

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ  
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»  
ВОЛЖСКИЙ ФИЛИАЛ МАДИ**

УТВЕРЖДАЮ  
Зав. кафедрой ИиТТП  
Петрова А.В.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ  
КУРСОВОЙ РАБОТЫ  
ПО ДИСЦИПЛИНЕ  
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

Направление подготовки  
***09.03.01 «Информатика и вычислительная техника»***

Направленность (профиль)  
***Автоматизированные системы обработки информации и  
управления в отраслях транспортно-дорожного комплекса***

Квалификация  
***Бакалавр***

Кафедра: ИиТТП

Чебоксары

## СОДЕРЖАНИЕ

|                                                           |    |
|-----------------------------------------------------------|----|
| ВВЕДЕНИЕ.....                                             | 4  |
| 1. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ.....                        | 6  |
| 2. СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ..... | 11 |
| 3. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ.....     | 13 |
| ПРИЛОЖЕНИЕ 1.....                                         | 14 |
| ПРИЛОЖЕНИЕ 2.....                                         | 23 |

## ВВЕДЕНИЕ

Важным фактором подготовки специалистов в области информационных технологий является умение программировать, используя современные языки, включающие объектные возможности, знакомство с основными методами и современными технологиями программирования, в том числе с использованием объектных библиотек конкретных языков.

Курсовой работе как одной из форм самостоятельной учебно-исследовательской работы отводится особая роль при формировании компетенции будущего бакалавра. Курсовая работа по дисциплине «Объектно-ориентированное программирование» представляет собой самостоятельную аналитическую и практическую работу студента и показывает знания и навыки работы, приобретенные в ходе изучения данной дисциплины.

**Цель курсовой работы** – систематизация и закрепление теоретических знаний в области объектной декомпозиции задачи и программирования на объектно-ориентированном языке, полученных за время обучения, приобретение опыта проектирования, разработки, отладки, развертывания и использования программных систем с использованием объектно-ориентированного подхода.

### **Задачи курсовой работы:**

- изучить теоретический материал и особенности применения на практике полученных теоретических знаний по дисциплине «Объектно-ориентированное программирование»;
- решить прикладную задачу из конкретной предметной области, создав программный продукт в одной из объектно-ориентированных инструментальных сред;
- результаты выполнения курсовой работы оформить в виде пояснительной записки, в соответствии с предъявляемыми требованиями. Вместе с пояснительной запиской студент предъявляет файл исходного текста

работающей программы (файлы исходных текстов работающих программ) на одном из доступных носителей.

## 1. ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Задание на курсовую работу определяет научный руководитель в соответствии с ее темой. Выполнение задания требует от студента теоретических знаний и умений осуществлять на практике моделирование, проектирование и программирование на основе объектно-ориентированной парадигмы.

Основной раздел работы должен содержать результаты решения конкретной прикладной задачи и отражать выполнение промежуточных заданий на этапах выполнения работы.

### Этап 1.

**Задание.** Определение темы, постановка целей и задач исследования в рамках курсовой работы.

#### Методические указания к этапу 1.

Тематика курсовых работ разрабатывается преподавателем по дисциплине «Объектно-ориентированное программирование», ежегодно дополняется и уточняется (приложение 1). Темы курсовых работ рассматриваются и утверждаются на заседании кафедры. Студент выбирает тему курсовой работы самостоятельно, из предложенного списка. Незначительное изменение темы разрешается только по согласованию с преподавателем.

### Этап 2.

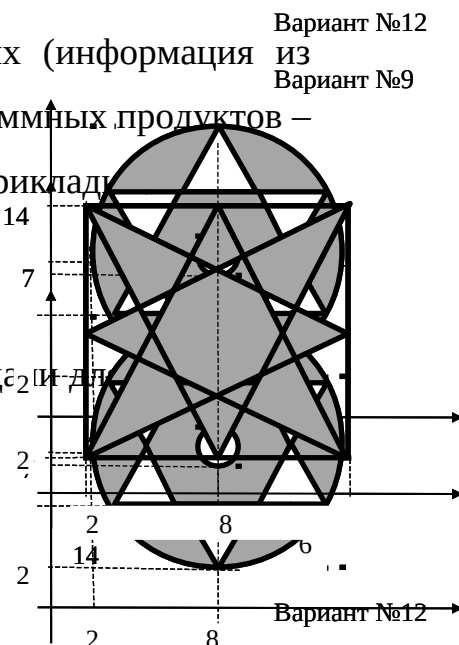
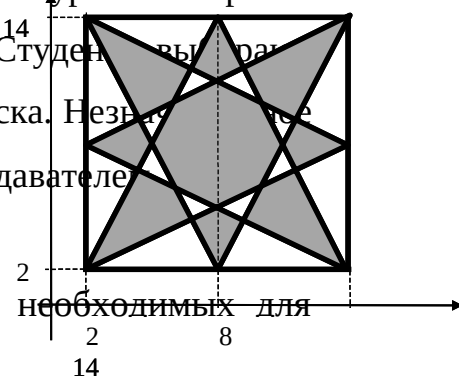
**Задание.** Поиск литературных и других источников, необходимых для выполнения курсовой работы.

#### Методические указания к этапу 2.

Необходимо провести поиск литературных и других (информация из Интернет) источников по теме исследования, а также программных продуктов – аналогов, позволяющих или решающих конкретную прикладную задачу, сформулированную в рамках курсовой работы.

### Этап 3.

**Задание.** Обработка и анализ материала, постановка задачи для программного продукта.



### Методические указания к этапу 3.

Необходимо проанализировать найденные литературные и другие источники, дать их сравнительную характеристику, указав при этом достоинства и недостатки существующих программных продуктов. На основании анализа необходимо составить предварительный список действий, которые будет выполнять разрабатываемый программный продукт.

#### Этап 4.

**Задание.** Выполнить анализ предметной области  $P_2$  в соответствии с вариантом задания на основании объектно-ориентированного подхода.

### Методические указания к этапу 4.

На этапе анализа необходимо выделить основные понятия предметной области, понять сущность и структуру исследуемого объекта, установить суть явлений, процессов и взаимосвязей переменных, характерных для объекта. Необходимо уяснить, в чем состоит проблема, для решения которой планируется создание программного продукта. В процессе анализа предметной области уточняется список действий, составленный на этапе 3, которые будут выполнять разрабатываемый программный продукт. Определяется также перечень действий, недопустимых для исследуемого объекта.

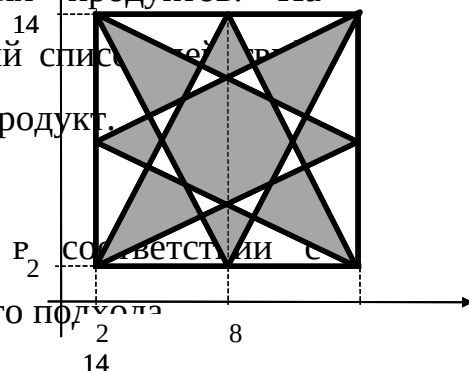
#### Этап 5.

**Задание.** Обосновать и разработать объектную модель предметной области в соответствии с вариантом задания.

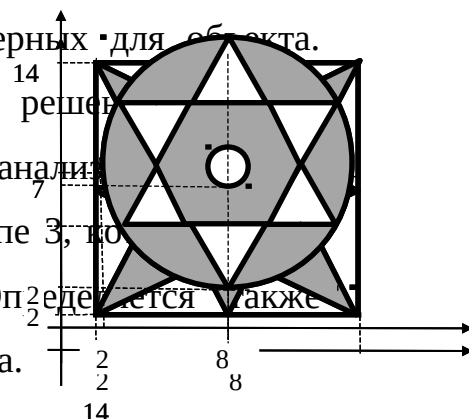
### Методические указания к этапу 5.

Модель предметной области должна представлять иерархию классов, включать в себя классы, соответствующие понятиям предметной области, выделенным на этапе анализа. В ходе выявления объектов из предметной области необходимо установить, какие связи существуют между ними. Интерес представляют два вида отношений: обобщение (отношение между подклассом и суперклассом) и агрегации (отношение между целым и частями). Между классами могут существовать и другие отношения, которые также являются исключительно важными.

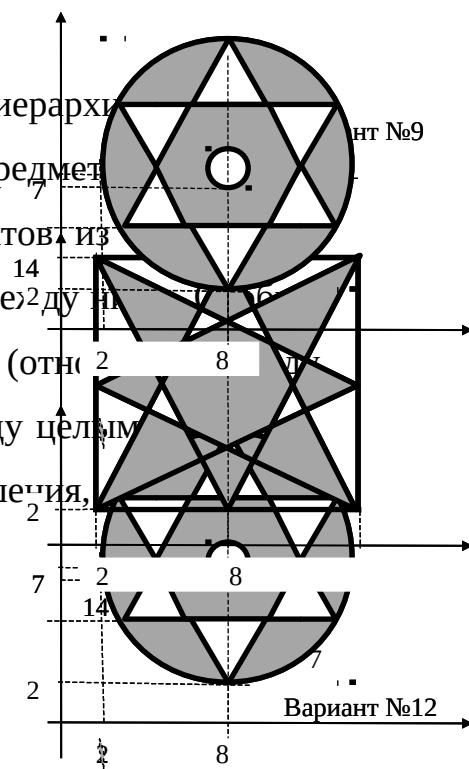
Вариант №9



Вариант №12



Вариант №12



Вариант №12

Классы в рамках объектно-ориентированного подхода имеют атрибуты (данные) и операции (методы, функции, выполняемые объектами). Однако, при моделировании предметной области, атрибуты и операции классов обычно не рассматриваются. На этом этапе следует сконцентрировать внимание на выявлении собственно объектов и отношений между ними.

#### **Этап 6.**

**Задание.** Выполнить проектирование программного продукта, реализующего созданную объектную модель и позволяющего решить задание своего варианта.

#### ***Методические указания к этапу 6.***

Каждый класс модели предметной области, созданной на предыдущем этапе, нужно дополнить описанием необходимых атрибутов и операций. Для атрибутов должны быть определены их типы. Для операций – состав и типы параметров, типы возвращаемых значений. Для всех атрибутов и операций класса необходимо определить область видимости: общедоступный (public), защищенный (protected) или закрытый (private).

Нередко на этапе проектирования появляются дополнительные классы, отсутствующие в модели предметной области. Такие классы позволяют более эффективно и гибко реализовать взаимодействие классов предметной области. Например, для удобства манипулирования объектами некоторого класса предметной области можно на этапе проектирования предусмотреть создание специализированного класса массива, который будет хранить набор объектов предметной области, и будет иметь операции для их сортировки, записи в файл, чтения из файла и т.д.

#### **Этап 7.**

**Задание.** Реализовать созданный проект в виде законченного приложения, используя объектную модель выбранного языка программирования и объектно-ориентированную технологию программирования соответствующей инструментальной среды.

#### ***Методические указания к этапу 7.***



На этом этапе нужно реализовать все спроектированные классы на выбранном языке программирования. Дополнительно нужно создать пользовательский интерфейс к приложению. Готовое приложение необходимо протестировать и исправить обнаруженные ошибки.

#### **Этап 8.**

**Задание.** Проанализировать полученные результаты решения прикладной задачи, сформулировать выводы по курсовой работе.

#### ***Методические указания к этапу 8.***

Следует сравнить полученные результаты с поставленными задачами и определить степень их выполнения и, в конечном итоге, степень достижения цели курсовой работы. На основе анализа результатов выполнения курсовой работы сформулировать выводы. Желательно отметить возможные направления и пути дальнейшего развития и совершенствования разработанного программного продукта.

#### **Этап 9.**

**Задание.** Оформить пояснительную записку к курсовой работе и созданный программный продукт.

#### ***Методические указания к этапу 9:***

Пояснительная записка оформляется в соответствии с требованиями, приведенными в данном методическом пособии, и сдается студентом на кафедру в установленный срок.

#### **Этап 10.**

**Задание:** Защитить полученные результаты выполнения курсовой работы.

#### ***Методические указания к этапу 10.***

На защите в устной форме излагается цель, задачи, использованные методы анализа, моделирования, проектирования и программирования программного продукта. Сообщаются полученные результаты. После устного сообщения студент демонстрирует сеанс работы с программным продуктом, его функциональные возможности.

### **В ходе выполнения курсовой работы студент должен:**

- посещать консультации научного руководителя в соответствии с календарным планом;
- регулярно отчитываться перед научным руководителем о выполнении соответствующих этапов работ;
- пользоваться библиографическим указателем и каталогом, сетью Интернет;
- знать основные методы сбора и анализа материала по теме;
- изучить литературу и другие источники по теме, собрать и проанализировать необходимый материал;
- изложить современное состояние вопроса в виде обзора литературы;
- выполнить анализ предметной области;
- обосновать и разработать объектную модель предметной области;
- спроектировать и реализовать программный продукт, решающий прикладную задачу;
- оформить курсовую работу в соответствии с требованиями.

### **Руководитель курсовой работы должен:**

- определить тему курсовой работы;
- обсудить содержание и структуру курсовой работы;
- проверить полноту библиографического списка, наличие и доброкачественность первичных материалов;
- проконтролировать календарный план выполнения работы и ее оформление;
- проследить уровень владения и использования студентом научной терминологии,
- помочь студенту изложить современное состояние вопроса;
- проверить полноту разработанной студентом объектной модели предметной области и ее адекватность решаемой прикладной задаче;

- проконтролировать качество разработки и реализации студентом программного продукта;
- помочь студенту проанализировать полученные результаты и сформулировать выводы по курсовой работе

## **2. СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ**

В состав пояснительной записки входят:

1. Титульный лист (оформляется по образцу, данному в приложении 2).
2. Содержание (с указанием страниц).
3. Введение.
4. Основной раздел (с разбиением на параграфы).
5. Заключение.
6. Список использованной литературы и других источников.
7. Приложение (Приложения).

**Во введении** определяется актуальность проблемы, предмет (объект), цель и задачи исследования.

### **Основной раздел.**

- 1) Обзор источников и аналогов.

Приводится обзор литературы и других источников, результат поиска аналогов с обязательными ссылками на найденные источники. Изучение источников по теме, как правило, предшествует сбору и анализу материала. Оно должно быть достаточно широким. Нельзя ограничиваться сведениями, почерпнутыми из учебников, обязательно используются монографии и статьи, опубликованные в научных журналах. Поиск литературы осуществляется студентом самостоятельно с помощью каталогов научной библиотеки, библиографических указателей и справочников, сети Интернет. В результате изучения литературы должно формироваться представление о современном состоянии вопроса, устанавливаются имеющиеся противоречия и нерешенные задачи. Обзор должен содержать логичное рассмотрение различных аспектов

темы исследования, выделяются основные теоретические и практические положения. Материал не должен представлять беспорядочное изложение точек зрения различных авторов. В основном используются источники последних 7-10 лет. В резюме выделяются дискуссионные вопросы.

## 2) Анализ предметной области.

При необходимости раздел иллюстрируется необходимым количеством рисунков, фотографий, таблиц, схем, диаграмм и графиков и т.д. Они должны не только иллюстрировать основные положения работы, но и служить наглядными доказательствами и обоснованиями для последующих заключений и выводов.

## 3) Объектная модель предметной области.

Описываются классы, соответствующие понятиям предметной области. Раскрываются их связи между собой и алгоритмы их взаимодействия в процессе решения прикладной задачи. Объектная модель изображается графически в виде диаграммы классов.

## 4) Проектирование программного продукта.

Подробно документируются классы предметной области. Определяется состав их атрибутов и операций. При необходимости добавляются дополнительные классы. Результаты проектирования графически изображаются в виде диаграмм, блок-схем.

## 5) Реализация программного продукта.

Поясняются особенности реализации основных операций классов и используемых алгоритмов. Описывается пользовательский интерфейс и последовательность работы с ним. Описание сеанса работы с программным продуктом целесообразно иллюстрировать копиями соответствующих окон пользователя.

**Заключение** в курсовой работе включает обсуждение полученных результатов, приводятся выводы по работе. Выводы должны полностью соответствовать цели работы и характеризовать ее результаты. Они являются

концентрацией основных положений работы и поэтому не могут развивать идеи, не вытекающие полностью из ее материалов.

В **список использованных источников** включаются обязательно все используемые работы, Интернет-ресурсы и др. по авторскому алфавиту. Список используемых источников оформляется в соответствии с библиографическими требованиями.

**Приложение(я)** включается в работу в случае необходимости. В приложениях приводятся листинги исходных текстов программ с основными комментариям, большие блок-схемы алгоритмов, таблицы, экспериментальных данных, результатов работы программы и т.д., если они занимают слишком много места в основном разделе работы. Материал приложений должен способствовать более четкому изложению материала, иллюстрировать отдельные положения и результаты курсовой работы.

Приложения последовательно нумеруются с названиями: Приложение 1, Приложение 2 и т.д.

### **3. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

Текст пояснительной записки к курсовой работе выполняется шрифтом 14 пт (Times New Roman), выравнивание по ширине, заголовки – жирный шрифт 14, межстрочный интервал – 1,5. Используются листы бумаги формата А4 с полями: левое – 2,5; правое – 1,5; верхнее 2,0; нижнее – 2,0. Текст печатается на одной стороне листа. При компоновке текста с иллюстрациями должно соблюдаться рациональное заполнение страниц. Нумерация страниц выполняется внизу страницы от центра, титульный лист (первая страница) не нумеруется.

Все страницы пояснительной записки скрепляются и оформляются в обложку. К пояснительной записке прикрепляется электронный носитель с файлами программ, представляемых и описанных в курсовой работе.

## ТЕМАТИКА КУРСОВЫХ РАБОТ

### **1. Иерархия типов: АСУ «Заработная плата сотрудников предприятия».**

- а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

ФИО, наименование цеха, зарплата, год рождения, дата поступления на работу, семейное положение, пол, количество детей, дата заболевания, дата выздоровления, оплата по бюллетеню (%), средний заработок.

- б) Определить задачи и реализовать методы для функционирования приложения.
- с) Обеспечить реализацию сортировки и фильтрации по разным полям.
- д) Обеспечить реализацию запросов для поиска по заданным критериям.
- е) Обеспечить реализацию агрегирующих запросов.

### **2. Иерархия типов: АСУ «Телефоны и абоненты».**

- а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

фамилия абонента, год установки телефона, номер телефона, номер телефона – дата, город, с которым был разговор, время разговора, номер телефона – дата – сумма оплаты.

- б) Определить задачи и реализовать методы для функционирования приложения.
- с) Обеспечить реализацию сортировки и фильтрации по разным полям.
- д) Обеспечить реализацию запросов для поиска по заданным критериям.
- е) Обеспечить реализацию агрегирующих запросов.

### **3. Иерархия типов: АСУ «Библиотека».**

- а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

шифр книги, автор, название, год издания, местоположение (номер стеллажа, шкафа и т.п.), количество экземпляров, шифр книги – дата выдачи книги Ф.И.О. читателя, название книги – автор стоимость, читатели-должники

- б) Определить задачи и реализовать методы для функционирования приложения.
- с) Обеспечить реализацию сортировки и фильтрации по разным полям.
- д) Обеспечить реализацию запросов для поиска по заданным критериям.
- е) Обеспечить реализацию агрегирующих запросов.

#### **4. Иерархия типов: АСУ «Продажа авиабилетов».**

- а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

наличие свободных мест.

фамилии пассажиров, номер рейса, маршрут, и дата; время вылета, стоимость и пункт назначения, пункты промежуточной посадки, дни полета, наличие свободных мест.

- б) Определить задачи и реализовать методы для функционирования приложения.
- с) Обеспечить реализацию сортировки и фильтрации по разным полям.
- д) Обеспечить реализацию запросов для поиска по заданным критериям.
- е) Обеспечить реализацию агрегирующих запросов.

#### **5. Иерархия типов: АСУ «Путевого лист для перевозки груза».**

- а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

название груза, категория груза (ящик, мешок, бочка, трубы, доски и т. д.), пожароопасность, вес, тип грузовика, пункт отправления и назначения, номер автомобиля, фамилия водителя, отметка о доставке.

b) Определить задачи и реализовать методы для функционирования приложения.

c) Обеспечить реализацию сортировки и фильтрации по разным полям.

d) Обеспечить реализацию запросов для поиска по заданным критериям.

Обеспечить реализацию агрегирующих запросов.

#### **6. Иерархия типов: АСУ «Обувной магазин».**

a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

артикул, наименование, количество, стоимость одной пары (Артикул начинается с буквы Д для дамской обуви, М для мужской, П для детской), город, название фабрики изготовителя –адрес фабрики, срок поставки обуви в магазин, наличие.

b) Определить задачи и реализовать методы для функционирования приложения.

c) Обеспечить реализацию сортировки и фильтрации по разным полям.

d) Обеспечить реализацию запросов для поиска по заданным критериям.

e) Обеспечить реализацию агрегирующих запросов.

#### **7. Иерархия типов: АСУ «Кафедра».**

a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

код направления, название дисциплины, блок(общетехнический, естественнонаучный, специальный, курс, семестр, признак отчетности (зачет, экзамен, курсовая работа, курсовой проект), преподаватель (кафедра, звание, степень, дисциплина), количество лекционных часов, количество



практических часов, количество часов на лабораторные работы, всего часов).

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

#### **8. Иерархия типов: АСУ «Интернет магазин».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

Товары: категория, модель, производитель, цены (описание, характеристики, внешний вид; магазины, где можно купить товар, их телефоны и адреса; аксессуары, их цены и где их купить).

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

#### **9. Иерархия типов: АСУ «Преподаватели».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

паспортные данные, адрес, телефон, дисциплины, кафедра, должность, звание, учёная степень, число ставок, зарплата, стаж.

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.

е) Обеспечить реализацию агрегирующих запросов.

#### **10. Иерархия типов: АСУ «Отдел кадров предприятия».**

а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

ФИО, паспортные данные, дом. и моб. телефоны, количество детей, возраст детей, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, стаж, направление работы, проект(ы), зарплата, форма зарплаты (почасовая, фиксированная).

б) Определить задачи и реализовать методы для функционирования приложения.

с) Обеспечить реализацию сортировки и фильтрации по разным полям.

д) Обеспечить реализацию запросов для поиска по заданным критериям.

е) Обеспечить реализацию агрегирующих запросов.

#### **11. Иерархия типов: АСУ «Отдел снабжения предприятия».**

а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.

б) Определить задачи и реализовать методы для функционирования приложения.

с) Обеспечить реализацию сортировки и фильтрации по разным полям.

д) Обеспечить реализацию запросов для поиска по заданным критериям.

е) Обеспечить реализацию агрегирующих запросов.

#### **12. Иерархия типов: АСУ «Ремонтная мастерская».**

а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

Заказы: тип заказа (гарантийный ремонт, не гарантийный ремонт), общая стоимость, скидка, товар(ы), их изготовители, модели (марки), серийные номера, описание неисправностей, необходимые ресурсы, клиент, дата получения заказа, срок завершения, дата выставления счёта и его оплаты, метод оплаты, дата поставки, метод и стоимость доставки.

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

### **13. Иерархия типов: АСУ «Магазин бытовой техники».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

название, компания-владелец, её контактные телефоны, адрес, товары и цены на них, описание товаров, основные отделы (категории товаров).

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

### **14. Иерархия типов: АСУ «Аукционы».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

продавцы, покупатели, предметы, аукционы (дата, время и место его проведения, его специфика (картины, ювелирные изделия), номер лота - продавец, стартовая цена, фактическая цена, краткое словесное описание

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

### **15. Иерархия типов: АСУ «Кинотеатры (Афиша)».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

Фильмы: название, описание, жанр (категория), длительность, популярность (рейтинг, число проданных билетов в России и в мире), показывается ли сейчас (сегодня, на текущей неделе), в каких кинотеатрах показывается, цены на билеты (в т.ч. средние).

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

### **16. Иерархия типов: АСУ «Кинотеатры (Размещение и сеансы)».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

Кинотеатры: название, адрес, описание: число мест (в разных залах, если их несколько), акустическая система, широкий экран, цены на билеты: детские и взрослые билеты в зависимости от сеанса (дневной, вечерний) и от категории мест (передние, задние и т.п.); сеансы показа фильмов (дата и время начала).

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

**17. Иерархия типов: АСУ «Ресторан».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

Список блюд по категориям, цена, название, вид кухни, категории (первое, второе, десерт; мясное, рыбное, салат и т. п.), является ли диетическим,

Компоненты блюд: тип (гарнир, соус, мясо и т.п.), калорийность, цена, время приготовления

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.
- e) Обеспечить реализацию агрегирующих запросов.

**18. Иерархия типов: АСУ «Бюро знакомств».**

- a) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

База потенциальных женихов и невест: пол, регистрационный номер, дата регистрации, сведения о себе, требования к партнеру (физические данные, характер, образование и т. д.).

- b) Определить задачи и реализовать методы для функционирования приложения.
- c) Обеспечить реализацию сортировки и фильтрации по разным полям.
- d) Обеспечить реализацию запросов для поиска по заданным критериям.

е) Обеспечить реализацию агрегирующих запросов.

### **19. Иерархия типов: АСУ «Обмен жилья».**

а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

База предложений по обмену: (район и адрес, характеристика дома и квартиры, запрашиваемая стоимость, площадь, планировка, этаж, координаты заявителя) и т. д.; требования к вариантам обмена, наличие положительных и отрицательных факторов. База спроса (требования покупателя к жилью: возможно, несколько вариантов, допустимые диапазоны, допустимая цена жилья, координаты заявителя)

б) Определить задачи и реализовать методы для функционирования приложения.

с) Обеспечить реализацию сортировки и фильтрации по разным полям.

д) Обеспечить реализацию запросов для поиска по заданным критериям.

е) Обеспечить реализацию агрегирующих запросов.

### **20. Иерархия типов: АСУ «Нарушители правил дорожного движения».**

а) Описать класс для формирования типизированного набора данных и класс для приложения.

Основные поля для описания классов:

фамилия водителя, номер автомобиля, пункт правил (от 1 до 10), дата нарушения, признак уплаты штрафа и дата, срок оплаты штрафа, нарушение первичное, повторное

б) Определить задачи и реализовать методы для функционирования приложения.

с) Обеспечить реализацию сортировки и фильтрации по разным полям.

д) Обеспечить реализацию запросов для поиска по заданным критериям.

е) Обеспечить реализацию агрегирующих запросов.

## ПРИЛОЖЕНИЕ 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования

**«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ  
ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»  
ВОЛЖСКИЙ ФИЛИАЛ**

Факультет автомобильных дорог и транспорта  
Кафедра гуманитарные и естественнонаучные дисциплины  
Направление подготовки информатика и вычислительная техника

### КУРСОВАЯ РАБОТА

по дисциплине «Объектно-ориентированное программирование»

**Иерархия типов: АСУ «\_\_\_\_\_»**

Рег. № \_\_\_\_\_

Выполнил: студент группы АСУ-16,  
Иванов И.И.

Дата сдачи «\_\_» \_\_\_\_\_ 20\_\_ г

Руководитель: \_\_\_\_\_

—

