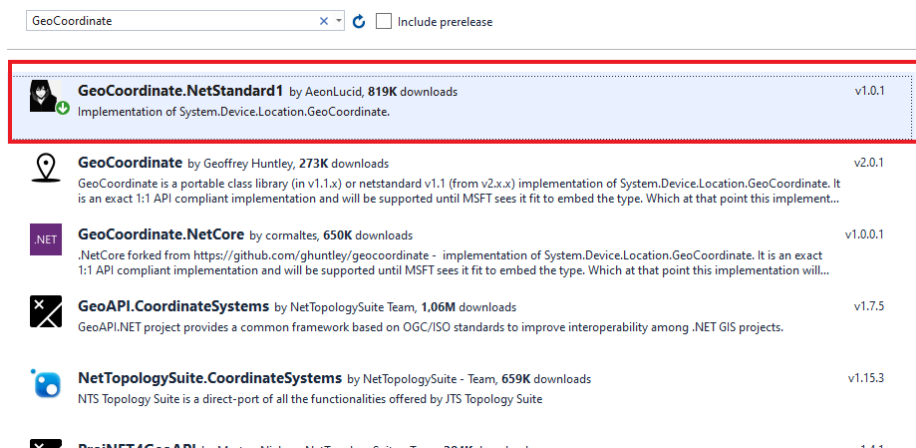


Часть 1. Объектно-ориентированная модель предметной области - 5 баллов

- Создайте новое консольное приложение .NET Core
- Добавьте в проект 2 класса со следующими исходными свойствами:
 - **Courier (курьер)** - name (имя), phone number (номер телефона), maximal acceptable weight of a parcel (максимально допустимый вес посылки, которую курьер может взять в кг), is available (bool) - признак доступности (свободен ли курьер в текущий момент времени)
 - **Order (заказ)** - number (номер), weight (вес в кг), delivery start date and time (дата и время начала доставки), delivery end date and time (дата и время окончания доставки). Дата начала и окончания доставки при создании заказа изначально не определены. Выберите соответствующим образом тип данных.
- Добавьте в проект внешнюю библиотеку **GeoCoordinate.NetStandard1** через менеджер пакетов NuGet.



После этого вы сможете использовать в программе специальный класс **GeoCoordinate** для управления геолокацией. Любая геопозиция (точка на карте) задается двумя базовыми атрибутами - широта (latitude) и долгота (longitude). В классе GeoCoordinate наряду с этими двумя свойствами есть еще ряд других (например, Altitude, Speed, Course), но в данной задаче они не нужны. Ниже показано несколько примеров использования данного класса:

```
// Coordinates of the Shabolovka HSE campus
var shabolovka = new GeoCoordinate(55.720099, 37.609212);
// Coordinates of the Pokrovka HSE campus
var pokrovka = new GeoCoordinate(55.753448, 37.648223);
// Returns the distance (in meters) between two points on the map
double distance = pokrovka.GetDistanceTo(shabolovka);
```

- Добавьте в класс Order два дополнительных свойства (две геопозиции - точка отправления заказа и точка назначения заказа). В класс Courier добавьте одно дополнительное свойство (текущая геопозиция курьера).
- Добавьте связующие свойства между двумя классами (у одного курьера много заказов в течение всего рассматриваемого периода времени, каждый заказ привязан максимум

к одному курьеру). Также возможен вариант, когда у заказа не назначен курьер (заказ только создан).

- Добавьте в проект класс репозитория для управления набором данных программы с двумя списками внутри (список курьеров и список заказов).
- Добавьте в класс репозитория следующие методы:
 1. `AddOrder` - принимает на вход три параметра (`weight` (вес заказа), `source` (откуда - геопозиция), `destination` (куда - геопозиция)), создает на их базе новый объект `Order`, назначая ему номер любым способом, но с гарантией уникальности (не должно быть двух заказов с одинаковым номером) и добавляет созданный объект в общий список заказов.
 2. `AssignCourier` - принимает в качестве параметра объект `Order` и назначает на него свободного курьера, который может взять данный заказ (вес заказа меньше или равен максимально допустимому весу, который готов взять курьер), и находящегося ближе остальных к точке отправления заказа (см. выше пример вычисления расстояния между двумя геопозициями). Одновременно с этим должны устанавливаться дата и время начала доставки (равными текущим дате и времени) и признак доступности курьера (курьеру назначен заказ, для новых заказов он временно недоступен). Если курьера, удовлетворяющего условиям выше, в списке нет, можно либо возвращать `false` из метода, либо генерировать исключение (остается усмотрение разработчика).
 3. `CompleteOrder` - принимает в качестве параметра объект `Order` и устанавливает в нем дату и время окончания доставки (равными текущим дате и времени), а также меняет признак доступности курьера, который был назначен на заказ (текущий заказ выполнен, курьер снова может назначаться на другие заказы).

Методы можно будет протестировать после создания тестового набора данных. Создавать его в коде программы (`hardcoding`) нет необходимости, можно сразу переходить к файловому вводу-выводу.

Часть 2. Файловый ввод-вывод - 3 балла

Реализуйте функции файлового ввода-вывода для набора данных программы. Вы можете использовать любые подходы, файлы любых форматов, с автоматической сериализацией или без нее. Методы чтения (загрузки) и сохранения данных следует разместить в классе репозитория. При проектировании формата файла не забудьте про то, что отдельные заказы могут быть не привязаны к курьерам.

При загрузке данных убедитесь, что все ссылки между объектами восстанавливаются необходимым образом.

Вызывайте операцию сохранения каждый раз, когда какие-либо данные в репозитории меняются.

Часть 3. Элементы абстрактного программирования - 2 балла

На текущий момент логика по назначению курьера на заказ реализована в классе репозитория. В данном задании предлагается вынести ее в отдельный класс под названием `MinimalDistanceStrategy` (стратегия назначения курьера по критерию минимального расстояния до точки отправления). Добавьте также второй схожий класс с другой стратегией назначения курьера. По этой второй стратегии на заказ назначается первый свободный курьер, готовый взять заказ (заказ проходит ограничение по весу), у которого в текущий день (используйте `DateTime.Today`) было меньше

всего заказов. Если таких курьеров несколько, берется первый по списку. Две стратегии должны быть в некоторой общей иерархии, что позволит легко переключаться между ними.

Ниже показан пример того, как после перестроения программы может выглядеть метод `AssignCourier` в классе репозитория.

```
public void AssignCourier(Order order)
{
    // Select a courier for a given order
    order.Courier = strategy.Select(order, Couriers);

    // Do other required actions ...
}
```

Отправка экзаменационного решения

По окончании времени экзамена завершите работу в среде программирования, сохранив все изменения, удалите из папки проекта временные папки `bin` и `obj`, добавьте все решение в архив формата `zip`.

Загрузите архив в Canvas в тот же раздел, откуда вы скачивали исходное задание.

Остановите сессию мониторинга `SafeExamApp`, загрузите отдельно файл лога сессии либо в Canvas в тот же раздел, что и задание, либо в облачное хранилище (в Canvas при этом необходимо оставить ссылку на файл в комментарии к загруженному проекту). Во втором случае убедитесь, что файл открыт по ссылке для всех.