

1. Hozzunk létre egy keretet (*Frame*), és ezen belül helyezzünk el egy panelt, valamint egy címkét (*Label*). Ha a felhasználó a panelre kattint az egérrel, a címkén jelenítsük meg az egérekattintás koordinátáit. A komponensek elhelyezésére használjunk egy megfelelő *LayoutManager* példányt, ne rögzítsük a pozíciókat és méreteket (ez a javaslat a legutolsó kivételével a következő feladatokra is érvényes).

Egészítsük ki a programot, olyan módon, hogy ne csak az egérekattintást figyeljük, hanem az egérmutató mozgását is. A címkére az aktuális esemény típusát is írjuk ki a koordináták mellé: amennyiben a felhasználó kattintott a „*clicked*” üzenet, amennyiben csak elmozdította a pointert a „*moved*” üzenet, amennyiben lenyomott gombbal mozdította a pointert a „*dragged*” üzenet jelenjen meg.

2. Hozzunk létre egy keretet, és ezen belül helyezzünk el egy többsoros szöveg megjelenítésére alkalmas komponenst (*TextArea*), egy egysoros szöveg bevitelére alkalmas szövegmezőt (*TextField*), valamint egy gombot. Ha a felhasználó a gombra kattint, vagy a szövegmezőn belül lenyomja az *enter* billentyűt, a szövegmező tartalmát hozzáadjuk a *TextArea* tartalmához, majd töröljük a szövegmezőből (lehetőséget adva egy új szöveg beírására).
3. Egy kereten belül egy címke szövegét változtassuk jelölőnégyzetek (*Checkbox*) segítségével: a címkén mindig az aktuálisan kijelölt jelölőnégyzeteknek megfelelő címkék szövegét jelenítsük meg. Alakítsuk át a programot, olyan módon, hogy egyszerre csak egy jelölőnégyzet legyen kiválasztható (a jelölőnégyzet komponensek „*radio button*” komponensekbe történő alakítása, a *CheckboxGroup* osztály segítségével).
4. Egy kereten belül helyezzünk el több különböző színű panelt, a színeket véletlenszerűen generálva. Ha az egérmutató belépik egy adott panel fölé, az illető panel véletlenszerűen színt vált.
5. Egy kereten belül helyezzünk el egy gombot, véletlenszerűen generált koordinátákra, a „*Push me!*” felirattal. Amikor a felhasználó megpróbál a gombra kattintani (az egérmutató a gomb fölé kerül), a gomb elmozdul (véletlenszerűen újrageneráljuk a koordinátákat, és áthelyezzük a gombot az új koordinátákra). A feladatnak elkészíthetjük egy olyan változatát is, amikor tényleg nem lehetséges a gomb lenyomása: az új koordináták generálásánál kiszűrjük annak a lehetőségét, hogy a gomb újra a mutató alá kerüljön.
6. Egy kereten belül helyezzünk el egy rajzvásznat (egy saját osztályt hozunk létre a *Canvas* osztályból származtatva), két *Choice* komponenst, egy jelölőnégyzetet (*Checkbox*), és egy gombot. A felhasználó a két *Choice* komponens segítségével kiválaszthat egy adott alakzattípust (pl. kör, négyzet stb.) és egy adott színt (pl. kék, piros stb.). A gomb lenyomásának hatására a vászonra kirajzoljuk a kiválasztott alakzatot a kiválasztott színnel. Amennyiben a jelölőnégyzet be van jelölve, az alakzat felületét is kitöltjük az illető színnel.

A programnak elkészíthetjük egy olyan változatát is, amelynek esetében nem szükséges a gomb lenyomása: bármelyik másik komponens állapotának változásakor frissítjük a rajzot. Ezen kívül a szín kiválasztására alkalmas *Choice* komponenst helyettesíthetjük olyan módon, hogy a felhasználó tetszőleges R, G, B értékeket meg tudjon határozni (pl. három szövegmező segítségével).

7. Az előbbi AWT grafikával kapcsolatos feladat (alakzatok kirajzolása) megoldása során elkészített program grafikus felületét írjuk át SWING komponensek felhasználásával. A programot egészítsük ki néhány új funkcionalitással. Egy *JSlider* segítségével legyen változtatható az alakzat mérete, olyan módon, hogy az meg is haladhassa a vászon (esetünkben *JPanel* komponens) aktuális méretét. Amennyiben az alakzat „kilóg” a vászonból, jelenjenek meg görgetősávok, amelyek segítségével változtathatjuk az éppen látható felületet (*JScrollPane* komponens alkalmazhatunk). A szín kiválasztására ezúttal egy külön grafikus felületet is biztosítsunk, a *JColorChooser* komponens felhasználásával.
8. Készítsünk egy egyszerű „autóverseny szimulátort”. Egy SWING eszköztár segítségével megvalósított grafikus felületen belül rajzoljunk ki egy versenypályát, és azon helyezzünk el néhány autót (az autókat egyszerű téglalapok segítségével is ábrázolhatjuk, de kis állományból beolvasott képeket is felhasználhatunk). Arra is lehetőséget adhatunk, hogy a felhasználó meghatározhassa az autók számát (pl. egy szövegmező segítségével). A verseny egy gomb lenyomására indul. Minden autót egy külön végrehajtási szál „irányít”, véletlenszerű időközönként változtatva azok pozícióját (a változtatás mértéke is lehet véletlenszerű). A verseny addig tart, ameddig az első autó beér a célvonalba. Készítsünk egy olyan változatot is, amelynek esetében minden autó beér a célba, majd egy párbeszédablak jelenik meg a beérkezés sorrendjével.
9. Készítsünk egy grafikus felülettel rendelkező egyszerű számológépet.

Opcionális: az elkészült számológépet kiterjesztve, hozzunk létre egy „Scientific Calculator” programot, amely az alapl műveleteken kívül további műveletek elvégzését is támogatja.

10. Készítsünk egy egyszerű rajzolóprogramot: egy kereten belül helyezzünk el egy rajzolásra alkalmas felületet és (minimum) két gombot/kapcsolót, amelyek különböző rajzolási műveletek elvégzését támogatják. Ha az első kapcsoló van lenyomva, akkor a felhasználó az egér segítségével tetszőleges görbe vonalakat rajzolhat a felületre: ha lenyomott gombbal mozdtja az egeret, kirajzolódik az egér pointer „útvonala”. A második kapcsoló szakaszok kirajzolására ad lehetőséget: ha a felhasználó kattint a felületre, elmentődnek a kattintás koordinátái, majd ha újra kattint akkor a pointer aktuális pozíciója és az előző kattintás pozíciója között kirajzolódik egy szakasz.

Opcionális: egészítsük ki a felületet további alakzatok kirajzolására lehetőséget adó kapcsolókkal, illetve építsünk be a programba törlési lehetőséget.