

# Beküldendő Házi Feladatok

A félév teljesítéséhez az idei félév alábbi házi feladatait kell **futtatható<sup>1</sup> és működőképes<sup>2</sup>** változatban beküldeni. Minden feladatot külön Netbeans IDE projektként kell megoldani és ezeket egy e-mailben tömörítve csatolmányként elküldeni **április 27. 23:59-ig** az [andris@aries.ektf.hu](mailto:andris@aries.ektf.hu) címre, „hétfő 8” tárggyal.

A határidő után beérkezett megoldásokat nem fogadom el. Ha kérdés van, kérdezzetek keressetek Hangouts-on(<http://google.com/+andrasperjesi>), vagy a fenti e-mailen!

## Domino

### Működési elv:

Adott egy véletlenszerűen generált  $n$  darab különböző dominót tartalmazó tömb. A cél létrehozni egy másik olyan rendezett tömböt amiben a dominók illeszkednek egymáshoz. A dominók illesztését NEM a leoptimalisabb módon kell elvégezni, NEM CÉL hogy az összes lehetséges dominót illesszük! A rendezett tömb létrehozásakor MINDIG A RENDEZETT TÖMBBE ELHELYEZETT LEGUTOLSÓ DOMINÓHOZ LINEÁRISAN KERESÜNK a véletlenszerűen generált tömbből még nem illesztett dominót.

### A program futása:

A program a *Domino* osztály felhasználásával a következőket valósítja meg:

1. Kérjük be az illesztendő dominók számát  $n$ -et ( $n$  1 és 24 közé eső egész szám kell legyen).
2. Generáljunk  $n$  különböző dominót véletlenszerűen egy  $n$  elemű *Dominoes Domino* típusú tömbbe - a dominók (nem)egyezőségének vizsgálatára használjuk az *equals()* metódust.
3. Végezzük el a dominók illesztését a fent leírtak szerint a *SortedDominoes Domino* tömbbe.
4. Írassuk ki a *Dominoes* és a *SortedDominoes* tömb elemeit.

A feladat megoldásához javasolt a [tanári megosztásokba feltöltött DominoProjekt](#)ben található *Domino* osztályt alapul venni, de saját *Domino* osztály is írható.

## BattleShip

### Működési elv:

A feladatban egy egyszemélyes torpedójátékot kell megvalósítani. Adva van egy 10x10-es tábla. A táblán 5 darab hajó van elhelyezve, ezek 1-2-3-4-5 egység hosszú hajók. A hajót alkotó cellák egy sort vagy egy oszlopot képeznek, és egy hajónak sincs olyan cellája mely egy másik hajó cellájával azonos pozíciót venne fel.

---

1 A program indításakor a program elindul:) és ha minden utasítását végrehajtotta, szabályosan hiba nélkül fejezi be futását

2 A program az adott feladatban meghatározott azon céloknak minimális részhalmazát megvalósítja melyből még kiderül, hogy a program eredetileg az adott feladat megoldására íródott:)

## A program futása:

1. A program kirajzolja a torpedótáblát a hajókkal
2. Lövési pozíciót kér be a felhasználótól,
3. Kiírja, hogy a pozíción volt-e valamely hajónak cellája, illetve hogy az adott hajó a lövéstől elsüllyedt-e (egy hajó akkor süllyedt el ha az összes celláját eltaláltuk).
4. A program mindaddig ismétli az 1-2-3-as lépéseket amíg -1-es értéket tartalmazó pozíciót nem adunk vagy az összes hajó el nem süllyedt.

Lévéen hogy a programot közösen megírtuk, nem szükséges saját torpedó programot írni, de a [feltöltött és közösen elkészített kódot](#) ismerni és érteni kell.

## Backgammon 1-1 koronggal

### Működési elv:

Adott egy 26 mezőből álló pálya. A mezők 0-25-ig számozódnak. *Léphető mezőknek* nevezzük az 1-től 24-ig sorszámozottakat, a 0-ás vagy 25-ös mezőre akkor kerül egy korong ha leütötték. A pálya 1-s pozícióján egy fehér korong található, a 24-esen egy fekete (*startpozíciók*).

A korongok véletlenszerűen kockadobással meghatározott lépést lépnek. A fehér korong a nagyobb pozíciók irányába lép, a fekete korong a kisebb pozíciók irányába. Ha bármelyik korong olyan pozícióra lép amin a másik korong áll, akkor ez a másik korong ütésre kerül és a saját startpozíciójára lép.

Az a korong nyer, amelyiket először vesszük le a pályáról. A korongot akkor vesszük le a pályáról, hogyha az aktuális dobással kilépne a tábláról – például ha a fehér korong a 21-es mezőn áll, 4-et vagy annál nagyobbat dobva az lelép a pályáról.

### A program futása

1. A program létrehoz egy fehér és egy fekete korongot a fentiekben leírt feltételeknek megfelelően.
2. A program véletlenszerűen sorsol egy számot 1 és 6 között a fekete és a fehér korongnak.
3. A magasabb véletlen számmal rendelkező korong kezd – érték egyezés esetén, addig sorsolunk újra amíg különböző véletlen számokat nem kapunk.
4. A korongok felváltva lépnek a fent leírtak szerint, program minden lépés után kiírja a játék állását, illetve ha valamelyik korong nyert, a játék eredményét.