

## TypeScript és CleanCode modul

### TypeScript

Készítse el az alábbi feladatokat TypeScriptben, ügyeljen a típusok megfelelő megadására:

#### 1. feladat

Készíts egy függvényt, ami kigenerál egy random számot, adott paraméterként megadott intervallumban (alsoHatar és felsoHatar) függvény neve legyen Rng.

**Függvény neve:** *Rng*

**Paraméterek:** *alsoHatar, felsoHatar*

**Visszatérési értéke:** *szám típusú változó*

#### 2. feladat

Készítsen egy univerzális tömb generátort, ami az előző függvény használva és 3 paraméter segítségével kigenerál egy random számokból álló tömböt!

**Függvény neve:** *TombGenerator*

**Paraméterek:** *meret, alsoHatar, felsoHatar*

**Visszatérési értéke:** *szám típusú tömb*

#### 3. feladat

Készítsen az előző generált random tömböt vizsgáló függvényt mely az előbb legenerált tömb minden elemét megduplázza!

**Függvény neve:** *Duplazo*

**Paraméterek:** *VizsgaltTomb(szám típusú tömb)*

**Visszatérési értéke:** *szám típusú tömb*

#### 4. feladat

Készítsen az előző generált random tömböt vizsgáló függvényt mely az először legenerált tömb minden elemét megvizsgálja és visszaadja hány prímszám található benne!

**Függvény neve:** *PrimekSzama*

**Paraméterek:** *VizsgaltTomb(szám típusú tömb)*

**Visszatérési értéke:** *szám típusú változó*

## 5. feladat

Készítsen az előző generált random tömböt vizsgáló függvényt mely az először legenerált tömb minden elemét megvizsgálja és visszaadja hány egyedi elem található benne!

**Függvény neve:** *EgyediElemek*

**Paraméterek:** *VizsgaltTomb(szám típusú tömb)*

**Visszatérési értéke:** *szám típusú tömb*

### UNIT Tesztelés:

Miután elkészítette a függvényeket TypeScriptben, fordítsa le az állományt az órán tanultak alapján JS kóddá, importálja az eredményeket (lefordult függvényeket) a „TsModuleNativ\_tesztfelulet.html” fájl megfelelő szegmensébe, illetve hívja meg a

- **TsModuleNativ\_teszt.js**

állományt is az ellenőrzéshez.

### Értékelés:

- 5 pont legalább 1 tesztet lefut és az eredmény jó!
- 10 pont a tesztesetek fele lefut és ezek eredménye jó! **(9+)**
- 15 pont majdnem az összes (1-2 hibás) tesztet lefut ezek eredménye jó, a függvények szerkezete is megfelelő **(16+)**
- 20 pont az összes tesztet lefut az eredményük jó, a függvények szerkezete is megfelelő.
- 25 pont az összes tesztet lefut az eredményük jó, a függvények szerkezete is megfelelő és a tiszta kód elveinek is megfelel az elkészített kód

### Feltöltés:

**A Modult egyben az összes, megoldáshoz felhasznált forrásfájl-t tartalmazva tömörítse (zip vagy rar) formátumban, és úgy töltsse fel a git felületre! ts\_modul néven, a megfelelő kiterjesztéssel!**

**A tömörített fájl tartalmazza legalább a következőket:**

- **TypeScript kódot tartalmazó ts állomány (KÖTELEZŐ)**
- **ts fájlból fordított js állomány (KÖTELEZŐ)**
- **A teszt html importálva a teszteket és a megoldott feladatokat**

**kötelező részek nélkül a modul eredménye 0 pont!**