

C++ parciális II - Éremtáblázat

Készítsen egy olimpiai játékok éremtáblázatát megvalósító osztályt. Az éremtáblázatot egy `map` adatstruktúrával ábrázoljuk, amelyben a kulcs a sportoló neve, az érték pedig egy olyan `vector` tároló, amely {medál, darabszám} elempárokat tartalmaz. A medál csupán három értéket vehet fel: arany, ezüst és bronz.

```
class Eremtablazat {
private:
    map<string, vector<pair<string, int>>> medalok;
public:
    Eremtablazat(const string &filename);
    void listazas() const;
    void ujMedal(const string &nev, const string &medalTipus);
    set<string> sportolok() const;
    int medalokSzama(const string &medalTipus) const;
    string legtobbMedal();
};
```

A konstruktor beolvassa a megadott fájlból az adatokat és elhelyezi a tárolóban. (2 pont)

Az `ujMedal` metódus egy új medált visz fel a tárolóba a megadott nevű sportolóhoz. Amennyiben már létezett olyan típusú medál (arany, ezüst vagy bronz), hozzáad 1-et a darabszámhoz. (1 pont)

A `listazas` metódus kilistázza a sportolókat az érmeikkel együtt alábbi formátumban, ahol a sportolókat egy-egy üres sor választja el. Ha egy sportolónak nincs például ezüst érme, akkor azt nem kell kilistázni, csak azokat az érmeiket, amelyekkel rendelkezik. (1 pont)

<sportoló neve>:

- arany: <darabszám>
- ezüst: <darabszám>
- bronz: <darabszám>

A `sportolok` metódus összegyűjti a tárolóban szereplő sportolók neveit és visszatéríti őket. (1 pont)

A `legtobbMedal` metódus visszatéríti a legtöbb medállal rendelkező sportoló nevét. Ha nincs ilyen, akkor térítsen vissza üres stringet, ha pedig több is van, elég egy sportoló nevét visszatéríteni. (1 pont)

A `medalokSzama` metódus meghatározza és visszatéríti, hogy egy adott típusú medálból összesen hány található a tárolóban. Amennyiben a paraméter üres sztring, kiszámítja, hogy összesen hány medál található a tárolóban. (2 pont)

Main függvény: az osztály minden függvényét hívja meg. (1 pont)