

TUDNIVALÓK:

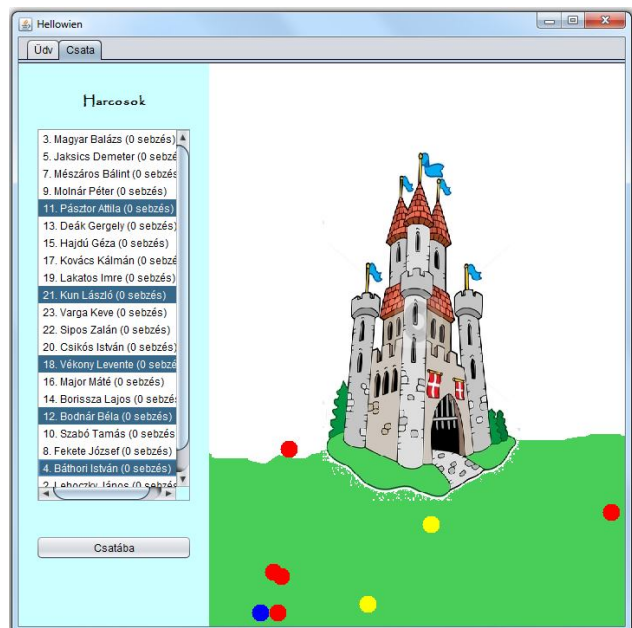
Most is és a következő gyakorlatokon is – akkor is, ha külön nem emeljük ki – az órán meg nem oldott feladatok HÁZI FELADAT-ként megoldandóak!!!

Ez fontos a tárgy sikeres teljesítéséhez!



1. feladat:

Egyre trendibb megünnepelni a Hellowien-t, nézzük meg, hogy mit is ünneplünk ekkor, és írjunk hozzá egy programot. A program induló felülete:



Kétségtelen, hogy a dátum egy kicsit hibás, de ekkora távlatból egy kis kerekítési hiba elmegy, a lényeg akkor is az, hogy „S nyögte Mátyás bús hadát Bécsnek büszke vára.”

Szóval, az első tabulátor-felületen látható az üdvözlő oldal, a másodikon a csatát „szimuláljuk”. Most csak egy-egy pötty jelzi a harcosokat, de ezt később lehet módosítani.

Ez a felület további két részből áll. A baloldalon a harcosok névsora látható, a jobboldalon a csatatér. Méretek: A teljes panel-felület: 700*650, a harcosok listáját tartalmazó rész 220 pixel szélességű.

A harcosok adatai egy Derby adatbázisba kerülnek.

A tábla neve: KATONAK.

Az adatbázisból létrehozott példányok egy rendezhető lista-modellbe kerülnek – a rendezési szempont a katonák sebzési értéke. Mint a fenti ábrán látható, mindegyik katonának van egy egyedi sorszámja is.

A rangtól függően más-más színű, de egyforma méretű pöttyök jellemzik őket. (A képen látható megoldásban a lovas parancsnok kék, a lovas piros, a gyalogos parancsnok zöld, a gyalogos sárga, a zászlós pink, bárki más fekete, de bármilyen más színt is kitalálhat.) Később esetleg sok minden másban is eltérhetnek egymástól.

id	nev	rang
1	Kinizsi Pál	lovas parancsnok
2	Lehoczky János	gyalogos parancsnok
3	Magyar Balázs	gyalogos parancsnok
4	Báthori István	lovas parancsnok
5	Jaksics Demeter	lovas parancsnok
6	Both János	lovas parancsnok
7	Mészáros Bálint	lovas
8	Fekete József	lovas
9	Molnár Péter	gyalogos
10	Szabó Tamás	gyalogos
11	Pásztor Attila	lovas
12	Bodnár Béla	lovas
13	Deák Gergely	gyalogos
14	Borissza Lajos	gyalogos
15	Hajdú Géza	lovas
16	Major Máté	gyalogos
17	Kovács Kálmán	gyalogos
18	Vékony Levente	gyalogos
19	Lakatos Imre	gyalogos
20	Csikós István	lovas
21	Kun László	zászlós
22	Sipos Zalán	zászlós
23	Varga Keve	zászlós

A katonákat a nevük, egyedi sorszámuk definiálja. Mindegyik lőtt, és mindegyiket meglőtték (persze, ha szerencséje van, akkor nem, de elvileg mindkettő lehet). Ha lőtt, akkor a sebzés értéke eggyel nő, ha meglőtték, akkor eggyel csökken, és ha negatívvá válik, akkor szegény katona meghal. Csak élő katona lőtt, ill. őt lehet meglőni.

A „Csatába” feliratú gomb hatására a listából kijelölt katonák harcba indulnak. Ez azt jelenti, hogy a csatatéren véletlen helyeken megjelennek az őket jellemző pöttyök. (Mivel akkor még nem volt légierő, ezért lehetőleg ne a levegőben jelenjenek meg, hanem – a háttérképtől függő magasságig, mondjuk, a panel fele (vagy valahány százalékanak) magasságáig.)

A gomb megnyomásakor szűnjön meg a kijelölés. (A gomb újbóli megnyomásakor a régi harcosok eltűnnek, újak kerülnek a csatatérre.)

Az a katona lőtt (vagy lövik meg), akire az egerrel rákattintunk. Ekkor valahány százalék eséllyel meglövik vagy ő lö. (Ezt a véletlen dönti el.)

Természetesen ennek megfelelően állandóan változik majd a listában látható sorrend, illetve ha egy katona meghal, akkor a listából is, csatatérről is töröljük.

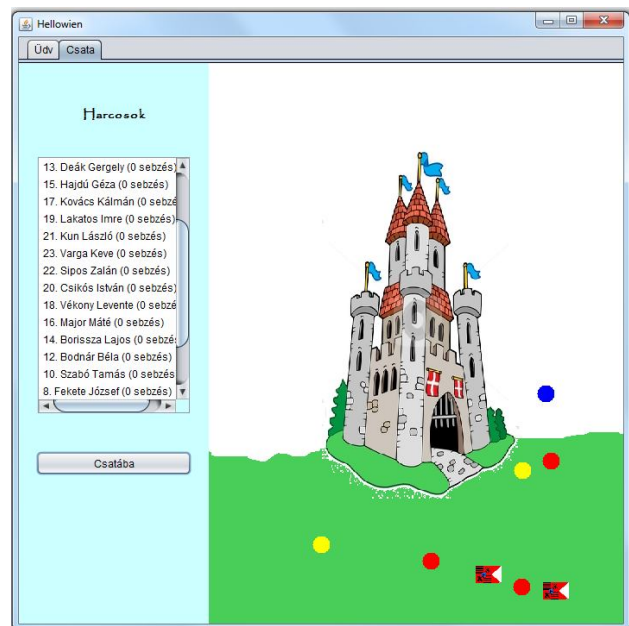
A megoldás során ezeket a fogalmakat (is) használjuk: adatbázisból való olvasás, singleton tervezési minta, gyártó függvény, rajzolás, leegyszerűsített MVC (ill. MVP) modell.

Továbbfejlesztések:

a) A zászlósok helyét egy-egy zászló jelezze.

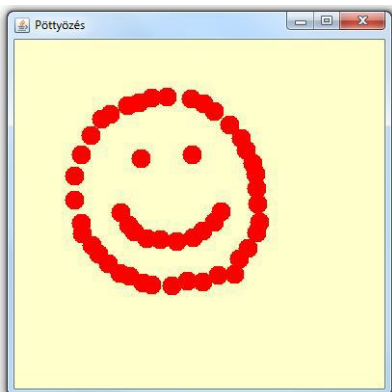
b) A többi katonához is rendelhet egy-egy kis képet, persze, a rangjuknak megfelelőt.

c) Indítsa be a fantáziáját, és még mindenféle egyebeket találjon ki: miben különbözhetnek még egymástól az egyes katonatípusok, mikor győztes a csata, stb.



PÖTTYÖS VARIÁCIÓK

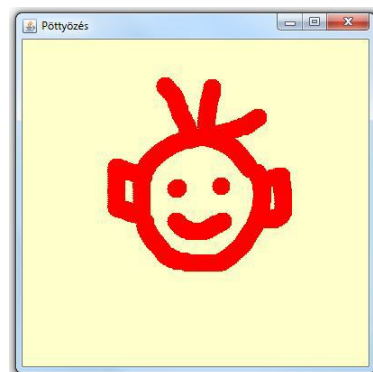
1. feladat:



tartjuk az egérgombot.

a/ Hozzunk létre egy 400x400-as felületű Swing alkalmazást, amelyben a felületre kattintva a kattintás helye, mint középpont körül megjelenik egy-egy kis piros pötty. Ezek a pöttyök az ablak bezárásáig „élnek” (azaz átméretezskor, alkalmazásváltáskor, stb. is.) – Ennél kevésbé béna rajzot is készíthet. ☺

b/ Talán könnyebb lenne rajzolni, ha nem kellene állandóan kattintgatni, hanem az egér mozgatásával tudnánk rajzolni, de csak akkor, ha nyomva



c/ Most ne rajzoljunk, hanem egérekattintásra csak egyetlen véletlen színű pötty jelenjen meg, az egér mozgatásával pedig lehessen arrébb vinni ezt a pöttyöt.

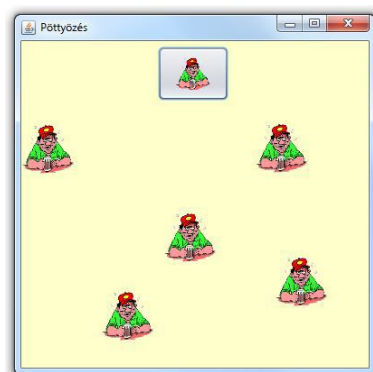


d/ Bár nagy eséllyel Ön szebben rajzol nálam, de azért adjunk lehetőséget a törlésre is, azaz rakjunk fel egy gombot, melynek hatására ki tudjuk törölni az összes pöttyöt.

e/ (HF) Jaj, már megint kocsma! Pöttyök helyett most kattintásra sörözgető figurák jelenjenek meg.

(A kép megtalálható a feladatok/gyak7_segedlet mappában.)

Segítség: Ld. a segédletek mappában lévő keprajzolas.txt fájlt.

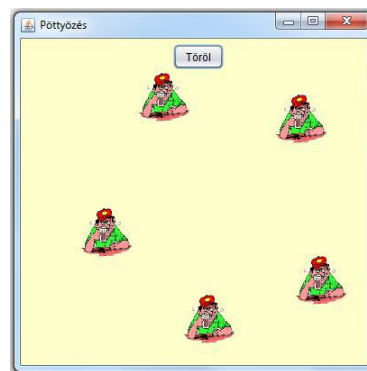


f/(HF játékos kedvéeknek)

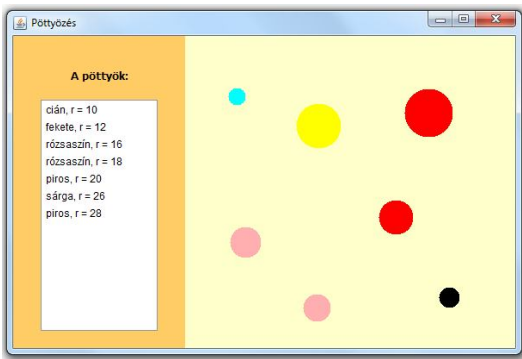
A törlés gombon is a söröző figura jelenjen meg a felirat helyett, és persze, a gombra kattintva tűnjenek el a többiek.

g/ Készítsen appletet valamelyik előző feladtból.

Segítség: A JFrame helyett JApplet form-ot hozzon létre, és erre húzza rá a JPanel-t



2. feladat



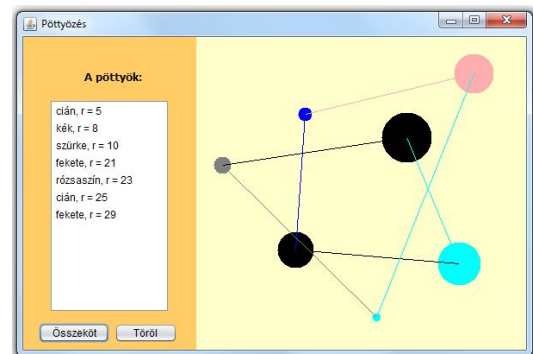
a/ A 600x400-as felület most két részre oszlik: a jobboldali 400x400-as felület hasonló szerepet játszik, mint az előző feladatban, vagyis erre kattintva a kattintás, mint középpont körül megjelenik egy véletlen sugarú és véletlen színű pötty, a baloldali listában pedig láthatjuk a pöttyök színét és sugarának nagyságát, mégpedig a sugár szerint növekvő sorrendben. Egy-egy újabb pötty létrehozásakor a pöttyről szóló leírás azonnal jó helyre kerüljön a listában.

Segítség:

1. Ha el is akarjuk nevezni a véletlen színeket, akkor nyilván nem az RGB színskálát kell használnunk, hanem a Color osztály fix színeit, vagy azok közül néhányat. A színeket és az elnevezéseket együtt kell kezelnünk, ezért hozunk létre erre is egy osztályt vagy enumot. (Az adatfeltöltéshez használhatja a gyakorlatok/gyak7/segitseget.txt fájlban lévő adatokat.)

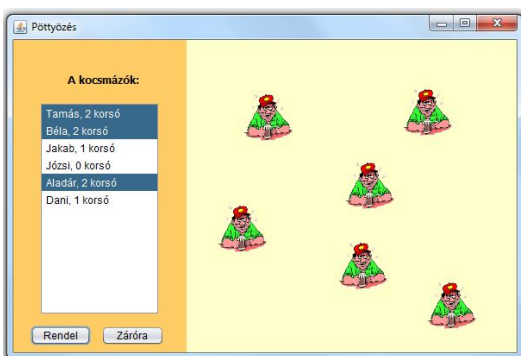
2. Használja a RendezhetőListModel-t (segedletek mappa).

b/ A baloldali felületre rakjunk fel egy „Töröl” és egy „Összeköt” feliratú gombot. A törlés hatására töröljük ki a listából is és a rajzfelületről is az összes pöttyöt. Az összekötés hatására pedig sorban kössük össze őket, mégpedig úgy, hogy a vonal mindig olyan színű legyen, mint az a pötty, ahonnan indul. Az utolsó pöttyöt kössük össze az elsóvel. Törléskor természetesen a vonalak is törlődjenek.



Segítség: A RendezhetőListModel-ben meg kell írni a clear() metódust. (Mivel később csak egyes elemeket is akarunk törölni, ezért célszerű megírni az elem törlésére vonatkozó metódust is, ld. segitseget.txt.

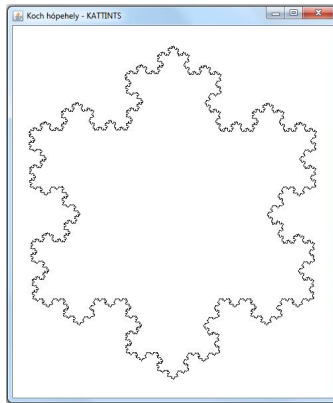
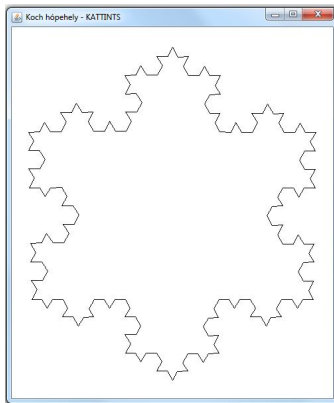
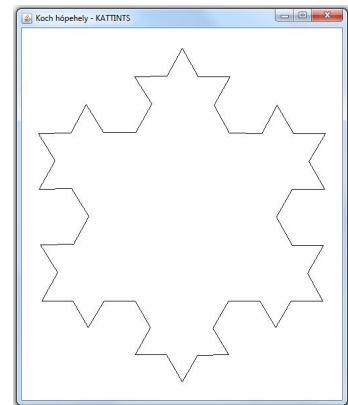
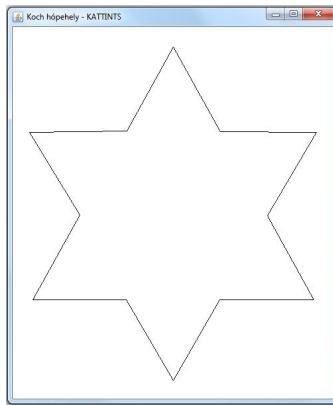
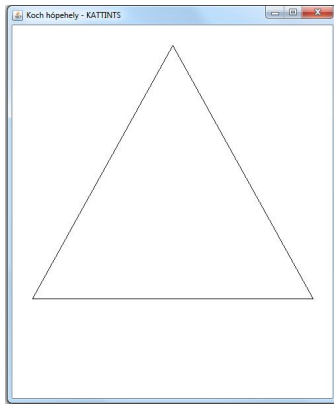
c/ Oldja meg, hogy ne az összes, hanem csak bizonyos pöttyöket lehessen törölni. Vagyis: a pöttyök felíratra (label) kattintva törölje ki a listában kijelölt pöttyöket (egyszerre többet is lehet) a listából is és a rajzról is. Figyeljen rá, hogy ha be volt kapcsolva az összekötés, akkor továbbra is csak a létező pöttyök között legyen vonal.



d/ Ismét jöjjenek a kocsmázók. A jobboldali felületre kattintva jelenjen meg egy sörivő, a neve, és az általa fogyasztott korsók száma pedig kerüljön a jobboldali listába (induláskor nulla korsó ☺). A kocsmázó nevét véletlenszerűen (de egyedi módon) válassza ki egy adott névsorból. Ha elfogytak a nevek, akkor adjon üzenetet, hogy már nincs hely a kocsmában.

A „Rendel” feliratú gombra kattintva a kijelölt emberek rendelnek egy újabb korsóval (a fogyasztott korsójuk száma növekszik eggyel). A kocsmázók felíratra kattintva a kijelöltek menjenek haza (ekkor nevük ismét „felszabadul”), a „Záróra” feliratú gombra kattintva pedig töröljünk mindenkit.

3. feladat



Az itt látható méretarány:

Frame: 500*600

Felső, bal-, jobboldaltól
való távolság: 30.

Alsó távolság: 150

Készítsen alkalmazást a Koch-hópehely (<http://hu.wikipedia.org/wiki/Koch-görbe>) alakulásának bemutatására. (Az frame legyen átméretezhető, változáskor az ábra is igazodjon a kerethez) Induláskor a felület üres, kattintásra jelenik meg a következő szint.

Az alapelv: kiindulunk egy szabályos háromszögből (az ábrán lévő nem szabályos, nyilván bármilyen háromszögből kiindulhatunk, csak nem lesz szabályos a kapott eredmény). Ez a 0. szint.

A következő lépésben a háromszög minden oldalát három részre bontjuk, és a középső rész fölé egy egyharmad oldalhosszúságú szabályos háromszöget rajzolunk úgy, hogy kimarad az eredeti vonalon lévő szakasz. Ez az 1. szint

A továbbiakban az előző szakaszban leírtakat ismételjük az összes vonalra. (Túl sokáig nem érdemes, sőt, nem is nagyon lehet, mert nagyon lelassul, és nagyon eszi a memóriát.)

Segítség:

A feladatot rekurzióval érdemes megoldani. Egy adott szintre fogalmazzuk meg a metódust, mégpedig úgy, hogy azon a szinten négy „vonalat” rajzoljon (az első szinthez tartozó ábrán a háromszög egy-egy oldala „fölköt” látható, hogy mi ez a négy vonal). Ténylegesen azonban nem sima vonalat, hanem a vonalrajzolás helyett ismét meghívjuk a metódust, de eggyel kisebb szinten.

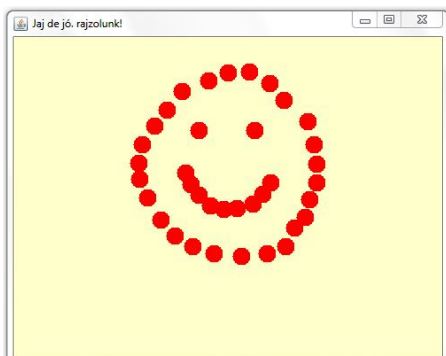
Ha a sarokpontok koordinátái sorban (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) , akkor a metódust meg kell hívunk az $((x_1, y_1), (x_2, y_2))$; $((x_2, y_2), (x_3, y_3))$; $((x_3, y_3), (x_4, y_4))$ és $((x_4, y_4), (x_5, y_5))$ szakaszokra.

Az (x_3, y_3) pont meghatározása igényel egy kis (középszintű) matek tudást. Akinek van ideje, ellenőrizze az interneten talált képletet:

$$x_3 = (\text{int}) (0.5 * (x_1 + x_5) + \text{Math.sqrt}(3) * (y_1 - y_5) / 6);$$
$$y_3 = (\text{int}) (0.5 * (y_1 + y_5) + \text{Math.sqrt}(3) * (x_5 - x_1) / 6);$$

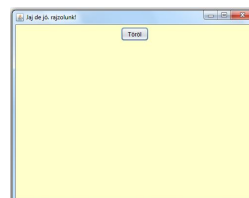
TOVÁBBI FELADATOK

1. feladat

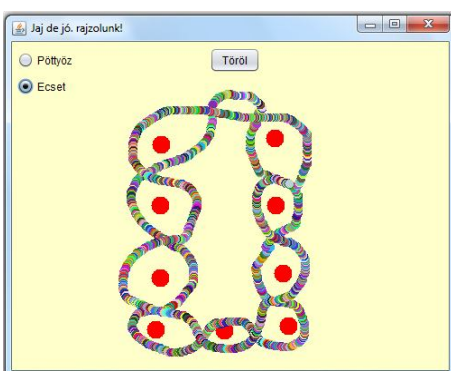
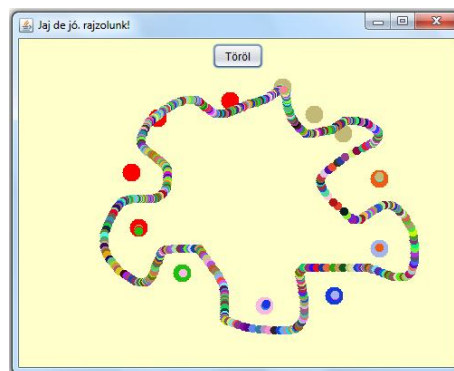


a/ Készítsen egy 500x400-as felületű Swing alkalmazást, amelyben a felületre kattintva a kattintás helye, mint középpont körül megjelenik egy-egy kis piros pötty. Ezek a pöttyök az ablak bezárásáig „élnek” (azaz átméretezéskor, alkalmazásváltáskor, stb. is.)

b/ Jó lenne szebb rajzot készíteni, de mi legyen a mostanival? Azonnal rakjunk fel egy törlés gombot! ☺
A gomb hatására tűnjön el az összes pötty!



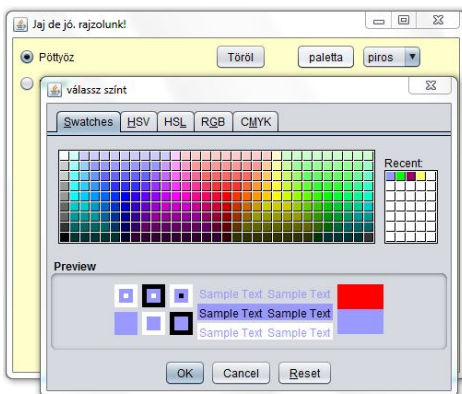
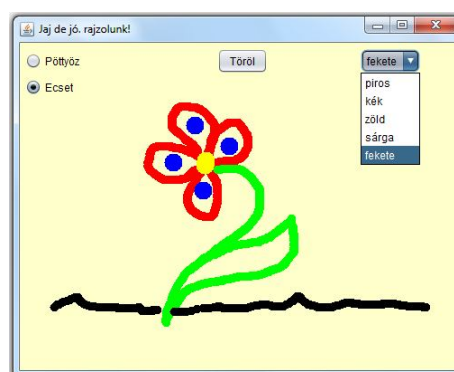
c/ Talán szebb rajzot tudnánk készíteni, ha nem kattintásra, hanem az egér mozgatására rajzolnánk. Hát, ha nem is szebb, de színesebb lett. Oldjuk meg, hogy kattintásra továbbra is egy-egy piros pöttyöt rajzoljon, egérmozgatásra pedig fele akkora sugarú véletlen színekből álló „ecsetnyomot”, vagyis szintén pöttyöket. Ecsetnyomot csak akkor húzzon, ha nyomva van az egér gombja, felemelt állapotban ne csináljon semmit.



d/ Néha bizony, egy ügyetlen kézrandulásra nehéz megkülönböztetni a kattintást az egérmozgatástól. Ezért jó lenne határozottabban megkülönböztetni a két eseményt. Rakjunk fel két rádiógombot, és a választástól függően vagy pöttyözzünk, vagy „ecsettel” rajzoljunk.

milyen színnel szeretnénk rajzolni. Ezért tegyünk fel a felületre egy combobox-ot, amely segítségével ki tudunk választani néhány színt.

e/ Látványos a véletlen szín, de mégis jobb lenne, ha mi választhatnánk meg, mit



f/ De miért csak néhány színt, amikor használhatnánk a teljes palettát? Használjuk is!

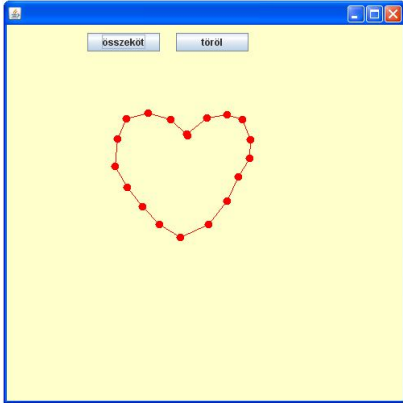
g/ De mielőtt bármit is csinálnánk, mentjük el ezt a „szép” virágot úgy, hogy legközelebb akár folytathassuk is.

Segítség: Szerializálni kell a Potty osztályt, és objektumként kiírni (beolvasni) a pottyok listát.

2. feladat

A feladat eleje ugyanolyan, mint az előző, azaz:

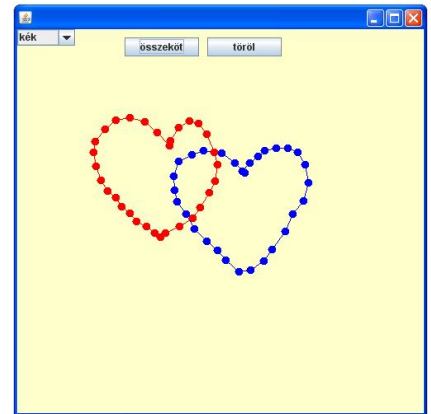
a/ Készítsen egy 500x500-as felületű Swing alkalmazást, amelyben a felületre kattintva a kattintás helye, mint középpont körül megjelenik egy-egy kis piros pötty. Ezek a pöttyök az ablak bezárásáig „élnek” (azaz átméretezéskor, alkalmazásváltáskor, stb. is.)



b/ A felület tetejére rakjon fel két gombot. Az egyik gomb megnyomásakor a kirajzolt pöttyöket kösse össze egy vonallal, a másik megnyomásakor törölje az egész rajzot, és tegye lehetővé egy új rajz készítését. (Figyeljen rá, hogy az összekötés után rajzolt pöttyök addig ne legyenek összekötve, amíg ismét az „összeköt” gombra nem kattint.)

c/ Rakjon fel egy újabb gombot, amelyre kattintva csak az összekötő vonal törlődik.

d/ Bővítse úgy a feladatot, hogy helyezzen fel a felületre egy combo-box-ot, amelyből ki tudja választani a rajzolandó pöttyök színét. Az esetleges összekötő vonal mindig legyen olyan színű, amilyen a pötty. Próbálja meg úgy megoldani, hogy két különböző színű pont között ne legyen összekötés.



e/ A színválasztó combo tartalmazza az összes alapszínt!

Segítség: Természetesen NE kézzel töltse fel, és kiválasztáskor se switch-case-t használjon.

Használja a DefaultComboModel<> osztályt, a színeket meg vagy enumként kezelje, vagy írjon egy Szin osztályt.

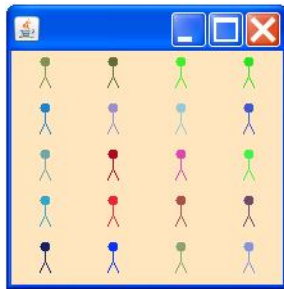
Töltse fel a combo-t a színek lista elemeivel, a kiválasztást pedig index alapján végezze.

f/ Még egyszer, önállóan is kipróbálhatja a színpalettát, és feltétlenül próbálja ki a fájlba-mentést, visszaolvasást.

g/ Növelje a rádió-gombok számát, és adjon rá lehetőséget, hogy az eddigiek mellett négyzet alakú ecsettel is rajzolhassunk.

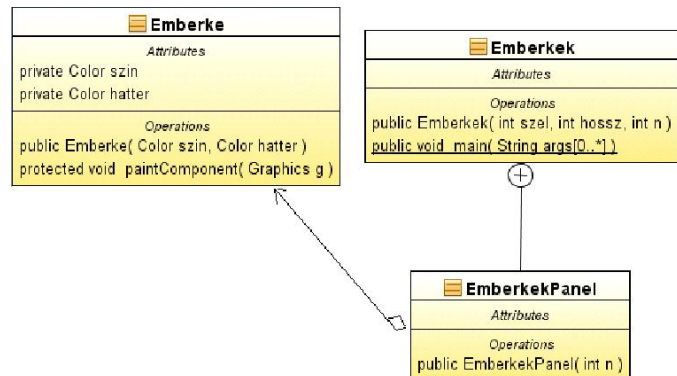
h/ Legyen radír funkció is. (háttérszínnel való festés).

3. feladat



a/ Rajzoljon pálcika emberkéket egy Swing felületre!

Segítség:



Az emberke egy JPanel komponensre van rajzolva, és egy másik panel GridLayout() elrendezésmenedzserének segítségével 5 sorba felrajzolva.

Csak hogy reagáljak az előadáson felvetett problémára: természetesen szabad kezét is rajzolni. ☺

b/ Egy emberkére kattintva, annak változzon meg a színe.

c/ Egy emberkére kattintva, töröljük őt a helyéről.

4. feladat

Az előző feladatok közül valamelyiket alakítsa appletté.

5. feladat

Készítsen egy 500x500-as alkalmazást, amelyen az egérmozgás hatására virágok „nyílnak”.

Képeket és az alapötletet innen nyerheti:

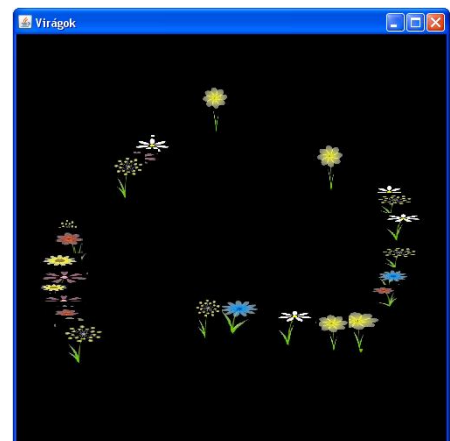
http://www.procreo.jp/labo/flower_garden.swf

(egérgombot nyomva tartva mozgassa az egeret ☺)

A képek kezeléséhez van egy kis leírás a segédlet könyvtárban.

Készítse el a feladat applet változatát is.

Képeket találhat a gyak7_segédlet mappában is.



6. feladat

Használja a fantáziáját, és készítsen egy grafikus appletet vagy alkalmazást. A grafika valamilyen esemény hatására jelenjen meg.

Tanulmányozza a HELP ide vonatkozó részeit, és a programban használjon fel minél több grafikus funkciót (színek, hátterek, különböző események...stb)