**1 feladat konzolos része**

**1. Készítsen konzolos alkalmazást a következő feladatok megoldására, melynek**

projektjét RealEstate néven mentse el!

VsCode megnyitása:

F1 -> Create Java Project -> No Build Tools -> (project name) RealEstate

**2. Készítsen saját osztályokat Ad, Seller és Category azonosítóval, melynek az adattagjainak**

**azonosítóit és láthatósági szintjét az osztálydiagramok szemléltetik! A privát**

**adattagokat egy lakat szimbólum különbözteti meg a publikusaktól. Ékezetes**

**azonosítókat is készíthet, illetve azokat kis- és nagybetűkkel is kezdheti.**

Az App.java *main* metódusában létrehozunk egy fájl osztályt, ami a kapott excel fájl útvonalát tartalmazza.

(Érdemes a projekt gyökérkönyvtárába bemásolni a realestate.csv -t. Akkor nem kell a teljes útvonal, csak a fájl neve.)

        File file = new File("realestates.csv");

Hozzuk létre az osztályokat a kapott adatokból:

Hozzuk létre a *Category.java* fájlt.

public class Category {

    int id;

    String Name;

    public Category(int id, String name) {

        this.id = id;

        Name = name;

    }

}

Hozzuk létre a *Seller.java* fájlt:

public class Seller {

    int id;

    String Name;

    String Phone;

    public Seller(int id, String name, String phone) {

        this.id = id;

        Name = name;

        Phone = phone;

    }

}

Hozzuk létre az *Ad.java* fájlt és a konstruktorait:

public class Ad {

    int Area;

    Category Category;

    String CreateAt; // itt a leírásban DateTime van, de ahhoz egy külön osztály kellene, és azt nem írja a feladat

    String Description;

    int Floors;

    Boolean FreeOfCharge;

    int id;

    String ImageUrl;

    String LatLong;

    int Rooms;

    Seller Seller;

public Ad() {

    }

    public Ad(int id, int area, Category category, String createAt, String description, int floors,

            Boolean freeOfCharge, String imageUrl, String latLong, int rooms, Seller seller) {

        this.id = id;

        this.Area = area;

        this.Category = category;

        this.CreateAt = createAt;

        this.Description = description;

        this.Floors = floors;

        this.FreeOfCharge = freeOfCharge;

        this.ImageUrl = imageUrl;

        this.LatLong = latLong;

        this.Rooms = rooms;

        this.Seller = seller;

    }

}

**3. Készítsen statikus metódust az Ad osztályhoz az adatforráshoz történő kapcsolódásra,**

**és az adatforrásban levő adatok betöltésére! A metódus neve utaljon az adatforrásból**

**betöltésre (pl.: LoadFromCsv). A metódus visszatérési értéke az Ad osztályból képzett**

**lista legyen! A metódus paramétere a betöltendő fájl neve legyen!**

Hozzunk létre az *Ap.java* osztályban egy listát ami tárolja a csv fájl sorait.

    static ArrayList<Ad> adList = new ArrayList<Ad>();

Írjuk meg a LoadFromCSV metódust:

public static ArrayList<Ad> LoadFromCsv(File file) throws Exception {

        try (Scanner sc = new Scanner(file)) {

            int count = 0;

            int areaSum = 0;

            String line = sc.nextLine();

            while (sc.hasNextLine()) {

                Ad ad = new Ad();

                line = sc.nextLine();

                String[] values = line.split(";");

                ad.id = Integer.parseInt(values[0]);

                ad.Area = Integer.parseInt(values[4]);

                ad.Category = new Category(Integer.parseInt(values[12]), values[13]);

                ad.CreateAt = values[8];

                ad.Description = values[5];

                ad.Floors = Integer.parseInt(values[3]);

                ad.FreeOfCharge = values[6].equals("1");

                ad.ImageUrl = values[7];

                ad.LatLong = values[2];

                ad.Rooms = Integer.parseInt(values[1]);

                ad.Seller = new Seller(Integer.parseInt(values[9]), values[10], values[11]);

                adList.add(ad);

                if (ad.Floors == 0) {

                    areaSum += ad.Area;

                    count++;

                }

            }

            double avg = (double) areaSum / count;

            System.out.print("\033[H\033[2J");

            System.out.printf("1. A földszinti ingatlanok átlagos alapterülete: %.2f m2", avg);

        }

        return adList;

    }

**4. Szükség esetén az Ad osztály adattagjainak beállítását konstruktor segítségével állítsa**

**be! A paraméterben kapott adatok az adatforrásoktól függően változóak lehetnek**.

Ez megtörtént az *Ad* osztály létrehozásakor.

Az *App.java*-ban hívjuk meg a *LoadFromCsv* metódust:

        Ad.LoadFromCsv(file);

**5. Olvassa be a választott adatforrás adatait és tárolja az adatokat az Ad osztály**

**segítségével egy olyan összetett adatszerkezetben, amely használatával a további**

**feladatok megoldhatók!**

Ez itt történt meg a *LoadFromCsv* metódusban.

try (Scanner sc = new Scanner(file)) {

            String line = sc.nextLine();

            while (sc.hasNextLine()) {

                Ad ad = new Ad();

                line = sc.nextLine();

                String[] values = line.split(";");

                ad.id = Integer.parseInt(values[0]);

                ad.Area = Integer.parseInt(values[4]);

                ad.Category = new Category(Integer.parseInt(values[12]), values[13]);

                ad.CreateAt = values[8];

                ad.Description = values[5];

                ad.Floors = Integer.parseInt(values[3]);

                ad.FreeOfCharge = values[6].equals("1");

                ad.ImageUrl = values[7];

                ad.LatLong = values[2];

                ad.Rooms = Integer.parseInt(values[1]);

                ad.Seller = new Seller(Integer.parseInt(values[9]), values[10], values[11]);

                adList.add(ad);

            }

        }

        return adList;

**6. Határozza meg és írja ki a minta szerint az eladásra kínált földszinti ingatlanok átlagos**

**alapterületét! Az eredményt a minta alapján két tizedesjegy pontossággal írja ki!**

Ez is a *LoadFromCsv*-ben történik:

az átlagszámításhoz kell az hogy hány értékből számolok átlagot ez a *count* változó

és kell az összes alapterület összege ez pedig a *areaSum* változó

 int count = 0;

 int areaSum = 0;

ez a rész végzi el az átlagolást, és ki is írja

ha az emelet egyenlő nullával akkor a lakás alapterületét hozzáadjuk az összes alapterülethez

if (ad.Floors == 0) {

                    areaSum += ad.Area;

                    count++;

                }

            }

            double avg = (double) areaSum / count;

            System.out.printf("1. A földszinti ingatlanok átlagos alapterülete: %.2f m2", avg);

**7. Készítsen DistanceTo néven valós értékkel visszatérő**

**metódust, amelynek segítségével adott ingatlan esetében**

**meg lehet állapítani egy GPS koordinátától való távolságot**

**Pitagorasz-tétel segítségével!**

Írjuk meg a *DistanceTo* metódust:

Az óvoda koordinátáit fogja megkapni paraméterként, ehhez az *App.java*-ban hívjuk meg:

        Ad.DistanceTo(47.4164220114023, 19.066342425796986);

vissza az *Ad.java*-ba:

létrehozunk két változót az egyik tartalmazza azt a példányt, ami a legkisebb távolságra van az ovitól, ez a *nearest*

a *minDistance* tartalmazza a távolság kezdőértékét, azért ez: Double.POSITIVE INFINITY mert be kell állítani egy nagy értéket, hogy mindenképp találjunk kisebbet. (legkisebb érték keresés tétel)

végigmegyünk az összes példányon egy *foreach* ciklussal,

a példány koordináta értékét szétbontjuk a *split*-el,

és eltároljuk *x* és *y* változókba,

a *distance* változóba eltároljuk a kiszámolt távolságot,

és ha az kisebb és tehermentes akkor felülírjuk az eredeti-t,

végül kiíratjuk az eredményt.

public static Ad DistanceTo(double xCord, double yCord) {

        Ad nearest = null;

        double minDistance = Double.POSITIVE\_INFINITY;

        for (Ad ad : adList) {

            String[] cords = ad.LatLong.split(",");

            double x = Double.parseDouble(cords[0]);

            double y = Double.parseDouble(cords[1]);

            double distance = Math.sqrt(Math.pow(Math.abs(xCord - x), 2) + Math.pow(Math.abs(yCord - y), 2));

            if (distance < minDistance && ad.FreeOfCharge == true) {

                minDistance = distance;

                nearest = ad;

            }

        }

        System.out.println();

        System.out.println("2. A zenevár óvodához légvonalban a legközelebbi tehermentes ingatlan adatai:");

        System.out.println("\tEladó neve     : " + nearest.Seller.Name);

        System.out.println("\tEladó telefonja: " + nearest.Seller.Phone);

        System.out.println("\tAlapterület    : " + nearest.Area);

        System.out.println("\tSzobák száma   : " + nearest.Rooms);

        System.out.println();

        return nearest;

    }

}