Ответьте как можно быстрее на следующие вопросы:   
**1. Как правильно обратиться к нестатическому методу из статического в контексте одного класса? self::nonStaticMethod() или static:: nonStaticMethod()? Аргументируйте своё решение.**

self::nonStaticMethod()

**2. Какой оптимальный способ привести строку вида «1155» к типу integer и float (отдельно)?**

**a)**

$var=**"1155"**;  
$int=(integer)$var;

b) $var=**"1155"**;  
$floar=*floatval*($var);

3. **Отрефакторьте следующий код:** [**http://i.imgur.com/3hI92pY.png**](http://i.imgur.com/3hI92pY.png)**.**

**function** greatherThetOne(int $number)  
{  
 **return** $number > 1 ? **true** : **false**;  
}

**4. Описан некоторый интерфейс, например, «ProductInterface». Интерфейс содержит описание следующих методов:   
a. public getPrice();   
b. public getName(): string;   
c. protected setPrice(): void;   
d. protected setName(): void;   
Что не так с этим интерфейсом?**

Интерфейс не может содержать не публичные методы, т.к в дальнейшем они должны быть реализованы в классе, реализующий этот интерфейс.

5. Имеется очень плохо реализованный метод, в котором содержится много условий для проверки корзины текущего пользователя. Например, в этом методе описана проверка минимальной суммы корзины: <http://i.imgur.com/zjqrD2A.png>. Количество условий может продолжаться до бесконечности, в зависимости от требований бизнеса.   
Не очень грамотный программист просто запихал эти условия в if-elseif спагетти. Какой подход или паттерн можно было бы применить в данном случае, для того, чтобы упростить код и сделать его более читаемым? Вариантов решения задачи может быть несколько.

**6. Где вы будете хранить бизнес-логику в типовом MVC приложении? В контроллере? В модели? Из чего может состоять слой модели (Model)?**

В MVC бизнес логику буду хранит в модели, т.к модель это ещё и модель предметной область.

Также это позволит использовать модель отдельно от конкретного приложения.