

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**  
**Директор высшей школы**  
**программной инженерии**  
**А.В. Малеев**

**Программа государственной итоговой аттестации  
Подготовка к сдаче и сдача государственного экзамена**

**по направлению:** Программная инженерия  
**профиль подготовки:** Разработка программно-информационных систем  
высшая школа программной инженерии МФТИ - Яндекс  
**курс:** 4  
**квалификация:** бакалавр

**семестр:** 7 (Осенний)

**Программу составил:** А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс  
04.06.2020

## 1. Цели и задачи

### Цели

Целью государственного экзамена является установление уровня подготовки обучающегося по математическим дисциплинам и соответствия результатов освоения обучающимся образовательной программы требованиям образовательного стандарта по направлению подготовки.

### Задачи

- оценка степени освоения обучающимися теоретических положений основных дисциплин, формирующих специальные знания в рамках освоения образовательной программы;
- оценка умения применять полученные знания для решения конкретных задач;
- оценка актуальности полученных студентами знаний и их соответствие требованиям потенциальных работодателей.

## 2. Перечень компетенций, уровень сформированности которых оценивается при проведении государственного экзамена

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.4 Грамотно, логично, аргументированно формирует собственные суждения и оценки
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
УК-6 Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни	УК-6.2 Способен планировать самостоятельную деятельность в решении профессиональных задач; подвергать критическому анализу проделанную работу; находить и творчески использовать имеющийся опыт в соответствии с задачами саморазвития
	УК-6.1 Определяет приоритеты профессиональной деятельности и способы ее совершенствования на основе самооценки
УК-10 Способен формировать нетерпимое отношение к проявлениям экстремизма, терроризма, коррупционному поведению и противодействовать им в профессиональной деятельности	УК-10.2 Знает причины, порождающие экстремизм, терроризм и коррупцию, возможные формы их проявления, принципы (правовые, административные, организационные и др.) противодействия экстремизму, терроризму и коррупции, формирования и реализации политики противодействия экстремизму, терроризму и коррупции, а также основы проведения антикоррупционных действий в различных областях жизнедеятельности
ПК-1 Способен самостоятельно или в качестве члена малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-1.3 Способен готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях
	ПК-1.2 Способен проводить научные исследования самостоятельно или в качестве члена малого научного коллектива
ПК-2 Способен формализовать и алгоритмизировать поставленную задачу	ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу

## 3. Перечень примерных вопросов, выносимых на государственный экзамен

1. Динамический массив. Амортизационный анализ. Учетная оценка времени добавления элемента в динамический массив (с удвоением заполненного буфера).
2. Связные списки. Стек, очередь, дек и их реализации.
3. Быстрая сортировка (QuickSort). Поиск порядковой статистики методом “Разделяй и властвуй” (QuickSelect).
4. Сортировка слиянием (MergeSort). Поразрядные сортировки.
5. Двоичная куча и сортировка кучей (HeapSort). Слияние  $k$  отсортированных массивов с помощью кучи.
6. Хеш-таблица, полиномиальная хэш-функция.
7. Динамическое программирование: общая идея, линейная динамика, матричная, динамика на отрезках.
8. RMQ. Sparse table. Дерево отрезков.
9. LCA: сведение к RMQ и метод двоичного подъёма.
10. Двоичное дерево поиска. Обходы в глубину и в ширину. Поиск ключа, наивные вставка и удаление ключа. AVL-дерево. Красно-чёрное дерево.
11. Декартово дерево. Декартово дерево по неявному ключу.
12. Минимальное остовное дерево: алгоритмы Прима и Крускала.
13. Максимальные потоки в сети. Методы: Форда-Фалкерсона; Эдмондса-Карпа (б/д).
14. Обход графа в глубину, ширину.
15. Поиск кратчайших путей в графе: алгоритмы Дейкстры, Форда-Беллмана, Флойда-Уоршелла.
16. Поиск сильно-связных компонент в графе.
17. Мосты и точки сочленения в графе.
18. Нахождение подстроки в строке: префикс-функция, алгоритм Кнута-Морриса-Пратта.
19. Стандартные контейнеры: `vector`, `deque`, `queue`, `priority_queue`, `set`, `map`, итераторы, компараторы.
20. Бор. Алгоритм Ахо-Корасик.
21. Суффиксное дерево.
22. Вычисление выпуклой оболочки множества точек в 2D и 3D.
23. Триангуляция Делоне. Диаграмма Вороного.
24. Длинная арифметика. Сложение, вычитание, хранение знака, умножение, деление. Алгоритм Карацубы.
25. Постановка задачи обучения с учителем (supervised learning), постановка задачи обучения без учителя (unsupervised learning). Постановка задачи классификации и метрики качества классификации. Наивный Байесовский классификатор.
26. Постановка задачи регрессии и метрики качества регрессии. Линейная регрессия. Теорема Гаусса-Маркова (формулировка). Проблема мультиколлинеарных признаков.  $L1$  и  $L2$  регуляризация, их влияние на веса признаков.
27. Проблема несбалансированных классов. Работа с категориальными признаками и пропущенными значениями. Mean encoding. Примеры простых алгоритмов, решающих стандартные задачи: kNN, наивный байесовский классификатор.
28. Логистическая регрессия. Понятия отступа (Margin). Эквивалентность решений полученных методом максимального правдоподобия и минимизации логистической функции потерь. Логистическая функция потерь, кросс-энтропия.
29. Решение задачи классификации методом опорных векторов. Понятия отступа (Margin). Hinge-loss. Оптимизационная задача в SVM (декларативно).
30. Процедура построения решающего дерева, критерии информативности: энтропийный, Джини. Бустинг (принцип построения ансамбля).
31. Bias-Variance decomposition (декларативно). Процедура bootstrap, алгоритм bagging. Метод случайных подпространств (RSM), Random Forest.
32. Градиентный бустинг. Принцип построения. Какие алгоритмы могут использоваться в качестве базовых алгоритмов.

33. Задача снижения размерности: алгоритмы PCA и t-SNE. Связь PCA и SVD. Может ли PCA давать не единственное решение?
34. Проблема переобучения. Пример, причины возникновения (идейно). Кросс-валидация. Мотивация использования train, val и test выборок. Понятие параметров и гиперпараметров моделей. Процедура подбора гиперпараметров. Понятие регуляризации (в общем случае). Способы регуляризации различных моделей (линейные модели, деревья, ансамбли, нейронные сети).
35. Метод градиентного спуска. Метод обратного распространения ошибки (backpropagation). Функции активации (Sigmoid, tanh, ReLU), их свойства и проблемы. Функции потерь в задаче многоклассовой классификации и регрессии. Методы регуляризации нейронных сетей: Dropout, Batch normalization, data augmentation.
36. Методы регуляризации нейронных сетей: Dropout, Batch normalization, data augmentation. Слабые стороны стохастического градиентного спуска. Способы доработки: Momentum, Nesterov momentum, RMSprop, Adam. Минусы данных подходов.
37. Рекуррентные нейронные сети (RNN) для упорядоченных данных. Основные принципы работы. Проблема затухающего градиента и ее возможные решения. Рекуррентные блоки: наивный (Vanilla RNN), LSTM, GRU, мотивация их использования. Функции активации в рекуррентных блоках.
38. Методы работы с изображениями. Почему линейные слои не получили широкого применения в задаче компьютерного зрения? Сверточные слои в нейронной сети. Мотивация их использования в задачах анализа изображений и сигналов. Одномерные и двумерные свертки (Conv1d и Conv2d). Max & average pooling.
39. Операционные системы и их компоненты. Ядро операционных систем. Системные вызовы и их отличия от обычных библиотечных функций. Способы реализации системных вызовов (прерывания, sysenter, syscall).
40. Целочисленная арифметика в представлении компьютера. Знаковые и беззнаковые значения, способы представления отрицательных значений. Целочисленное переполнение и его контроль. Длинная целочисленная арифметика.
41. Вещественная арифметика. Представления с фиксированной и плавающей точкой. Стандарт IEEE754. Специальные вещественные значения, определенные стандартом IEEE754 и операции над ними.
42. Процессы и потоки. Сходства и различия между ними. Реализация многозадачности и алгоритмы планирование задач в операционных системах.
43. Проблема многопоточной синхронизации. Атомарные переменные и объекты блокировки. Неблокирующие структуры данных и их реализация.
44. Message Passing Interface (MPI) Существующие реализации, задачи MPI как среды программирования. Жизненный цикл MPI программы. Создание и завершение MPI процессов. Организация потока ввода-вывода, параметры указываемые MPI программе.
45. Понятие ускорения и масштабируемости параллельных программ. Закон Амдала. Оценка эффективности параллельных программ. Ярусно-параллельная форма программы.
46. Распределенные файловые системы. Роли элементов системы, обеспечение отказоустойчивости. Алгоритмы чтения и записи в распределенных файловых системах. Репликация данных.
47. Модель вычислений MapReduce. Пары ключ-значение в реализациях MapReduce. Основные стадии вычислений и дополнительные элементы модели.
48. Соединение данных (операция Join) в модели MapReduce. Модель вычислений и оптимизации данных.
49. Итеративные вычисления на больших объемах данных. Модель ленивых вычислений и структура хранения данных в реализации Spark (RDD). Кэширование результатов вычислений и итеративные вычисления.
50. Распределенные диспетчеры сообщений. Репликация и реализация отказоустойчивости. Семантики доставки сообщений.
51. Степени изоляции транзакций. Принципы атомарности, согласованности, изолированности и устойчивости (правила ACID). Применение (commit) и отмена транзакций.

52. Теорема Фишера-Линч-Патерсона (FLP-теорема), CAP-теорема и их применение. Распределенные системы хранения конфигураций.
53. Выбор процесса-лидера или машины-лидера в распределенных системах. Алгоритм консенсуса.

#### **4. Порядок сдачи государственного экзамена**

Государственный экзамен представляет собой итоговый контроль уровня знаний и умений студента, которые он должен продемонстрировать для подтверждения соответствия приобретенных им компетенций нормативным требованиям.

К государственному экзамену допускается обучающийся, освоивший дисциплины, которые покрываются программой экзамена, и не имеющий по ним академических задолженностей.

Формат экзамена - устный.

Экзаменационный билет состоит из двух вопросов, выбор которых осуществляется учащимися из разделов, по одному вопросу из двух разных разделов. Опрос студентов о выборе вариативных разделов проводится не менее чем за одну неделю до даты государственного экзамена.

Экзаменационные билеты разрезаны на несколько частей, сгруппированных по разделам экзамена, студент вытягивает разные части билета из соответствующих стопок. На подготовку к устному ответу студенту отводится 1 астрономический час.

Устная часть экзамена включает в себя ответ студента на вопросы экзаменационного билета. После завершения устного ответа члены ГЭК могут задавать дополнительные и уточняющие вопросы, в том числе требующие дополнительного времени на подготовку.

#### **5. Описание материально-технической базы, необходимой для проведения государственного экзамена**

Аудитория для проведения аттестационного испытания, оснащенная местами для обучающихся и экзаменационной комиссии.

В случае действия ограничительных мер, запрещающих проведение экзамена в присутственной форме - работоспособные дистанционные рабочие места для всех обучающихся и экзаменаторов.

#### **6. Перечень рекомендуемой литературы**

Основная литература

1. Архитектура компьютера, [учеб. пособие для вузов] / Э. Таненбаум, Т. Остин. — Санкт-Петербург, Питер, 2020.— URL: <https://ibooks.ru/bookshelf/361850/reading> (дата обращения: 26.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
2. Наука моделирования и программирования задач математики, информатики и техники из информационных, интеллектуальных и сервисных ресурсов / Е. М. Лаврищева. – Москва: МФТИ, 2023.

Дополнительная литература

#### **7. Рекомендации обучающимся по подготовке к государственному экзамену**

При подготовке к устной части государственного экзамена обучающимся рекомендуется вспомнить темы математических дисциплин, входящие в программу устной части государственного экзамена, используя при необходимости конспекты лекций и рекомендуемую литературу. После повторения каждой темы обучающемуся рекомендуется самостоятельно написать формулировки и доказательства теорем, содержащихся в программе устной части государственного экзамена, без использования литературы и вспомогательных средств.

Если возникают вопросы, которые студент не может самостоятельно решить с помощью рекомендуемой литературы, эти вопросы рекомендуется задать на консультации, проводимой преподавателем кафедры по соответствующей дисциплине.

## **8. Методика и критерии оценки государственного экзамена**

За каждую из двух частей экзамена выставляется отдельная оценка по 10-балльной шкале. Оценки ниже 3 считаются неудовлетворительными и блокирующими: в случае получения неудовлетворительной оценки за любую часть экзамена, экзамен считается сданным на оценку "неудовлетворительно", и до сдачи другой части экзамена обучающийся не допускается.

Ответы на устную часть по каждому из разделов программы оцениваются отдельно по 10-балльной шкале. Оценка за часть экзамена, которая проводится в один день, определяется как среднее арифметическое из оценок за различные разделы. Результат оценки за часть округляется до целого значения по стандартным арифметическим правилам и объявляется обучающемуся в день проведения части экзамена.

Итоговая оценка выставляется по 10-балльной шкале и определяется как среднее значение из двух оценок, полученных за разные части экзамена. Способ округления - стандартный арифметический.

отлично (10) – правильный, четкий и уверенный ответ на оба вопроса билета и дополнительные вопросы;

отлично (9) – даны правильные ответы на оба вопроса билета и дополнительные вопросы с незначительными неточностями;

отлично (8) – даны ответы на оба вопроса билета и дополнительные вопросы после небольших исправлений и наводящих вопросов экзаменаторов;

хорошо (7) – даны ответы на оба вопроса билета, но нет верного ответа на один из дополнительных вопросов;

хорошо (6) – есть недочеты в ответе на один из вопросов билета и нет верного ответа на один из дополнительных вопросов;

хорошо (5) – есть недочеты в ответах на оба вопроса билета и нет верного ответа на один из дополнительных вопросов;

удовлетворительно (4) – есть недочеты в ответах на оба вопроса билета или нет ответа ни на один из дополнительных вопросов;

удовлетворительно (3) – нет ответа на один из вопросов билета, но есть ответы на дополнительные вопросы (возможно с недочетами);

неудовлетворительно (2) – нет ответа на один из вопросов билета и на дополнительные вопросы;

неудовлетворительно (1) – нет ответа ни на один из вопросов билета.

## **9. Особенности проведения государственной итоговой аттестации для инвалидов и лиц с ограниченными возможностями здоровья**

Для обучающихся из числа инвалидов государственная итоговая аттестация проводится с учетом особенностей их психофизического развития, их индивидуальных возможностей и состояния здоровья (далее – индивидуальные особенности).

9.1. При проведении ГИА обеспечивается соблюдение следующих общих требований:

- проведение государственной итоговой аттестации для инвалидов в одной аудитории совместно с обучающимися, не имеющими ограниченных возможностей здоровья, если это не создает трудностей для обучающихся при прохождении ГИА;
- присутствие в аудитории ассистента (ассистентов), оказывающего обучающимся инвалидам необходимую техническую помощь с учетом их индивидуальных особенностей (занять рабочее место, передвигаться, прочитать и оформить задание, общаться с членами ГЭК);
- пользование необходимыми обучающимся инвалидам техническими средствами при прохождении ГИА с учетом их индивидуальных особенностей;
- обеспечение возможности беспрепятственного доступа обучающихся инвалидов в аудитории, туалетные и другие помещения, а также их пребывания в указанных помещениях.

9.2. По письменному заявлению обучающегося инвалида продолжительность сдачи обучающимся инвалидом государственного аттестационного испытания может быть увеличена по отношению к установленной продолжительности его сдачи:

- продолжительность сдачи государственного экзамена, проводимого в письменной форме, – не более чем на 90 минут;
- продолжительность подготовки обучающегося к ответу на государственном экзамене, проводимом в устной форме, – не более чем на 20 минут.

9.3. Обучающийся инвалид не позднее, чем за 3 месяца до начала проведения ГИА подает письменное заявление о необходимости создания для него специальных условий при проведении государственных аттестационных испытаний с указанием особенностей его психофизического развития, индивидуальных возможностей и состояния здоровья. К заявлению прилагаются документы, подтверждающие наличие у обучающегося индивидуальных особенностей (при отсутствии указанных документов в дирекции института).

В заявлении обучающийся указывает на необходимость (отсутствие необходимости) присутствия ассистента на государственном аттестационном испытании, необходимость (отсутствие необходимости) увеличения продолжительности сдачи государственного аттестационного испытания по отношению к установленной продолжительности.

## **10. Примеры контрольных заданий, билетов**

Примеры заданий приведены в приложении

**Примеры вопросов, выносимых на государственный экзамен и типовые контрольные задания.**

1. Предпосылки появления и развития ВС.
2. Технология программирования и основные этапы ее развития
3. Проблемы разработки сложных программных систем
4. Режим реального времени.
5. Основные принципы объектно-ориентированного программирования
6. Информационные модели.
7. Основные принципы объектно-ориентированного программирования
8. Информационные модели.
9. Определение класса. Компоненты классов
10. Классификация ОС
11. Жизненный цикл и этапы разработки программного обеспечения
12. Доступность компонентов класса. Конструкторы, деструкторы
13. Признаки классификации ВС.
14. Ускорение разработки программного обеспечения. Технология RAD
15. Критерии качества программы. Диалоговые программы. Дружественность.
16. Режим разделения времени
17. Управление процессами
18. Нисходящая и восходящая разработка программного обеспечения
19. Структурное и «неструктурное» программирование
20. Дружественные классы
21. Управление памятью
22. Компоненты – данные и компоненты функции
23. Жизненный цикл программы.
24. Определение компонентных функций
25. Сети межсоединений.
26. Наследование классов
27. Классификация языков программирования. Языки высокого и низкого уровня.
28. Основные эксплуатационные требования к программным продуктам
29. Модели представления данных (особенности иерархической, сетевой модели  
1. данных).
30. Принципиальные различия между многомашинными и многопроцессорными  
2. ВС.
31. Стиль оформления программы
32. Режим разделения времени.
33. Правила ввода-вывода для основных структур программирования.
34. Метрика программного обеспечения.
35. Принципы классификации архитектур ВС.
36. Способы записи алгоритма.
37. Методы распределения памяти с использованием дискового пространства
38. Множественное наследование
39. Реляционная модель данных: основные элементы и понятия. Индексирование.  
3. Понятие и характеристика индексов.
40. Программа как реализация алгоритма на языке высокого уровня.
41. Стандартные типы данных.
42. Операции реляционной алгебры: объединение, вычитание, пересечение,  
4. произведение, выборка, проекция.
43. Базовые алгоритмические структуры: следование, ветвление и повторение  
5. (циклы).
44. Особенности однопрограммных режимов работы ВС.
45. Управление вводом-выводом



46. Классификация программных продуктов по функциональному признаку
47. Нормализация данных. Цели нормализации. Нормальные формы.
48. Компьютеры SIMD.
49. Связи. Виды связей. Примеры.
50. Обработка прерываний
51. Объектно-ориентированные языки и системы программирования
52. Совместимость в ВС.

### **Примерные практические задания государственного экзамена**

1. Решить задачу: разработать модуль для работы с векторами вида  $a(a_1, a_2)$ . Реализовать операции сложения, вычитания, нахождения длины вектора. Используя модуль, переписать из данного файла в другой те векторы, длины которых больше заданного  $K$ .
2. Спроектировать базу данных методом нормальных форм или методом «сущность-связь». Указать структуры полученных таблиц, первичные и внешние ключи. Школа – успеваемость (Класс, ученик, предмет, оценка, дата)
3. Решить задачу: удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.
4. Создать приложение "Калькулятор" с использованием технологии RAD.
5. Решить задачу: сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в файле:

Type Stud = Record

fam:string[12];

num:121..126;

otm1, otm2, otm3: 2..5;

end;

Var Kurs: File of Stud;

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости их студентов.

6. Спроектировать базу данных методом нормальных форм или методом «сущность-связь». Указать структуры полученных таблиц, первичные и внешние ключи. Авторынок (Модель, марка, производитель, цена, год выпуска, страна изготовления)
7. Создать WEB страницу с использованием стандартных компонентов.
8. Сдвинуть логически содержимое регистровой пары HL на два разряда влево.
9. Решить задачу: заполнить файл целыми числами из отрезка  $[-1000; 1000]$  с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.
10. Умножить содержимое ячейки памяти  $6000H$  на 5. Результат — в ячейку  $6001H$ .
11. Создать WEB страницу с использованием скриптов.
12. Точно и однозначно сформулировать условие задачи, решение которой приведено ниже.

```

Program Example_File_1;
Var F : File Of Real; G : File Of Integer;
    S : String; A : Real; B : Integer;
Begin
Write('Имя файла? '); ReadLn(S); Assign(F, S);
Write('Имя файла? '); ReadLn(S); Assign(G, S);
ReSet(F); ReWrite(G);
While Not Eof(F) Do
Begin
Read(F, A); B := Trunc(A); B := Ord(Odd(B)) * B; Write(G, B)
End; Close(F); Close(G)
End.

```

13. Непрерывно, начиная с числа 01h, наращивать на единицу содержимое ячейки памяти с адресом 6000h до появления признака переноса. Результат поместить в регистр В.

14. Разработать класс, реализующий понятие множества. Функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками.

Множество реализовать на основе:

- а.) двунаправленного списка (обеспечить «распечатку» содержимого как в прямом, так и в обратном направлениях);
- б.) открытого хеширования (хеш-функцию построить, например, на основе атрибута «цвет», при поиске дополнительно указывать категорию).

15. Спроектировать базу данных методом нормальных форм или методом «сущность-связь». Указать структуры полученных таблиц, первичные и внешние ключи. Мебельный салон (Модель, тип, производитель, материал, цена)

16. Создать приложение с использованием стандартных компонентов

17. Создать приложение с использованием графики.

18. Создать приложение с использованием диалоговых компонент.

19. Разработать класс, реализующий понятие

- а.) «открытого» стека;
- б.) «открытой» очереди;
- в.) «открытого» дека.

Контейнер реализуется на основе массива, память под который выделяется и освобождается динамически, размер массива (максимальная вместимость контейнера) задается как параметр конструктора. «Открытость» контейнера подразумевает возможность просматривать элементы в контейнере, осуществлять поиск (функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками). Объекты класса «фигура» хранятся в контейнере по ссылке (указателю).

20. Сложить число 1510 и 6010 и поместить результат в ячейку с адресом 6000h.

21. Создать приложение с использованием VBA в Excel.

22. Получить дополнительный код числа из ячейки памяти с адресом 6001h. Результат — в ячейку 6002h.

23. Разработать класс, реализующий понятие множества с дубликатами. Функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками. Множество реализовать на основе

а.) двунаправленного списка (обеспечить «распечатку» содержимого как в прямом, так и в обратном направлениях);

б.) открытого хеширования (хеш-функцию построить, например, на основе атрибута «цвет», при поиске дополнительно указывать категорию).

24. На основе списка разработать класс, реализующий понятие

а. «открытого» стека;

б.) «открытой» очереди;

в.) «открытого» дека

(т.е. разрешается просматривать элементы в контейнере, осуществлять поиск).

Функция поиска ищет фигуры с заданными геометрическими и/или графическими характеристиками.

25. Спроектировать базу данных методом нормальных форм или методом «сущность-связь».

Указать структуры полученных таблиц, первичные и внешние ключи. Авиа – самолеты (тип, кол-во мест, цена, дата выпуска, изготовитель, страна изготовления)