

Домашнее задание на позицию “Стажера MVC”

| | |
|--|---|
| Фамилия Имя Отчество | |
| Возраст | |
| Электронная почта | |
| Телефон | |
| Текущее место работы (если не работаете, то дата увольнения) | |
| Ожидаемый уровень дохода | На этапе стажировки, и по ее завершению |
| Дата выдачи задания | |
| Дата завершения задания | |

Напутствие

Уважаемый кандидат, ниже представлены задачи и вопросы которые проверяют различные критерии Вашего опыта. Перед тем как начать выполнение, необходимо заполнить карточку о себе в начале документа. По мере решения заданий у вас могут возникать вопросы, в этом случае вы их можете задавать указывая вопросы в комментариях к непонятному месту в тексте. При решении заданий допустимо использовать любые источники информации: литература, интернет, звонок другу. Не допускается передача решения данного задания другим лицам.

Все задания могут быть запакованы только в архив ZIP. Все файлы должны быть простыми текстовыми - не допускается использовать Word или другие специальные редакторы.

Все задания нужно рассматривать как одно целое, т.к. требования одного задания могут повлиять на решение другого задания. От претендентов так же ожидаются вопросы к заданию. **Без уточняющих вопросов выполненное решение рассматриваться не будет.**

Желаем успеха и надеемся вам будет интересно и у вас все получится

Wildberries

Задание 1

Расскажите нам, почему вам интересно работать в компании Wildberries и почему хотите стать специалистом MVC. Задание является обязательным (ответ нужно расположить ниже и выделить другим цветом).

Задание 2

1. Необходимо разработать классы-провайдеров (посредник для доступа к хранилищу данных), которые имеют только один открытый метод, при вызове которого возвращается список доступных номиналов монет. Каждая монета имеет номинал и признак, заблокирована или нет. При этом первый класс должен возвращать данные из xml документа, а второй из базы данных MS SQL. Задача так же разработать xml документ и заполнить его номиналами с примера в изображении;
2. Необходимо разработать классы-провайдеров, которые имеют только один открытый метод, при вызове которого возвращается список доступных напитков. Каждый напиток имеет стоимость, кол-во, название. При этом первый класс должен возвращать данные из xml документа, а второй из базы данных MS SQL. Задача так же разработать xml документ и заполнить его напитками на свое усмотрение, но не меньше 10 вариантов;
3. Используя UnityContainer необходимо научить систему получать нужный провайдер как для номиналов монет, так и для напитков кофе. Что выбирать должно задаваться в конфигурационных файлах, по умолчанию данные должны браться из XML;
4. Все провайдеры данных должны иметь в обязательном порядке деструктор;
5. Необходимо в коде привести пример использования using для использования интерфейса IDisposable;

Задание 3

1. Необходимо разработать MVC-приложение по дизайну
2. Используя принципы DDD (design driven development) для разделения на уровни (presentation, application, domain, infrastructure).
3. Для взаимодействия между уровнями необходимо использовать DI и IoC-контейнер (например, UnityContainer);
4. Обязательно использование JQuery;
5. Список доступных монет должен браться из провайдера, созданного в задании №2;
6. Блоки монет и напитков отрисовывать при помощи шаблонизатора (например jsRender)
7. Если монета разблокирована, показать ее как зеленый квадрат с номиналом, если монета заблокирована, показать ее как красный круг с номиналом, используя стили.

8. Если выбрать элемент который заблокирован, кнопка “Пополнить” блокируется, иначе она разблокирована;
9. После того, как нажата кнопка “Пополнить”, нужно увеличить сумму баланса. Используя стили, фон у поля баланса должен быть красный, если баланс равен нулю, и зеленым, если он не равен нулю. В самом поле сумму тоже можно менять, но позволяет вводить только целые числа больше нуля или ноль, используя валидацию данных.
10. Так же нужно показать список напитков, но при этом список должен быть горизонтальным и быть в одну строку. Если места не хватает должна появляться горизонтальная прокрутка. В случае если денег хватает на напиток, его фон должен быть зеленым, иначе красным. При желании в ячейки можно добавить картинки напитков.
11. Когда выбран доступный напиток, кнопка Купить активируется, иначе она заблокирована.
12. При нажатии кнопки “Купить”:
 - a. на сервер отправляется запрос покупки
 - b. экшн должен принимать модель с полями (для биндига модели сделать свой биндер)
 - i. id товара
 - ii. количество (для данного примера всегда будет 1)
 - c. количество купленного предмета в хранилище уменьшается
13. В ответ вернуть оставшиеся предметы:
 - a. используя LINQ его обязательно отфильтровать и оставить только напитки, где кол-во больше нуля (при отображении в форме), а так же отсортировать по возрастанию стоимости. Список монет тоже должен быть отсортирован с помощью LINQ в порядке возрастания;
 - b. результат выборки закешировать (на минуту) с помощью redis
14. Сайт необходимо разместить на любом бесплатном хостинге для тестирования, пример кода отправлен нам;
15. Пример структуры БД нужно нарисовать в качестве графической схемы в любой из нотаций;

Номиналы для пополнения

Сумма на балансе

15 р 10 р 5 р

3 р 2 р 1 р

Пополнить

25

Сдача

Чай 10 р Кофе 15 р Молоко 12 р Кокао 8 р

Купить

Задание 4

1. Нужен ответ в один абзац, что такое паттерны проектирования и разработки, какие они дают преимущества в целом;
2. Необходимо придумать и разработать структуру классов предметной области (используя Domain Driven Design? Мы понимаем, что это не простая тема, но у нас есть практика её применения. А для вас это будет плюсом) на выбор: автотранспорт, библиотека на языке C#. При этом пример обязан содержать: интерфейс или интерфейсы; абстрактный класс или классы и абстрактные методы. Так же в этом примере необходимо показать принцип работы полиморфизма;
3. Необходимо подготовиться к устному вопросу изучить по одному паттерну GOF (не singleton) и паттерну SOLID. Ответ должен содержать назначение паттерна, его плюсы и минусы, а так же объяснить принцип работы.