

Programozás 1 (2018/2019 I.)

1. zárthelyi vizsga

2018. október 24.

- Hozz létre egy új projektet a saját **neptun kód**oddal elnevezve és állítsd be a python 3.6-os fordítót.
- Nyomtatott és írott segédeszköz használata megengedett, de **digitális segédeszköz használata TILOS!**
- A megoldásra **80 perc** áll rendelkezésre.
- Maximálisan **10 pont** szerezhető, feladatonként 2.

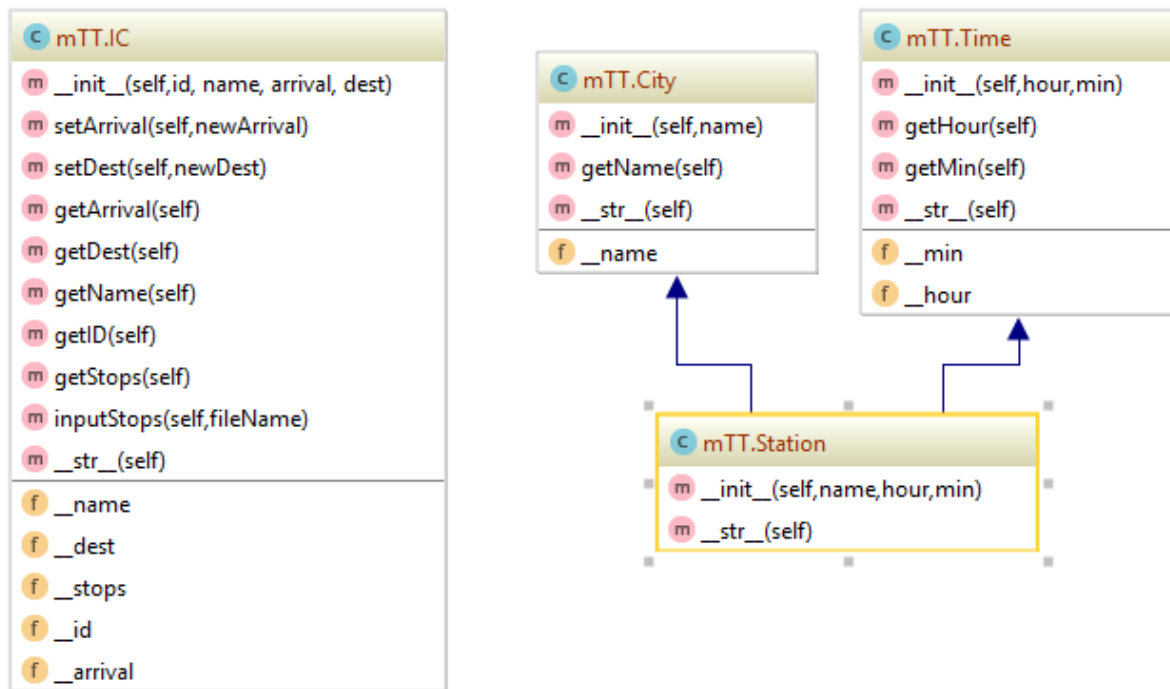
1. Feladat

InterCity vonatok által érintett állomások menetrendjét összeállító programot kell írni. Ehhez a csatolt UML diagramban szereplő osztályokat implementálja az alábbi leírások alapján:

- Készítse el a *Time*, *City* és a *Station* osztályokat a diagramban látható adattagok és metódusok alapján. A *get* metódus mindig a nevében szereplő adattag értékével tér vissza, az *__str__* metódusok pedig az osztályok adattagjait (mindet összekonkatenálva, ha több van) adják vissza sztring-ként. (2 pont)
- Írja felül a *Station* osztályba a *<, >*, *<=* és *>=* operátorokat, úgy hogy az állomásokat időpont alapján (óra, perc) rendezze, azon belül (hyper gyors vonatok esetében) pedig az állomás neve szerint ABC rendbe. (2 pont)
- Írja meg az *IC* (InterCity) osztályt a diagramban látható adattagok (név, azonosító, kiinduló város, cél város) és metódusok alapján. A *stops* listát az *inputStops* metódussal lehessen *Station* objektumokkal fájlból feltölteni (*stops1652.txt** és *stops617.txt**). Kivételkezelés szükséges! (2 pont)
- Írja meg az *IC* osztályba az *__str__* függvényt is, amely kiírja az adott vonat minden fontos paraméterét és a menetiránynak megfelelő sorrendbe az állomásokat. A fő programrészben használja a *stops1652.txt*; *stops617.txt* és az *IC.txt*-ben található adatokat és írja a képernyőre a létrehozott IC objektumok adatait (lásd *output1*). (2 pont)
- A fő programban valósítsa meg az eredeti feladatot, tehát, az összes IC vonat által érintett városnak készítse el a menetrendjét, vagyis írja ki minden városhoz, hogy mikor melyik vonat halad át rajta (lásd *output2*). Plusz egy pontért rendezze a városok menetrendjét városok szerint ABC rendbe, azon belül pedig időrendbe. (2pont + 1)

*A bemenet sorai az egyes járatok útvonaladatait írják le (nem feltétlenül sorban). A sorokban a járatszám, a kiinduló állomástól való távolság, az érintett állomás neve és az állomás elérésének az ideje (óra és perc alakban).

UML diagram:



output1:

```
Pava IC(1652) from: Budapest-Nyugati to: Nyiregyhaza:
8:23; Budapest-Nyugati
8:37; Kobanya-Kispest
8:43; Ferihegy
8:29; Zuglo
9:38; Szolnok
9:18; Cegled
10:21; Puspokladany
10:41; Hajduszoboszlo
11:02; Debrecen
11:34; Nyiregyhaza
```

```
Hajdu IC(617) from: Nyiregyhaza to: Budapest-Nyugati:
9:21; Nyiregyhaza
9:53; Debrecen
10:31; Puspokladany
10:11; Hajduszoboszlo
11:43; Cegled
11:22; Szolnok
12:21; Kobanya-Kispest
12:37; Budapest-Nyugati
12:15; Ferihegy
12:29; Zuglo
```

```
Nyirseg IC(1023) from: Tokaj to: Debrecen:
Ismeretlen információ.
```

output2:

Budapest-Nyugati:

12:37 - Hajdu IC(617)

8:23 - Pava IC(1652)

Kobanya-Kispest:

12:21 - Hajdu IC(617)

8:37 - Pava IC(1652)

Ferihegy:

12:15 - Hajdu IC(617)

8:43 - Pava IC(1652)

Zuglo:

12:29 - Hajdu IC(617)

8:29 - Pava IC(1652)

Szolnok:

11:22 - Hajdu IC(617)

9:38 - Pava IC(1652)

Cegled:

11:43 - Hajdu IC(617)

9:18 - Pava IC(1652)

Puspokladany:

10:21 - Pava IC(1652)

10:31 - Hajdu IC(617)

Hajduszoboszlo:

10:11 - Hajdu IC(617)

10:41 - Pava IC(1652)

Debrecen:

11:02 - Pava IC(1652)

9:53 - Hajdu IC(617)

Nyiregyhaza:

11:34 - Pava IC(1652)

9:21 - Hajdu IC(617)