

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
профессионального образования
«Санкт-Петербургский государственный университет аэрокосмического
приборостроения»

Центр организационно-методического обеспечения магистерской подготовки



«УТВЕРЖДАЮ»

Ректор ГУАП

Ю.А. Антохина

«13» 04 2015

ПРОГРАММА ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ ПО ПРИЕМУ В
МАГИСТРАТУРУ НА НАПРАВЛЕНИЕ ПОДГОТОВКИ

02.04.03 «Математическое обеспечение и администрирование
информационных систем»

Санкт-Петербург 2015

Рассмотрено и рекомендовано к использованию на заседании
Координационного совета по магистерской подготовке в ГУАП

09.04.2015 протокол № 04/КС

Программа согласована с выпускающей кафедрой;

Ответственный за ОП 02.04.03 кафедры № 43

доцент, к.т.н., доцент каф. № 43  А.А. Ключарев

Программа соответствует федеральному государственному образовательному
стандарту высшего образования по направлению 02.04.03

Директор ЦОМОМП

 Е.Г.Семенова

1 ОБЩИЕ ПОЛОЖЕНИЯ ПО ПРОВЕДЕНИЮ ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ

ПО ПРИЕМУ В МАГИСТРАТУРУ НА НАПРАВЛЕНИЕ

02.04.03 «Математическое обеспечение и администрирование информационных систем»

1.1 Настоящая Программа, составленная в соответствии с федеральным государственным образовательным стандартом ВО по направлению 02.04.03 «Математическое обеспечение и администрирование информационных систем», устанавливает содержание вступительных испытаний с целью определения подготовленности претендентов и наличия способностей для продолжения образования в магистратуре по направлению 02.04.03.

1.2 В качестве вступительного испытания для претендентов на обучение в магистратуре ГУАП в соответствии с СТО ГУАП. СМКО 2.72 - «Магистерская подготовка в ГУАП», установлен междисциплинарный экзамен, проводимый в письменной или устной форме.

1.3 Решение экзаменационной комиссии заносится в протокол.

2 ПРОГРАММА ДЛЯ ПРОВЕДЕНИЯ ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ

2.1 Программа вступительного испытания, содержит вопросы в объеме требований, предъявляемых образовательным стандартом высшего профессионального образования уровня подготовки бакалавра по направлению, соответствующему направлению магистратуры.

2.2 Конечной целью вступительного испытания является определение уровня знаний и компетенций абитуриента по 100-балльной шкале.

Основное вступительное испытание предназначено для определения степени подготовленности абитуриента к обучению по выбранному направлению магистерской подготовки, для определения уровня его знаний и компетенций. По результатам основного вступительного испытания приемная комиссия определяет проходной балл для зачисления абитуриентов на данное направление магистерской подготовки.

Целью предварительного вступительного испытания является определение степени подготовленности абитуриента к обучению по выбранному направлению магистерской подготовки, владение им основными понятиями и терминологией в данной области. Экзаменационная комиссия выставляет претенденту оценку по 100-балльной шкале. Успешно прошедшими предварительное вступительное испытание считаются лица, набравшие не менее 60 баллов. При наборе меньшего числа баллов абитуриент не допускается к прохождению основного вступительного испытания.

Рекомендуется следующая система оценивания результатов предварительного вступительного испытания по следующей 100-балльной квантованной шкале:

– 100 баллов - в ответе отражены основные концепции и теории по данному вопросу, проведен их критический анализ и сопоставление, описанные теоретические положения иллюстрируются практическими примерами и экспериментальными данными. Абитуриентом формулируется и обосновывается собственная точка зрения на заявленные проблемы, материал излагается профессиональным языком с использованием соответствующей системы понятий и терминов.

– 80 баллов - в ответе описываются и сравниваются основные современные концепции и теории по данному вопросу, описанные теоретические положения иллюстрируются практическими примерами, абитуриентом формулируется собственная точка зрения на заявленные проблемы, однако он испытывает затруднения в ее аргументации. Материал излагается профессиональным языком с использованием соответствующей системы понятий и терминов.

– 60 баллов - в ответе отражены лишь некоторые современные концепции и теории по данному вопросу, анализ и сопоставление этих теорий не проводится. Абитуриент испытывает значительные затруднения при иллюстрации теоретических

положений практическими примерами. У абитуриента отсутствует собственная точка зрения на заявленные проблемы.

Материал излагается профессиональным языком с использованием соответствующей системы понятий и терминов.

– 40 баллов - ответ не отражает современные концепции и теории по данному вопросу. Абитуриент не может привести практических примеров. При изложении материала не используются понятия и термины соответствующей научной области.

3 ПЕРЕЧЕНЬ ВОПРОСОВ ДЛЯ ПРОВЕДЕНИЯ ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ

3.1 Перечень вопросов для проведения предварительного вступительного испытания

1. Концепции структурного программирования.
2. Основные конструкции процедурных языков программирования.
3. Основные типы данных, используемые в процедурных языках программирования.
4. Классификация структур данных и их особенности.
5. Принципы построения и реализации структур данных: различных линейных структур данных, деревьев, графов.
6. Принципы основных алгоритмов обработки данных: поиск, сортировка, хеширование, сжатие данных.
7. Принципы оценки структур данных и алгоритмов.
8. Базовые положения объектно-ориентированного программирования.
9. Основные понятия и модели объектно-ориентированного программирования.
10. Методы и алгоритмы проектирования объектно-ориентированных программ.
11. Классификация языков объектно-ориентированного программирования, их архитектура, выразительные средства и технология применения.
12. Основные синтаксические конструкции языков LISP и PROLOG.
13. Механизм выполнения функциональных и логических программ.
14. Принципы создания функциональных и логических программ.
15. Методы управления эффективностью функциональных и логических программ.
16. Модели жизненного цикла ПО.
17. Критерии качества программного обеспечения.
18. Модели и методы проектирования, используемые в современных системах автоматизации разработки программного обеспечения.
19. Методы тестирования и отладки программного обеспечения.
20. Основные понятия в области компьютерной графики.
21. Принципы построения современных графических систем.
22. Основные способы и форматы представления изображений.
23. Основы геометрического моделирования графических объектов.
24. Основные модели цвета в компьютерной графике.
25. Принципы реализации эффектов фотореалистичной графики.
26. Способы развертки и закраски поверхностей.
27. Способы удаления невидимых линий и поверхностей.
28. Основные принципы фрактальной графики.
29. Основные понятия в области человеко-машинного взаимодействия.
30. Психологические аспекты человеко-машинного взаимодействия
31. Аппаратные средства человеко-машинного взаимодействия.
32. Модели и прикладные аспекты человеко-машинного взаимодействия.
33. Формальные методы описания диалоговых систем.
34. Основные форматы представления аудиовизуальных данных.
35. Примеры языков описания диалога и виртуальных моделей.
36. Принципы создания интерактивных аудиовизуальных презентаций.
37. Основные функции ОС.
38. Принципы построения ОС.

39. Основные разновидности ОС.
40. Способы организации управления памятью в ОС.
41. Способы планирования процессов в многозадачных ОС с однопроцессорной и мультипроцессорной архитектурами.
42. Основные механизмы коммуникации и синхронизации процессов.
43. Основные функции подсистем ввода-вывода в ОС.
44. Современные стандарты в области ОС.
45. Направления развития ЭВМ с нетрадиционной архитектурой.
46. Принципы выбора архитектуры и комплексирования современных ЭВМ, комплексов и систем.
47. Основные понятия реляционной алгебры и реляционного исчисления.
48. Основные синтаксические конструкции языка SQL.
49. Методы формирования запросов средствами операторов реляционной алгебры и с помощью выражений исчисления кортежей.
50. Теория зависимостей и методы нормализации баз данных.

3.2 Перечень вопросов для проведения основного вступительного испытания

1. Тенденции развития архитектур вычислительных систем и комплексов
2. Стандартные типы данных. Принципы типизации данных; стандартные типы данных.
3. Номенклатура операций. Понятие выражения; номенклатура Си – операций; понятие приоритета операции.
4. Управляющие конструкции. Представление основных управляющих структур программирования; способы реализации ветвлений и циклов; вложение управляющих конструкций; ограничения, накладываемые на структуру программы концепциями структурного программирования.
5. Процедуры и функции. Концепции процедур и функций; механизмы передачи параметров; процедурный тип данных; область видимости; директивы описаний процедур и функций; рекурсивные определения и алгоритмы; программирование рекурсивных алгоритмов.
6. Структурные типы данных. Массивы; утверждения о массивах; данные типа «структура»; файлы; индуктивные функции на последовательностях (файлах, массивах).
7. Ссылочный тип данных. Использование динамической памяти; типизированные и нетипизированные указатели; операции над указателями; процедуры и функции распределения динамической памяти; динамические структуры данных; линейные списки: основные виды и способы реализации; линейный список как абстрактный тип данных; очереди и стеки; двоичные деревья.
8. Общие сведения, сложность алгоритмов. Понятие алгоритма. Основные принципы анализа сложности и эффективности алгоритмов. Порядковые статистики. NP-сложные задачи. Труднорешаемые задачи.
9. Спецификация, представление и реализация абстрактных типов данных. Скалярные данные различного типа. Составные данные: массив, запись (структура), множество. Динамические структуры данных: однонаправленный линейный список, двунаправленный линейный список, однонаправленный циклический список, двунаправленный циклический список, мультисписки (иерархические списки). Алгоритмы вставки, удаления, поиска в динамических структурах данных. Линейные структуры данных: стек, очередь, дек и операции, осуществляемые с ними.
10. Деревья, их представление и основные операции над ними. Понятие дерева. Понятие леса. Представления бинарных деревьев: списки, массив. Методы обхода деревьев. Представление выражений с помощью деревьев: префиксная, инфиксная и постфиксная запись выражений. Сжатие данных с помощью деревьев, кодовые деревья, оптимальные префиксные коды. Представление сильноветвящихся деревьев.

11. Графы, их представление и основные алгоритмы. Способы представления графа: матрица смежности, матрица инцидентности, списки смежных вершин, список ребер. Алгоритмы поиска на графах: поиск в глубину, поиск в ширину. Минимальное остовное дерево. Кратчайший путь.
12. Файлы, их организация, представление и обработка. Понятие файла. Организация файлов. Представление файлов с использованием В-деревьев. Основные операции обработки файлов.
13. Алгоритмы поиска. Исчерпывающий поиск: перебор с возвратом, метод ветвей и границ, динамическое программирование. Быстрый поиск: бинарный и последовательный поиски в массивах, хеширование и разрешение коллизий. Использование деревьев в задачах поиска: бинарные, случайные бинарные, оптимальные и сбалансированные по высоте (АВЛ) и рандомизированные деревья поиска. Алгоритмы поиска на графах: поиск в глубину, поиск в ширину.
14. Алгоритмы сортировки, внутренняя и внешняя сортировка. Алгоритмы внутренней сортировки: подсчетом, простым включением, методом Шелла, простым извлечением, распределением, слиянием, пузырьковая, сортировка Хоара. Алгоритмы внешней сортировки. Оптимальная сортировка.
15. Основные концепции ООП. Децентрализация, классификация, инкапсуляция, полиморфизм, контракты.
16. Объявление и определение класса. Данные-члены и функции-члены класса. Создание экземпляров классов, ссылки и указатели на объекты. Статические и нестатические члены класса.
17. Инкапсуляция. Определение прав доступа. Друзья класса.
18. Конструкторы. По умолчанию, инициализирующий конструктор, список инициализации и конструктор копирования. Деструкторы – назначение, определение и использование.
19. Наследование. Одиночное, множественное и виртуальное. Область видимости членов класса.
20. Шаблоны классов. Параметризация и специализация шаблонов классов.
21. Операторные функции. Функции в пространстве имен и как члены класса.
22. Полиморфизм. Таблица виртуальных функций, абстрактные классы.
23. Исключения. Реализация обработки исключительных ситуаций, спецификация исключений.
24. История развития и причины появления СУБД. Основные термины и понятия в области баз данных и СУБД. История развития СУБД: System R, СУБД класса Desktop, файл-серверная и клиент-серверная архитектуры, основные производители современных СУБД и их продукты. Основные недостатки файловых систем, причины появления СУБД. Архитектура и функции СУБД.
25. Модели данных. Основные термины и понятия реляционных баз данных. Отношения и их свойства, ключи отношений. Реализация отношений в базах данных, типы данных в языке SQL, операторы языка SQL для создания, удаления, модификации таблиц базы данных.
26. Реляционная алгебра. Булевы операции над отношениями. Операция выборки, ее свойства. Операция проекции, ее свойства. Операция соединения, ее свойства. Операция деления. Операция переименования атрибутов. Операции эквисоединения, θ -выбоки и θ -соединения.
27. Оператор выборки в языке SQL. Основные разделы оператора SELECT. Агрегатные функции в операторе выборки языка SQL. Объединение, пересечение, разность запросов в языке SQL. Запросы с подзапросами в языке SQL. Экзистенциальные запросы в языке SQL. Виды соединений в языке SQL. Трехзначная логика и обработка NULL-значений в языке SQL.
28. Средства манипулирования данными в языке SQL. Представления: создание, фрагментация данных, ограничения по использованию. Управляющие конструкции, процедурное программирование в языке SQL. Хранимые процедуры: создание, вызов,

рекурсия, ограничения по использованию. Триггеры: виды, создание, ограничения по использованию.

29. Базовые понятия и определения. История развития Internet. Обзор технологий, используемых в Internet. Элементы SGML.

30. Язык HTML и CSS. Базовая структура документа в формате HTML. Каскадные таблицы стилей.

31. Динамический HTML. Объектная модель документа (DOM). DOM 2 Core. Объектная модель документа. DOM 2 HTML. Модель событий динамического HTML. Клиентские языки сценариев. JavaScript – основные положения. Встроенные возможности языка JavaScript.

32. Серверные языки сценариев. Web сервер. Клиент-серверное взаимодействие. ASP.

33. Средства Java для разработки Web приложений. Платформа Java. Основные понятия. Синтаксис и особенности языка программирования Java. Возможности Java для мультимедийных web приложений – Апплеты.

34. XML и XSL. Обзор XML. Язык XSL. XSL Transformation.

35. Понятие вычислительного процесса. История развития теории вычислительных процессов. Вклад отечественных ученых в развитие теории вычислительных процессов. Место теории вычислительных процессов в ряду других фундаментальных и прикладных дисциплин. Мировоззренческие экономические и правовые аспекты.

36. Алгоритмические основы теории вычислительных процессов и структур. Определение и основные свойства алгоритмов. Уточнения понятия алгоритма. Рекурсивные функции. Машины Тьюринга. Формальные системы.

37. Элементы теории абстрактных автоматов как формальных моделей организации вычислительных процессов на ЭВМ. Определение и классификация конечно-автоматных моделей вычислительных процессов. Операции над конечно-автоматными моделями вычислительных процессов, синтез и анализ моделей. Линейно-ограниченные модели вычислительных процессов. Модели вычислительных процессов в виде МП-автоматов.

38. Теория формальных языков и их применимость к процессам синтеза и анализа вычислимых процедур. Определение и классификация моделей вычислительных процессов на основе формальных языков. Модели вычислительных процессов на основе регулярных языков. Модели вычислительных процессов на основе контекстно-свободных и контекстно-связанных языков.

39. Основные понятия теории трансляции программ и инструментальных средств организации вычислений в различных операционных средах. Основы построения моделей трансляторов и моделей программ. Инструментальные средства организации вычислений при трансляции программ.

40. Элементы теории схем программ. Основные определения и свойства моделей схем программ. Классификация моделей схем программ.

41. Сколько раз будет выведено на экран значение переменной i в соответствии с приведенным фрагментом программы?

```
for (int i=0; i<=10; i++)
    cout << ++i << endl;
```

1. 9

2. 10

3. 5

4. 6

42. Функция f определена следующим образом:

```
void f(int &a, int b, const int * c) {a++; b++; c--;}
```

Необходимо определить значения переменных X , Y и Z после выполнения операторов:

```
int X=0, Y=1, Z=2;
```

```
f(X, Y, &Z);
```

1. $X=1, Y=2, Z=1$

2. $X=0, Y=1, Z=2$

3. $X=1, Y=1, Z=2$

4. X=1, Y=1, Z=1

43. Укажите, что выполняется в приведенном фрагменте кода:

```
int *p1 = new int(5); //1
int *p2 = new int[5]; //2
int *p3 = new int;    //3
```

1. строка 1 и 2 - выделение памяти под 5 величин типа `int`, `p1`, `p2` - имя массива; строка 3 - объявление указателя на `int`

2. строка 1 и 2 - объявление указателя на `int` и инициализация выделенной динамической памяти значением 5; 3 - объявление указателя на `int`

3. строка 1 - объявление указателя на `int` и инициализация выделенной динамической памяти значением 5; строка 2 - выделение памяти под 5 величин типа `int`, `p2` - имя массива; строка 3 - объявление указателя на `int`

4. разницы при объявлении переменных нет

44. Дан пример функции, возвращающей сумму двух целых величин

```
#include <iostream>
using namespace std;
int sum(int a=4, int b=5);
```

```
int main()
{
    int a = 2, b = 3, c, d;
    c = sum(a, b);
    cin >> b;    // ввод b=10
    d = sum(b);
    b = sum();
    return 0;}

```

45. Чему будет равно значение переменных `b`, `c`, `d` после выполнения программы?

1. `b=5`, `c=5`, `d=14`

2. `b=9`, `c=5`, `d=15`

3. `b=9`, `c=5`, `d=14`

4. ошибка - при вызове функции `sum` обязательно должно указываться 2 параметра

46. Классический пример рекурсивного вызова функции - вычисление факториала

```
char factorial(int n)
{
    cout<<i++;
```

```
    if (n==0)
        return 1;
    return n*factorial(n-1);
}
```

.....

```
int k=factorial(6);
```

Какое значение будет возвращено при вызове функции `factorial(6)` и сколько раз будет выполнена функция `factorial` в этом случае?

1. 720; 6

2. 720; 7

3. -48; 6

4 -48; 7

5 208; 7

7. программа работать не будет

47. Чему будет равняться значение переменной `k` после выполнения операторов

```
int a=4;
int b=5;
if (a=b)
    k=1;
```


else

 k=0;

1. 0

2. 1

3. ошибка в программе

48. Необходимо найти ошибки в следующем фрагменте программы (синтаксические, логические, ...). :

```
int A[10], B[10], *C;
```

```
for(i=1;i<=10;i++)
```

```
{
```

```
    cin>>A[i];
```

```
    cin>>B[i];
```

```
}
```

```
for (i=1;i<=10;i++)
```

```
    C[i]=A[i]+B[i];
```

Каждое возможное исправление/добавление в строчке кода считается за ошибку.

Определить количество ошибок, которое будет выдано на этапе компиляции.

1. 1

2. 2

3. 3

4. 7

5. 11

6. 10

49. В теле функции может быть указан оператор {... return 1;}, если она возвращает значение типа:

1. void

2. void*

3. char

4. любой тип

5. int

6. double

50. Перегрузка функций - использование одного и того же имени для разных функций.

При этом действуют следующие ограничения:

1. перегруженные функции должны находиться в одной области видимости;

2. перегруженные функции должны находиться в разной области видимости;

3. не могут перегружаться функции, имеющие совпадающие тип и число аргументов, но разные типы возвращаемых значений;

4. не могут перегружаться функции, имеющие разное число аргументов и одинаковые типы возвращаемых значений;

5. не могут перегружаться функции, если их списки формальных параметров различаются только применением модификаторов const и volatile или использованием ссылки &, а типы возвращаемых значений одинаковые;

6. не могут перегружаться функции, если их списки формальных параметров различаются только применением модификаторов const и volatile или использованием ссылки &.

51. В C/C++ к целочисленным типам данных относят:

1. int

2. char

3. float

4. double

5. bool

6. void

52. Сколько указателей объявляется в следующей строчке:

```
int * x1, x2, x3;
```

1. 1
 2. 2
 3. 3
53. Что означает наличие NOT NULL при описании атрибута в таблице?
1. Атрибут является обязательным.
 2. СУБД автоматически присваивает значения этому атрибуту.
 3. Атрибут может принимать только числовые значения целого типа.
 4. Атрибут не может иметь неопределенных значений.
 5. Атрибуту всегда по умолчанию присваивается значение NOT NULL.
54. Какие из утверждений справедливы к понятию «внешний ключ»?
1. Значения внешнего ключа для каждой строки таблицы должны отличаться?
 2. В таблице может быть несколько строк, имеющих одинаковое значение ключа.
 3. В таблице может не быть ни одного внешнего ключа.
 4. В каждой таблице может быть только один внешний ключ.
 5. В таблице может быть несколько внешних ключей.
55. В каких из перечисленных ниже случаях необходимо задавать ограничение на уровне таблицы?
1. Если проверяемое выражение содержит ссылку на атрибут из другой таблицы.
 2. Если внешний ключ состоит из нескольких атрибутов.
 3. Если проверяемое выражение включает несколько атрибутов.
 4. Если внешний ключ ссылается на поле в родительской таблице, имеющее тип Timestamp.
 5. Если первичный ключ является составным.
56. Какое из определений соответствует понятию «Возможный ключ»?
1. Возможный ключ – это набор атрибутов, однозначно идентифицирующий кортеж отношения или строку таблицы.
 2. Возможный ключ – это набор атрибутов, по которому пользователи могут делать запрос к базе данных.
 3. Возможный ключ – это набор атрибутов, по которому выполнена индексация в таблице.
 4. Возможный ключ – это набор атрибутов, по которому одна родительская таблица связана с одной подчиненной (дочерней) таблицей.
57. В операторе выбора «SELECT» могут использоваться следующие агрегатные функции:
1. Not In()
 2. IN
 3. Count()
 4. Avg()
 5. Exists()
 6. Min()
58. Внешнее соединение – это:
1. Соединение, использующее внешние структуры-индексы, инвертированные списки.
 2. Соединение с отношением, физически хранящимся на внешнем источнике (другой сервер БД или другой файл).
 3. Соединение, которое выполняется на уровне модели трехуровневой архитектуры.
 4. Соединение, при котором в результирующее отношение попадают даже те кортежи из присоединяемого отношения, для которых нет соответствующих кортежей в соединяемом отношении.
59. Какие из перечисленных ниже операторов относятся к языку манипулирования данными?
1. CREATE PROCEDURE
 2. CREATE TABLE

3. SELECT
 4. ALTER TABLE
 5. CREATE TRIGGER
 6. UPDATE
 7. DROP TABLE
 8. INSERT INTO
 9. DELETE
60. Какие высказывания справедливы по отношению к оператору INSERT?
1. Это оператор ввода данных в БД.
 2. Этот оператор позволяет ввести только по одной строке сразу в несколько таблиц БД.
 3. Этот оператор позволяет ввести только одну строку в одну таблицу БД.
 4. Этот оператор позволяет ввести одну или несколько строк, но только в одну таблицу.
 5. Этот оператор позволяет ввести сразу несколько строк в несколько таблиц БД.
 6. Это оператор ввода данных в заданную таблицу БД.
61. Какие из приведенных определений являются корректными?
1. Транзакцией называется последовательность запросов к базе данных, которую выполняет одна программа или один пользователь.
 2. Транзакцией называется последовательность операций над базой данных, соответствующая некоторому бизнес-процессу.
 3. Транзакцией называется последовательность операций над базой данных, трактуемая как единая и неделимая единица работы. Все операции, составляющие транзакцию, либо выполняются, либо отвергаются.
 4. Транзакцией называется последовательность операций над базой данных, которая переводит базу данных из одного непротиворечивого (корректного) состояния в другое непротиворечивое (корректное) состояние.
62. Какие из перечисленных высказываний справедливы по отношению к понятию «уровень изолированности»?
1. Если уровень изолированности не задан, то транзакция выполняется с максимальным уровнем изолированности данной транзакции от других.
 2. Уровень изолированности определяет связь между двумя серверами баз данных, на которых хранятся две реплики.
 3. Уровень изолированности определяется для параллельно выполняемых транзакций, чтобы можно было оптимизировать план выполнения множества транзакций.
 4. Уровень изолированности определяет количество одновременно работающих с базой данных приложений.
 5. Это понятие не имеет отношения к базам данных.
 6. Уровень изолированности задается для транзакции, которая допускает некоторую некорректность в просматриваемых данных.
63. Выберите все корректные высказывания по отношению к термину «X-блокировка».
1. X-locks.
 2. Блокировка записи.
 3. Монопольная блокировка.
 4. Блокировка чтения.
 5. exclusive locks.
 6. Разделяемая блокировка.
 7. Предупредительная блокировка.
64. Выберите все корректные высказывания по отношению к термину «S-локировка».
1. Блокировка чтения.
 2. Блокировка без взаимного доступа.
 3. Монопольная блокировка.
 4. S-locks.

5. Shared locks.
6. Разделяемая блокировка.
65. Какие проблемы параллельного доступа возникают при работе с базами данных?
 1. Невозможность повторного чтения тех же данных.
 2. Чтение неподтвержденных данных (грязных данных).
 3. Удаление важных данных.
 4. Запись неподтвержденных данных.
 5. Наличие фантомов.
66. Какие системные очереди используются для выбора процессов, которым будет выделен очередной квант времени?
 1. Очереди ожидания ресурсов
 2. Очереди готовых процессов
 3. Очереди сообщений процессам
 4. Очереди обработки событий
67. Когда адресное пространство делится на сегменты, которые затем делятся на страницы речь идет о
 1. Сегментном распределении
 2. Страничном распределении
 3. Сегментно-страничном распределении
 4. Распределении перемещаемыми разделами
 5. Распределению фиксированными разделами
68. После многократного выделения и освобождения непрерывных областей памяти могут остаться только неиспользуемые области памяти малого размера. Это явление называется
 1. Фрагментацией
 2. Спулингом
 3. Свопингом
 4. Разделением
 5. Подкачкой
69. С чем связано использование в ряде ОС множества очередей готовых процессов?
 1. Со множеством используемых процессами ресурсов
 2. Со множеством механизмов коммуникации и синхронизации
 3. Со множеством приоритетов
 4. Со множеством файловых систем
 5. Со множеством потоков
70. Где могут выполняться системные задачи в ОС с симметричным мультипроцессированием на компьютере с несколькими процессорами (ядрами)?
 1. Только на выделенном процессоре (ядре)
 2. На нескольких выделенных процессорах (ядрах)
 3. На всех процессорах (ядрах), кроме выделенного процессора (ядра)
 4. На любом процессоре (ядре)
71. Где могут выполняться прикладные задачи в ОС с симметричным мультипроцессированием на компьютере с несколькими процессорами (ядрами)?
 1. Только на выделенном процессоре (ядре)
 2. На нескольких выделенных процессорах (ядрах)
 3. На всех процессорах (ядрах), кроме выделенного процессора (ядра)
 4. На любом процессоре (ядре)
72. Какие механизмы обычно используются для организации критических секций в параллельных потоках?
 1. Сигналы
 2. Сообщения
 3. Мьютексы
 4. Сокеты
 5. Таймеры

73. Какая файловая система из перечисленных не использует механизм журнализации?
1. NTFS
 2. JFS
 3. XFS
 4. ZFS
 5. EXTFS
74. Для чего в современных файловых системах используются битовые карты?
1. Для ускорения поиска используемых и не используемых имен файлов
 2. Для ускорения поиска свободных и занятых элементов, используемых для размещения файлов
 3. Для ускорения поиска свободных записей в файле
 4. Для ускорения чтения файлов в системах с расщеплением данных на несколько физических дисков
75. Что операционная система отдельно выделяет потоку в рамках одного процесса?
1. Файлы
 2. Мьютексы
 3. Семафоры
 4. Оперативную память
 5. Процессорное время
76. Какие операции над элементами характерны для структур данных «очередь» и «стек»?
1. Занесение элемента, извлечение элемента и очистка
 2. Занесение элемента, извлечение элемента и просмотр
 3. Занесение элемента и извлечение элемента
 4. Занесение элемента, извлечение элемента, просмотр, сортировка и удаление текущего элемента
 5. Поиск элемента и сортировка
77. Какие позиции стека доступны для занесения новых элементов?
1. Только вершина и дно стека
 2. Только вершина стека
 3. Все позиции, кроме дна стека
 4. Занесение элемента возможно в произвольную позицию
 5. Только дно стека
78. Просмотр линейного однонаправленного списка даёт следующий результат: 2, 4, 6, 8, 10, 12. Известно, что данные заносились в начало списка. Как они были упорядочены перед вводом в список?
1. По возрастанию
 2. В шахматном порядке
 3. Данные не были упорядочены
 4. По убыванию
79. Имеется двоичное дерево (не являющееся деревом поиска), содержащее целые числа. Прямой обход дерева даёт следующий результат: 2, 4, 6, 8, 10, 12, 14. Какое число содержится в корне дерева?
1. 6
 2. 14
 3. 8
 4. 2
 5. 10
80. Имеется двоичное дерево поиска, содержащее целые числа от 1 до 7. Каким будет результат симметричного обхода дерева?
1. 4, 2, 1, 3, 6, 5, 7
 2. 7, 6, 5, 4, 3, 2, 1
 3. 1, 2, 3, 4, 5, 6, 7
 4. 1, 3, 2, 5, 7, 6, 4

5. 4, 2, 6, 1, 3, 5, 7
81. Имеется AVL-сбалансированное двоичное дерево, содержащее 31 узел. Какова высота этого дерева?
1. 4 уровня
 2. 5 уровней
 3. 3 уровня
 4. 1 уровень
 5. 2 уровня
82. Имеется неупорядоченный массив целых чисел из 10 элементов. Сколько операций сравнения потребуется для установления факта отсутствия искомым данных в этом массиве?
1. 9
 2. 0
 3. 10
 4. 5
 5. 1
83. Имеется упорядоченный массив целых чисел из 9 элементов. Сколько операций сравнения потребуется при двоичном (бинарном) поиске для нахождения искомого ключа, если он находится точно в середине массива?
1. 0
 2. 9
 3. 5
 4. 1
 5. 8
84. Была произведена Быстрая сортировка (Хоара) упорядоченного по убыванию массива размером n . Какова минимальная временная сложность $T_{\min}(n)$ указанного алгоритма?
1. $O(n \cdot \log n)$
 2. $O(n^2)$
 3. $O(n \cdot n)$
 4. $O(n)$
 5. $O(1)$
85. Как называется сортировка, происходящая в оперативной памяти?
1. сортировка таблицы адресов
 2. полная сортировка
 3. сортировка прямым включением
 4. внутренняя сортировка
 5. внешняя сортировка
86. Есть несколько способов представления графа в памяти ЭВМ. Какой из способов приведенных ниже не относится к ним?
1. матрица инцидентности
 2. матрица смежности
 3. список ребер
 4. массив инцидентности
87. Как называется граф, содержащий только дуги?
1. ориентированным
 2. неориентированным
 3. простым
 4. смешанным
88. Найти длину пути из вершины a до вершины d графа, заданного следующей матрицей смежности:

	a	b	c	d
a		2	1	
b				3
c				
d			6	

1. 1
2. 2
3. 3
4. 5
5. 6

89. Какие вершины графа, заданного следующей матрицей смежности, будут содержаться в минимальном остовном дереве этого графа:

	a	b	c	d
a		2	1	
b				3
c				
d			6	

1. a, b, c
 2. a, b, c, d
 3. b, c
 4. a, c
 5. b, d
90. Как будет выглядеть хеш-таблица длиной 5 при хеш-функции $h(x) = x^2 \bmod 5 + i$ (метод разрешения коллизий – линейное опробование, i - номер попытки разрешить коллизию), если в нее добавляются следующие данные: 3, 9, 4, 5?
1. 5, 4, 9, «пусто», 3
 2. 5, 9, 4, «пусто», 3
 3. 3, 9, 4, 5, «пусто»
 4. «пусто», 3, 4, 5, 9
91. Наиболее важным свойством библиотеки классов ADO .NET является то, что классы спроектированы для работы в _____ режиме, что важно в современном веб-ориентированном мире.
1. Отключенном.
 2. Подключенном.
92. Для доступа к удаленным источникам данных из приложения клиента необходимо предоставить параметры соединения:
1. Имя машины, на которой запущен сервер баз данных.
 2. Имя машины клиента.
 3. Регистрационные данные для входа.
 4. Имя источника данных.
 5. Имя провайдера.
93. Вызов хранимой процедуры с командным объектом предусматривает указание:
1. Имени хранимой процедуры.
 2. Только входных параметров процедуры.
 3. Только выходных параметров процедуры.
 4. Параметров процедуры.
94. Класс DataSet в технологии ADO .NET был спроектирован как _____ контейнер данных.
1. Отключенный.
 2. Подключенный.
95. В классе DataSet технологии ADO .NET отсутствует понятие:
1. Соединения с базой данных.
 2. Отношения (связей) данных.
 3. Ограничения данных.
96. Ключевое отличие между классом DataSet и устаревшим иерархическим объектом RecordSet состоит:
1. В способе представления отношений.
 2. В способе хранения данных.

97. Разработка приложения, взаимодействующего с данными, часто предполагает 16
разделение приложения на:
1. Уровни.
 2. Потоки.
 3. Ветви.
 4. Пространства имен.
98. Общим требованием при написании многоуровневой системы, будь то клиентское Windows- или веб-приложение, является передача как можно меньшего объема
1. Данных между уровнями.
 2. Программного кода между уровнями.
 3. Документации между уровнями.
99. Для чтения данных из внешнего источника и вставки их в объект DataSet существует два основных способа:
1. Использование адаптера данных.
 2. Чтение и запись данных в формате XML.
 3. Чтение и запись данных в формате SQL.
 4. Использование коммутатора данных.
100. Объект DataSet является контейнером данных, который позволяет организовать:
1. Межуровневые потоки данных приложения.
 2. Межуровневые соединения для приложения.