# 8. gyakorlat

## Megírtuk az első ZH-t és beadandót

Most már tudjuk az alapokat. Most jön a képernyőn való megjelenítés, a konzolos kiíráson túl.

Jön a grafikus ablak, továbbra is JAVA-ban. Keretrendszer, objektum orientált. Egy osztálykönyvtárt felhasználva tudunk grafikusan megjeleníthető objektumokat készíteni. Ilyen a Swing.

## Swing

Félév végéig ezzel fogunk foglalkozni.

Egyablakos program készítésén keresztül ismerkedünk meg vele.

* Egyszerű Java Application kell hozzá, nincs speciális projekt típus külön erre.
* A $javax.swing$ csomagban vannak ezek a $J\*$ komponensek. Pl.: $javax.swing.JFrame$
* $new JFrame\left(​\right);$ – Még nem jelenik meg semmi. A program lefut, és egyből megáll-
* Minden tulajdonságot setter függvénnyel be kell állítani. Például a láthatóságot: $new JFrame\left(​\right).setVisible\left(true\right)$
* Ekkor egy minimális méretű névtelen ablakot kapunk a bal felső sarokban $\left(0; 0\right)$.
* Az ablak bezárása után sem áll meg a program. Ez érdekes dolog. Hol tart most a program futása?
	+ Az ablak láthatóság tulajdonsága hamisra változott a bezárással. A main függvény futása megtörtént.
	+ Több szálon futunk. A virtuális gép addig nem áll le a program, amíg van futó szála.
	+ Jelen esetben a rejtett ablak, amit bezártunk, még fut. A main függvény már nem. De a rejtett ablak szála a fontos.
	+ Ha nem állítanánk láthatóvá létrehozás után, nem futna örökké a programunk.
	+ Tehát a szál akkor indul el, mikor először láthatóvá állítjuk.
* Természetesen több ablakot is létrehozhatunk egy **for** ciklusban vagy egyszerűen egymás után.
* Eseménykezeléssel fog működni sok minden. Megnyomunk egy gombot, beírunk egy szöveget, események jönnek létre. Ezeket az eseményeket a háttérben futó szál váltja ki. Ha van eseménykezelő hozzárendelve, a program lekezeli az eseményeket. A bezárás gombra kattintás is egy ilyen esemény. Ekkor leállíthatjuk a programot.
* Sok mindent kell állítani egy-egy ablakon, ezért célszerű referenciát készíteni az ablakról. Vagy jobb: Saját osztályt készíteni, ami rendelkezik a kívánt működéssel: JFrame f **=** **new** OurFirstSwingFrame**();**
* Ebben az osztályban a $this$ kulcsszóval tudjuk elérni az ablakot. Nem is muszáj kiírni.
* Cím: setTitle**(**"Our first swing frame"**);**
* Méret: **import** java**.**awt**.**Dimension**;** majd setSize**(new** Dimension**(**600**,** 400**));**
	+ vagy setSize**(**600**,** 400**);**
* Hely: setLocation**(new** Point**(**600**,** 400**));**
* Bezáráskor lépjen ki: setDefaultCloseOperation**(**EXIT\_ON\_CLOSE**);**
	+ (A DISPOSE\_ON\_CLOSE a szemétgyűjtőnek adja.)
	+ (A HIDE\_ON\_CLOSE az alapértelmezett, elrejti.)
	+ (A DO\_NOTHING\_ON\_CLOSE nem csinál semmit.)
	+ Ha kilépés előtt szeretnénk valamit csinálni, akkor a bezárás eseményt kell kezelnünk.
* Vezérlőelemek hozzáadása: Container-hez kell hozzáadni, amit a getContentPane**();** segítségével kapunk meg.
	+ Gomb: getContentPane**().**add**(new** JButton**());**
	+ Feliratos gomb: getContentPane**().**add**(new** JButton**(**"Button"**));**
		- vagy:
		- JButton btn **=** **new** JButton**();**
		- btn**.**setText**(**"Button"**);**
		- getContentPane**().**add**(**btn**);**
	+ A gombok egymásra kerülnek, mert alapértelmezésből a teljes konténert kitölti.
		- Méret beállítása egyenként: btn**.**setSize**(new** Dimension**(**90**,** 30**));**
		- Layout beállítása: setLayout**(new** FlowLayout**());**
		- Van még GridLayout, stb. Az alapérelmezett a BorderLayout.
	+ Panel:
		- JPanel panel **=** **new** JPanel**();**
		- panel**.**setLayout**(new** GridLayout**(**3**,** 5**));**
		- getContentPane**().**add**(**panel**,** BorderLayout**.**CENTER**);**

## HF: Számológép felületét létrehozni.

Mögötte levő logikát nem.

Fontos: Közös viselkedésű elemek legyenek együtt: A számok és műveletek ciklusben legyenek elkészítve és listában tárolva.

Próbáljunk ki többféle vezérlőelemet.