к ней)
3.1.5.1	Диаграммы 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/
3.6	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, 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 2005)
6	Результаты работы программы
	Литература
	Смоделировать в программе различные операции (кроме рассмотренных на пр. зан. №5), выполняемые в банке со счетами, например: 1) перечисление денег с одного счета на другой; 2) кассир банка должен идентифицировать клиента.

* Примечание. В Графическом интерфейсе первым окном должна быть заставка, содержащая информацию титульного листа проекта (см. выше с. 1). 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

См. материалы:

- П_3_№5, стр. 7-15, 25-32, 34 - самостоятельно, 37, листинги 5.2, 5.7.
- 6. Компьютерные модели реальных и концептуальных систем, разработанные в соответствии с парадигмой ООП (с. 6)

3.3. Распространить чертежную программу (см. разд. 3.3, 3.4 и 3.5 – Материалы к Практик. Зан. №7, стр. 10-17) на любые другие классы **Shape**. Например, добавить класс **Octagon (и другие)**, для этого достаточно сделать его производным от **Shape** и включить в него реализацию метода **DrawYourself**. Затем нужно добавить объект **Octagon** к чертежу, причем метод **DrawDrawing** остается при этом неизменным: он не зависит от того, с какой фигурой ему приходится иметь дело, главное, чтобы она была производной от класса **Shape** (т. е. имела свой метод **DrawYourself**).

3.4. Компьютерные игры содержат объекты различных типов, которые движутся, действуют и отображаются на экране: космические корабли, пришельцы, машины, монстры и т. д. Можно создать общий базовый класс для элементов игры, скажем, **GameElement**, который, наряду с другими методами, будет содержать абстрактный метод **DrawYourself**. Несмотря на то, что набор содержит совершенно разные игровые элементы, каждый из которых рисуется на экране по-своему, для их вывода достаточно просто пройти по списку, вызывая метод **DrawYourself** (см. Материалы к Практик. Зан. №7, стр. 17).

3.5. База данных сотрудников включает данные о разных служащих: программистах, секретарях, маркетологах, уборщицах, директорах и т. д. Несмотря на различия, у них есть и общие атрибуты — имя, адрес, дата рождения, и методы — начисление зарплаты (**CalculateSalary**) и т. д. Можно создать базовый класс **Employee**, производными от которого будут остальные классы. Снабдив **Employee** общими атрибутами и методами (**CalculateSalary**), при начислении зарплаты достаточно пройти по всему списку, вызывая метод **CalculateSalary**. В итоге, не обязательно знать, кем именно является конкретный служащий, поскольку подходящий метод вызывается посредством динамического связывания (см. Материалы к Практик. Зан. №7, стр. 17).

3.6 Моделирование работы четырех светофоров на перекрестке

3.7 Программу «Моделирование работы банка» наполнить другим содержанием, а именно:

3.7.1 Моделирование работы склада

3.7.2 Моделирование работы отдела кадров

3.7.3 Моделирование работы Дома Мод

3.7.4 Моделирование Салона Красоты

3.7.5 etc. , etc.

4 Авторские темы – предлагаются студентами

NB

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

(см. на web-page)

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