KissDraw

Rajzoló program
(Keep It Simple and Stupid)

Programfejlesztési dokumentáció

Angster Erzsébet
Készült: 2003. április 1.
Utolsó módosítás: 2005. február 5.

Tartalomjegyzék

1. Célkitűzés

Készítendő egy rajzolóprogram, mely segítségével különböző színű síkidomok (egyenes, téglalap, ellipszis) és szabadkézi rajzok tehetők egy "rajzlapra". A rajzlap háttérszíne megváltoztatható. Az egyes síkidomok legyenek kijelölhetők és törölhetők (egyszerre több is). Legyen meg a lehetőség a rácsra igazításra, valamint a korlátlan visszavonásra, és annak visszavonására is. A rajzlapokat lehessen lemezre menteni és az elmentetteket betölteni. Óvjuk a használót attól, hogy könnyen elveszítse rajzait. Az esetleges adatvesztés előtt (kilépés, másik rajz betöltése stb.) mindig meg kell kérdezni a használót, nem akarja-e menteni a módosított rajzot.

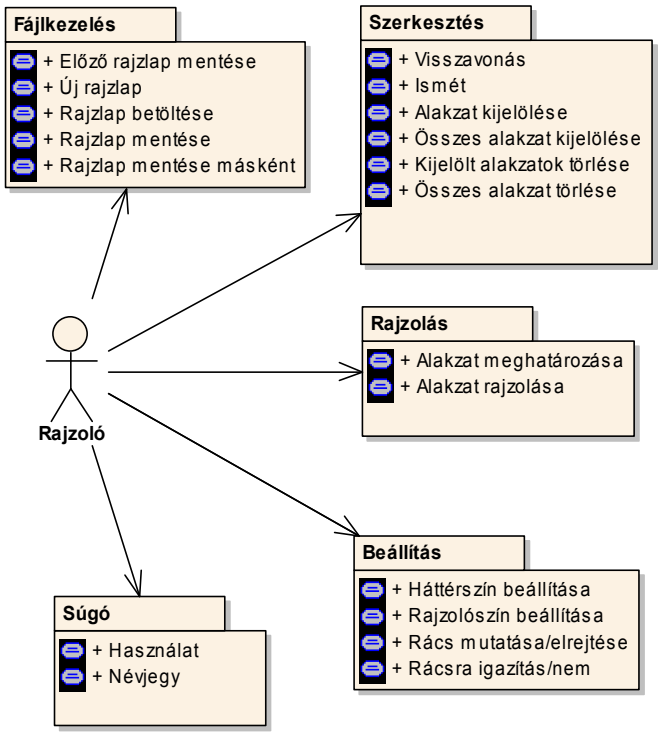
2. Követelmények

2.1. Fogalomtár

Magyar	Angol	Jelentés
Egyenes	Line	Két tetszőleges pontot a legrövidebb úton összekötő vonal.
Ellipszis	Oval, Ellipsis	Egy téglalapba írható ovális (lapított kör). Egy téglalap meghatározza az oválist.
Eszköztár	Toolbar	Ikonokat tartalmazó sor, mely a program fontosabb funkcióit teszi elérhetővé.
Ismét	Redo	A visszavonás (undo) visszavonása. Megkapjuk az előzőleg visszavont állapotot.
Kijelölés	Select	Egy alakzatnak a többitől való megkülönböztetése abból a célból, hogy valamilyen műveletet elvégezzünk vele (pl. törlés).
Rajzlap	Drawing paper DrawPanel	Képernyőterület, melyre egérrel rajzolunk. A lapon síkidomokat helyezünk el.
Rajzolószín	Draw color	A következőként rajzolt alakzat ilyen színű lesz.
Rács	Grid	Adott távolságra levő függőleges és vízszintes egyenesek.
Rácsra igazítás	Align to grid	A funkció beállításkor az ezután rajzolt síkidomok kezdő és végpontja rácspontra fog esni, függetlenül attól, hogy a rács látszik vagy sem.
Síkidom, alakzat	Figure	Kétdimenziós alakzat, például egyenes, téglalap, ellipszis, szabadkézi rajz stb.
Téglalap	Rectangle	Derékszögű négyszög. A határoló egyenesek párhuzamosak a képernyő széleivel.
Visszavonás	Undo	Visszatérés a rajzolás előző állapotához. Olyan, mintha az utolsó kirajzoló művelet nem történt volna meg.

2.2. Használati esetek (funkcionális követelmények)

A használatieset diagramot a következő ábra mutatja. Az alkalmazásnak egyetlen használója van, a rajzoló. A használati eseteket logikailag csoportosítottuk.

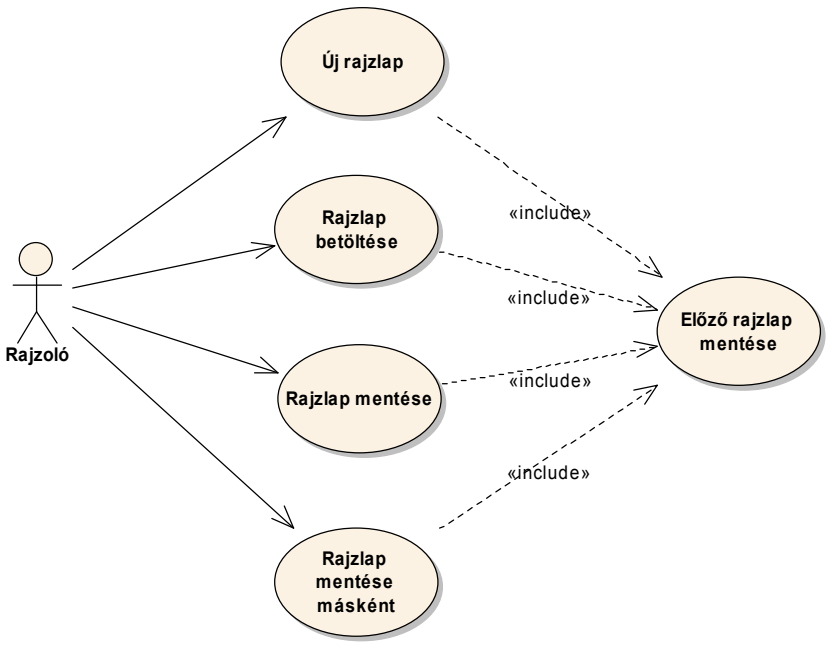


KissDraw használatieset diagramja

A következőkben részletezzük a használati eseteket.

Fájlkezelés

A csomag a rajzlap betöltésével és elmentésével kapcsolatos használati eseteket fogja össze. HE diagramja az ábrán látható. A mindenkori fájlnév (rajzlap neve) az alkalmazás státuszsorában látható. Ha a rajlapnak nincs neve, akkor a "Névtelen kiírás jelenik meg".



Új rajzlap
Végrehajtásra kerül az "Előző rajzlap mentése" HE. Ezután megjelenik egy üres, fehér rajzlap, "Névtelen" fájlnévvel.

Rajzlap betöltése
Végrehajtásra kerül az "Előző rajzlap mentése" HE. Ezután megjeleník egy fájldialógus, melyből a Rajzoló választhat betöltendő rajzot. Két eset lehetséges:

- Kiválasztott egy rajzot: Betöltésre kerül a kiválasztott rajzlap (Hiba1).
- Mégse: Nem történik betöltés, marad az előző rajzlap.

Rajzlap mentése
Ha a rajzlapnak van neve, akkor kiírás a lemezre. Ha nem volt még neve, akkor meghívásra kerül a "Rajzlap mentése másként" HE.

Rajzlap mentése másként
Felkínáljuk egy fájldialógust. A Rajzoló két gombot nyomhat le:

- Mentés: Ha a kiválasztott fájl még nem létezik, Rajzlap mentése a megadott néven és vége. Ha már létezik, akkor kérdés: "Felülírjam?" Igen esetén Rajzlap mentése, és vége. Nem esetén marad a fájldialógus, újra lehet választani.
- Mégse: Nem történik semmi.

Előző rajzlap elmentése (tartalmazott használati eset)
Ha az aktuálisan szerkesztett rajzlap módosult, megkérdezi a Rajzolót, hogy elmentse-e. Lehetséges válaszok:

- Igen: Meghívásra kerül a "Rajzlap mentése" HE. Az eredetileg végrehajtandó HE folytatódik.
- Nem: A rajzlap nem lesz elmentve. Az eredetileg végrehajtandó HE folytatódik.
- Mégse: Nem történik semmi; az eredetileg végrehajtandó HE is megszakad.

Szerkesztés

Visszavonás, Ismét
A következő műveletek számítanak: Alakzat rajzolása, Alakzat kijelölése, Összes alakzat kijelölése, Kijelölt alakzatok törlése, Összes alakzat törlése
Új rajzlap betöltésekor ez a lehetőség is alapállapotba kerül, nincs visszavonandó művelet.

Alakzat kijelölése
Ha az egér egy alakzat határvonala felett „húz el”, akkor az egér formája megváltozik. Ilyenkor egérkattintásra az alakzat ki lesz jelölve, illetve ha már ki volt jelölve, akkor a kijelölés megszűnik. A kijelölt alakzat vonala körül kis karikák jelennek meg.

A többi HE értelemszerű.

Rajzolás

Alakzat meghatározása

Kiválasztható a következőként rajzolandó alakzat (az alakzat típusa, mint egyenes, téglalap stb).

Alakzat rajzolása

Egérrel kattintunk az alakzat kezdőpontján, majd vonszoljuk az egeret. Az alakzat végpontja az egér koordinátája lesz annak felengedésekor. Az alakza típusa, valamint a kezdő- és végpont meghatározza az alakzat rajzát.

Súgó

Használat

Megjelenik egy (nem modális) ablak egy egyszerű szövegterülettel, benne az alkalmazás egyszerű leírása.

Névjegy

Megjelenik egy modális ablak, rajta az alkalmazás és a fejlesztő alapvető adatai.

Beállítás

A háttérszínt és a rajzolószínt színdialógussal lehet kiválasztani. A rács mutatása/elrejtése, valamint a rácsra igazítás/igazítás megszüntetése jelölőmezőkkel történik.

2.3. Nemfunkcionális követelmények

Egyfelhasználós, eseményvezérelt, desktop alkalmazás. Az alkalmazás induláskor legyen teljes képernyős. A használata legyen egyszerű, és a színek legyenek kellemesek.

Módszertanok:

- Egységesített eljárás, UML

Fejlesztési környezet:

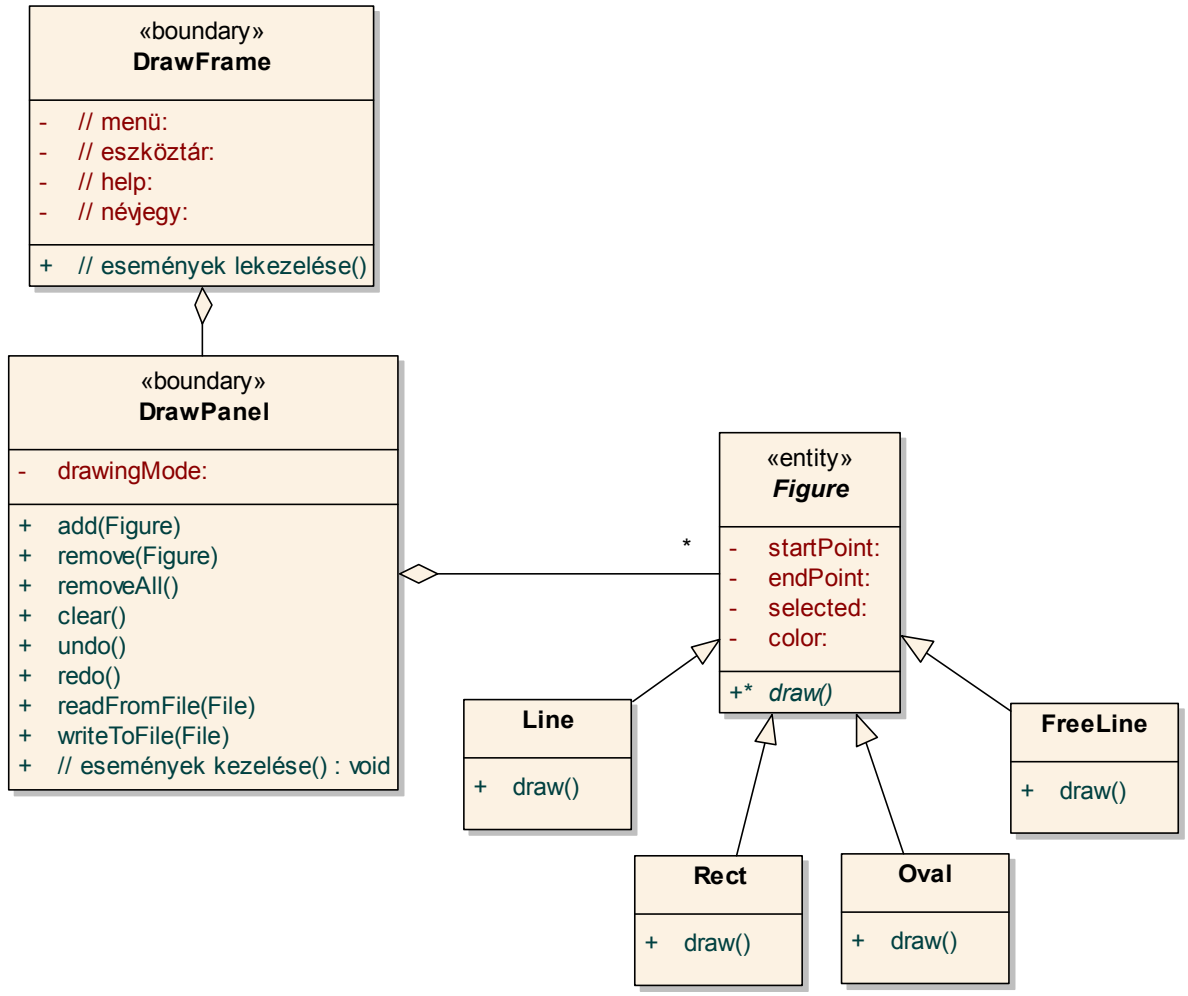
- IBM PC kompatibilis számítógép, Windows vagy Linux operációs rendszer
- JBuilder alapváltozata, JDK
- CASE eszköz: Enterprise Architect (http://www.sparxsystems.com.au/)

Futtatási környezet:

- IBM PC kompatibilis számítógép, Windows vagy Linux operációs rendszer
- JRE futtatórendszer

Gondoljunk a funkciók jövőbeni bővítésére: a fejlesztői dokumentáció alapján a továbbfejlesztés legyen egyszerű!

2.4. Szakterületi osztálydiagram



KissDraw – szakterületi osztálydiagram

Az alkalmazás keretében (DrawFrame) egy rajzlapot (DrawPanel) helyezünk el. A rajzlap síkidomokat (Figure) tartalmaz. Bármely síkidom lehet egyenes (Line), téglalap (Rectangle), ellipszis (Oval) vagy szabadkézi rajz (FreeLine).

- **Figure**: A síkidomok (alakzatok) absztrakt ősosztálya. Információhordozó osztály, eseményekre nem reagál. Egy síkidomnak van kezdő és végpontja, van színe, és ki lehet választani. A síkidom kirajzolását (draw) az utód osztályban kell megadni.
- **Line, Rect** stb.: Síkidomok. A síkidom típusától függően itt a kezdő és végponton kívül más adatok is lehetnek (pl. szabadkézi rajz esetén a többi pont). Minden síkidomnak van saját kirajzoló metódusa. A DrawPanel a kirajzoló metódusok segítségével rajzolja a síkidomokat a rajzlapra. A síkidomok listája bővíthető (pl. lehetne háromszög, szöveg stb.).

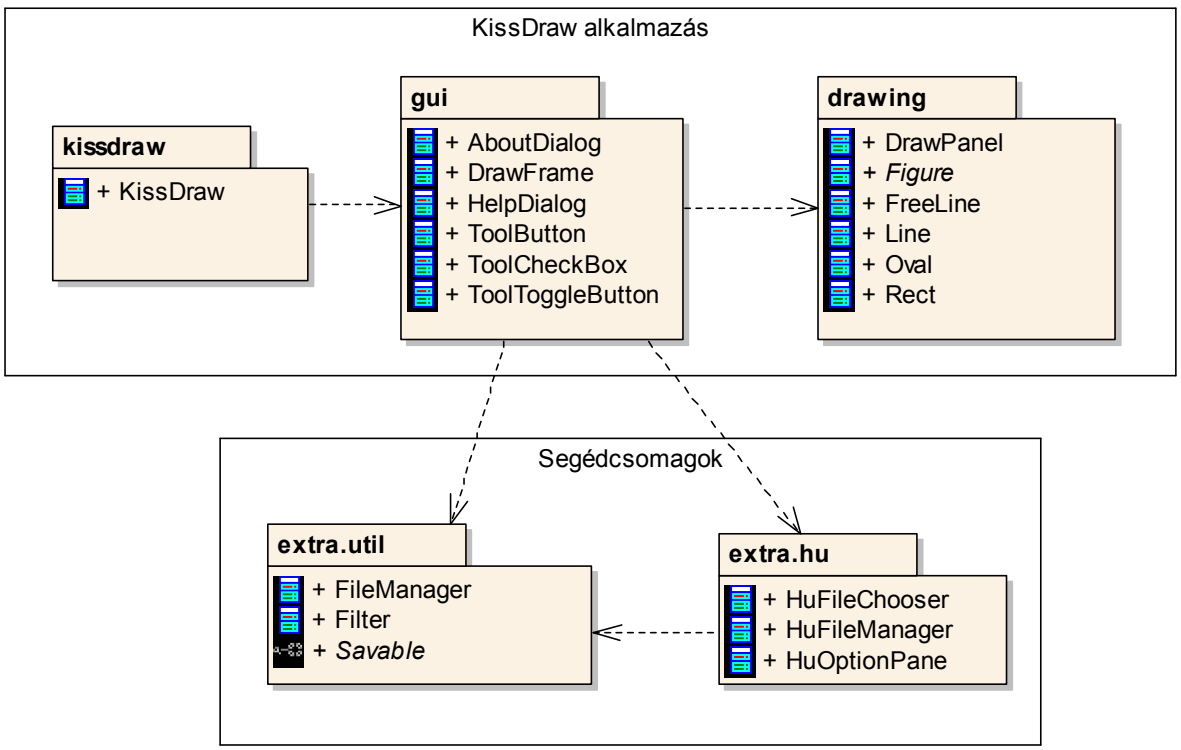
- **DrawPanel**: Rajzlap; határ osztály. A rajzlap viselkedése a rajzolási módtól (drawingMode) függ. A rajzolási mód lehet:
  - Egyenes mód,
  - Téglalap mód,
  - Ellipszis mód, vagy
  - Szabadkézi rajz mód: Egér segítségével kirajzolható a lapra az aktuális alakzat. Az alakzat kezdő- és végpontját az egér koordinátái adják meg az egér lenyomásakor, illetve felengedésekor.
  - Kijelölő mód: Ha az egér egy alakzat határvonala felett „húz el”, akkor az egér formája megváltozik. Ilyenkor egérkattintásra az alakzat ki lesz jelölve, illetve ha már ki volt jelölve, akkor a kijelölés megszűnik.

  A rajzpanelnek különböző parancsok küldhetők:
  - a kijelölt, illetve az összes alakzat törlése;
  - a rács megjelenítése, eltüntetése;
  - igazítás rácsra, illetve az igazítás megszüntetése. Ha az igazítás érvényben van, akkor a következő kirajzolt alakzat rácsra igazodik.

  A rajzpanel lemezre menthető. A betöltést/mentést kívülről lehet indítani. A betöltési és mentési folyamatot az osztály definiálja (readFromFile, writeToFile).
- **DrawFrame**: Az alkalmazás főablaka. Határosztály, van menüje és eszköztára. A rajzlapra vonatkozó műveleteket innen lehet indítani. Innen lehet segítséget is kérni.

3. Programterv

3.1. Csomagfelépítés



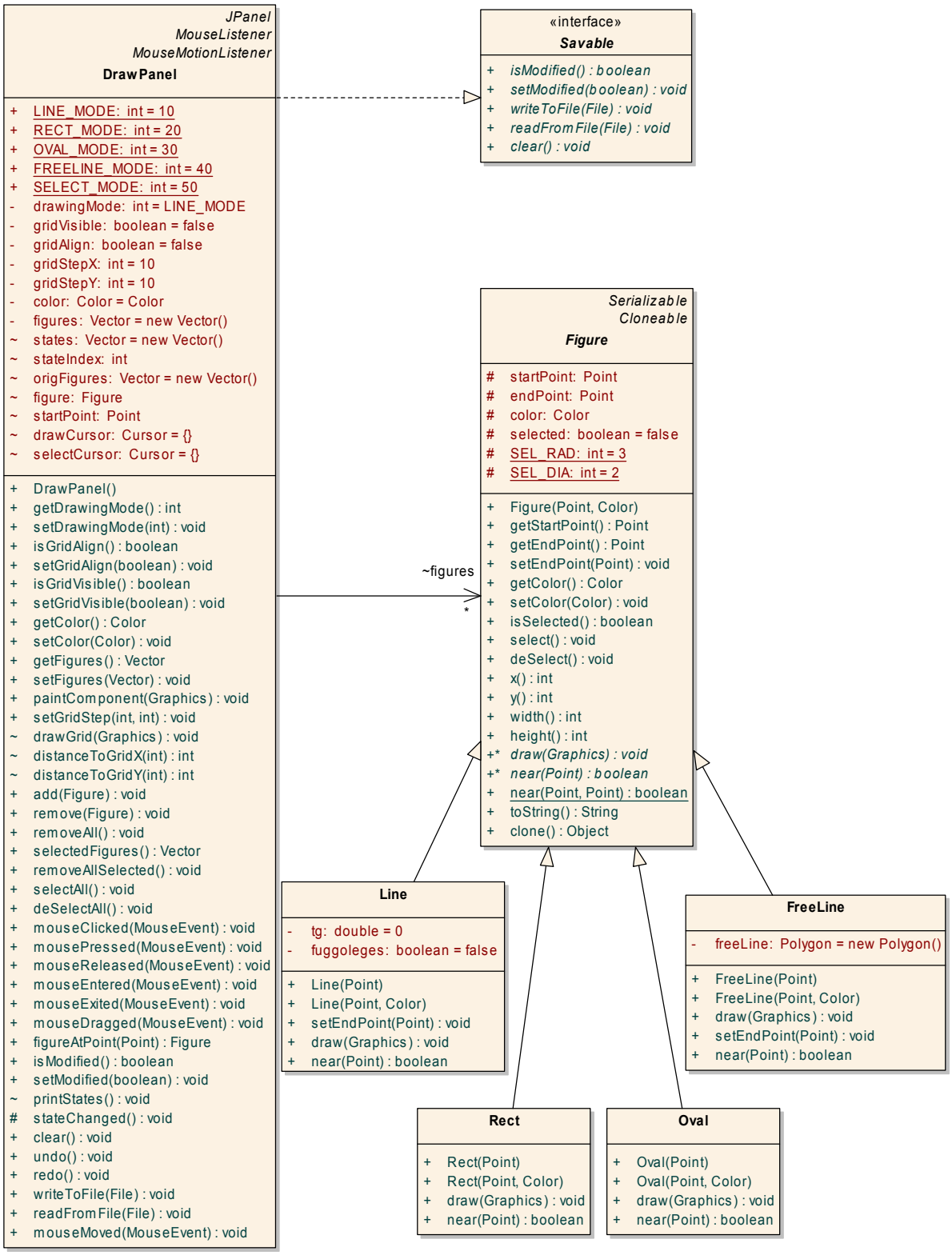
A KissDraw csomagfelépítése

Az alkalmazás három saját és két segédcsomagból épül fel:

- kissdraw: Csak az indító osztályt tartalmazza. Ez a program belépési pontja. A KissDraw osztálynak mindössze az a feladata, hogy létrehozza az alkalmazás fő keretét, a DrawFrame egy példányát.
- gui: Tartalmazza az alkalmazás főablakát, a menü és eszköztár megvalósításához szükséges osztályokat, valamint a segítség és névjegy dialógusokat.
- drawing: A rajzspecifikus elemeket tartalmazza.
- extra.util: A FileManager osztály tetszőleges Savable objektum lemezre való mentését és betöltését menedzseli. Szükség esetén kommunikál a felhasználóval.
- extra.hu: A HuOptionPane osztály a javax.swing.JOptionPane osztály mintájára magyar dialógusokat definiál. A HuFileChooser egy magyarul „beszélő” JFileChooser, a HuFileManager pedig egy magyar FileManager.

Most megadjuk egyenként az egyes csomagok részletes osztálydiagramját, valamint az egyes osztályok fontosabb feladatainak (adatainak, metódusainak) leírását.

3.2. drawing csomag (alkalmazáslogika)



drawing csomag

drawing.Figure osztály

Absztrakt osztály, ebből öröklődik az összes alakzat.
Fontosabb adatok:

- startPoint, endPoint: Point
  Minden alakzatnak van egy kezdő és egy végpontja (például a startPoint-nál kezdik el az alakzat rajzolását, és az endPoint-nél engedik fel az egeret).
- color: Color
  Az alakzat színe.
- selected: boolean
  true esetén ki van választva. Ekkor az alakzat bizonyos helyein köröcskék jelennek meg.
- SEL_RAD
  A kijelölő köröcske sugara.

Fontosabb metódusok:

- x(), y(): int
  Megadja a bal felső sarok koordinátáit.
- width(), height(): int
  Megadja a startPoint és endPoint közötti vízszintes és függőleges vetületet.
- draw()
  Absztrakt metódus. Kirajzolja az alakzatot.
- near(Point): boolean
  Megadja, hogy a megadott pont közel van-e (1-2 pontnyira) az alakzathoz. Például ha az egérkurzor közel lesz az alakzat vonalához, akkor megváltozik az egérkurzor alakja.
- near(Point,Point): boolean
  Statikus metódus. Megadja, hogy a megadott két pont közel van-e egymáshoz.

drawing.Line osztály

Egy egyenest határoz meg. Felülírja a Figure osztály megfelelő metódusait (draw, near...)

drawing.Rect osztály

Egy téglalapot határoz meg. Felülírja a Figure osztály megfelelő metódusait (draw, near...)

drawing.Oval osztály

Egy ellipszist határoz meg. Felülírja a Figure osztály megfelelő metódusait (draw, near...)

drawing.FreeLine osztály

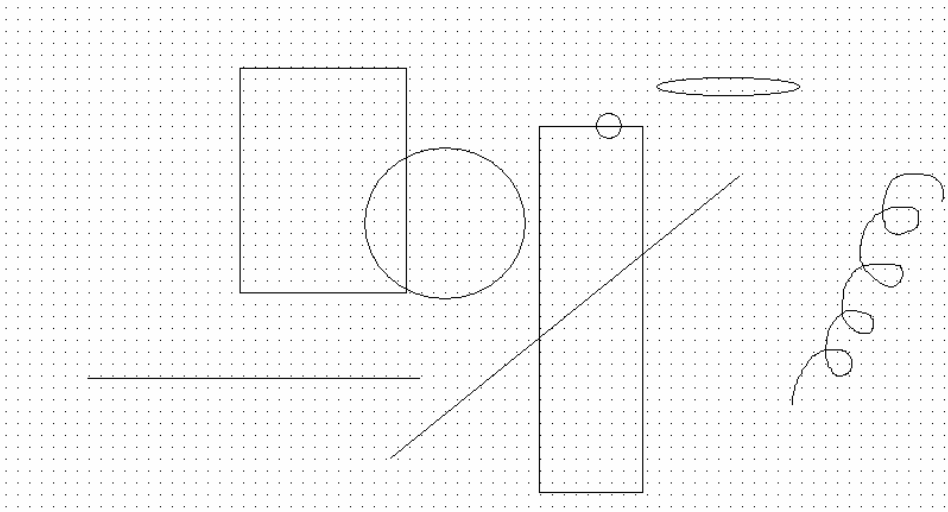
Egy szabadkézi vonalat határoz meg. A vonal az egér húzásának vonalában keletkezik. A vonal pontjait egy poligon tárolja. Felülírja a Figure osztály megfelelő metódusait (draw, near...).

drawing.DrawPanel osztály

Rajzlap, amelyre rajzolunk. A JPanelből származik. Implementálja a Savable interfészt, így a rajzlapot tudja kezelni a FileManager (elmenthető, visszaolvasható).

Fontosabb adatok:

- Rajzolási mód: int
  `LINE_MODE, RECT_MODE, OVAL_MODE, FREELINE_MODE, SELECT_MODE.`
  Mindig az aktuális rajzolási mód szerint történik a rajzolás.
- gridVisible: boolean
  Látható-e a rács. A rács pontokból áll, ahol a pontok gridStep=10 pontnyira vannak egymástól.
- gridAlign: boolean
  Van-e rácsra igazítás. Rácsra igazítás esetén az alakzatok kezdő és végpontjai rácspontok lesznek.
- color: Color.
  Az aktuális rajzolószín.
- figures: Vector.
  Az alakzatok listája.



A rajzpanel lekezeli a billentyű- és egéreseményeket.
Egéresemények:

- Ha `SELECT` módban vagyunk, akkor az egérkurzor alatti alakzatot kattintással kiválaszthatjuk.
- Nem `SELECT` módban (`LINE_MODE`, `RECT_MODE`, `OVAL_MODE`, `FREELINE_MODE`) az egérrel rajzolhatunk.

Billentyűesemények:

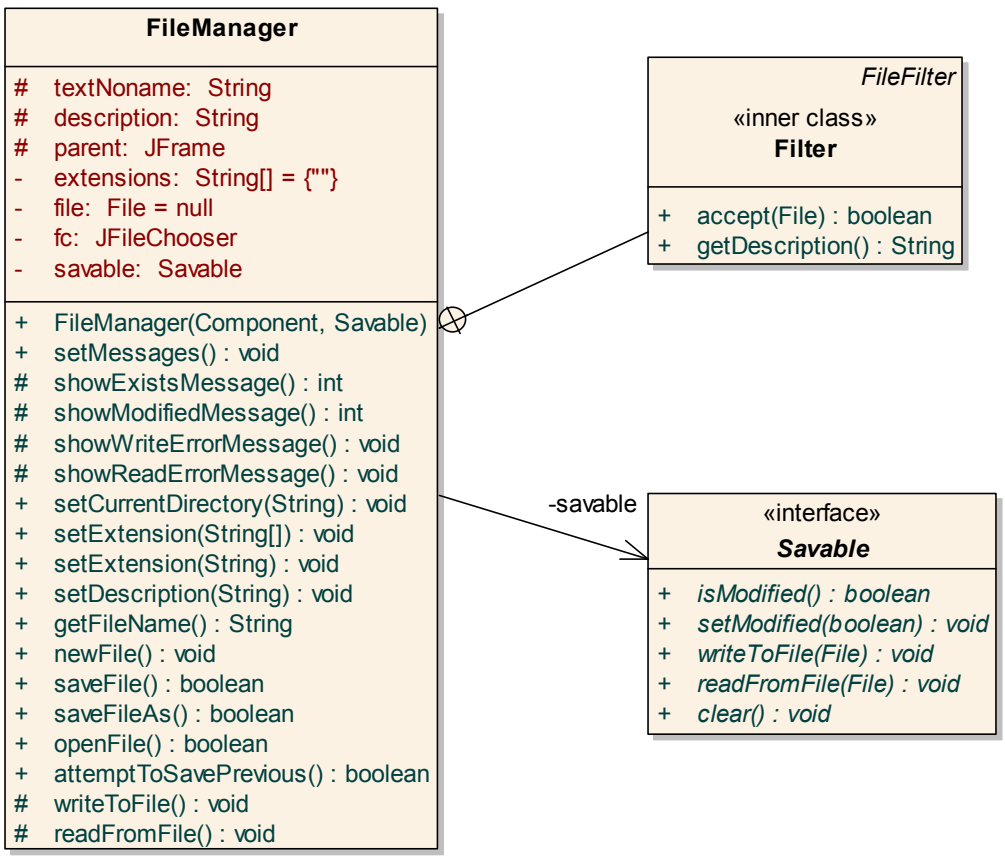
- delete: a kijelölt alakzatok törlése.

Fontosabb metódusok:

- selectAll()
- deSelectAll()
  Összes alakzat kiválasztása, illetve a kiválasztás megszüntetése.

- removeAllSelected ()
  Az összes kiválasztott alakzat törlése.
- selectedFigures(): Vector
  Visszaadja a kiválasztott alakzatokat.
- add(Figure)
- remove(Figure)
  Alakzat hozzáadása a rajzlaphoz, illetve levétele arról.
- clear()
  A rajzlap alapállapotba hozása. A rajzlap fehér színű és üres lesz. Az állapotokat elfelejti (redo és undo).
- undo()
  Legutolsó művelet visszavonása.
- redo()
  Ismét.
- writeToFile(File)
- readFromFile(File)
  A rajzlap kiírása a megadott fájlba, illetve visszaolvasása onnan.

3.3. extra csomag (külső könyvtár)



extra.util csomag

extra.util.Savable interfész

A FileManager osztály csak Savable objektumokat tud kezelni. A következő metódusokat kell implementálni:

- isModified():boolean
  Megadja, hogy az elmentendő objektumot módosították-e a legutóbbi mentés óta.
- setModified(boolean)
  Ezután az isModified értéke false lesz. Ha módosították az elmentendő objektumot, akkor az objektumnak gondoskodnia kell e metódus meghívásáról.
- writeToFile(File)
- readFromFile(File)
  A rajzlap kiírása a megadott fájlba, illetve visszaolvasása onnan.
- clear()
  Az objektum alapállapotba hozása, vagyis "kiürítése" (mintha az objektum most jött volna létre).

extra.util.FileManager osztály

A FileManager osztály egy objektum elmentését, betöltését menedzseli a felhasználó bevonásával. Az osztályban megtalálhatók az egyszerűbb szövegszerkesztőkben is megszokott műveletek, mint Megnyitás, Mentés, Mentés másként stb. A FileManager gondoskodik például arról, hogy ha az objektumot a legutóbbi mentés óta módosították, akkor megkérdezi a felhasználót, hogy el akarja-e menteni a pillanatnyi állapotot.
A FileManager objektumhoz a konstruktorában rendeljük hozzá a menedzselt objektumot.
A FileManager objektum fájldialógust kínál fel, hogy megadják azt a fájlt, amelybe kiíródik, illetve amelyből beolvasásra kerül majd a menedzselt objektum. A fájldialógusban be lehet állítani az induló mappát, a lehetséges fájlkiterjesztéseket és a leírást.
A FileManager használatának feltétele, hogy a menedzselt fájl implementálja a Savable interfészt.

Metódusok:

- FileManager(parent: Component, savable: Savable)
  Konstruktor. A parent objektum az üzenetek kiírásához kell. Az egyes dialógusok a parent (szülő komponens) közepén jelennek meg.
- setCurrentDirectory(path: String)
  Beállítja a fájldialógus induló mappáját.
- setExtension(ext: String[])
  Beállítja a fájldialógus lehetséges fájlkiterjesztéseit.
- setExtension(ext: String)
  Beállítja a fájldialógus egyetlen lehetséges fájlkiterjesztését.
- setDescription(description: String)
  Beállítja a fájldialógusban megjelenő leírást.
- getFileName(): String
  Visszaadja a fájl aktuális, teljes útvonalát.
- newFile()
  Új fájl létrehozása.

- saveFile(): boolean

  Fájl mentése. Ha még nincs hozzárendelve fizikai fájl, akkor meghívjuk a "Mentés másként" funkciót.
- saveFileAs(): boolean

  Mentés másként. Bekérünk egy fájlnevet. Ha megadják, akkor abba a fájlba mentünk. A visszaadott érték true, ha volt mentés, false, ha nem.
- openFile(): boolean

  Fájl betöltése. Előtte felkínáljuk mentésre az előző fájlt, ha azt módosították.
- attemptToSavePrevious(): boolean

  Az előző fájl mentése, ha azt módosították és a felhasználó akarja menteni. A visszatérési érték
  - true, ha a mentés el van intézve (akár van mentés, akár nem);
  - false, ha a felhasználó visszavonja eredeti akaratát.
- writeToFile()

  A savable objektum kiírása a file-ba. Ha nincs a fájlnak kiterjesztése, akkor az az alapértelmezett lesz – amit a setExtension metódussal elsőként adtak meg.
- loadFromFile()

  A savable objektum beolvasása a file-ból.
- exit()

  A filemanager befejezi működését, alapállapotba kerül. Az objektumot még elmenti, ha módosították és ezt a felhasználó igényli. A visszatérési érték
  - true, ha a kilépés el van intézve (akár van mentés, akár nem);
  - false, ha a felhasználó visszavonja eredeti akaratát.

extra.util.HuOptionPane osztály

A HuOptionPane a javax.swing.JOptionPane osztály mintájára magyar dialógusokat definiál. A részletezéstől eltekintünk.

Példák:

```
HuOptionPane.showMessageDialog(this,"Enter-re kezdődik...");
```

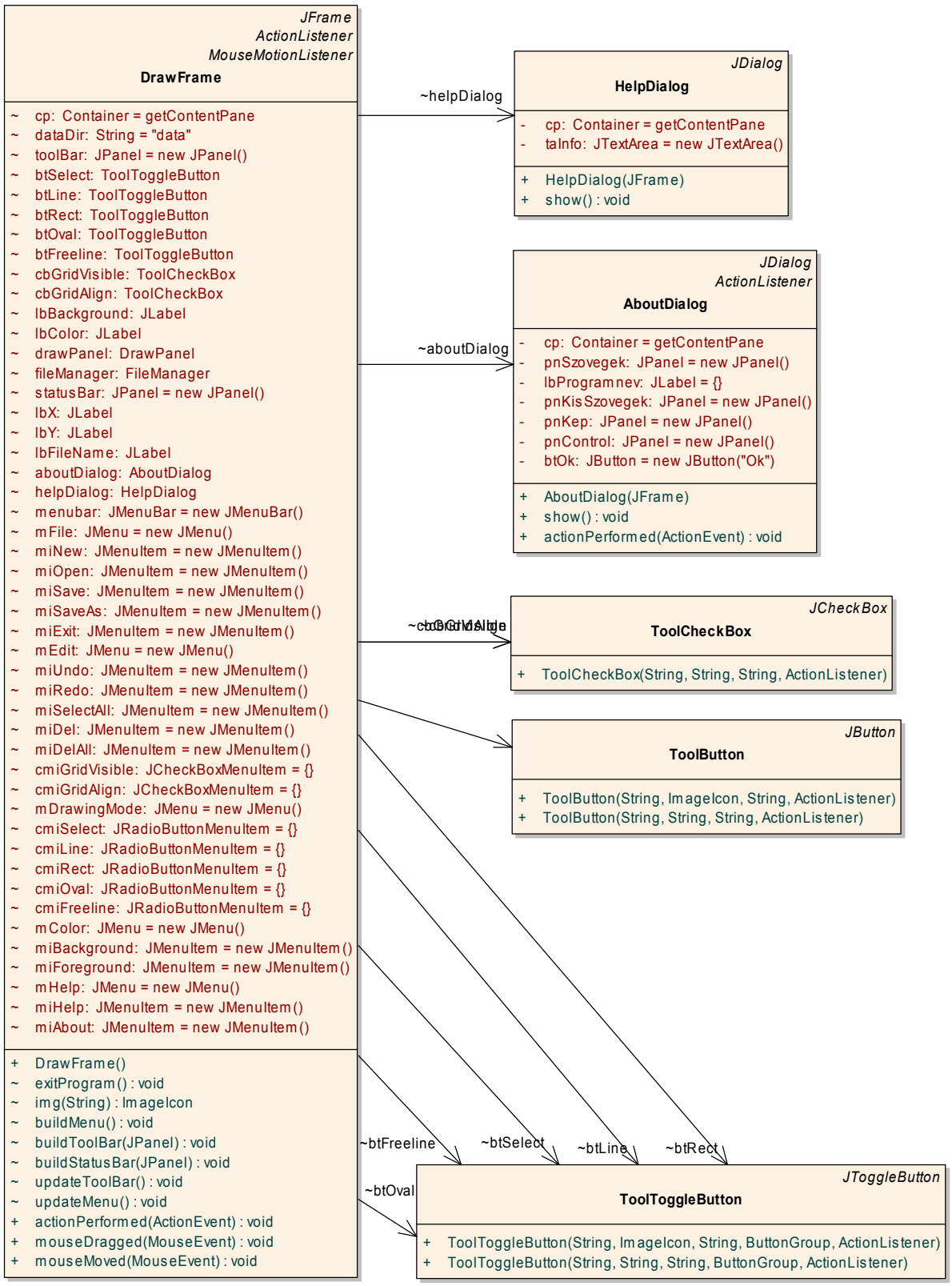

```
int option = HuOptionPane.showConfirmDialog(parent,
    getFileName() + " fájl megváltozott. Menti?",
    "Figyelmeztetés!",JOptionPane.YES_NO_CANCEL_OPTION,
JOptionPane.WARNING_MESSAGE);
```

extra.hu.HuFileManager osztály

A FileManager osztály leszármazottja. Mindössze annyiban különbözik tőle, hogy ez egy magyarul „beszélő” FileManager.

extra.hu csomag

3.4. gui csomag (felhasználói interfész)



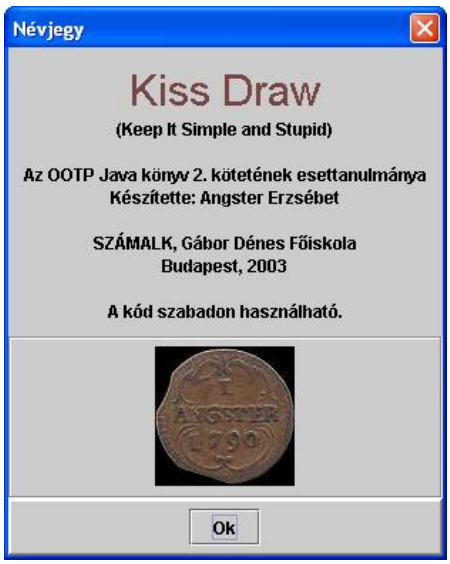
gui csomag

gui.HelpDialog osztály

A segítséget megjelenítő dialógus ablak. Egy görgethető szövegterületet tartalmaz. Az ablakot csak a jobb felső becsukó ikonnal lehet bezárni. Csak eltűnik, a legközelebbi kérésre pedig megjelenik.

gui.AboutDialog osztály

A program névjegyét megjelenítő dialógus ablak.



gui.ToolCheckBox

Olyan JCheckbox, melynek létrehozáskor kezdő paramétereket lehet adni: akcióparancs, szöveg, eszköztipp, figyelő. Eszköztár felépítésében segít.

Konstruktor:

- ToolCheckBox(String command, String text, String tip, ActionListener listener)

gui.ToolButton

Olyan JButton, melynek létrehozáskor kezdő paramétereket lehet adni: akcióparancs, kép, eszköztipp, figyelő. Eszköztár felépítésében segít.

Konstruktor:

- ToolButton(String command, ImageIcon img, String tip, ActionListener listener)

gui.ToolToggleButton

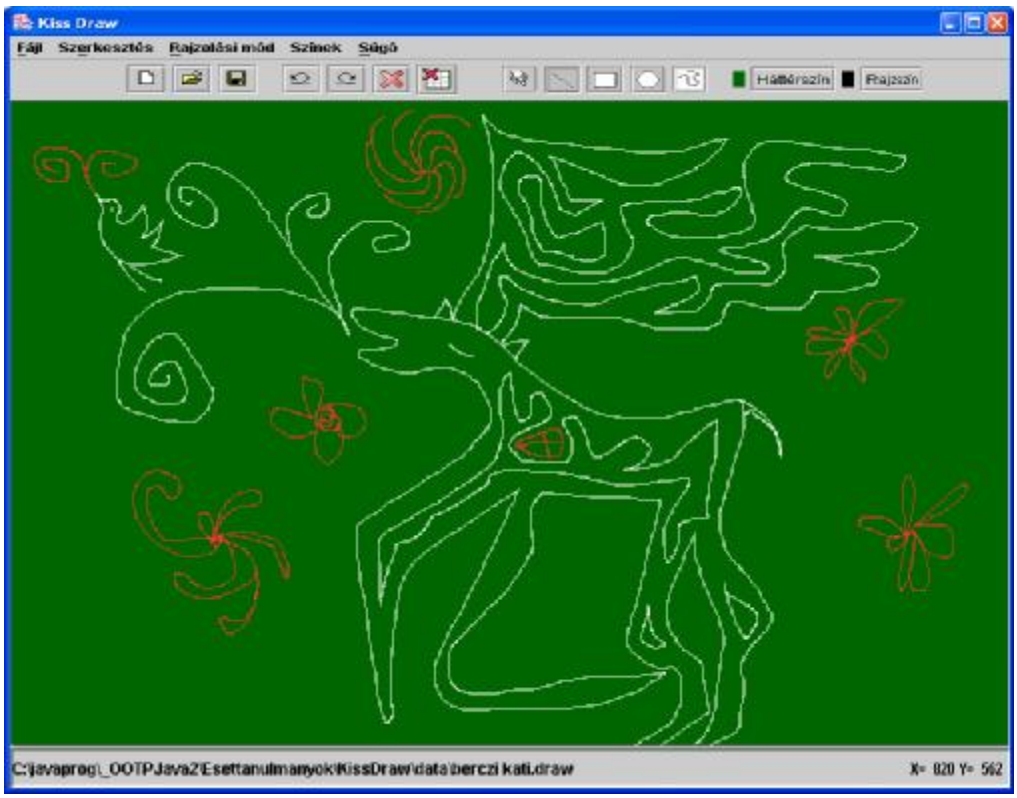
Olyan JToggleButton, melynek létrehozáskor kezdő paramétereket lehet adni: akcióparancs, kép, eszköztipp, figyelő. Eszköztár felépítésében segít.

Konstruktor:

- ToolToggleButton(String command, ImageIcon img, String tip, ButtonGroup bg, ActionListener listener)

gui.DrawFrame osztály

Az alkalmazás főablaka, határ osztály. Elhelyezünk rajta egy rajzlapot (DrawPanel). Felül van egy menü, alatta az eszköztár. A rajzlap alatt található a státuszsor.
A főablak egér- és billentyűeseményekre reagál. A rajzlapra vonatkozó műveleteket innen lehet elérni.
A főablakból lehet kérni a használati útmutatót (HelpDialog) és a névjegyet (AboutDialog).



Fontosabb adatok:

- String dataDir = "data"
  Az adatok induló könyvtára.
- DrawPanel drawPanel
  A rajzlap.
- FileManager fileManager
  A drawPanel fájlkapcsolata.
- HelpDialog helpDialog
- AboutDialog aboutDialog
  Súgó és névjegydialógusok.

Konstruktor:

- public DrawFrame()
  Felépíti a menüt, eszköztárat és a státuszsort. Létrehozza a rajzlapot. A rajzlapon figyeljük az egér koordinátáit. Létrehozza a fájlmenedzsert, mely a rajzpanel mentését/betöltését felügyeli. Létrehozza a dataDir könyvtárat, ha az nem létezik. Létrehozza a súgót és a névjegyet.

- void exitProgram()

  Kilépés a programból. Ha volt módosítás, akkor megengedjük a mentést. Ha nem gondolta meg magát, akkor tényleg kilépünk.
- ImageIcon img(String fName)

  Létrehozza az fName fájlban tárolt képet. A KisDraw osztály könyvtárában van a resources könyvtár, és abban az fName nevű fájl.
- void buildMenu()

  A menü felépítése.
- void buildToolBar(JPanel tb)

  Az eszköztár (toolBar) felépítése.
- void buildStatusBar(JPanel sb)

  A státuszsor (statusbar) felépítése.
- void updateToolBar()

  Beállítja az eszköztár gombokat a drawPanel értékei alapján.
- void updateMenu()

  Beállítja a menü gombjait a drawPanel értékei alapján.

Menüterv

Az egyes menüpontok mellett az általuk indított akcióparancs neve található.

```
Fájl
  Új (new)
  Megnyitás... (open)
  Mentés (save)
  Mentés másként (saveAs)
  --------
  Kilépés (exit)

Szerkesztés
  Visszavonás (undo)
  Ismét (redo)
  --------
  Összes kijelölése (del)
  --------
  Kijelöltek törlése (selectAll)
  Összes törlése (delAll)
  --------
  Rács látszik (gridVisible)
  Rácsra igazítás (gridAlign)

Rajzolási mód
  Kijelölés (select)
  --------
  Egyenes (line)
  Téglalap (rect)
  Ellipszis (oval)
  Szabadkézi (freeline)

Színek
  Háttérszín (background)
  Rajzolószín (foreground)

Súgó
  Használati útmutató (help)
  Névjegy (about)
```

Eszköztár terv

Az egyes eszköztár elemek mellett az általuk indított akcióparancs neve található.
Csoportokban:

```
Új (new)
Megnyitás... (open)
Mentés (save)
--------
Visszavonás (undo)
Ismét (redo)
Kijelöltek törlése (selectAll)
Összes törlése (delAll)
--------
Kijelölés (select)
Egyenes (line)
Téglalap (rect)
Ellipszis (oval)
Szabadkézi (freeline)
--------
Háttérszín (background)
Rajzolószín (foreground)
```

Akcióparancsok (menü, eszköztár)

Az akcióparancsok a menüből, illetve az eszköztárból elérhetők. A legtöbb parancs értelemszerűen meghívja a FileManager, illetve a DrawPanel (rajzlap) megfelelő metódusát; a leírásokat lásd ott. Minden egyes parancs lekezelése után frissítjük az eszköztárat (updateToolBar) és a menüt (updateMenu).

- new
  fileManager.newFile()
- open
  fileManager.openFile()
- save
  fileManager.saveFile()
- saveas
  fileManager.saveFileAs()
- exit
  exitProgram()
- undo
  drawPanel.undo()
- redo
  drawPanel.redo()
- del
  drawPanel.removeAllSelected()
- selectAll
  drawPanel.selectAll()
- delAll
  drawPanel.removeAll()
- gridVisible
  drawPanel.setGridVisible(bt.isSelected())

- gridAlign
  drawPanel.setGridAlign(bt.isSelected())
- select
  drawPanel.setDrawingMode(drawPanel.SELECT_MODE)
- line
  drawPanel.setDrawingMode(drawPanel.LINE_MODE)
- rect
  drawPanel.setDrawingMode(drawPanel.RECT_MODE)
- oval
  drawPanel.setDrawingMode(drawPanel.OVAL_MODE)
- freeline
  drawPanel.setDrawingMode(drawPanel.FREELINE_MODE)
- background
  // drawPanel háttérszínének beállítása színválasztó dialógus segítségével.
- foreground
  // drawPanel rajzolószínének beállítása színválasztó dialógus segítségével.
- help
  helpDialog.show()
- about
  aboutDialog.show()
- egérmozgás a rajzlapon: koordináták megjelenítése a státuszsorban.

4. Továbbfejlesztési tervek

A programot természetesen a végtelenségig lehetne még fejleszteni. Néhány funkció megvalósítása azonban meglehetősen kézenfekvő lenne. Az itt felsorolt ötletek megvalósítását valamint további ötletek felsorolását a kedves Olvasóra bízom:

- Különböző vonalvastagság használata
- Kitöltőszín használata (a kijelölt alakzat kifestése a megadott kitöltőszínnel)
- Tömör alakzatok rajzolása
- Szövegírási lehetőség a rajzlapon
- Rajzlap nyomtatása
- Összes alakzat kijelölése
- Csoportos kijelölés
- A kijelölt alakzatok mozgatása
- Menüpontok szürkítése, ha a hozzátartozó funkciónak éppen nincs értelme.
- Az utoljára betöltött fájlok újranyitása (reopen, a fájl menü végén)
- Az alkalmazás kinézetének megadása (look & feel)
- Az alkalmazás bizonyos adatainak beállítása, a beállítások elmentése. Például lehetséges fájlkiterjesztések, ablakméret, kinézet, alapértelmezett háttérszín stb.
- Biztonsági másolat készítése a rajzokat tartalmazó data könyvtárról.
- Hibafájl készítése. Ha a program hibával áll le, akkor a hibát felírja a fájlba.

Találjon ki további lehetőségeket!