

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
"Национальный исследовательский университет  
"Высшая школа экономики"**

Факультет компьютерных наук  
Департамент программной инженерии

**Рабочая программа дисциплины  
Проектирование архитектуры программных систем**

для образовательной программы «Программная инженерия»  
направления подготовки 09.03.04 «Программная инженерия»  
уровень - бакалавр

Разработчик программы:

Мицюк А. А., старший преподаватель, [amitsyuk@hse.ru](mailto:amitsyuk@hse.ru)

Одобрена на заседании департамента программной инженерии «\_\_»\_\_\_\_\_ 2018 г.  
Руководитель департамента Авдошин С. М. \_\_\_\_\_

Утверждена Академическим советом образовательной программы  
«\_\_»\_\_\_\_\_ 2018 г., № протокола \_\_\_\_\_

Академический руководитель образовательной программы  
Шилов В. В. \_\_\_\_\_

Москва, 2018

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.*



## 1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины «Проектирование архитектуры программных систем» устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов образовательной программы «Программная инженерия» направления подготовки 09.03.04 «Программная инженерия», изучающих дисциплину «Проектирование архитектуры программных систем».

Программа разработана в соответствии с

- образовательным стандартом НИУ ВШЭ по направлению 09.03.04 «Программная инженерия» (уровень подготовки: бакалавр, [http://www.hse.ru/data/2015/05/20/1097268782/Бакалавриат\\_ОС\\_Программная%20инженерия.pdf](http://www.hse.ru/data/2015/05/20/1097268782/Бакалавриат_ОС_Программная%20инженерия.pdf));
- образовательной программой НИУ ВШЭ по направлению подготовки 09.03.04 «Программная инженерия» (уровень подготовки: бакалавр);
- рабочим учебным планом университета по направлению 09.03.04 «Программная инженерия», утвержденным в 2018 г.;
- международным образовательным стандартом Software Engineering 2014. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (<http://www.acm.org/binaries/content/assets/education/se2014.pdf>).

**Примечание:** Настоящая программа является развитием программы дисциплины «Проектирование архитектуры программных систем» 2017/2018 учебного года (<https://www.hse.ru/edu/courses/214346550>), которая была разработана профессором, д. ф.-м. н. М. И. Кумсковым ([mkumskov@hse.ru](mailto:mkumskov@hse.ru)).

## 2 Цели освоения дисциплины

Целями освоения дисциплины «Проектирование архитектуры программных систем» являются получение теоретических знаний и практических навыков работы с требованиями к информационным системам (ИС) как входных данных для проектирования, а также получение теоретических знаний и практических навыков при выполнении основных задач архитектора и проектировщика в рамках *унифицированного процесса* (УП, UP) проектирования и визуального моделирования с использованием *унифицированного языка моделирования UML*, в том числе согласно методологии IBM Rational Unified Process (IBM RUP).

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать основные понятия, используемые при проектировании и создании приложений, информационных систем (ИС), включая лучшие практики разработки программного обеспечения (ПО); основы визуального моделирования при проектировании и разработке архитектуры ПО, способы преобразования требований к ИС на основе сценариев использования в архитектурные представления ПО; основные задачи, выполняемые архитектором и разработчиком при проектировании по методологии IBM RUP.
- Уметь строить проектные модели ПО и ИС с использованием диаграмм визуальной нотации UML (Унифицированный язык моделирования); проектировать логическое представление архитектуры на UML; пошагово строить на UML диаграммы классов, пред-



ставляющие модель предметной области и ключевые абстракции проекта; с использованием паттернов трансформировать модель предметной области; идентифицировать и разрабатывать спецификации сценариев использования ИС для создания классов-участников реализации; использовать диаграммы взаимодействия, состояний и активности при проектировании ПО.

- Иметь навыки (приобрести опыт) пошагового построения проекта ПО как последовательного набора UML диаграмм в инструментальном CASE средстве; реализации лучших практик разработки программного обеспечения, включая построения компонентной архитектуры ПО и применение архитектурных механизмов анализа; использования визуальных диаграмм UML при проектировании архитектурных представлений.

В результате освоения дисциплины студент осваивает следующие компетенции:

ПК-2	Способен к формализации в своей предметной области с учетом ограничений используемых методов исследования
ПК-3	Способен использовать методы и инструментальные средства исследования объектов профессиональной деятельности
ПК-4	Способен обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности
ПК-6	Способен формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта
ПК-9	Способен создавать программное обеспечение для ЭВМ и систем различной архитектуры
ПК-10	Способен проектировать, конструировать и тестировать программные продукты
ПК-12	Способен моделировать, анализировать и использовать формальные методы конструирования программного обеспечения
ПК-13	Способен оценивать временную и емкостную сложность программного обеспечения
ПК-14	Способен создавать программные интерфейсы
ПК-15	Способен использовать операционные системы, сетевые технологии, средства разработки программного интерфейса, применять языки и методы формальных спецификаций, системы управления базами данных
ПК-16	Способен использовать различные технологии разработки программного обеспечения
ПК-19	Способен понимать стандарты и модели жизненного цикла

## 4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части дисциплин профессионального цикла (блок Б.ПЦ.Б) рабочего учебного плана направления 09.03.04 «Программная инженерия» подготовки бакалавра на 2018-2019 учебный год. Дисциплина предлагается студентам в третьем и четвертом модулях третьего года обучения. В рамках курса запланированы 32 часа лекций и 32 часа семинаров (практических занятий), т. е. всего 64 аудиторных часа. Кроме этого, 126 часов отводятся под самостоятельную работу студентов. Таким образом, всего в рамках дисциплины запланировано 190 часов. Итоговый экзамен проводится в конце четвертого модуля.

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:



- Владеть: навыками решения задач из следующих разделов современной математики: математический анализ, линейная алгебра и аналитическая геометрия, математическая логика, дискретная математика, знаниями основных алгоритмов при работе с массивами, списками и деревьями, навыками разработки программ и программных систем с использованием баз данных.
- Знать: основные направления, проблемы, теории и методы современной математики, в том числе основы математического анализа, линейной алгебры, дискретной математики и математической логики, основные понятия информатики, используемые при проектировании и разработке программ.
- Уметь: решать стандартные задачи математического анализа, линейной алгебры, дискретной математики и математической логики и применять идеи, использованные в их решениях, для решения аналогичных задач, составлять решения задач на алгоритмических языках, включая C/C++.
- Иметь навык: программирования на языке высокого уровня с поддержкой объектно-ориентированной парадигмы типа C++, Java или C#.

Изучение дисциплины базируется на знаниях, полученных студентами ранее при освоении учебных дисциплин «Программирование» и «Введение в программную инженерию» первого года обучения, а также «Конструирование программного обеспечения», «Алгоритмы и структуры данных» второго года обучения.

Основные положения дисциплины должны быть использованы в дальнейшем при подготовке курсовых и выпускных квалификационных работ.

## 5 Предварительный тематический план учебной дисциплины

№	Название темы	Всего часов	Аудиторные часы		Самостоятельная работа
			Лекции	Семинары	
1	Архитектура программных систем. Унифицированный процесс разработки.	11	2	2	7
2	Унифицированный язык моделирование UML. Диаграммы UML. Процесс выявления требований к ИС. Прецеденты (сценарии использования). Эскиз и спецификация прецедента. Диаграмма прецедентов UML.	13	2	2	9
3	Классы и объекты. Отношения между классами. Диаграммы классов UML. Модель предметной области. Шаблоны проектирования.	13	2	2	9
4	Анализ и проектирование — дисциплина унифицированно-го процесса. Рабочий поток анализа. Задача архитектора: «Архитектурный анализ».	11	2	2	7
5	Рабочий поток анализа. Задача разработчика: «Анализ прецедента». Объектная динамика при проектировании. Диаграммы взаимодействия UML. Диаграмма «Представление классов участников».	13	2	2	9
6	Архитектурные представления. Пакеты в UML. Рабочий поток проектирования. Задачи архитектора «Идентификация проектных элементов», «Идентификация проектных механизмов».	11	2	2	7



7	Рабочий поток проектирования. Задачи проектировщика «Проектирование прецедентов», «Проектирование подсистем».	13	2	2	9
8	Рабочий поток проектирования. Задача разработчика «Проектирование классов». Проектирование поведения ИС. Диаграммы состояний, диаграммы деятельности языка UML. Другие языки моделирования.	13	2	2	9
9	Документирование проекта архитектуры программной системы.	13	2	2	9
10	Нефункциональные требования в архитектуре ИС. Объектный язык ограничений OCL.	11	2	2	7
11	Лучшие практики разработки информационных систем. CASE-средства визуального моделирования. Прямое и обратное проектирование.	11	2	2	7
12	Архитектурные стили, шаблоны. Плюсы и минусы различных архитектурных стилей.	13	2	2	9
13	Шаблоны проектирования GRASP. Шаблоны проектирования GoF. Адаптация и поддержка архитектуры ИС. Тестирование и рефакторинг.	13	2	2	9
14	Реализация и развертывание ИС. Рабочий поток реализации. Влияние особенностей аппаратного обеспечения на программную архитектуру. Диаграммы развертывания языка UML.	11	2	2	7
15	Архитектурные стили сетевых и распределенных программных систем.	10	2	2	6
16	Проектирование защищенных ИС. Проектирование и разработка программной системы — командная работа.	10	2	2	6
<b>Итого:</b>		190	32	32	126

## 6 Формы контроля знаний студентов

Тип контроля	Форма контроля	3-й год		Параметры
		3	4	
Текущий	Проектная работа		●	Выполнение проекта по заданной постановке задачи в инструментальном CASE-средстве
	Домашние задания и работа на семинарах	●	●	Задачи периодически выдаются для закрепления пройденных тем на семинарах (практических занятиях)
Итоговый	Экзамен		●	Письменный экзамен продолжительностью 90 минут

### 6.1 Критерии оценки знаний и навыков

**Текущий контроль** предусматривает периодическое выполнение студентами *заданий на практических занятиях и домашних заданий*. Задания могут быть рассчитаны на выполнение индивидуально и в небольших группах (2-5 студентов). Задания выполняются письменно



или с использованием программных средств. Студенты, не сдавшие какое-либо из домашних заданий, получают за это задание оценку 0.

Кроме периодических заданий, студенты в четвертом модуле выполняют *проектную работу*. Данная работа выполняется по заданной постановке задачи (примеры постановок приведены в Приложении) в течение нескольких недель. В ходе проекта следует выполнить *следующие этапы*, каждый из которых оценивается отдельно в интервале от 0 до 10:

1. Выявить требования к информационной системе (ИС) в виде прецедентов (сценариев использования) системы и модели предметной области – построить соответствующие диаграммы на языке UML, подготовить документ *Видение* – оценка  $O_{(1)}$ .
2. Подготовить спецификации прецедентов по известному шаблону, включая основной и альтернативные потоки событий, предусловия и постусловия – оценка  $O_{(2)}$ .
3. Идентифицировать классы участники реализации прецедентов - *граничные классы, управляющие и классы-сущности*, - и распределить по ним поведение прецедента для каждого потока событий, построив соответствующие диаграммы взаимодействия на языке UML – оценка  $O_{(3)}$ .
4. Построить диаграммы классов на языке UML для классов участников реализаций прецедентов, отражающие спроектированную архитектуру, подготовить отчет с описанием спроектированной архитектуры – оценка  $O_{(4)}$ .

Проектирование ИС на языке UML производится с использованием CASE-средства, выбранного для проведения работ.

**Итоговый контроль** предусматривает экзамен в конце четвертого модуля, который проводится в письменной форме. Экзамен предполагает выполнение задания на проектирование и/или ответ на теоретические вопросы по программе курса. Продолжительность выполнения письменного задания — 90 минут.

## 6.2 Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов. Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

**Накопленная оценка** (от 0 до 10) учитывает результаты студента по текущему контролю следующим образом:

$$O_n = 0,5 \cdot O_p + 0,4 \cdot O_d + 0,1 \cdot O_a,$$

где составляющие накопленной оценки имеют следующее значение:

- $O_p$  — оценка за проектную работу, которая вычисляется по формуле  $O_p = (O_{(1)} + O_{(2)} + O_{(3)} + O_{(4)}) / 4$ , где  $O_{(i)}$  — оценка за  $i$ -й этап проекта. Например, если за этапы проекта получены следующие оценки – оценка  $O_{(1)} = 9$ , оценка  $O_{(2)} = 8$ , оценка  $O_{(3)} = 6$ , оценка  $O_{(4)} = 9$ , тогда оценка  $O_p = (9 + 8 + 6 + 9) / 4 = 32 / 4 = 8$ .
- $O_d$  — это *среднее* значение оценки по всем домашним заданиям (ДЗ).  $O_d$  принимает значение от 0 до 10. Если какое-то ДЗ не было выполнено (сдано), за него выставляется оценка 0. Например, если из 5-ти заданий было сдано только 4 со следующими оценками: 7, 8, 6, 5 соответственно, то тогда  $O_d = (7 + 8 + 6 + 5 + 0) / 5 = 26/5 = 5,2$ .
- $O_a$  — это значение оценки за работу на семинарских (практических) занятиях. Студенты могут получать на семинарах небольшие тесты/задания, которые должны выполнять-





ся на занятии, кроме того дополнительные баллы могут выставляться за активное участие в работе группы на семинаре, выступления с результатами выполнения ДЗ и т. д.

**Результирующая оценка** за дисциплину рассчитывается следующим образом:

$$O_p = 0,5 \cdot O_n + 0,5 \cdot O_э,$$

где  $O_n$  — накопленная оценка,  $O_э$  — экзаменационная оценка. Способ округления *результующей* ( $O_p$ ), *накопленной* ( $O_n$ ) и *экзаменационной* ( $O_э$ ) оценок по учебной дисциплине — арифметический. Оценки текущего контроля ( $O_{п}$ ,  $O_{д}$ ,  $O_{а}$ ) не округляются. При пересдаче неудовлетворительной оценки формула вычисления оценки студента не изменяется.

**Таблица соответствия оценок по десятибалльной и пятибалльной системам:**

По десятибалльной шкале	По пятибалльной шкале
1 – неудовлетворительно 2 – очень плохо 3 – плохо	неудовлетворительно – 2
4 – удовлетворительно 5 – весьма удовлетворительно	удовлетворительно – 3
6 – хорошо 7 – очень хорошо	хорошо – 4
8 – почти отлично 9 – отлично 10 – блестяще	отлично – 5

## 7 Содержание дисциплины

**Примечание:** Возможны незначительные изменения в содержании отдельных тем.

### Тема 1 Архитектура программных систем. Унифицированный процесс разработки.

Вводная информация. Понятие архитектуры программной информационной системы (ИС). Роль архитектуры в контексте процесса разработки программного обеспечения. Проект и архитектура. Цель проектирования архитектуры ИС. Парадигмы проектирования: структурная, объектно-ориентированная, функциональная, логическая.

Объектно-ориентированный анализ и проектирование (ООАиП). Унифицированный процесс (УП) разработки программного обеспечения (ПО). Методология IBM Rational Unified Process (IBM RUP) как пример унифицированного процесса.

**Литература:** [Леффингуэлл и др.], [Соммервилл], [Арлоу и др.], [Taylor et al.], [Крачтен]

### Тема 2 Унифицированный язык моделирование UML. Диаграммы UML. Процесс выявления требований к ИС. Прецеденты (сценарии использования). Эскиз и спецификация прецедента. Диаграмма прецедентов UML.

Основные принципы визуального моделирования. Сложность ПО и архитек-



турные представления. Статические и динамические диаграммы языка UML. Репозиторий модели CASE-средства. Прямое и обратное проектирование ПО и структуры базы данных (БД). Стереотипы UML и их использование.

Заинтересованные лица, свойства ИС, классификация свойств. Формулировка проблемы. Диаграмма причинно-следственных связей. Матрица трассировки требований. Выявления трудоемкости реализации прецедента и свойств ИС. Понятие риска. Понятие объема работ проекта (Score). Инструментальная поддержка процесса проектирования.

Известные подходы к идентификации прецедентов. Первичные и вторичные акторы. Эскиз и спецификация прецедента. Выявление прецедентов в бизнес-процессах. Типовая структура спецификации. Основной и альтернативные потоки действий. Шаблоны выявления прецедентов ИС. Учет бизнес-правил при составлении спецификации прецедента. Пред-условия и пост-условия прецедента. Визуальное представление на UML модели прецедентов (сценариев использования). Словарь проекта. Концепция ИС. Модель прецедентов.

**Литература:** [Ларман], [Леффингуэлл и др.], [Соммервилл], [Арлоу и др.], [Крачтен], [Fowler]

### **Тема 3 Классы и объекты. Отношения между классами. Диаграммы классов UML. Модель предметной области. Шаблоны проектирования.**

Понятия класса и объекта. Отношения между классами и их визуальное представление с помощью UML. Ассоциация и ее разновидности. Понятия навигации ассоциации, наследования, зависимости. Отличие агрегации от композиции. Понятие кратности ассоциации. Шаблоны действий при создании и преобразовании UML диаграмм классов. Ассоциативные классы и их использование в проекте.

Процесс формирования визуальной модели. Регистрируемые события и объекты. Связь модели предметной области с задачей идентификации функциональных требований к ИС. Используемые UML-диаграммы, шаблоны при их построении. Понятие состояния объекта и его визуальное представление. Поиск акторов в модели предметной области.

**Литература:** [Ларман], [Кумсков], [Арлоу и др.], [Крачтен], [Fowler]

### **Тема 4 Анализ и проектирование — дисциплина унифицированного процесса. Рабочий поток анализа. Задача архитектора: «Архитектурный анализ».**

Обзор дисциплины «Анализ и проектирование». Входные и выходные артефакты (рабочие материалы) дисциплины. Роли и задачи дисциплины. Назначение и задачи этапов дисциплины. Используемые диаграммы языка UML в задачах проектирования.

Роль архитектора в проекте и выполняемые им задачи. Ключевые абстракции и их идентификация. Диаграммы UML для визуализации ключевых абстракций. Архитектурные механизмы, их назначение. Архитектурные шаблоны и их использование. Понятие слоя и его представление в языке UML. Структура «реализация прецедента» (use case realization) и ее представление в визуальной модели.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Fowler]





**Тема 5 Рабочий поток анализа. Задача разработчика: «Анализ прецедента». Объектная динамика при проектировании. Диаграммы взаимодействия UML. Диаграмма «Представление классов участников».**

Выявление классов-участников. Применение прецедентов для проектирования объектной динамики. Стереотипы «граничный» («boundary»), «управляющий» («control»), «сущность» («entity») классов-участников реализации. Шаблоны для идентификации классов.

Диаграммы последовательности и коммуникации UML. Правила использования стереотипов «boundary», «control», «entity» при проектировании объектной динамики.

Диаграмма классов-участников (VOPC). Правила идентификации операций (ответственностей) классов. Правила идентификации отношений между классами – ассоциаций и зависимостей.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Fowler]

**Тема 6 Архитектурные представления. Пакеты в UML. Рабочий поток проектирования. Задачи архитектора «Идентификация проектных элементов», «Идентификация проектных механизмов».**

Понятие заинтересованных лиц. Детальность проекта. Архитектурные шаблоны и их использование в архитектурном проекте. Использование модели предметной области при решении задачи проектирования. Пакеты в UML. Связи между пакетами UML и их использование в представлении архитектуры.

Задача архитектора «Идентификация проектных элементов». Определение классов, подлежащих декомпозиции. Упаковка концептуальных классов по пакетам проекта. Зависимости между пакетами. Архитектурный шаблон «Слои» и его использование при проектировании ИС.

Задача архитектора «Идентификация проектных механизмов». Понятие проектного шаблона. Представление шаблонов в CASE-средстве. Типы архитектурных механизмов и их представление в визуальной модели.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Fowler]

**Тема 7 Рабочий поток проектирования. Задачи проектировщика «Проектирование прецедентов», «Проектирование подсистем».**

Задача проектировщика «Проектирование прецедентов». Реализация прецедента на этапе проектирования. Применение архитектурных механизмов. Представление компонентов в проекте. Использование интерфейсов на диаграммах последовательности UML.

Задача проектировщика «Проектирование подсистем». Отличие пакетов и подсистем. Структурный класс UML и его использование. Операции интерфейса и их реализация в проекте подсистемы. Диаграмма последовательности UML. Динамическое и статическое представление дизайна подсистемы. Использование проектных механизмов. Зависимости классов и пакетов.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Fowler]



**Тема 8 Рабочий поток проектирования. Задача разработчика «Проектирование классов». Проектирование поведения ИС. Диаграммы состояний, диаграммы деятельности языка UML. Другие языки моделирования.**

Проектные классы. Ответственности и операции классов. Выбор атрибутов классов. Атрибуты и понятие состояния объекта. Шаблоны выявления состояний. Специальные состояния – начальное и конечное. Супер состояние и его использование. Диаграммы состояний UML. Условия на диаграмме состояний. Правило перехода из состояния в состояние и из запись на диаграмме.

Диаграммы деятельности языка UML. Их использование в рамках унифицированного процесса. Семантика деятельности. События, исключения.

Конечные автоматы. Конечные автоматы в рамках унифицированного процесса. Конечные автоматы и язык UML. Состояния и переходы. Другие языки моделирования поведения: сети Петри, модели BPMN.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Соммервилл], [Taylor et al.], [Martin]

**Тема 9 Документирование проекта архитектуры программной системы.**

Документирование проектируемой архитектуры ИС. Способы описания и документирования архитектуры ИС. Архитектурный документ как артефакт проекта. Представление архитектуры N+1(4+1). Документ программной архитектуры (Software Architecture Document, SAD). Архитектурные представления. Архитектурные факторы. Архитектурные решения. Применение и оформление диаграмм UML в документации проекта. Государственный стандарт ГОСТ Р 57100-2016. Использование этого стандарта.

**Литература:** [Ларман], [Крачтен], [ГОСТ-1]

**Тема 10 Нефункциональные требования в архитектуре ИС. Объектный язык ограничений OCL.**

Нефункциональные требования. Эффективность, сложность, масштабируемость, расширяемость, адаптируемость, безопасность, зависимости системы. Учёт нефункциональных требований в проекте архитектуры ИС. Трассируемость требований в архитектуре при проектировании.

Объектный язык ограничений OCL. Возможности OCL и область применения. Синтаксис и типы выражений. OCL в диаграммах UML.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Соммервилл], [Леффингуэлл и др.]

**Тема 11 Лучшие практики разработки информационных систем. CASE-средства визуального моделирования. Прямое и обратное проектирование.**

Лучшие практики разработки ИС. Назначение лучших практик. Связь практик с визуальным моделированием. Инструментальная поддержка практик. Понятие компонента архитектуры ИС. Интерфейс компонента. Версионный контроль основных артефактов проекта. Понятие конфигурации.

CASE-средства визуального моделирования. Прямое и обратное проектирование. Различия рисования и визуального моделирования. Репозиторий CASE-средства и синхронизация его содержимого на UML диаграммах. Понятие каркасного кода при прямом проектировании. Сравнение возможностей CASE-средств визу-



ального моделирования.

**Литература:** [Ларман], [Арлоу и др.], [Крачтен], [Соммервилл], [Taylor et al.], [Martin]

## **Тема 12 Архитектурные стили, шаблоны. Плюсы и минусы различных архитектурных стилей.**

Понятия архитектурного стиля, шаблона, фреймворка. Отличия архитектурного шаблона от шаблона проектирования. Популярные стили и шаблоны архитектуры ИС. Шаблон «Слой». Трёхзвенная (многозвенная) архитектура. Шаблоны «модель-представление-контроллер» (MVC), «сенсор-вычислитель-контроллер» (SCC). Стили «программа и подпрограммы», «виртуальная машина», «клиент-сервер», «каналы-и-фильтры», «классная доска», «издатель-подписчики», «событийная архитектура», «пиринговая архитектура», «конечный автомат». Плюсы и минусы различных архитектурных стилей. Соответствие стиля задаче. Описание и оценка архитектурного стиля. Архитектурные фреймворки: 4+1, TOGAF, RM-ODP, SOMF.

**Литература:** [Taylor et al.], [Martin], [Соммервилл], [Гамма и др.], [ГОСТ-1]

## **Тема 13 Шаблоны проектирования GRASP. Шаблоны проектирования GoF. Адаптация и поддержка архитектуры ИС. Тестирование и рефакторинг.**

Шаблоны проектирования GRASP. Шаблоны: Creator, Information Expert, Low Coupling, Controller, High Cohesion, Polymorphism, Pure Fabrication, Indirection, Protected Variations. Шаблоны проектирования GoF. Шаблоны: Adapter, Factory, Singleton, Strategy, Composite, Facade, Observer и другие. Связь между шаблонами проектирования и архитектурными стилями. Применение шаблонов проектирования. Принципы проектирования: SRP, OCP, LSP, ISP, DIP.

Адаптация и поддержка архитектуры ИС. Тестирование и рефакторинг при проектировании архитектуры. Итеративность и инкрементальность при проектировании архитектуры ИС. Элементы гибких процессов проектирования.

**Литература:** [Ларман], [Соммервилл], [Гамма и др.], [Fowler], [Martin]

## **Тема 14 Реализация и развертывание ИС. Рабочий поток реализации. Влияние особенностей аппаратного обеспечения на программную архитектуру. Диаграммы развертывания языка UML.**

Реализация и развертывание ИС. Рабочий поток реализации унифицированного процесса. Учёт особенностей реализации и развертывания при проектировании архитектуры ИС. Связь программного и аппаратного обеспечения. Влияние особенностей аппаратного обеспечения на программную архитектуру. Артефакты реализации. Диаграммы развертывания языка UML. Применение диаграмм развертывания для проектирования процесса развертывания и адаптации ИС.

**Литература:** [Арлоу и др.], [Taylor et al.], [Martin], [Соммервилл]

## **Тема 15 Архитектурные стили сетевых и распределенных программных систем.**

Проектирование архитектуры сетевых и распределенных программных систем. Архитектурный стиль «пиринговая архитектура» (peer-to-peer). Архитектурный стиль Representational state transfer (REST). Связь архитектурного стиля



REST и устройства всемирной сети Интернет. Архитектурный шаблон «Удаленный вызов процедур» (RPC). Веб-сервисы. Стил «Сервис-ориентированная архитектура приложений» (SOA). Проектирование архитектуры мобильных и встроенный приложений.

**Литература:** [Taylor et al.], [Martin]

## **Тема 16 Проектирование защищенных ИС. Проектирование и разработка программной системы — командная работа.**

Защищенность как ключевое нефункциональное требование к ИС. Проектирование защищенных ИС. Принципы и шаблоны проектирования архитектуры для обеспечения защищенности программной системы. Контроль доступа.

Проектирование и разработка программной системы — командная работа. Роли и распределение задач в команде. Значение организации процесса при проектировании.

Ещё раз о лучших практиках разработки и проектирования: гибкость, итеративность, инкрементальность процесса. История создания и развития языка UML. Группа Object Management Group (OMG). Заключительные замечания.

**Литература:** [Taylor et al.], [Martin], [Соммервилл]

## **8 Образовательные технологии**

Проводятся лекции для потока и практические занятия (семинары) в группах. Для лучшего изучения материала на практических занятиях предлагаются задачи, некоторые из которых выполняются прямо на занятиях, другие задания выдаются для самостоятельного закрепления материала. Задания выполняются с использованием инструментальных CASE-средств, описанных в разделе 10.4 данной программы. Используется система LMS.

## **9 Оценочные средства для текущего контроля и аттестации студента**

### **9.1 Тематика заданий текущего контроля**

Примерное условие задачи для проектной работы, экзамена и домашних работ.

Пусть задана постановка задачи (*примеры формулировок задач — см. Приложение*).  
Необходимо:

1. выявить требования к информационной системе в виде модели прецедентов (сценариев использования) системы и построить соответствующие диаграммы на UML;
2. определить классы модели предметной области и построить соответствующие диаграммы на UML;
3. подготовить спецификацию заданного прецедента по известному шаблону, включая основной и альтернативные потоки событий, предусловия и постусловия;
4. идентифицировать классы-участники реализации заданного прецедента — граничные классы, управляющий и классы сущности, — и распределить по ним обязанности для каждого потока событий прецедента, построив соответствующие диаграммы взаимодействия на UML;



5. построить диаграмму классов на UML для классов-участников реализации прецедента.

**Примечание:** При оценке задания основное внимание уделяется — соответствию построенного проекта ИС условию задачи (требования выявлены верно), точности соблюдения процесса проектирования, использованию шаблонов проектирования, а также совместимости отдельных диаграмм и представлений между собой.

## 9.2 Вопросы для оценки качества освоения дисциплины

Примерный перечень вопросов к итоговому контролю (экзамену) для самопроверки:

1. Сервер приложений в архитектуре ИС. Построение ИС без сервера приложений.
2. Многозвенная архитектура ИС. Ее представление в проекте.
3. Итерационная и каскадная разработка ИС. Правила организации итераций. Назначение фаз при разработке ИС.
4. Управление требованиями, основные создаваемые документы и их содержание.
5. Лучшие практики разработки ИС.
6. Назначение лучших практик разработки программного обеспечения.
7. Связь лучших практик разработки ПО с визуальным моделированием.
8. Программные инструменты для поддержки лучших практик разработки ПО.
9. Компонент, использование компонентов при проектировании ПО
10. Конфигурация сборки ПО.
11. Понятие архитектуры ИС. Архитектурные представления ПО.
12. Формирование требований к информационной системе.
13. Функциональные требования к ПО. Приведите примеры.
14. Атрибуты качества ПО. Приведите примеры.
15. Документы, которые создаются при определении требований к ПО, их содержание.
16. Модель предметной области.
17. Шаги процесса формирования визуальной модели ПО.
18. Кто использует регистрируемые события и объекты?
19. Связь модели предметной области с задачей идентификации функциональных требований.
20. Какие UML диаграммы создаются при создании модели предметной области, перечислите какие используются шаблоны проектирования?
21. Понятие состояния объекта и его визуальное представление на UML.
22. Как можно определить экторов ИС по модели предметной области?
23. Каковы подходы к идентификации первичных и вторичных экторов?
24. Эскиз и спецификация прецедента. Их состав, область применения, в чем сходство и различия?
25. Выявление прецедентов в бизнес-процессах.
26. Типовая структура спецификации прецедента.
27. Основной и альтернативные потоки прецедента.
28. Шаблоны выявления прецедентов ИС.
29. Учет бизнес правил в прецедентах.
30. Предусловия и постусловия прецедента.
31. Визуальное представление на UML модели прецедентов.
32. Процесс выявления требований к ИС.
33. Заинтересованные лица, свойства ИС, классификация свойств.
34. Формулировка проблемы, решаемой ИС в концепции ИС.
35. Матрица трассировки требований и ее использование при работе с требованиями.
36. Обзор дисциплины «Анализ и проектирование» – входные и выходные артефакты (рабочие материалы).
37. Роли и задачи дисциплины «Анализ и проектирование». Назначение и задачи этапов.
38. Используемые диаграммы UML в задачах дисциплины «Анализ и проектирование».
39. Задача «Архитектурный анализ». Роль и выполняемые шаги.
40. Ключевые абстракции и их идентификация.
41. Диаграммы UML для визуализации ключевых абстракций.





42. Архитектурные механизмы, их назначение.
43. Архитектурные шаблоны и их использование.
44. Понятие слоя и его представление на UML.
45. Структура «реализация сценария использования» (use case realization) и ее представление в визуальной модели.
46. Понятие класса и объекта. Диаграмма классов на UML.
47. Отношения между классами и их визуальное представление.
48. Понятия навигации, наследования, зависимости.
49. Отличие агрегации от композиции.
50. Понятие кратности ассоциации.
51. Шаблоны создания и преобразования диаграмм классов на UML.
52. Задача «Анализ прецедентов». Роль и выполняемые шаги.
53. Концептуальные классы, их стереотипы и назначение в визуальной модели.
54. Диаграммы взаимодействия UML и их назначение.
55. Диаграмма последовательности UML и ее использование в задаче.
56. Понятие ответственности класса.
57. Диаграмма классов в задаче и ее связь с диаграммой коммуникаций.
58. Диаграмма состояний на UML. Понятие состояния объекта.
59. Шаблоны выявления состояний.
60. Специальные состояния – начальное и конечное. Супер состояние и его использование.
61. Условия на диаграмме состояний.
62. Правила перехода из состояния в состояние и их запись на диаграмме.
63. Задача «Идентификация проектных элементов». Роль и выполняемые шаги.
64. Понятие подсистемы и ее представление на UML.
65. Подсистема и компонент.
66. Понятие интерфейса и его представление на UML.
67. Понятие пакета и правила их использование для структуризации визуальной модели.
68. Зависимости пакетов и правила их выявления.
69. Задача «Идентификация проектных механизмов». Роль и выполняемые шаги.
70. Представление механизмов в визуальной модели.
71. Задача «Проектирование сценария использования». Роль и выполняемые шаги.
72. Задача «Проектирование подсистем». Роль и выполняемые шаги.
73. Построить модель предметной области по постановке задачи.
74. Найти сценарии использования ИС по постановке задачи.
75. Построить модель анализа по заданному сценарию использования.
76. Диаграмма деятельности на UML.
77. Представление архитектуры N+1.
78. Типичные разделы документа программной архитектуры.
79. Понятие архитектурного стиля.
80. Архитектурные стили и их особенности.
81. Шаблоны проектирования GRASP и GoF. Их связь с архитектурой ИС.
82. Рабочий поток реализации.
83. Диаграммы развертывания UML.
84. Архитектурные стили сетевых и распределенных ИС. REST-архитектура.
85. Защищенность как ключевое нефункциональное требование к ИС.





## 10 Учебно-методическое и информационное обеспечение дисциплины

### 10.1 Базовый учебник

[Ларман] *Ларман Крэг*. Применение UML 2.0 и шаблонов проектирования, 3-е изд. — Пер. с англ. — М.: ООО «И. Д. Вильямс», 2018. — 736 с.

### 10.2 Основная литература

[Арлоу и др.] *Арлоу Джим, Нейштадт Айла*. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование, 2-е издание. — Пер. с англ. — СПб.: Символ-плюс, 2018. — 624 с.

[Крачтен] *Крачтен Филипп*. Введение в Rational Unified Process. — М.: Вильямс, 2002.

[Леффингуэлл и др.] *Леффингуэлл Дин, Уидриг Дон*. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. — Пер. с англ. — М.: Вильямс, 2002. — 448 с.

[Taylor et al.] *Taylor Richard, Medvidovic Nenad, Dashofy Eric*. Software Architecture: Foundations, Theory, and Practice. — Wiley, 2010. — 712 p.

### 10.3 Дополнительная литература

[Fowler] *Fowler Martin*. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3-rd edition. — Addison-Wesley Professional, 2003. — 208 p.

[Гамма и др.] *Гамма Эрих, Хелм Ричард, Джонсон Ральф, Влссидес Джон*. Приемы объектно-ориентированного проектирования. Паттерны проектирования. — Пер. с англ. — СПб.: Питер, 2014. — 366 с.

[Кумсков] *Кумсков М. И*. Базы Данных и процессы их создания. Введение. — М.: Мехмат МГУ, 2004. — 136 с.

[Martin] *Martin Robert*. Clean Architecture: A Craftsman's Guide to Software Structure and Design. — Prentice Hall, 2017. — 432 p.

[Соммервилл] *Соммервилл Иан*. Инженерия программного обеспечения. 6-е изд. — Пер. с англ. — М.: Вильямс, 2002. — 624 с.

[ГОСТ-1] Государственный стандарт ГОСТ Р 57100-2016. Системная и программная инженерия. Описание архитектуры.

### 10.4 Программные средства

Практические задания дисциплины могут выполняться с использованием следующих CASE-средств:

- Sparx Enterprise Architect версии 12 и новее (<https://sparxsystems.com>),
- Modelio версии 3 и новее (<https://www.modelio.org/>).



## Примеры задач для выполнения контрольных и домашних заданий, проектной работы

### Задача 1. «Автоматизация поликлиники»

Районный отдел здравоохранения принял решение разработать ИС для учета заболеваемости в районе, загруженности медицинского персонала в поликлиниках и в районной больнице. Информационная система должна поддерживать ответы на следующие запросы (за текущий период):

- Сколько пациентов обратилось в поликлиники района?
- По каждому типу болезни, сколько новых диагнозов поставлено?
- Сколько больничных листов выдано по данному типу заболевания?
- Сколько койко-мест свободно (занято) в каждом отделении районной больницы на данное число?
- Какова загруженность врачей данной специальности по поликлиникам района?

В поликлиниках ведется запись пациентов к врачам-специалистам, ИС должна поддерживать эту работу регистратуры – учитывать приемы у врача. На приеме у врача пациенту ставится диагноз по его жалобе. На приеме пациенту может быть выписан (открыт) или закрыт больничный лист, также может быть выписано направление на лечение в районную (областную) больницу. По результату приема пациента медицинская сестра в регистратуре (по документам, предоставленным врачом, в конце рабочего дня) или сам врач (в своем кабинете) вносит в ИС данные о приеме данного пациента.

По направлению на лечение проводится прием в больницу, пациент размещается в соответствующем (диагнозу) отделении больницы, ему выделяется больничная койка (койко-место). После прохождения лечения пациент выписывается из больницы и направляется в поликлинику, где ему закрывают (или продлевают) больничный лист. ИС должна регистрировать поступление и выписку пациентов по отделениям больницы.

### Задача 2. «Комбинат питания»

Предприятие владеет сетью небольших кафе и ресторанов – пунктов питания (ПП), - по всему городу. Руководство решило ввести централизованный учет движения денежных средств, включая возможность расчета средней стоимости чека как характеристики пункта питания.

Основной услугой ПП является простое обслуживание клиента. Клиент заказывает, используя меню, состав своего обеда (ужина) как набора блюд и напитков, официант проводит обслуживание, выписывает чек, клиент оплачивает чек (наличными) и



по своему усмотрению может оставить чаевые официанту. Каждый ПП самостоятельно проводит закупку продуктов для своих нужд. Согласно рецепту на приготовление каждого блюда расходуется определенный состав продуктов (мясо, рыба, овощи, фрукты, специи, масла...). За состав блюд и рецептуру отвечает шеф-повар данного кафе. Себестоимость блюда складывается из состава продуктов по рецепту и времени повара и его помощников при приготовлении блюда. Цена блюда определяется как его себестоимость умноженная на ценовой коэффициент, фиксированный для данного ресторана (кафе).

Рестораны и кафе могут проводить (по предварительному заказу клиента) «корпоратив»: празднование юбилея, организацию свадебного ужина, праздника, корпоративного мероприятия. В этом случае кафе закрыто для обычных посетителей и арендная плата включается в себестоимость мероприятия. Для каждого корпоратива составляется собственный список блюд, под который специально закупаются продукты и напитки. Кроме стоимости блюд и аренды помещения стоимость корпоратива определяется составом официантов, обслуживанием гостей, и составом поваров, проводящих приготовление блюд. Корпоратив при заказе оплачивается авансовым платежом в кассу ПП, а по завершения мероприятия клиенту (оформлявшего корпоратив) выставляется счет за полный состав услуг и блюд, который клиент оплачивает (за вычетом суммы аванса).

Внедряемая информационная система (ИС), имеющая «клиентские терминалы» в каждом кафе (ресторане), должна вести учет продуктов (для каждого ПП), потребляемых клиентами блюд, числа обслуженных клиентов, числа проводимых корпоративов и т.п. ИС должна иметь возможность формирование различных отчетов, включая расчет среднего чека по ПП, средней стоимости корпоратива, занятости персонала в ПП и на корпоративах, а также доходов и расходов как по ПП, так и по периодам времени.

### **Задача 3. «Документы муниципалитета»**

Руководству муниципалитета требуется автоматизировать работу своей канцелярии в плане контроля исполнения решений, принятых по письмам граждан. Новая информационная система «ДОКУМЕНТ» должна предоставлять информацию о полученных письмах и запросах граждан, о принятых по ним решениях, об ответах и их датах, о постановке решения «на контроль». На любой запрос (письмо, жалоба, обращение, заявление и т.д.) сотрудники муниципалитета обязаны ответить в течении 10 календарных дней.

Система «ДОКУМЕНТ» должна вести регистрацию входящих писем, их классификацию (по внутренним правилам), назначение отдела, которому поручено заниматься письмом, назначение контрольного срока исполнения ответа. Класс письма присваивает письму канцелярия при его регистрации как входящего документа. Каждый отдел отвечает за обработку писем заданных ему классов.

После регистрации в системе «ДОКУМЕНТ» бумажные документы – с соответствующей пометкой: номер и дата, - передаются в отделы для обработки. В отделе его руководитель назначает исполнителя ответа, - одного из своих сотрудников, - и заносит это



решение в систему «ДОКУМЕНТ». Начальник может просмотреть в системе «ДОКУМЕНТ» исполнение поручений по датам, сотрудникам, видам писем и распечатать соответствующий отчет.

По документу назначенный сотрудник готовит ответ, текст которого сохраняется в системе. После занесения ответа в систему канцелярия распечатывает ответ на бланке муниципалитета и подписывает документ в соответствующем отделе. После подписи ответ регистрируется как выходящий с номером и датой и передается в экспедицию муниципалитета для доставки на почтовое отделение. Сотрудник делает пометку в системе «ДОКУМЕНТ» о выполненном поручении по данному письму, и канцелярия подтверждается «закрытие письма». Исходные (поступившие) письма в систему не заносятся – ведется только учетная информация о их движении по отделам. Кроме простого письма-ответа возможны следующие действия по документам.:

- Подготовлен проект решения «необходимо проведение мероприятия муниципалитетом»,
- Подготовлен проект решения «необходимо проведение мероприятия смежными организациями».

Проект решения выносится на очередное заседание муниципалитета начальником соответствующего отдела. Если решение принято, то планируемые мероприятия по письмам должны быть поставлены канцелярией на контроль с указанием ответственного лица, на контроль ставится и само письмо в новой системе. Документ – обращение гражданина, - снимается с контроля при успешном завершении соответствующего мероприятия. Об этом событии готовится ответ заявителю, чье письмо послужило причиной назначения мероприятия. Ответ готовит тот же, сотрудник, который подготавливал первый ответ по письму и проект решения по проведению мероприятия.

Список сотрудников и список отделов новая система «ДОКУМЕНТ» должна брать из уже существующей кадровой системы муниципалитета «Кадры». Информация о планируемых мероприятиях (по которым приняты решения муниципалитета) передается в существующую систему «Поручения и проекты» муниципалитета, где ее средствами отслеживаются этапы мероприятия/проекта. После завершения мероприятия по системе «Поручения и проекты» канцелярия отслеживает это событие и снимает с контроля письмо в системе «ДОКУМЕНТ». Руководство муниципалитета имеет доступ к системе для получения отчетов и проведения оперативного поиска. Доступ пользователей к системе авторизован.

#### **Задача 4. «Театральные кассы»**

Предприятие владеет сетью театральных касс, распространяющих (по договору с владельцами кинотеатров и концертных залов) билеты на зрелищные мероприятия, проводимые в театрах, кинотеатрах, концертных залах (зрелищных помещениях – ЗП) районного центра Московской области. Кассы расположены как в самих театрах и кинотеатрах, так и в специальных бутиках, расположенных по всему городу в людных местах. По договору с владельцами ЗП предприятие распространяет (продает) билеты на мероприятия.



Театры имеют свой собственный репертуар, в кинотеатрах показ одного и того же фильма может проводиться сразу в нескольких местах. Не всегда кинотеатр или зрительный зал заполняется полностью – часть билетов может остаться не проданной. Однако при гастролях популярных артистов или театральных коллективов все билеты распродаются задолго до начала мероприятия. Театральные кассы могут принимать по телефону предварительный заказ на бронирование билетов на то или иное мероприятие. Стоимость бронирования составляет 10% от стоимости заказанных билетов. Забронированные билеты должны быть выкуплены в кассе театра или кинотеатра за 30 минут до начала мероприятия. Государственные учреждения города имеют постоянную бронь на все мероприятия в размене 15% мест. Эта бронь и условия ее использования определяются договором между мэрией и предприятием. На высоко популярные мероприятия (например, гастроли звезд эстрады, оперы, балета) этот процент брони может быть доведен до 60% по «просьбе» из мэрии города. Билеты этой брони (невыкупленные) начинают продавать за 30 минут до начала мероприятия. В этом случае наценка за бронь не взимается.

Руководство предприятия решило автоматизировать учет проданных билетов по ЗП города на различные виды зрелищных мероприятий, а также ввести централизованный учет бронирования билетов. Если в кинотеатре имеется несколько зрительных залов, то каждый из них является зрелищным помещением (ЗП), подлежащим учету (с собственным составом зрительских мест и собственным составом киносеансов).

Состав мероприятий определяется владельцем (управляющим) зрелищного предприятия - ЗП. Один из важных характеристик эффективности управления ЗП, является среднемесячный доход с одного места зрителя по всем ЗП одного владельца. Состав ЗП в городе достаточно динамичен – появляются новые небольшие ЗП, а большие старые залы преобразуются в несколько залов (ЗП).

Информационная система должна вести учет проданных билетов по помещениям и иметь возможность формирования отчетов по доходам от продажи билетов на кинопоказы, театральные представления, концерты. В перспективе, эта ИС будет использоваться рекламным агентством – партнером предприятия, - для формирования показателей эффективности рекламных компаний. Эти показатели продажи билетов в ЗП тех районах города, где проводилась рекламная компания (например, премьеры кинофильма) на наружных рекламных площадках данного района города, в местных газетах, по радио и местному телевидению.

### **Задача 5. «Таксопарк»**

Фирма «Бешеная черепаха» (г. Одесса) услуги такси по предварительным заказам. При этом часть водителей зачислены в штат и пользуются машинами самой фирмы. Другая часть - обслуживает клиентов на собственных автомобилях, которые сдают часть выручки в фирму за предоставленный заказ. Все водители имеют номер-позывной. Клиенты делают заказ по телефону: сообщают время и дату подачи такси по указанному адресу.



Диспетчер проводит оценку стоимости услуги – определяет базовую цену заказа, - и сразу сообщает ее клиенту. За 20-30 минут клиенту сообщается марка, цвет и номер автомобиля, назначенного на заказ. Когда машина прибывает на место диспетчер сообщает клиенту об этом. Если машина подается с опозданием, то это снижает качество обслуживания и может привести в будущем к отказу клиента от услуг «Бешенной черепахи». Поэтому водителям стараются быть на месте за 10-15 минут до срока. С другой стороны, за простой автомобиля – например, если клиент вышел к машине позже указанного срока, - производится дополнительная оплата. При отсутствии «своих» машин (в часы пик) диспетчер может передать заказ дружественной такси-компании - за фиксированный процент от заказа.

Клиент может изменить маршрут и место назначения поездки. Водитель по прибытию на место сообщает диспетчеру данные о реальном маршруте и о простоях по требованию клиента. Диспетчер проводит перерасчет базовой цены и сообщает ее водителю. Водитель получает указанную сумму с клиента и при запросе – оформляет чек.

Руководство решило автоматизировать процесс учета выполненных заказов и их стоимости. От новой системы требуется предоставление информации по доходам от услуг и по загрузке машин – как собственных, так и внешних водителей. При некачественном обслуживании – не верно указаны данные по заказу и, например, машина пришла не туда, - нужно знать имя диспетчера, принявшего заказ. Нужен учет опозданий машин – кто, когда и почему, - и требуется статистика по случаям передачи заказа в другой таксопарк.

Кроме этого следует решить проблему быстрого автоматического оповещения клиентов о назначенных на его заказ машинах и о том, что машина уже на месте - без участия диспетчера. Для этого будет использоваться смс-сообщение на мобильный телефон клиента (если указан мобильный телефон) и требуется формирование автоматического голосового сообщения при автонаборе телефонного номера клиента – при формировании, что машина уже подана. Для отправки смс следует использовать интерфейс, предоставленной компанией «Лайф-Телекон», с которой заключен соответствующий договор. Таксист может использовать свой мобильный телефон (с номером «Лайф-Телекон») для доступа к данным по «свободным» заказам, и он может с телефона указать «свое согласие» взять конкретный заказ.

### **Задача 6. «Информационные материалы»**

Корпорация «Знания-сила» занимается сдачей в аренду информационных материалов (ИМ) (аудио-книг, DVD/CD-дисков содержащих аналитические материалы ряда государственных агентств) через разветвленную сеть «Точек Контакта» - (ТК). Материалы имеют внутрифирменные грифы «важности» и выдаются исключительно юридическим лицам, согласно их статусу и привилегиям доступа к материалам с данным грифом. До сих пор учет ИМ велся на бумажных носителях. Масштабы бизнеса выросли и появилась потребность автоматизировать учет движения ИМ - от момента получения копии ИМ от поставщика (от гос. агентства), до их списания, - с учетом всех «пунктов





перемещения ИМ» и ведения списка заказчиков, имевших доступ к ИМ. Вам поручено разработать для этого учета информационную систему «Еще не вечер».

Одной из целей руководства при автоматизации своих работ является выявление предпочтений организаций-заказчиков в различных регионах по видам ИМ (ТК расположены в различных городах, а в разных районах одного и того же города возможен различный «спрос» на ИМ одного типа). Для этого будет использоваться ИС центрального офиса. Задача системы «Еще не вечер» сбор текущей информации о выдаваемых ИМ в ТК и передача его в центр. Система должна отслеживать положение в фирме конкретного экземпляра ИМ, включая его перемещение из ТК в центр и из центра в ТК.

Каталог, содержащий описание имеющихся в корпорации ИМ и из экземпляров, ведется в центральном офисе на основе существующей системы «БД ИМ», из которой система «Еще не вечер» должна уметь читать необходимые данные. Корпорация «Знания-сила» централизованно заказывает («закупает») экземпляры ИМ у «поставщиков» на основе заявок ТК, и затем, по получению заказа, централизованно распределяет их по ТК. Теперь эти заявки ТК должны вестись через систему «Еще не вечер». Оформление и учет заказов на ИМ, выданных гос. агентствам («поставщикам») ведется через существующую систему «Заказы», которая получает заявки ТК из системы «Еще не вечер» и описания из системы «БД ИМ».

Заказчики через ТК получают на время («берут в аренду») экземпляры ИМ, оплачивают эту аренду и возвращают ИМ до контрольного срока. Просрочка оплачивается по отдельному тарифу. Система «Еще не вечер» должна уметь выявлять должников по ТК – для этого должна быть возможность формировать соответствующие отчеты. Система «Еще не вечер» не должна вести денежную информацию – она учитывает только копии ИМ и их текущее «положение» (на центральном складе, в конкретном ТК, у конкретного заказчика и т.п.)

ИМ ведутся по названиям, темам, агентствам-авторам, каждый ИМ может иметь несколько копий (конкретная кассета/диск). Система должна по ИМ учитывать, где какие копии находятся в настоящий момент. Доступ всех пользователей к системе авторизован.

Клиенты не могут терять ИМ – это документы строгого учета. ИМ может «списываться» (сниматься с учета и уничтожаться) из-за потери или отсутствия актуальности на текущий момент. ТК передает такие ИМ в центральное хранилище корпорации, где они или списываются по акту, или перенаправляются в ТК других регионов. Эти движения ИМ система «Еще не вечер» должна отслеживать. Акты списания ИМ также подлежат учету в системе.

Информация о выданных / сданных / отправленных ИМ будет использоваться в ИС центрального офиса (Центр\_ИС), где она будет агрегироваться и анализироваться средствами этой ИС «Центр\_ИС». Сотрудники могут формировать различные отчеты в системе «Центр\_ИС».



### **Задача 7. «Мастерские авто обслуживания»**

Предприятие владеет сетью автомастерских в Москве и в нескольких городах Московской области. Некоторые из них являются станциями технического обслуживания (СТО) и могут выполнять сервис как отечественных автомобилей, так и иномарок квалифицированными мастерами-автомеханиками. Некоторые мастерские имеют специализацию: кузовной ремонт, шиномонтаж и т.п.

Руководство предприятия приняло решение ввести централизованный учет выполненных работ, автоматизировать учет движения и расхода запасных частей, а также ввести учет трудозатрат сотрудников предприятия. Цель внедрения ИС - сокращение накладных расходов, оптимизация складских запасов материалов и комплектующих, а также оперативный учет движения денежных средств.

Владельцы авто – клиенты - могут обращаться в СТО как по предварительному заказу, так и «в живую», приехав на станцию. Перед началом обслуживания с клиентом проводится согласование состава работ и списка запасных частей и материалов. За все работы и их качество отвечает мастер, назначаемый на сервис данного автомобиля. Мастер должен иметь соответствующий сертификат на обслуживание данной марки авто (АвтоВАЗ, фольксваген, Форд, Дэу, хёндай, Нисан, Митсубиси). После проведения работ клиент принимает авто и проводит оплату сервиса. Оплата проводится только наличными в кассу, клиент получает акт приема-сдачи работ и счет фактуру.

Предприятие имеет централизованный склад, откуда по заявкам СТО материалы пересылаются на станцию под заказ работ для данной марки авто. Каждое СТО и мастерская имеет небольшой собственный склад для хранения расходуемых материалов таких как масла, фильтры, свечи, прокладки и т.п. Периодически СТО высылают заявку на центральный склад о поставке партии расходуемых материалов по списку. Эта заявка выполняется в течение 2-3 дней – материалы поставляются и приходятся на складе СТО.

Предприятие периодически делает крупные централизованные закупки оригинальных запчастей и расходуемых материалов у поставщиков и дилеров. Эти заказы обычно выполняются за три-четыре поставки на центральный склад в течение 2-4-х недель. Оплата поставщику проводится отдельно по каждой поставке материалов, после их получения на центральный склад.

Планируется, что на каждой СТО будет иметь хотя бы одно клиентское место ИС, а сервер базы данных будет размещен в центральном офисе. Информационная система должна отвечать на вопросы о загрузке различных СТО и мастерских, о доходах и расходах по мастерским, по видам работ, по расходуемым материалам и комплектующим - по их движению на предприятии: заказ – поставка – центральный склад – СТО – сервис авто. Таким образом, кроме учета движения материалов ИС должна учитывать движение денежных средств – расходы, доходы - по предприятию и ее станциям (мастерским) и иметь возможность формировать соответствующие отчеты.