

Java alacsony szintű grafika labor

Készítette: Goldschmidt Balázs, BME IIT, 2015.

A feladatok megoldásához felhasználandó osztályok leírásait az alábbi URL-en találja meg:

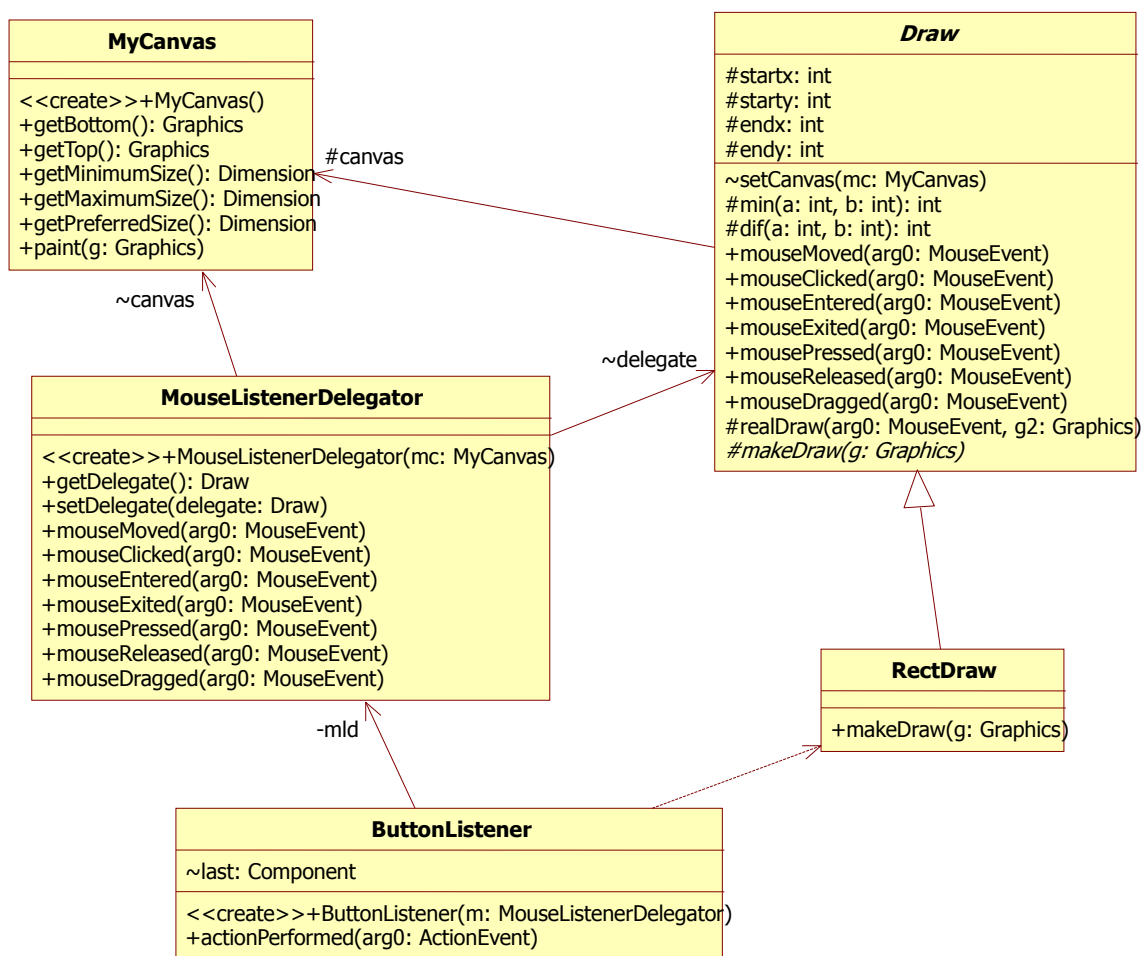
<http://docs.oracle.com/javase/8/docs/api/>

<http://docs.oracle.com/javase/8/docs/api/java/awt/Graphics.html>

i. Alapprogram

Készült egy egyszerű Swing-es alkalmazás, ami kezdetleges rajzoló funkciót valósít meg.

Az egyes osztályok kapcsolatait az alábbi UML osztálydiagram szemlélteti:



A program vázát a **Main** osztály alkotja (az ábrán nincs feltüntetve), ez hoz létre egy *JFrame*-et, helyezi el a widgeteket és regisztrálja a listenereket. Az alapprogramban egy listener van, amelyik a rajzolando alakzat váltását vezényli, a **ButtonListener**. Ez gombnyomás hatására lecseréli az aktuális rajzoló rutint egy másikra. A rutinok a **Draw** osztály leszármazottai, a jelenleg elérhető egyetlen ilyen a **RectDraw**, ami téglalapot tud rajzolni. A *Draw* valósítja meg az egérrel kapcsolatos események feldolgozását, pl. egérgomb megnyomása, lenyomott egérgommbal mozgatás, stb.

A rajzolás egy **MyCanvas** nevű komponensen történik, alacsonyszintű grafikai rutinok hívásával. Egy *MyCanvas* objektumok három kép áll a rendelkezésére: *image*, *bottom*, *top*. Az *image*-et arra használjuk, hogy az éppen megjelenítendő képet előállítsuk, majd egy *drawImage* metódussal a képernyőn megjelenítsük. A *bottom* tartalmazza a már megrajzolt képet, a *top* pedig egy efelett lévő átlátszó réteg, amire az átmeneti ábrákat tehetjük. A *Draw* absztrakt osztály a leszármazottaiban megvalósított *makeDraw* metódust hívja meg a konkrét új ábra létrehozásához, de attól függően, hogy nyomva tartjuk-e az egérgombot vagy már elengedtük, a *MyCanvas* objektum *top* illetve *bottom* rétegére teszi az elkészült ábrát.

A **MouseListenerDelegator** feladata az, hogy kényelmesen tudjuk az egyes rajzoló rutinokat cserélni: a *MyCanvas*-ben egy *MouseListenerDelegator*-t regisztrálunk, és ez utóbbiban tudjuk kényelmesen cserélgetni a *Draw* leszármazottakat.

Az osztályok dokumentációja és forráskódja a labor weboldalán elérhető.

Feladat: tölts le az alapprogram eclipse csomagját, importálja az eclipse-be, és próbálja ki!

ii. Vonalrajzolás

Készítsen a *RectDraw* osztály forráskódja alapján **LineDraw** osztályt, amivel vonalat lehet rajzolni. A vonal a téglalaphoz hasonlóan csak akkor véglegesítődjön, ha elengedtük az egérgombot. A vonal kezdete a lenyomáskor eltárolt érték (*startx*, *starty*), vége az éppen aktuális kurzorpozíció (*endx*, *endy*).

Az osztály egy példányát a *ButtonListener actionPerformed* metódusában kell a megfelelő esemény hatására beállítani (*MouseListenerDelegator setDelegate* metódusa).

iii. Körrajzolás

Készítsen a *RectDraw* osztály forráskódja alapján **CircleDraw** osztályt, amivel kört lehet rajzolni. A kör a téglalaphoz hasonlóan csak akkor véglegesítődjön, ha elengedtük az egérgombot. A kör középpontja a lenyomáskor eltárolt érték (*startx*, *starty*), a kör pereme mindig legyen az aktuális kurzorpozíción (*endx*, *endy*).

Az osztály egy példányát a *ButtonListener actionPerformed* metódusában kell a megfelelő esemény hatására beállítani (*MouseListenerDelegator setDelegate* metódusa).

iv. Színváltás

Készítsen színváltást lehetővé tevő komponenst (**ColorChanger**). Ez valósítsa meg az *ActionListener* interfészt, és legyen a "color" nevű *JButton*-hoz regisztrálva. Ha a gombot megnyomják, jelenjen meg egy *JColorChooser* ablak (*JColorChooser.showDialog* metódus), és a kiválasztott szín legyen az új rajzoló szín. Ehhez természetesen a rajzoló színt is el kell tárolni, célszerű ezt a *Draw* osztály statikus attribútumaként megtenni, hogy a *ColorChanger* kényelmesen hozzáférhessen.

v. Görberajzolás

A fentiek alapján írjon görberajzoló komponenst (**CurveDraw**). Ez a fentiekkel ellentétben folyamatosan rajzoljon. Ehhez közvetlenül a *bottom*-ra kell tenni az ábrát, vagyis a *makeDraw* mellett felül kell definiálni a *Draw mouseDragged* metódusát is.

Az osztály egy példányát a *ButtonListener actionPerformed* metódusában kell a megfelelő esemény hatására beállítani (*MouseListenerDelegator setDelegate* metódusa).