

Ответы к РК

1. Приведите пример «качественной» информации. Ответ поясните
Основные свойства «качественной информации»:

- репрезентативность
- содержательность
- достаточность
- доступность
- актуальность
- своевременность
- точность
- достоверность
- устойчивость

Примером может быть ресурс «википедия».

2. Предположим, вы работаете в крупной компании, занимающейся поиском в интернете. Вам нужно проанализировать некоторое количество сайтов и выделить среди них те, которые предоставляют наиболее «качественную» информацию. По каким признакам вы их будете оценивать? Ответ поясните
Вопрос подразумевает ответ о свойствах качественной информации, указанных выше.

Помимо этого, можно также рассказать о таких параметрах, как PageRank, ТИЦ (тематический индекс цитирования).

3. Предположим, вы работаете в крупной компании, занимающейся поиском в интернете. Вам нужно проанализировать некоторое количество сайтов и выделить среди них те, которые являются спамом и предоставляют наименее «качественную» информацию. По каким признакам вы их будете оценивать? Ответ поясните

Вопрос подразумевает ответ о свойствах качественной информации, указанных выше.

Помимо этого, можно также рассказать о таких параметрах, как PageRank, ТИЦ (тематический индекс цитирования).

4. Как вы считаете, какое распределение имеет наибольшую энтропию: равномерное или нормальное? Ответ поясните

Вопрос относится к понятию энтропии. Согласно определению энтропии, равномерное распределение соответствует наибольшей энтропии.

5. Как вы считаете, какое сообщение несет в себе больше информации: свет включен или ветер северо-восточный? Ответ обоснуйте

Вопрос относится к понятию информации.

Количество информации можно рассчитать по формулам:

$I = \log N$, где N – количество возможных состояний

Или формула Хартли: $I = \sum -\frac{1}{k} \log \frac{1}{k}$, где $1/k$ – вероятность события.

Применим их в нашем случае:

Свет: 2 состояния («вкл.» и «выкл.»). Поэтому $I = \log 2 = 1$, или $I = -1/2 \log(1/2) - 1/2 \log(1/2) = 1$

*Ветер: 8 состояний («С», «СВ», «СЗ», ...). Поэтому $I = \log 8 = 3$, или $I = \sum_{i=1..8} -\frac{1}{8} \log \frac{1}{8} = -8 * 1/8 * (-3) = 3$.*

Логарифм в данном случае взят двоичный.

Видно, что сообщение «ветер северо-восточный» несет в себе больше информации.

6. Поясните, в чем основные преимущества шаблона проектирования MVC (Model-View-Controller). В каких случаях и с какой целью его следует использовать?

Шаблон MVC (Модель-Контроллер-Представление) служит для разделения представления данных от их обработки и хранения.

В рамках этой модели разделены пользовательский интерфейс, модель данных и взаимодействие с пользователем.

Модель отвечает за предоставление знаний

Представление отвечает за отображение этих знаний/данных, то есть часто за графический интерфейс.

Контроллер обеспечивает связь между пользователем и системой.

7. Для чего существует инкапсуляция? Приведите пример.

Инкапсуляция – сокрытие реализации для более удобного использования предоставленного интерфейса. Пользователь взаимодействует с объектом только через предоставленный ему интерфейс.

За счет инкапсуляции достигается максимальная локализация изменений реализации. Примеры в лекциях.

8. Для чего существует полиморфизм? Приведите пример.

Возможность объектов с одинаковой спецификацией иметь различную реализацию. Примеры в лекциях.

9. Для чего существует наследование? Приведите пример.

Механизм ООП, позволяющий описать новый класс на основе уже существующего, свойства и функциональность нового класса заимствуются у родительского. Примеры в лекциях.

10. Предположим, перед вами стоит задача разработки крупной системы. Какую из двух моделей (каскадной и спиральной) вы предпочтете? Почему?

Ответ: спиральная модель и описание ее преимуществ.

11. Для чего существуют шаблоны проектирования? Какие из шаблонов проектирования вам знакомы и в каких случаях их следует использовать?

Ответ: примеры шаблонов «Наблюдатель», «Одиночка», «Абстрактная фабрика», «Мост», описания есть в лекциях.

12. Представьте, что в SOA вместо слабого связывания используется сильное связывание. Какие вы видите недостатки в этом?

Сильное связывание подразумевает доступ к части реализации. Изменение реализации станет затруднительнее, поскольку ее придется менять не только на уровне поставщика сервиса, но и его потребителей.

13. Вы проектируете поисковую систему. Опишите, какую архитектуру вы будете использовать и почему.

Трехуровневая архитектура. Тонкий клиент. На клиенте только часть представления (пользовательский интерфейс), все сложные вычисления – на удаленных серверах.

14. Вы проектируете социальную сеть. Опишите, какую архитектуру вы будете использовать и почему.

Аналогично предыдущему.

15. Предположим, вы – системный архитектор. Перед вами стоит задача – спроектировать систему, которая должна удовлетворять следующим требованиям:

- а. Иметь общее хранилище данных. Количество данных – 5 ТБ
- б. К хранилищу должно обращаться множество пользователей с терминалов/маломощных РС. Пользователи должны иметь возможность чтения, изменения, обработки данных, вызова процедур .

Какую архитектуру вы предпочтете использовать и почему?

Тонкий клиент, поскольку он предполагает хранение всех данных и осуществление сложных вычислительных операций на удаленных серверах.

16. Предположим, вам нужно написать библиотеку (систему классов) парсеров различных форматов: XML, JSON. Опишите вашу систему классов.

Объявим абстрактный класс:

AbstractParser:

- *createParser() //конструктор*
- *Elements parse(string) // метод для разбора строки, возвращает некоторое дерево Elements (структуру элементов)*
- *Elements getElements() //метод, возвращающий уже существующее дерево Elements*

Объявим два наследника этого класса:

XmlParser:AbstractParser

JSONParser:AbstractParser

В них будет своя реализация метода parse(string).

17. Предположим, вам нужно написать библиотеку (систему классов) сериализаторов различных форматов: XML, Binary, SOAP. Опишите вашу систему классов.

Объявим абстрактный класс:

AbstractSerializer:

- *createSerializer() //конструктор*
- *serialize(Object, file) // метод для сериализации объекта в конкретный файл*
- *Object deserialize(file) // метод для десериализации объекта из файла*

Объявим три наследника:

XMLSerializer:AbstractSerializer

BinarySerializer:AbstractSerializer

SOAPSerializer: AbstractSerializer

У каждого из них все методы абстрактного класса будут реализованы в соответствии с форматом файлов.

18. Спроектируйте схему БД/систему классов для сервиса аренды машин.
Manufacturer(id PK, name) - производитель
Model(id PK, name, manufacturer_id FK) – модель машины
Category(id PK, name) – категория машины
Car(id PK, name, model_id FK, mileage, category_id FK) – машина (одна машина относится к одной категории и модели)
Client(id PK, name, address, passport) – данные клиента
Status(id PK, name) – статус заказа
Booking(id PK, date_from, date_to, car_id FK, client_id FK, status_id FK) – заказы.
В заказе указываются даты «от» и «до», id машины, клиента и статуса заказа.
19. Спроектируйте схему БД/систему классов для социальной сети
School(id PK, name) //школа
Work (id, name) // работа
Status(id, status, update_time) // статус в профиле
Profile(id PK, name, gender, birth, favourites, school_id FK, work_id FK, status_id FK) // профиль пользователя: имя, пол, дата рождения, любимые фильмы-цитаты-книги, школа, работа и статус. Следует заметить, что в рамках данной схемы у пользователя одновременно могут быть указаны только одна школа, одна работа и один статус. Для возможности указания многих, следует выделить дополнительную сущность.
Friends(id PK, are_friends, profile_id_1 FK, profile_id_2 FK) // друзья: ссылки на 2 профиля и флаг друзья.
Данная схема сильно упрощена и может быть существенно расширена.
20. Спроектируйте схему БД/систему классов для сервиса отслеживания погоды.
Calendar (id PK, date) // календарная дата
Type(id PK, name) //типы локаций (пустыня, тайга,...)
Location(id PK, type_id FK, name, description) //локация (город). Можно расширить сущностью «Страна» и ссылкой на сущность
StandartForecast(id PK,summary, desacription) //стандартный прогноз («облачно, небольшие дожди», “солнечно, без осадкой”)
LocationWeather(location_id FK PK,calendar_id FK PK, summary_id FK, temperature, wind) //погода в конкретной регионе: в одном регионе одновременно может быть только один прогноз.
21. Предложите модель/структуру данных для хранения миллиарда телефонных номеров. Ответ обоснуйте
Префиксное дерево - <http://en.wikipedia.org/wiki/Trie>
22. Представьте, что ваша система работает параллельно на нескольких машинах. Машины соединены в локальную сеть. Система содержит некоторые объекты, состояние которых постоянно меняется и его нужно отслеживать. Если у объекта А на машине РС1 изменилось состояние, то машина РС2 должна неким образом об этом узнать. Опишите механизм, при помощи которого вы осуществите такую функциональность.

Предполагался ответ, касающийся механизма сериализации (для передачи состояния одного объекта между различными машинами в сети).

23. Спроектируйте систему классов для описания форм Windows (у формы есть размер, имя, на ней могут присутствовать кнопки(buttons), радиокнопки(radio buttons), чекбоксы(check boxes), текстовые поля (text areas)). Опишите основные свойства и методы.

Объявим базовый класс Widget:

Свойства

- *Size* //размер
- *Position* //положение на экране
- *Name* // название
- *Controls[]* //элементы управления, расположенные на нем

Методы:

- *create(name, size, position)*
- *draw()*
- *hide()*
- *show()*
- *remove()*
- *addControl(Control)*

Класс Control (элемент управления):

Свойства

- *Name*
- *Size*
- *Position*

Методы:

- *scale(size)*
- *refresh()*

Помимо этого, допустим, у нас имеются различные виды кнопок(button, radiobutton, checkbox) и текстовых полей(textbox, textarea) .

Для кнопок объявим базовый класс ButtonBase, где будут описаны основные методы, свойства и события для кнопок (например,click(), clicked). Аналогично поступим с классом для текстовых полей TextBoxBase.

В результате получим следующую систему классов для элементов управления:

Button: ButtonBase, Control

RadioButton: ButtonBase, Control

CheckBox:ButtonBase, Control

TextBox: TextBoxBase, Control

TextArea:TextBoxBase, Control

24. Спроектируйте систему классов/схему БД для электронной почты.

Folder(id PK,parent_id FK, name, description) // директория для письма, parent_id – ссылка на родительскую директорию

Address(id PK, name, user_name) // список контактов (имя реальное, имя виртуальное)
Message(id PK, folder_id FK, sender_id FK, datetime, subject, body, is_unread, is_junk, is_star, priority) //письмо. Следует заметить, что письмо всегда имеет ОДНОГО отправителя и ОДНОГО или МНОГО адресатов. В данном случае, в этой сущности присутствует только отправитель (одно письмо-один отправитель)
MessageAddress(message_id FK PK, address_id FK PK, header_type)
//дополнительная сущность для осуществления связи многие-ко-многим (письмо может иметь более одного адресата). header_type – тип заголовка (cc, to)
Attachment(id PK, filename, content, extension, size) //приложение к письму. Приложений может быть опять более одного
MessageAttachment(message_id FK PK, attachment FK PK) //связь многие-ко-многим для письма и приложения.