**Разработка библиотеки классов**

Для того чтобы при продаже товаров можно было быстро рассчитывать стоимость с учетом скидок, необходимо разработать библиотеку классов. Данная библиотека будет подключаться к основному проекту и должна быть представлена в виде .dll файла. Чтобы система правильно интегрировалась вам необходимо обязательно следовать правилам именования библиотек, классов и методов в них. В случае ошибок в рамках именования ваша работа не может быть проверена и ваш результат не будет зачтен. Классы и методы должны содержать модификатор public, чтобы внешние приложения могли получить к ним доступ.

В качестве названия для библиотеки необходимо использовать: WSUniversalLib. Вам необходимо загрузить исходный код проекта с библиотекой в отдельный репозиторий с названием, совпадающим с названием проекта.

**Класс расчета стоимости**

Метод должен рассчитывать стоимость проданных товаров, учитывая систему скидок. Величина скидки зависит от типа товара и от стоимости покупки. Стоимость должна быть округлена до рублей.

Процент скидки для различных типов товаров (product\_type):

Тип товара 1 – 3%,

Тип товара 2 – 5%,

Тип товара 3 – 1%.

При этом нужно учитывать, что скидка дается только в том случае, если стоимость купленных товаров превышает 100 000 рублей.

Если после применения скидки в зависимости от типа товара стоимость превышает 300 000 рублей, то дается дополнительная скидка 2% независимо от типа товара.

При этом если в качестве параметров метода будут приходить несуществующие типы товаров или другие неподходящие данные, то метод должен вернуть -1.

Например, необходимо продать изготовить 5 единиц товара 2 типа ценой 70 000 рублей. Стоимость без учета скидки будет равна 350 000. Так как товар типа 2 и стоимость покупки превышает 100 000 рублей, на него дается скидка 5% и стоимость становится равной 332 500. Так как после применения скидки с учетом типа товара стоимость превышает 300 000, дается дополнительная скидка 2% и стоимость становится равной 325 850.

Спецификация метода представлена в отдельном файле в ресурсах.

**Разработка модульных тестов (Unit-tests)**

Для выполнения процедуры тестирования созданного вами метода библиотеки WSUniversalLib, возвращающего стоимость товаров, вам необходимо создать отдельный проект модульных тестов.

В рамках проекта разработайте тесты, максимально полно покрывающие функционал метода. Ничего страшного, если ваш метод работает не совсем идеально и тесты могут быть не пройдены в связи с этим - в данном модуле это не так важно.

Обратите внимание, что имена тестов должны отражать их суть, т.е. вместо TestMethod1() тест следует назвать, например, GetCostForProduct\_NonExistentProductType() для тестирования случая передачи несуществующего типа товара.

Необходимо разработать модульные тесты, которые на основании исходных данных можно условно разделить на 2 группы следующим образом: 10 методов низкой сложности и 5 методов высокой сложности.

**Разработка тестовых сценариев (Test-cases)**

Для выполнения процедуры добавления нового пользователя Вам нужно описать четыре сценария.

Необходимо, чтобы варианты тестирования выполняли проверку работы приложения при использовании различных исходных данных. Для описания тестовых сценариев в ресурсах предоставлен шаблон testing-template.docx.