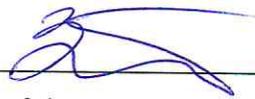


Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и кибербезопасности

УТВЕРЖДАЮ

Директор ИКНК

 Д.П. Зегжда
«04» октября 2023 г.

ПРОГРАММА

вступительного испытания для поступающих в магистратуру

по направлению подготовки / образовательной программе

**02.04.03 Математическое обеспечение и администрирование
информационных систем**

**02.04.03_01 Разработка и математическое обеспечение
интеллектуальных информационных систем**

Код и наименование направления подготовки / образовательной программы

Санкт-Петербург

2023

АННОТАЦИЯ

Программа междисциплинарного вступительного экзамена в магистратуру содержит перечень дисциплин направления **02.03.03 Математическое обеспечение и администрирование информационных систем**, перечень тем (вопросов) по дисциплинам, рекомендуемую литературу для подготовки к экзамену и примеры тестовых заданий.

Вступительное испытание оценивается по стобалльной шкале и состоит из междисциплинарного экзамена в объеме требований, предъявляемых государственными образовательными стандартами высшего образования к уровню подготовки бакалавра по направлению, соответствующему направлению магистратуры, проводимого очно в письменной форме или дистанционно (**максимальный балл – 100**);

Минимальное количество баллов, подтверждающее успешное прохождение междисциплинарного экзамена – **50 баллов (50%)**.

Руководитель ОП

Доцент ВШТИИ ИКНК, к.ф.-м.н.

В.Г. Пак

Составители:

Директор ВШТИИ ИКНК, к.т.н.

Доцент ВШТИИ ИКНК, к.ф.-м.н.

Доцент ВШТИИ ИКНК, к.т.н.

В.А. Мулюха

И.Н. Белых

О.Ю. Сабинин

Программа рассмотрена и рекомендована к изданию ученым советом ИКНК (протокол № 1/23 от «04» октября 2023 г.).

1. ДИСЦИПЛИНЫ, ВКЛЮЧЁННЫЕ В ПРОГРАММУ МЕЖДИСЦИПЛИНАРНОГО ЭКЗАМЕНА

- 1.1. Объектно-ориентированное программирование;
- 1.2. Базы данных;
- 1.3. Структуры и алгоритмы обработки данных.

2. СОДЕРЖАНИЕ УЧЕБНЫХ ДИСЦИПЛИН

2.1. Объектно-ориентированное программирование

Темы (вопросы):

1. Основные положения разработки событийно-управляемых приложений.

Взаимодействие: приложение – ресурсы вычислительной системы – операционная система – пользователь. Организация традиционной программы с заданным порядком выполнения (характерной для однозадачных ОС). Приложение, управляемое событиями. Основные понятия: событие, выявление (мониторинг) внешних событий, сообщение, очереди сообщений, обработчик сообщения.

2. Системные функции и управление памятью.

Интерфейс прикладного программирования (Win 32 API) – связь приложения с “внешним миром”. Механизм RPC (LPC). Защищенное адресное пространство приложения. Средства манипулирования памятью.

3. Унифицированный интерфейс пользователя.

Элементы управления. Элементы пользовательского интерфейса. Аппаратно-независимый ввод/вывод. Графический интерфейс Windows (GDI). Аппаратная независимость графического вывода. Графические концепции и графические примитивы. Понятие контекста устройства. Режимы отображения. Система координат. Инструменты для «рисования».

4. Объектная модель Windows.

Типы объектов. Понятие дескриптора. Окно – объект Windows. Окно, структура данных для описания окна, дескриптор (handle) окна, класс окна, создание экземпляра окна, активное окно, понятие фокуса. Иерархия и стили окон. Диспетчер окон (Window Manager). Визуальные компоненты интерфейса: рамка, заголовок окна, системное меню, клиентская область. Составные части Windows-приложения.

5. Концепция объектно-ориентированного программирования. Отличия процедурного программирования от ООП.

Понятие класса. Объявление класса. Оператор sizeof и размер класса. Спецификаторы доступа. Создание экземпляра класса. Реализация методов класса. Доступ к public членам класса посредством объекта, селектор «.»». Указатель this. Анатомия вызова нестатического метода класса. Конструкторы. Понятие конструктора. Конструктор по умолчанию. Конструктор с параметрами. Перегрузка конструкторов. Конструктор с параметрами по умолчанию. Конструкторы и модификатор explicit. Конструкторы базовых типов. Динамическое создание объектов и вызов конструктора. Деструктор. Специфика объявления деструктора как метода класса. Создание и уничтожение объектов с разным способом хранения.

6. Конструктор копирования.

Создание копий. Передача объектов в качестве параметров функции. Возвращение объекта по значению. Проблемы, которые могут возникнуть при использовании конструктора копирования по умолчанию и способы их решения. Указатель на класс. Указатель на объект класса. Доступ к членам класса посредством указателя. Селектор «->». Указатель this.

7. Массивы и классы.

Массивы объектов класса. Массивы указателей на объекты класса. Поля битов. Специфика использования.

8. Наследование.

Виды наследования. Простое (single) наследование. Объявление производного класса. Спецификатор `protected` в базовом классе. Спецификаторы наследования. Порядок вызова конструкторов и деструкторов. Передача параметров базовому классу при конструировании. Специфика передачи параметров конструктору копирования базового класса. Открытое наследование. Правила объектно-ориентированного проектирования. Пример ошибочного построения иерархии классов.

9. Полиморфизм.

Раннее и позднее связывание. Виртуальные функции. Механизм вызова виртуальной функции. Виртуальные деструкторы. Чисто виртуальные функции и абстрактные классы. Наследование интерфейса и наследование реализации при открытом наследовании.

10. FRIEND (привилегированные) функции и классы.

Внешняя friend-функция. Friend-функция, являющаяся методом другого класса. Friend-класс.

11. Перегрузка операторов.

Перегрузка операторов для базовых и пользовательских типов. Правила перегрузки унарных и бинарных операторов. Порядок поиска компилятором функции. Формы перегрузки операторов. Перегрузка оператора с помощью метода класса. Перегрузка с помощью метода класса. Специфика перегрузки оператора присваивания. Оператор присваивания и нетривиальные классы. Оператор присваивания и перегрузка. Оператор `[]` с проверкой выхода за границы массива. Оператор `++` (`--`). Оператор `()` и функциональные объекты.

12. Внедряемые объекты.

Конструирование и уничтожение встроенных объектов. Передача параметров конструктора встроенным объектам. Список инициализации. Специфика инициализации константных объектов и ссылок. Порядок инициализации членов класса. Пример «ассоциативного массива». Указатели на объекты в

качестве членов данных класса. Предварительное неполное объявление класса (forward reference). Ссылки на указатели.

13. Статические члены класса.

Ключевое слово `static`. Статические данные. Статические функции: для доступа извне к `private` или `protected static` -данным класса, для «косвенного» создания объекта.

14. Множественное наследование.

Создание объектов производных классов при множественном наследовании. Полиморфизм и множественное наследование. Проблемы, возникающие при множественном наследовании. Множественное наследование – основа COM.

15. Обобщенное программирование.

Шаблоны. Объявление шаблона. Инстанцирование шаблона. Обобщенное программирование. Обобщенные алгоритмы. Шаблоны функций. Способы обобщения функций, выполняющих одинаковые действия, но оперирующих данными разных типов. Создание функции по заданному шаблону и ее вызов. Шаблоны классов. Ключевое слово `typename`. Эмуляция шаблона одномерного защищенного массива (`vector`). Введение понятия итератора. Эмуляция шаблона двухсвязного списка (`list`). Реализация итератора для двухсвязного списка.

16. Потоки ввода-вывода.

Система ввода-вывода C++. Поток C++ с точки зрения системы. Поток C++ с точки зрения прикладной программы. Классы ввода/вывода. Стандартные объекты `cin` и `cout`. Буферизованный ввод-вывод. Состояние потока. Форматированный ввод-вывод. Флаги ввода-вывода и методы для работы с ними. Манипуляторы ввода-вывода. Файловый ввод-вывод. Неформатированный ввод-вывод.

Литература для подготовки:

1. Лафоре Р. Объектно-ориентированное программирование в C++. Питер, 2018. 928 с.

2. Прата С. Язык программирования C++. Лекции и упражнения. Диалектика-Вильямс, 2018. 1248 с.

2.2. Базы данных

Темы (вопросы):

1. Основы построения баз данных.

Базы данных (БД) и системы управления базой данных (СУБД). Выбор системы управления базами данных. Жизненный цикл базы данных. Реляционные СУБД. Распределенные СУБД на инвертированных файлах. Гипертекстовые и мультимедийные БД. XML-серверы. Объектно-ориентированные БД. Коммерческие БД. Языковые средства современных СУБД. Язык SQL: общая характеристика, основные разделы и команды.

2. Базовые понятия реляционных баз данных.

Понятие отношения, домена, атрибута, кортежа, ранга (степени) отношения, схемы отношения. Графическая интерпретация отношений. Особенности двумерных таблиц, моделирующих отношения. Понятие возможного ключа, первичного ключа. Эквивалентные схемы отношений. Взаимосвязь между отношениями: основное отношение, подчиненное отношение, понятие внешнего ключа.

3. Основы реляционной алгебры.

Общее понятие реляционной алгебры как абстрактной алгебры, объекты операции, замкнутость операций относительно объектов. Классификация операций реляционной алгебры. Теоретико-множественные операции – объединение, пересечение, разность отношений, расширенное декартово произведение. Специальные операции – горизонтальная выборка, проекция, условное соединение, деление. Примеры выполнения запросов.

4. Проектирование баз данных на основе теории нормализации.

Проектирование реляционных баз данных. Теория функциональных зависимостей и нормальных форм баз данных. Алгоритмы декомпозиции и синтеза при проектировании нормальных форм для реляционных баз данных.

5. Проектирование баз данных на основе модели Сущность-Связь.
6. Базовые возможности языка SQL.

Литература для подготовки:

1. Дейт К. Введение в системы баз данных. Диалектика, 2019. 1328 с.
2. Кириллов В.В., Громов Г.Ю. Введение в реляционные базы данных. ВНУ, 2017. 464 с.
3. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика". Вильямс, 2017. 1440 с.
4. Кузнецов С.Д. Базы данных: языки и модели. Корона-Принт, 2008. 720 с.

2.3. Структуры и алгоритмы обработки данных

Темы (вопросы):

1. Понятия алгоритма и абстрактного типа данных:
 1. Понятие алгоритма. Требования к алгоритмам;
 2. Способы записи алгоритмов;
 3. Абстрактные типы данных (АТД). Типы и структуры данных. Виды структур данных.
2. Анализ вычислительной сложности алгоритмов:
 1. Время работы алгоритма: основные определения. Правила вычисления времени работы алгоритма. Анализ времени работы алгоритмов с вызовами процедур;
 2. Анализ рекурсивных алгоритмов. Оценка решений рекуррентных соотношений.

3. Нелинейные структуры данных:
 1. Графы (орграфы), способы их задания. Представление графов структурами данных;
 2. Структуры Вирта;
 3. АД графов, его основные операторы;
 4. Задача обхода графа (орграфа). Алгоритм обхода в глубину. Глубинный остовный лес;
 5. Алгоритм обхода в ширину. Остовный лес;
 6. Деревья. АД деревьев, его основные операторы. Реализации деревьев структурами данных;
 7. Задача обхода дерева. Алгоритмы поиска в глубину;
 8. Деревья выражений. Обход деревьев выражений в глубину. Способы записи выражений;
 9. Алгоритм обхода дерева в ширину.
4. Внутренняя сортировка:
 1. Задача сортировки элементов (записей). Модель внутренней сортировки. Простые алгоритмы сортировки, их вычислительная сложность;
 2. Алгоритм быстрой сортировки Хоара, его вычислительная сложность в худшем и в среднем;
 3. АД множеств, его основные операторы. Сбалансированные деревья. Реализация АД множеств с помощью сбалансированных деревьев, 2-3-деревья. Поиск элемента в 2-3-дереве;
 4. Процедуры вставки элемента в 2-3-дерево и удаления элемента из 2-3-дерева. Пирамидальная сортировка на 2-3-деревьях;
 5. АД очередей с приоритетами. Реализация очередей с приоритетами с помощью частично упорядоченных деревьев. Реализация частично упорядоченных деревьев массивом «куча»;

6. Пирамидальная сортировка на частично упорядоченных деревьях.
5. Внешняя сортировка:
 1. Модель внешней обработки данных. Файловая структура данных. Внешняя сортировка слиянием;
 2. Внешняя сортировка многоканальным слиянием;
 3. Внешняя многофазная сортировка слиянием. Оптимальное начальное распределение записей исходного файла.
6. Алгоритмы поиска:
 1. Двоичные деревья и их представления структурами данных. Код Хаффмана;
 2. Деревья двоичного поиска. Реализация АД множества деревьями двоичного поиска. Время выполнения операторов АД множества на деревьях двоичного поиска;
 3. Красно-чёрные деревья, операции над ними. Время выполнения операций;
 4. Метод ветвей и границ. Решение задачи коммивояжёра и задачи о ранце методом ветвей и границ;
 5. Поиск с возвратом: общее описание алгоритма, применение.
7. Файлы
 1. Хранение данных в файлах: простая организация данных. Ускорение работы с данными в файлах;
 2. Хешированные файлы;
 3. Индексированные файлы;
 4. Внешние деревья поиска: m -арные и B -деревья. Операции с B -деревьями. Время выполнения операций.

8. Основы теории сложности алгоритмов

1. Понятие сложности алгоритма. Сложность в худшем. Временная и пространственная сложности. Алгебраическая сложность;
2. Асимптотические оценки сложности алгоритмов;
3. Понятие сложности в среднем.

Литература для подготовки:

1. Абрамов С.А. Лекции о сложности алгоритмов. МЦНМО, 2012. 248 с.
2. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. Вильямс, 2018. 400 с.
3. Вирт Н. Алгоритмы и структуры данных. ДМК Пресс, 2016. 274 с.
4. Вирт Н. Алгоритмы + структуры данных = программы. Мир, 1985. 406 с.
5. Кормен Т., Штайн К., Ривест Р., Лейзерсон Ч. Алгоритмы. Построение и анализ. Диалектика, 2019. 1328 с.

3. ПРИМЕР ЭКЗАМЕНАЦИОННОГО БИЛЕТА (ТЕСТОВОГО ЗАДАНИЯ)

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности

ВСТУПИТЕЛЬНОЕ ИСПЫТАНИЕ

по направлению подготовки/образовательной программе

02.04.03 Математическое обеспечение и администрирование информационных
систем

02.04.03_01 Разработка и математическое обеспечение интеллектуальных
информационных систем

Код и наименование направления подготовки / образовательной программы

УТВЕРЖДАЮ

Руководитель ОП

 В.Г. Пак

«04» октября 2023 г.

ПРИМЕР ТЕСТОВЫХ ЗАДАНИЙ

Междисциплинарный экзамен состоит из 20 тестовых заданий,
максимальный балл за задание - 5.

1. Какова функция роста по n времени работы фрагмента алгоритма,
представленного псевдокодом?

```
l:=0
for i:=1 to  $\lceil \sqrt{n^2 + 1} \rceil$ 
do
begin
z:=i
repeat
begin
l:=l+1
z:=[z/5]
end
until z<=0
end
```

- A. $O(n \log n)$
- B. $O(n^2 \log n)$**
- C. $O(n^4 \frac{1}{2} \log n)$
- D. $O(n^{1/2} \log n)$

2. Что делает приведённый ниже псевдокод алгоритма?

```

Function FF(n, T)
  Begin
    For p:=1 to maxnodes do
      begin
        L:=header[p]
        if not Empty(L) then
          begin
            node:=First(L)
            if node.Label=n then return(p)
            while node.Next<>0 do
              begin
                node:=Next(L, node)
                if node.Label=n then return(p)
              end
            end
          end
        end
      end
    return(0)
  End FF

```

Выберите один ответ:

- A. находит число сыновей узла n в дереве T, реализованным массивом указателей на родителей
- B. находит самого левого сына узла n в дереве T, реализованном массивом указателей на родителей
- C. находит родителя узла n в дереве T, реализованным массивом указателей на родителей
- D. находит самого левого сына узла n в дереве T, реализованным списками сыновей
- E. находит число сыновей узла n в дереве T, реализованном списком сыновей
- F. находит родителя узла n в дереве T, реализованным списками сыновей**

3. Укажите правильный порядок обработки логических операторов (при условии, что скобки не используются).

Выберите один ответ:

- A. AND, OR, NOT
- B. OR, NOT, AND
- C. NOT, OR, AND
- D. NOT, AND, OR**

4. Сколько вариантов группировок будет обработано в результате применения оператора ROLLUP(a,b,c). Введите число.

Правильный ответ: 4

5. Способность объекта скрывать детали своей реализации называется:

- A. абстракция
- B. полиморфизм
- C. наследование
- D. инкапсуляция**

6. В каком случае произойдет ошибка при инициализации переменной:

```
class A {
private:
    int a;
protected:
    int b;
public:
    int c;
};

class B : private A {
    B()
    {
        a = 0;
        b = 0;
        c = 0;
    }
};
```

- A. ни в каком
- B. a=0;**
- C. b=0;
- D. c=0;

7. Имеется следующее начальное распределение записей исходного файла по 4 вспомогательным файлам: 108, 100, 85, 50; 5-й вспомогательный файл - пустой. Сойдётся ли при таком начальном распределении алгоритм внешней многофазной сортировки слиянием?

A. сойдётся не оптимально

B. сойдётся оптимально

C. невозможно определить

D. не сойдётся