zh-2023-06-20-gy

Határidő Nincs megadva határidő **Pont** 40 **Kérdések** 1
Elérhető 2023. jún 20, 14:45 - 2023. jún 20, 16:50 körülbelül 2 óra **Időkorlát** Nincs
Engedélyezett próbálkozások Korlátlan

Ez a kvíz már nem érhető el, mivel a kurzus befejeződött.

Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
LEGUTOLSÓ	1. próbálkozás	118 perc	0 az összesen elérhető 40 pontból *

* Néhány kérdés még nem lett értékelve

Ezen próbálkozás eredménye: **0** az összesen elérhető 40 pontból *
Beadva ekkor: 2023. jún 20, 16:46
Ez a próbálkozás ennyi időt vett igénybe: 118 perc

1. kérdés **Még nincs értékelve / 40 pont**

Programozási nyelvek Java ZH, 2023.06.20.

Feltételek

- A feladat megoldását önállóan, más segítsége nélkül kell elkészíteni.
  - Kommunikáció csak az oktatókkal megengedett.
  - Az elkészített megoldást nemcsak a ZH végéig, hanem egészen a ZH napjának végéig nem szabad megosztani mással (pl. fórumba vagy publikus verziókezelő rendszerbe felöltés).
  - A megoldás elkészítéséhez használható a Java API és a JUnit dokumentációja. Ez a Canvasből letölthető, kicsomagolható.
- Az elkészített megoldást **zip** formátumba csomagolva kell feltölteni a Canvasbe.
  - A `zip` tartalmazza a forrásfájl(oka)t. A `jar` fájlok ne kerüljenek bele.
  - A **ZH végén kb. 10 percet érdemes fenntartani** a kód tisztázására, fordíthatóvá tételére, tömörítésére, beküldésére.

Időjárás előrejelzés

Készítsd el az alábbiakban és a `ForecastTestSuite` osztály által hivatkozott **strukturális tesztelőkben** meghatározott programot, ami az elmúlt napok adatait felhasználva megjósolja a következő napi időjárást!

Milyen az idő?

Készíts `weather.data.WeatherType` felsorolási típust az alábbi konstansokkal:

- `SUNNY` (napos),
- `CLOUDY` (felhős),
- `RAINY` (esős)

Ez jelképezi egy adott nap időjárását.

Napi adatok

Készíts osztályt `weather.data.WeatherData` néven, ami egy napi mérés adatait tárolja! Legyen egy `type` nevű `WeatherType` típusú és egy `temp` nevű egész adattagja!

A két paraméteres konstruktor értelemszerűen állítsa be az adattagokat! Amennyiben a `temp` hőmérséklet érték nem esik -20 és 50 közé, váltson ki `IllegalArgumentException`-t!

Győződj meg róla, hogy a `WeatherData` osztály adattagjait példányosítás után már ne lehessen módosítani!

Az osztály szöveges reprezetációja `{type} - {temp} C` formátumú legyen, pl. `SUNNY - 23 C`.

Sorszámozott adatok

Származtass `weather.data.IndexedWeatherData` osztályt a `WeatherData` osztályból, ami a nap sorszámát jelképező egész értéket is tárolja (szintén módosíthatatlan) adattagként.

A szöveges reprezentációja kapja meg a nap sorszámát is az elejére, pl. `1. SUNNY - 23 C`.

Előrejelzés interfész

Készíts `interface`-t `weather.forecast.Forecaster` néven! Az interfész tartalmazza a `forecast()` metódust, ami változó számú `WeatherData` paraméterben kapja meg az elmúlt napok értékeit és egy új `WeatherData` objektummal tér vissza!

Egyszerű előrejelzés

A `weather.forecast.SimpleForecaster` osztály implementálja a `Forecaster` interfészt! A `forecast` metódus a következőképp nézzen ki:

- Amennyiben az esős (`RAINY`) napok száma több volt, mint a napsütéses (`SUNNY`) napok száma, a metódus jósoljon esős (`RAINY`) időt, különben pedig az utolsó nappal megegyezőt
- A jósolt hőmérsékleti érték a tömbbeli `temp` értékek átlaga legyen

Komolyabb előrejelzés

A `weather.forecast.AdvancedForecaster` osztály a `SimpleForecaster`-hez hasonlóan jósolja az időt, de reálisabb eljárással:

- a `type` értéket az utolsó nap alapján állítja elő, mindig a soron következő `enum` konstansra:
  - ha borús volt az ég (`CLOUDY`), másnap valószínűleg elered az eső (`RAINY`)
  - ha esett az eső (`RAINY`), az feltételezhetően el fog állni, így a jóslat `SUNNY`
  - ha sütött a nap (`SUNNY`), feltehetőleg be fog borulni (`CLOUDY`)
- a `temp` értéket az `átlag + (utolsó napi hőmérséklet - első napi hőmérséklet) / napok száma` képlet segítségével számolja ki

Dinamikus előrejelzés

A `weather.forecast.DynamicForecaster` osztály naponként szimulálva ad előrejelzést.

Az osztály egy listában tárolja az elmúlt napok időjárását.

Írj `recordDailyData(WeatherData)` metódust, ami hozzáadja a paraméterében kapott adatokat a listához!

Terheld túl a metódust `recordDailyData(int, WeatherType)` szignatúrával, ami egy `IndexedWeatherData` objektumot ad a listához, a paraméterek mellett a lista hosszával sorszámozva a napot.

Az osztály konstruktorában kapjon egy `Forecaster` típusú objektumot, és tárolja is el a `forecaster` nevű privát adattagjában!

A `nextDayForecast()` metódus hívja meg a tárolt objektum `forecast()` metódusát a tárolt napi adatokat tömbbé alakítva!

Funkcionális tesztelő

Készíts `weather.forecast.DynamicForecasterTest` osztályt egyetlen metódussal,

ami a `DynamicForecaster` osztály előrejelzéseit ellenőrzi.

1. Hozz létre egy `DynamicForecaster` objektumot `SimpleForecaster` előrejelzővel.
2. Adj hozzá egy napi adatot a következő `SUNNY 24 °C` értékkel
3. Tesztelendő, hogy a `nextDayForecast()` metódus 24 °C-ot jósol napos idővel.
4. Adj hozzá egy 26 °C-os, felhős napot!
5. Tesztelendő, hogy a `nextDayForecast()` metódus 25°C-ot jósol felhős idővel.
6. Adj hozzá két új, 30°C-os, de `RAINY` adatot!
7. Tesztelendő, hogy a `nextDayForecast()` metódus 27°C-ot jósol esős idővel.

zh.zip (https://canvas.elte.hu/files/2320819/download)

Kvízeredmény: **0** az összesen elérhető 40 pontból