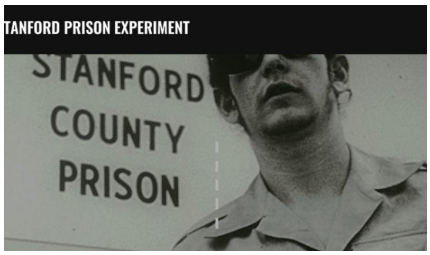
Börtönkísérlet



Szimulálja az igencsak megdöbbentő stanfordi börtönkísérletet egy Java programmal. (https://hu.wikipedia.org/wiki/Stanfordi_börtönkísérlet)

A kísérletben résztvevő személyek véletlenszerűen kapják a fogoly vagy a börtönőr szerepét.

A kísérleti **személy** a *nevével* és egy egyedi *sorszám*mal azonosítható. Minden személy számára egyformán telnek a napok. Az *ujabbNap()* metódus hatására az *eltelt napok száma* eggyel nő. Minden személynek van egy saját *napló*ja, melybe időnként bejegyzéseket kell írnia. A *naplobejegyzes()* során a naplóba beírja, hogy ma hanyadik nap van, majd kicsit eltérő módon folytatja a fogoly, ill. a börtönőr. A fogoly azt írja még be, hogy mekkora a fásultsági mértéke (pl. a beletörődés mértéke: 3), a börtönőr pedig azt, hogy mekkora az agresszivitása (pl. az agresszió mértéke 2).

A kísérlet során a **fogoly** *fasul()*, ekkor *fásultsági mérték*e eggyel növekszik. Előfordulhat, hogy időnként *lazad()*, ekkor fásultsági mértéke eggyel csökken.

A **börtönőr** viszont egyre agresszívabbá válik. Az *agresszioNovekedes()* metódus hatására az *agresszió mértéke* a metódus paraméterében megadott értékkel növekszik. Ha az agresszió mértéke elér egy, az összes börtönőrre egyformán érvényes *határ*t, akkor *gonoszkodik()*. Ha ez már igaz, akkor a naplóba az agresszió mértékét jelző szám mellé kerüljön még be a „már gonoszkodik” szöveg is.

A **napló** a *tulajdonos* személlyel azonosítható, és *bejegyzések* (string lista) vannak benne. A napló létrejöttekor azonnal kerüljön be a legelső bejegyzés, az, hogy ez kinek a naplója. A *beir()* metódus hatására kerül be a bejegyzések közé a metódus paraméterében lévő string.

…

Az alaposztályok kódrészletei (getter/setter nélkül):

```
public class Naplo {

    private Szemely tulaj;
    private List<String> bejegyzesek = new ArrayList<>() ;

    public Naplo(Szemely tulaj) {
        this.tulaj = tulaj;
        bejegyzesek.add(tulaj + " naplója");
    }

    void beir(String szoveg) {
        bejegyzesek.add(szoveg);
    }
```

```java
public class Szemely {

    private String nev;
    private int sorSzam;

    private Naplo naplo;

    private static int utolso;
    private static int elteltNapokSzama;

    public Szemely(String nev) {
        this.nev = nev;
        utolso++;
        this.sorSzam = utolso;
    }

    public static void ujabbNap(){
        elteltNapokSzama++;
    }

    public void naploBejegyzes(){
        this.naplo.beir("Ma a(z) " + elteltNapokSzama + ". nap van.");
    }

    @Override
    public String toString() {
        return sorSzam + ". " + nev ;
    }


public class BortonOr extends Szemely{

    private int agresszioMerteke;
    private static int hatar;

    public BortonOr(String nev) {
        super(nev);
    }

    public void agresszioNovekedes(int agresszio) {
        agresszioMerteke += agresszio;
    }

    public boolean gonoszkodik(){
        if(this.agresszioMerteke > hatar) return true;
        return false;
    }

    @Override
    public void naploBejegyzes() {
        super.naploBejegyzes();
        String temp = " az agresszió mértéke: " + agresszioMerteke;
        if(this.gonoszkodik()) temp += " már gonoszkodik.";
        this.getNaplo().beir(temp);
    }
```

```java
public class Fogoly extends Szemely implements Comparable<Fogoly>{

    private int fasultsagiMertek;
    private static boolean rendezve;

    public Fogoly(String nev) {
        super(nev);
    }

    public void fasul(){
        fasultsagiMertek++;
    }

    public void lazad(){
        this.fasultsagiMertek--;
    }

    @Override
    public void naploBejegyzes() {
        super.naploBejegyzes();
        this.getNaplo().beir(" a beletörődés mértéke: " + fasultsagiMertek);
    }

    public int getFasultsagiMertek() {
        return fasultsagiMertek;
    }

    @Override
    public int compareTo(Fogoly t) {
        return t.fasultsagiMertek - this.fasultsagiMertek;
    }

    @Override
    public String toString() {
        if(rendezve) return this.getNev() + " fásultsága: " + fasultsagiMertek;
        else return super.toString();
    }
```