

Квалификационные задания для .NET-разработчиков

Данные задания разработаны для облегчения проверки Ваших знаний, и для Вашего удобства, так как Вы можете их выполнить в любое удобное для Вас время и в обстановке, которую Вы считаете оптимальной. Ответы на задания Вы можете составить в свободной форме и прислать нам вместе с Вашим резюме на почтовый ящик hell@parcsis.org. После проверки Ваших заданий и резюме, мы обязательно сообщим о нашем мнении.

Обратим ваше внимание на то, что заданий много, а свободного времени для их реализации традиционно мало. Каждое задание служит целью проверить глубину Ваших знаний в определенной области. Ни одно из заданий не является обязательным, однако, выполнение каждого из них добавляет Вам ценности при рассмотрении Вашей кандидатуры и дает нам представление о том, какие задачи для Вас более привычны и понятны.

В общем, это все. Перейдя на следующую страницу, Вы увидите наши задания.

Тестовое задание 1:

Реализовать серверную часть приложения для хранения списка родственников.

Серверная часть (сервис) должна предоставлять следующие функции:

1. Основная часть.
 - *Выдача списка родственников. Детализация: Фамилия, Имя, Отчество, Дата рождения, Адрес. С возможностью фильтрации строк по любой комбинации полей*
 - *Детализация по конкретному родственнику (карточка родственника)*
 - *Добавление родственника*
 - *Редактирование родственника*
 - *Удаление родственника*
2. Карточка родственника.
 - *Выдача развернутой информации о человеке.*
 - *Выдача списка родственников для данной персоны.*
 - *Добавление человека как родственника персоны.*
 - *Удаление человека из списка родственников персоны.*
 - *Изменение родственных связей данного человека (удаление связей, определение новых, как прямых=я отец моего сына, так и обратных=я сын моего отца).*

Сервис необходимо реализовать на базе http. На вход подаются параметры посредством Get или Post. На вывод отдается результат в формате XML или JSON в зависимости от параметра (наименование параметра придумаете сами) отвечающего за формат вывода с сервиса. Функциональность сервиса продемонстрировать при помощи реализации сценариев использования в Unit-test-ax (NUnit или MS Quality Tools). Тесты инициируют http-запросы на сервис, эмулируя действия пользователя.

Доп. задание (необязательно): реализовать WPF-клиент для работы с сервисом.

Тестовое задание 2:

Реализовать клиент-серверное приложение «Удаленный менеджер загрузок».

Сервис на базе WCF с подключающимся к нему WPF клиентом и/или http сервис и html-страницы. Задачи загрузки физически выполняются на сервере, на котором расположен сервис. Загруженные файлы остаются на сервере, предполагается, что Вы физически с него их заберете.

Требования:

- *Хранение очереди загрузок.*
- *Параллельная загрузка нескольких задач.*
- *Пауза/Возобновление загрузки для задачи.*
- *Изменение количества одновременных задач «на лету».*
- *Клиентское приложение продолжает отвечать на действия пользователя при осуществлении запросов на сервер.*

Тестовое задание 3:

База данных имеет следующую структуру:

Customers	Orders
<i>Id (int, identity)</i>	<i>Id (int, identity)</i>
<i>Name (varchar)</i>	<i>CustomerId(int, FK – Customers Id)</i>
	<i>Date (datetime)</i>
	<i>Sum (money)</i>

Составить запрос в одну строку к базе данных:

1. Выводящий количество заказов для каждого клиента по дням за последние 2 месяца.

Результат вида:

Вася 05.06.2010 5
Вася 04.06.2010 0
Петя 05.06.2010 1
Петя 04.06.2010 4

2. Выводящий максимальную сумму заказа в рамках дня для каждого клиента по дням за последнюю неделю.

Результат вида:

Вася 05.06.2010 20
Вася 04.06.2010 0
Петя 05.06.2010 10
Петя 04.06.2010 400

Тестовое задание 4:

Составить регулярное выражение, выделяющее из текста примера следующие именованные группы:

1. Тип якоря в тексте (пункт, статья)
2. Нумерация якоря в тексте с учетом вложенности (5, 2.3 -> 2 и вложенное 3)
3. Якорь действия (после слов)
4. Якорное выражение (после каких именно слов)
5. Действие (дополнить)
6. Параметр действия (словами, пунктом)
7. Операнд действия (какими словами)

Примечание: Регулярное выражение должно быть одно. Именованная группа, это named capture. Т.е. то что можно спросить в качестве результата регулярного выражения по ключу-имени. В данном конкретном задании, необходимо получить машинопонятную инструкцию что делать с изменяемым текстом.

Пример (для первого абзаца текста):

1. TextAnchorType = «пункт»
2. AnchorNumeration = [] {2,1}
3. ActionAnchor = «после слов»
4. AnchorExpression = «в здании администрации»
5. Action = «дополнить»
6. ActionParameter = «словами»
7. ActionOperand = «", а также сотрудникам федеральных органов государственной власти, организаций, непосредственно и постоянно взаимодействующих с администрацией края, иными организациями, расположенными в здании администрации края,»

Текст:

пункт 2.3 после слов "в здании администрации" дополнить словами", а также сотрудникам федеральных органов государственной власти, организаций, непосредственно и постоянно взаимодействующих с администрацией края, иными организациями, расположенными в здании администрации края,";

пункт 2.7 после слов "председатели (заместители) постоянных комитетов Совета народных депутатов" дополнить словами ", руководитель аппарата, его заместитель, руководители структурных подразделений краевого Совета народных депутатов,";

пункт 5.4 после слов "не находящихся на балансе управления делами администрации края" дополнить словами ", и краевого Совета народных депутатов, структурных подразделений администрации края, иных организаций, размещенных в здании администрации края."

пункт 5 после слов "краевого бюджета" дополнить словами "и иных не запрещенных законодательством источников";

статью 6 после слов "предусмотренных адресной инвестиционной программой," дополнить словами "и иных не запрещенных законодательством источников."

Положение дополнить пунктом 2.8 следующего содержания: "Один экземпляр акта приемки в эксплуатацию, копия паспорта и проектная документация должны храниться в органах архитектуры местного самоуправления. Эти материалы передаются застройщиками (инвесторами) безвозмездно".

Вопросы на проверку базовых знаний C#.NET:

1. Платформа .NET. Почему «платформа»? В чем принципиальное отличие C# от C++.
2. Опишите паттерн одиночка (*синглтон, singleton*)
3. Чем отличается абстрактный класс от интерфейса (*Interface*). Расскажите о ситуациях, когда у Вас вставал выбор, что использовать, и на основании каких факторов был сделан выбор.
4. Боксинг и анбоксинг. Что это такое? В каких случаях выполняется? Влияние на производительность.
5. Опишите (кратко) архитектуру приложения, где вами были использованы основные принципы ООП.
6. Многопоточность. Какие аспекты необходимо учитывать при разработке приложения, реализующего многопоточный доступ к одним ресурсам; с пользовательским интерфейсом? Какие существуют инструменты для управления коллекцией потоков?
7. Дженерики. В чем отличие от «обычных» коллекций? Как реализуется расширенный функционал самой платформой?
8. Сериализация. Что такое сериализация? Какие есть виды сериализации? Как можно управлять Xml-сериализацией. Как менять внешний вид результирующего документа?
9. Что можно сделать в клиент-серверном приложении для передачи в качестве параметра объекта, который не поддерживает Xml-сериализацию?
10. Понятия «аутентификация» и «авторизация». В чем разница?
11. Какие проблемы у Вас возникали и в чем особенность отладки WCF-сервиса и WCF-клиента?
12. Опишите опыт создания WPF приложений. Что такое WPF и в чем отличие от GDI+? Различие биндинга на интерфейс коллекции List<T> и ObservableCollection<T>.

Вопросы на проверку глубоких знаний .NET

1. Сценарий выполнения приложения на .NET.
2. Опишите опыт создания .NET приложений под виртуальные машины .NET отличные от стандартных (т.е. не Desktop вариант на Microsoft Windows). В чем особенности и чем они обусловлены? Какие проблемы у Вас возникали (могут возникнуть)?
3. В чем плюсы и минусы паттерна одиночка? Почему его иногда называют антипаттерном?
4. Паттерны проектирования. Какие из паттернов Вы знаете. Расскажите о последних случаях применения паттернов вами при разработке или рефакторинге кода. Что было сделано в архитектуре приложения? Какие проблемы удалось решить?
5. Какие сложности возникают при работе с интерфейсом пользователя из не интерфейсного потока? Как решаются эти сложности?
6. Сериализация. Каким образом можно уменьшить объем передаваемых данных в случае бинарной сериализации? Особенности сериализации объектов типа XmlDocument. Какие проблемы возникают? Как решаются?
7. Рассмотрим архитектуру: Сервер, предоставляющий клиентам набор WCF-сервисов, поддерживает свой собственный механизм аутентификации, которая является необходимой для выполнения всех команд, кроме первой, имеющей сигнатуру: `bool Authorize(string Login, string Password)`.
8. Каким образом можно обеспечить параллельную работу нескольких клиентов?
9. WPF. Какие сложности у Вас возникали при разработке приложений с использованием WPF? Как решались?

Благодарим Вас за время, потраченное на выполнение заданий!