Activity Tracker projektfeladat

A feladatban egy úgynevezett Activity Tracker alkalmazás szimulációját kell megvalósítanod, mellyel nyilvántarthatod sportolási tevékenységeidet, és lekérdezheted eredményeidet.

Különböző sportolási tevékenységek (activity) vannak, mint a kerékpározás, túrázás, kosárlabda, stb.

Ezek közül vannak olyanok, melyekhez tartozhat út (track), és van, amelyhez nem. Az út útpontokból (trackpoint) áll. Egy útpont tartalmaz koordinátákat és magasságot. A tevékenységeket listában kell tárolnod, és különböző műveleteket kell ezekkel végezned.

A következő típusokat kell implementálnod:

`ActivityType` enum:

Négyféle típus lehet: BIKING, HIKING, RUNNING, BASKETBALL.

`IActivity` interface:

Van egy `CalculateDistance()` metódusa, amivel az adott sport közben megtett távot tudod lekérdezni, valamint egy `TypeOfActivity` property-je, amivel a tevékenység típusát kaphatod meg.

`Coordinate` osztály:

Két read-only property-je van: a szélességi és hosszúsági fok (`Latitude`, `Longitude`), előbbi értéke a [-90,90] intervallumon, utóbbi értéke a [-180,180] intervallumon mozoghat. Dobj `ArgumentException` kivételt, ha kívül esik valamelyik érték.

`TrackPoint` osztály:

Van egy koordináta és egy emelkedés adattagja. Tartalmaz egy `CalculateDistanceFrom(TrackPoint point)` metódust, mely egy másik ponttól vett távolságot adja vissza:

```
        public double CalculateDistanceFrom(TrackPoint trackPoint)
        {
            const int radiusOfEarth = 6371;
            double thisLatitude = Coordinate.Latitude;
            double thisLongitude = Coordinate.Longitude;
            double otherLatitude = trackPoint.Coordinate.Latitude;
            double otherLongitude = trackPoint.Coordinate.Longitude;

            Double latitudeDistance = (Math.PI * (otherLatitude - thisLatitude)) / 180.0;
            Double longitudeDistance = (Math.PI * (otherLongitude - thisLongitude)) /
180.0;
            Double a = Math.Sin(latitudeDistance / 2) * Math.Sin(latitudeDistance / 2)
                    + Math.Cos((Math.PI * (thisLatitude)) / 180.0) * Math.Cos((Math.PI *
(otherLatitude)) / 180.0)
                    * Math.Sin(longitudeDistance / 2) * Math.Sin(longitudeDistance / 2);
            Double c = 2 * Math.Atan2(Math.Sqrt(a), Math.Sqrt(1 - a));
            double distance = radiusOfEarth * c * 1000;
            double height = Elevation - trackPoint.Elevation;
```

```
            distance = Math.Pow(distance, 2) + Math.Pow(height, 2);

            return Math.Round(Math.Sqrt(distance), 5);
        }
```

`Track` osztály:
Trackpointok listáját tárolja. Le lehet kérdezni

- az összemelkedést: azokat az egymást követő trackpoint-ok távolságát kell összeadni, amelyek szintemelkedést mutatnak
- az össztávolságot: azokat az egymást követő trackpoint-ok távolságát kell összeadni, amelyek szintcsökkenést mutatnak
- egy olyan koordináta objektumot, ami a track objektumban előforduló legnagyobb Latitude és legnagyobb Longitude értéket tartalmazza
- egy olyan koordináta objektumot, ami a track objektumban előforduló legkisebb Latitude és legkisebb Longitude értéket tartalmazza
- a pontokat, mint függvényt elképzelve a bennfoglaló téglalap területét. Ehhez használd fel az előző két Coordinate értékeket, mintha a téglalap egyik oldala max. latitude és min. latitude közötti távolságnyi lenne, a másik oldala pedig max. longitude és min. longitude közötti távolságnyi lenne.

`ActivityWithTrack` osztály:

Implementálja az `IActivity` interface-t. Van egy `Track` típusú adattagja. A `CalculateDistance()` metódusa a megadott track össztávolságát adja vissza, míg a `TypeOfActivity` a típust.

`ActivityWithoutTrack` osztály:

Implementálja az `IActivity` interface-t. Mivel itt nincs `track` adattag, a `CalculateDistance()` metódusa mindig nullát ad vissza. A `TypeOfActivity` property-vel a típus kérdezhető le.

`ActivitiesManager` osztály:

`IActivity`-k listáját tartalmazza. Le lehet kérdezni a track-es, illetve a track nélküli listaelemeket. Valamint típusonként külön-külön az össztávolságot egy `Report`-okat tartalmazó listában.

`Report` osztály:
Egy DTO osztály, melynek egy típus és egy távolság adattagja van.