

- Írj *KisGepjarmu* interfészt, ami egy *haladhatItt* metódust deklarál. A metódus egy logikai értékkel térjen vissza (haladhat -e ezen az úton a gépjármű), és egy egész számot (sebességet) kérjen paraméterként.
- Készíts egy absztrakt *Jarmu* osztályt.
 - Egy járműnek legyen aktuális sebessége (int) és rendszáma. Az aktuális sebesség látszódjon a leszármazott osztályokban is (használd a lehető legszűkebb láthatóságot), míg a rendszám adattagot csak ebből az osztályból lehessen elérni. Írj konstruktort két paraméterrel, ami beállítja az adattagokat.
 - Készíts egy *gyorshajtottE* absztrakt metódust, ami egy sebességkorlátot (int) kér paraméternek, és logikai értékkel tér vissza attól függően, hogy az adott jármű gyors-hajtott -e.
 - Készíts *toString* metódust, ami az alábbi módon alakítja szöveges formára az objektumot: "rendszám - X km/h" (ahol rendszám a jármű rendszáma, X pedig az aktuális sebessége)
- Készíts egy *Robogo* osztályt, ami a *Jarmu* osztályból származik és implementálja a *KisGepjarmu* interfészt.
 - A robogónak legyen egy maximális sebesség adattagja. A robogó konstruktora a rendszámot, az aktuális sebességet és a maximális sebességet kérje el paraméterül, és ez alapján hozza létre az objektumot.
 - A *gyorshajtottE* metódus nézze meg, hogy a jármű aktuális sebessége fölötté van -e a paraméterként kapott korlátnak, és ennek megfelelően térjen vissza logikai értékkel.
 - A *haladhatItt* metódust hamis értékkel térjen vissza, ha a robogó maximális sebessége nagyobb, mint a kapott paraméter, ellenkező esetben igaz legyen a visszatérés.
 - Bővítsd ki az örökölt *toString* metódust, hogy az alábbiakat adja vissza: "Robogo: rendszám - X km/h" (ahol rendszám a jármű rendszáma, X pedig az aktuális sebessége) Használd fel az ősoosztály *toString* metódusát is!
- Készíts egy *AudiS8* osztályt, ami a *Jarmu* osztályból származik.
 - Az Audinak legyen egy lézerblokkoló (boolean) paramétere. Konstruktora a rendszámát, az aktuális sebességét kérje el, és hogy van -e lézerblokkolója, és ezek alapján hozza létre az objektumot.
 - A *gyorshajtottE* metódus nézze meg, hogy a jármű aktuális sebessége fölötté van -e a paraméterként kapott korlátnak, és ennek megfelelően térjen vissza logikai értékkel. Ha a jármű rendelkezik lézerblokkolóval, úgy ehelyett mindig hamissal térjen vissza.

- Bővítsd ki az örökölt *toString* metódust, hogy az alábbiakat adja vissza: "Audi: rendszám - X km/h" (ahol rendszám a jármű rendszáma, X pedig az aktuális sebessége). Használd fel az *Ősosztály* *toString* metódusát is!
- Készíts egy *Országut* futtatható osztályt.
 - Az osztálynak legyen egy tárolója (tetszőleges kollekció, pl. lista), amiben Jarmu típusú objektumokat tárol.
 - Legyen tovább egy statikus *jarmuvekJonnek* metódusa. Ez egy fájl elérési útját várja paraméterül. A metódus feladata, hogy a fájlból beolvasott sorokat feldolgozza, és létrehozzon belőlük Robogo és AudiS8 objektumpéldányokat, amiket hozzáad a tárolóhoz.
 - Legyen egy statikus *kiketMertunkBe* metódus is, ami végigmegy a tárolón, és kiírja egy "buntetes.txt" fájlba a benne lévő járművek szöveges formában, és az Audi típusúakra pluszban kiírja azt is, hogy gyorskijáratottak -e, míg a robogó típusúakra azt, hogy haladhatnak -e ezen az úton. Mindkét esetben 90 legyen a paraméter.
 - A main metódusban hívd meg a *jarmuvekJonnek* metódust egy parancssori argumentumból bekért elérési úttal, majd hívd meg a *kiketMertunkBe* metódust is.
 - Minden esetleges kivételt (főleg az *IOException*, de figyelj a bemenet feldolgozása közben tömbtúlindekselésre és a számok átalakítása közben fellépő hibákra is) kezelj le vagy kivétel specifikációval, vagy try blokkban!

Egy minta fájl felépítése az alábbi: (típus;rendszám;aktuális sebesség;blokkoló/max.sebesség)

```
robogo;"";40;60
audi;AAA-111;200;true
robogo;"";80;65
audi;AAA-111;130;false
```