Программа вступительного испытания по предмету
«Математическое обеспечение и администрирование информационных систем»
для поступающих на основную образовательную программу магистратуры
«Математическое обеспечение и администрирование информационных систем»
направления подготовки
02.04.03 «Математическое обеспечение и администрирование информационных систем»

Раздел 1. Организационно-методический раздел

Длительность вступительного испытания

180 минут (3 астрономических часа).

Форма вступительного испытания

- Вступительный экзамен проводится в письменной форме без использования каких-либо информационных источников во время экзамена.
- В зависимости от порядка проведения испытания, задание выполняется в электронной форме, либо в простой письменной (от руки) форме.
- В случае выполнения в электронной форме с использованием программных средств, не предоставляющих возможности ввода формул и форматированного текста, математические формулы рекомендуется вводить в формате (La)TeX.
- В случае выполнения экзамена в простой письменной форме, ответы должны быть написаны аккуратно и разборчиво, затем отсканированы или сфотографированы для передачи проверяющим в качестве, не допускающем неоднозначного прочтения написанного.

Структура и содержание вступительного испытания

Экзаменационное задание включает три раздела.

1. **Научно-исследовательская часть.** Абитуриент представляет краткое эссе на тему своей научно-исследовательской деятельности с описанием методологии, проделанной работы и полученных результатов, а также планов на дальнейшую научную работу.
2. **Теоретическая часть.** Абитуриент письменно отвечает на два полученных от экзаменаторов вопроса из разных тем раздела 2. В вопросах, включающих те или иные утверждения, достаточно их формулировок, доказательства приводить не требуется.
3. **Практическая часть.** Абитуриент получает практическую задачу, аналогичную приведённым примерам в Разделе 4, и решает её с использованием любого языка программирования на своё усмотрение.

Рекомендуемый объём по 1 части — не более 500 слов, по 2 части (в совокупности по двум вопросам) — 400–800 слов.

Критерии оценивания вступительного испытания

Вступительное испытание оценивается по шкале от 0 до 100 баллов.

Научно-исследовательская часть (от 0 до 30 баллов):

Критерий	Максимальный балл
Есть четко сформулированная решаемая проблема, указана её актуальность	5
Указаны используемые методы и технологии, релевантные решению обозначенной проблемы	10
Представлены результаты работы, способствующие решению обозначенной проблемы	10
Соответствие тематики и результатов НИР тематике выбранной образовательной программы	5

Итого: от 0 до 30 баллов по данной части.

Теоретическая часть (от 0 до 40 баллов)

Каждый из двух вопросов оценивается следующим образом (от 0 до 20 баллов каждый):

Качество ответа	Количество баллов
Полный корректный ответ	20
Незначительные локальные неточности, опечатки/описки	15
Неточности, не нарушающие ход рассуждения и изложения	10
Неточности, влияющие на ход рассуждения или изложения	5
Ответ, демонстрирующий непонимание смысла вопроса, его тематики	0
Ответ с содержательными (в т.ч. математическими) ошибками	0

Практическая часть (от 0 до 30 баллов)

Критерий	Максимальный балл
Корректность понимания задания, выбор подходящего алгоритма	5
Работоспособность программного кода, в т.ч. на корректных, но вырожденных (например, пустых) входных данных	15
Аккуратность оформления программного кода	10

Раздел 2. Содержание теоретической части экзамена

Тема 1. Математические основы программирования

1. Понятие алгоритма и его уточнения: машины Тьюринга, нормальные алгоритмы Маркова, рекурсивные функции. Эквивалентность данных формальных моделей алгоритмов. Понятие об алгоритмической неразрешимости. Примеры алгоритмически неразрешимых проблем.
2. Понятие сложности алгоритмов. Классы P и NP. Полиномиальная сводимость задач. Формулировка теоремы Кука-Левина об NP-полноте задачи выполнимости булевой формулы. Примеры NP-полных задач, подходы к их решению. Субоптимальные решения неразрешимых проблем и вычислительно сложных задач.
3. Примеры эффективных (полиномиальных) алгоритмов: быстрые алгоритмы поиска и сортировки; полиномиальные алгоритмы для задач на графах и сетях (поиск в глубину и ширину, о минимальном остове, о кратчайшем пути, о назначениях).
4. Алгебра логики. Булевы функции, канонические формы задания булевых функций. Понятие полной системы. Минимизация булевых функций.
5. Исчисление предикатов первого порядка. Понятие интерпретации. Выполнимость и общезначимость формулы первого порядка. Формальные языки и способы их описания. Классификация формальных грамматик. Их использование в лексическом и синтаксическом анализе.
6. λ -исчисление, правила редукции, нормальная формы и правила ее достижения. Каррирование в λ -исчислении, пример использования каррирования в функциональных языках (Haskell или в любом языке семейства ML).

Тема 2. Вычислительные машины, системы и сети

1. Докомпьютерные вычислительные устройства. Первые компьютеры 1930–40-х. Гарвардская (Эйкена) и Принстонская (Фон Неймана) архитектуры ЭВМ. Поколения ЭВМ.
2. Архитектура современных компьютеров. Организации памяти и архитектура процессора современных вычислительных машин. Организация шин. Кэш-память.
3. Конвейеризация, предсказание переходов, спекулятивное и внеочередное выполнение. Конфликты на конвейерах. Особенности конвейеризации для CISC- и RISC-процессоров.
4. Виртуальная память. Адресные преобразования в основе организации виртуальной памяти.
5. Особенности физической архитектуры локальных сетей на примере Ethernet: коаксиальные кабели, витые пары, оптоволоконные сети.
6. Сеть Internet, доменная организация, семейство протоколов TCP/IP версий 4 и 6. Адресация и маршрутизация. Иерархия ISO/OSI на примере протоколов Internet.

Тема 3. Языки и системы программирования. Технология разработки программного обеспечения

1. Языки программирования. Примеры процедурных, функциональных и логических языков программирования. Особенности приведённых в пример языков, решаемые с их помощью задачи. Процедурные языки программирования. Основные управляющие конструкции, структура программы.
2. Работа с данными: переменные и константы, типы данных (булевский, целочисленные, плавающие, символьные, типы диапазона и перечисления, указатели), структуры данных (массивы и записи)
3. Процедуры (функции): вызов процедур, передача параметров (по ссылке, по значению, по результату), локализация переменных, побочные эффекты. Обработка исключительных ситуаций.
4. Языки программирования со статической и динамической типизацией. Сильная и слабая типизация. Языки сценариев.
5. Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы. Понятие об объектном окружении. Рефлексия. Библиотеки классов. Машинно-ориентированные языки, язык ассемблера. Представление машинных команд и констант.
6. Макросредства, макровыводы, языки макроопределений, условная макрогенерация, принципы реализации.
7. Модульное программирование. Типы модулей на примерах «юнитов» (units) C и/или модулей Modula, Pascal, Rust или Golang. Роль транслятора и редактора связей в сборке ПО.

Тема 4. Инструменты разработки ПО

1. Командные оболочки, языки сценариев командных оболочек. Командные оболочки различных семейств операционных систем (Windows, Unix-подобных).
2. Инструменты обработки текстовых данных и файловых систем на примере find, sed, ггер.
3. Системы программирования (СП, IDE), типовые компоненты СП: языки, трансляторы, редакторы связей, отладчики, специализированные текстовые редакторы. Автодополнение, поиск определений, понятие рефакторинга.
4. Встроенная документация на примере Doxygen, Javadoc и/или Python docstrings.
5. Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик. Инструменты генерации лексических и синтаксических анализаторов: lex и yacc (или аналогичные инструменты), ANTLR, парсер-комбинаторы.

6. Системы сборки ПО и управления модулями. Универсальные системы (Make) и ориентированные на конкретные языки (CMake, Cargo, Maven и т.д.)
7. Конфигурационное управление. Системы управления версиями. Централизованное и распределённое управление версиями. Git Workflow, GitHub Workflow. Системы CI/CD.

Тема 5. Методологические вопросы разработки ПО

1. Терминология: «программная инженерия» и «технология программирования». Причины возникновения программной инженерии как науки.
2. Процесс разработки ПО и организация работы команды. Водопадная модель, V-модель, инкрементальная, спиральная модели.
3. Гибкие методологии. Agile-манифест, Scrum Guide. История и основные положения экстремального программирования. Современные гибкие методологии.
4. Управление проектом, понятие проекта. Творческие и индустриальные проекты. Типичные фазы индустриального проекта. Сбор требований, планирование. План.
5. Организационная структура проекта. Отслеживание и оперативное управление проектом. Сдача. Определение циклической разработки (Round-Trip Engineering), варианты реализации.
6. UML CASE-системы: описание функциональности, примеры известных систем.

Тема 6. Базы данных

1. Основные понятия реляционной и объектной моделей данных.
2. CASE-средства и их использование при проектировании базы данных (БД), нотация IDEF.
3. Нормальные формы схемы БД с первой по третью, нормальная форма Бойса-Кодда, пример нормализации схемы
4. Транзакции в СУБД. Блокировка. Взаимная блокировка, сериализуемость транзакций.
5. Язык SQL. Средства определения и изменения схемы БД, определения ограничений целостности. Средства манипулирования данными.

Тема 7. Средства организации высокопроизводительных и распределённых вычислений

1. Параллельное программирование над разделяемой памятью. Парадигмы SPMD, SIMD и MIMD. Стандартный интерфейс OpenMP.
2. Распределённое программирование над распределённой памятью. Стандартный интерфейс MPI и его использование в различных языках программирования.

3. Векторные вычисления с использованием GPGPU и возможностей современных центральных процессоров.

4. Высокоуровневые подходы к обработке больших объёмов данных. Обработка массивов данных с использованием подхода MapReduce. Вычисления на графах с использованием Pregel и аналогичных вычислительных систем.

Раздел 3. Примеры практических заданий

1. Без использования библиотечных контейнеров реализуйте на любом языке программирования с поддержкой классов класс «неизменяемый список». Такой список, будучи единожды сконструированным, не может изменять своё содержимое, тем не менее, в этом классе должны присутствовать операции добавления и удаления элемента по произвольной позиции, получения элемента по позиции. Подумайте, что должны возвращать такие операции и как эффективно использовать память.

2. Дан неориентированный граф в виде матрицы смежности / матрицы инцидентности / списка смежности / списка рёбер (способ задания графа и представление входных данных в программе — на ваш выбор). Напишите программу, проверяющую, связный ли это граф.

3. В БД имеются таблицы с аэропортами, рейсами из этих аэропортов и пассажирами, купившими билеты на эти рейсы.

```
create table airport (  
    id integer primary key,  
    name varchar(100)  
);
```

```
create table flight (  
    id integer primary key,  
    departure_airport_id integer,  
    departure_time timestamp,  
    foreign key(departure_airport_id) references airport(id)  
);
```

```
create table passenger (  
    id integer primary key,  
    name varchar(150)  
);
```

```
create table booking (  
    passenger_id integer,  
    flight_id integer,  
    foreign key(passenger_id) references passenger(id),  
    foreign key(flight_id) references flight(id)  
);
```

Напишите запрос, выдающий количество пассажиров, зарегистрированных на рейсы, отправляющиеся из аэропорта с заданным названием.

Раздел 4. Информационные источники

Список обязательной литературы

1. Ахо, Сети Р., Ульман Дж. Компиляторы: принципы, техника реализации и инструменты. М., 2001.
2. Вигерс К.И. Разработка требований к программному обеспечению. Русская редакция, 2004.
3. Воеводин В.В., Воеводин Вл. В. Параллельное программирование. СПб.: БХВ-Петербург, 2002.
4. Дейт К.Дж. Введение в системы баз данных. М.: Вильямс, 1999.
5. Гласс Г., Эйбле К. Unix для программистов и пользователей. СПб.: БХВ-Петербург, 2004.
6. Кознов Д.В. Основы визуального моделирования. ИНТУИТ.РУ, БИНОМ. Лаборатория знаний, 2008.
7. Кнут Д. Искусство программирования. Т. 1 – 3. М., СПб., Киев: ИД «Вильямс», 2000.
8. Кузнецов С.Д. Базы данных: языки и модели. Учебник. М.: Бином-Пресс, 2008.
9. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы, построение и анализ. М.: МЦНМО, 2000.
10. Липаев В.В. Программная инженерия. Методологические основы. М.: Государственный Университет – Высшая школа экономики, 2006.
11. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы, СПб.: Питер, 2000.
12. Сомервилл И. Инженерия программного обеспечения. М.: Вильямс, 2002.
13. Стивенс Р., Раго С. UNIX. Профессиональное программирование. СПб.: Символ-Плюс, 2007.
14. Таненбаум Э. Современные операционные системы. 4-е изд. СПб.:Питер, 2021.
15. Таненбаум Э., Ван Стен М. Распределенные системы. Принципы и парадигмы. СПб.:Питер, 2003.
16. Танненбаум Э., Узеролл Д. Компьютерные сети. СПб.: Питер, 2003.
17. Танненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2021.
18. Терехов А.Н. Технология программирования. М.: ИНТУИТ.РУ, 2007.
19. С.М. Львовский. Набор и верстка в системе LATEX. ЛитРес, 2017.
20. Фредерик Брукс. Мифический человеко-месяц или как создаются программные системы. СПб.: «Символ-Плюс», 2000.

Список дополнительной литературы

1. Гласс Г., Эйбле К. Unix для программистов и пользователей. СПб.: БХВ-Петербург, 2004.
2. Дейтел Г. Введение в операционные системы. М.: Мир, 1987.
3. Когаловский М.Р. Энциклопедия технологий баз данных. М.: Финансы и статистика, 2002.
4. Королев Л.Н. Архитектура процессоров электронных вычислительных машин, М.:

Издательский отдел ВМиК МГУ, 2003.

5. Соломон Д., Руссинович М. Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP, Windows 2000. СПб.: Питер, 2005.
6. Гергель В. П. Современные языки и технологии параллельного программирования. Издательство МГУ, 2012.
7. Корнеев В.В. Параллельные вычислительные системы. М.: Нолидж, 1999.
8. Дэвид М. Харрис, Сара Л. Харрис Цифровая схемотехника и архитектура компьютера : RISC-V [пер. с англ.] – М.: ДМК Пресс, 2022. – 810 с.

Перечень иных информационных источников

1. Pro Git book, интернет-ресурс. <https://git-scm.com/book/ru/v2>
2. The Scrum Guide, интернет-ресурс. <https://www.scrum.org/resources/scrum-guide>