

Developer Specification – Sorting Framework

1. Projekt célkitűzése

A jelen projekt célja egy **stratégiai tervezési mintára (Strategy Pattern)** épülő, moduláris és bővíthető keretrendszer megvalósítása, amely lehetővé teszi különböző **rendezési algoritmusok** kipróbálását, teljesítményük összehasonlítását és gyakorlati alkalmazását. A rendszer képes:

- heterogén típusú adatok (egész számok és szöveges elemek) **fájlokból történő beolvasására**,
- a felhasználó által megadott algoritmus szerinti **sorrendezésre**,
- az eredmények **fájlba történő kiírására**,
- valamint a rendezési folyamat **időbeli teljesítményének mérésére és visszajelzésére**.

Ezáltal a keretrendszer mind oktatási, mind kutatási, mind pedig ipari környezetben alkalmazható.

2. Architektúra

2.1 Tervezési minta

A rendszer a **Strategy Pattern** elvét követi:

- A `SortAlgorithm` absztrakt osztály deklarálja az algoritmusok interfészét (`sort` metódus).
- A különböző algoritmusok önálló osztályként implementálják az interfészt, biztosítva a **cserélhetőséget és a bővíthetőséget**.

2.2 Fő komponensek

1. **Rendezési algoritmusok (`SortAlgorithm`)**
 - `QuickSort` – klasszikus rekurzív gyorsrendezés
 - `MergeSort` – oszd meg és uralkodj alapú rendezés
 - `CombSort` – (jelenleg még nem implementált)
 - `BinaryInsertionSort` – bináris keresésre épülő beszúrásos rendezés
2. **`ExecutionTimer`**
 - Statikus segédosztály, amely bármely függvény végrehajtási idejét képes mérni nagy pontosságú időzítő segítségével.
3. **`FileProcessor`**
 - Fájlbeolvasás és -írás kezelése.
 - Megkülönbözteti a numerikus és szöveges adatokat.
 - Alapvető hibakezeléssel rendelkezik (pl. `FileNotFoundException`).

4. **SortingData**

- A fő alkalmazási logika aggregátora.
- Feladata az adatok beolvasása, rendezése, az eredmény kiírása, valamint a futási idő jelentése.

5. **main()**

- Parancssori interfész (argparse).
 - Felhasználói paraméterek feldolgozása.
 - Algoritmus kiválasztása és futtatása.
-

3. Működés folyamata

1. **Input kezelés**

- Az `FileProcessor.read_file()` soronként olvassa a fájl tartalmát.
- Konverziós logika: ha egy sor egész számmá alakítható, akkor `int`, egyébként `str`.

2. **Adatok rendezése**

- A `SortingData.sort_file()` a felhasználó által kiválasztott algoritmust alkalmazza.
- Az `ExecutionTimer` mérése alapján a futási idő is naplózásra kerül.

3. **Output kezelés**

- Az eredmény az `FileProcessor.write_file()` segítségével íródik fájlba.
- A kimeneti formátum megőrzi a típusjellegzetességeket (számok vagy szövegek).

4. **Felhasználói visszajelzés**

- Konzolra: algoritmus neve + futási idő.
 - Fájlba: a teljes, rendezett adatsor.
-

4. Parancssori interfész (CLI)

Példa futtatás

```
python sorter.py --input bemenet.txt --output kimenet.txt --algorithm quick_sort
```

Paraméterek

- `--input` → Bemeneti fájl neve *(kötelező)*
- `--output` → Kimeneti fájl neve *(kötelező)*
- `--algorithm` → Rendezési algoritmus kiválasztása

Érvényes opciók:

- `quick_sort`
 - `merge_sort`
 - `comb_sort`
 - `binary_insertion_sort`
-

5. Implementációs részletek

5.1 Hiányosságok és teendők

- `CombSort` implementáció hiányzik.
- `FileProcessor.write_file` → kiterjedt hibakezelés szükséges (pl. jogosultsági hibák).
- **Alkalmazás-szervezés** → célszerű `Application` osztály bevezetése, amely egységesen kezeli a futási folyamatot.
- **MergeSort optimalizálása** → `list.pop(0)` jelenlegi implementációja kvadrátikus időbonyolultságot okoz.
- **Tesztelés** → Unit tesztek és edge case-ek lefedése jelenleg hiányzik.

5.2 Kódstílus

- PEP8 konvenciók részben teljesülnek.
 - Nyelvhasználat vegyes (magyar és angol kommentek). Egységesítés szükséges.
-

6. További fejlesztési irányok

- **További algoritmusok:** HeapSort, RadixSort, TimSort.
 - **Benchmark modul** több algoritmus párhuzamos összehasonlítására.
 - **Grafikus felület (GUI) vagy webes interfész.**
 - **Logger integráció** a `print` utasítások helyett.
 - **Automatizált tesztelés** (pl. `pytest`).
-

7. Példa bemenet és kimenet

Bemenet (input.txt):

5, 3, 7, 2

Futtatás:

```
python sorter.py --input input.txt --output output.txt --algorithm merge_sort
```

Kimeneti fájl (output.txt):

2, 3, 5, 7

Konzol output:

Rendezés MergeSort-el: 0.000123 mp