



Сессия 3

Разработка библиотеки классов

Общие требования

В связи со стремительным развитием нашей системы было решено вынести некоторый важный функционал за рамки основного проекта и сделать библиотеку классов, которую мы сможем подключать к любому нашему проекту в случае расширения. Данная библиотека будет подключаться к основному проекту и должна быть представлена в виде .dll/.jar файла или папки с файлом .ру.

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 (если это реализуемо в рамках платформы), чтобы внешние приложения могли получить к ним доступ.

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

Класс аналитики

Вам необходимо создать класс с названием Analytics, который будет позволять проводить аналитику различных процессов в рамках компании.

Реализуйте метод, который принимает в себя список объектов даты и времени по совершенным покупкам/заказам в рамках нашей компании, а возвращает список дат (без времени), отсортированный в порядке уменьшения частоты заказов. Это необходимо, чтобы наша компания могла прогнозировать наиболее высокий спрос на следующий год для обеспечения более качественного оказания услуг.

Возвращаемые данные должны содержать только даты для первого числа каждого месяца и 00:00 минут. Например, вам поступили следующие данные: 2019-12-12 14:43, 2019-12-01 15:05, 2019-11-04 09:01, а, значит, самый популярный месяц - декабрь. Вам необходимо вернуть следующие данные: 2019-12-01 00:00, 2019-11-01 00:00. В случае совпадения характеристик популярности сперва нужно вывести более ранние месяцы.

Прогноз строится на основе предыдущего года. Так что данные Вам будут выдаваться строго за предыдущий год.

Спецификация метода представлена в отдельном файле в ресурсах.

Разработка модульных тестов (Unit-tests)

Для выполнения процедуры тестирования созданного вами метода библиотеки CompanyCoreLib, возвращающего упорядоченный список популярных месяцев, 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, `PopularMonths_NullList()` для тестирования случая передачи пустой коллекции дат.

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 тестирования удаления услуг Вам нужно описать пять сценариев. Удаление может быть выполнено, а может быть отклонено согласно требованиям предметной области. Необходимо, чтобы варианты тестирования демонстрировали различные исходы работы алгоритма. Для описания тестовых сценариев в ресурсах предоставлен шаблон `testing-template.docx`.

