OKJ gyakorló feladatsor

Mindegyik feladatot GitHub-on is el kell tárolni, a feladatban megjelölt commitokkal. Lokálisan készíts repót, majd a legvégén töltsd fel a remote repóba. Ha nem sikerül, akkor is legyen meg, ami és ahogy sikerült. A legvégén a leadáshoz kell a GitHub linket is mellékelni.

**A feladatokon sorban haladj!**

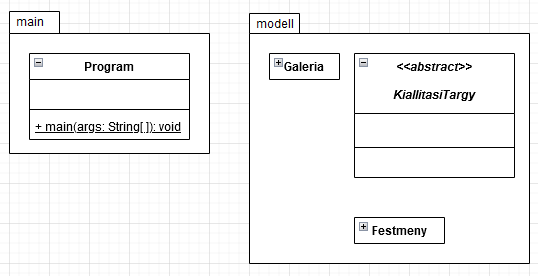
**Olvasd végig a feladatokat, mielőtt nekikezdenél!!!**

Mind a kettő feladathoz kell .gitignore és readme.md fájl. Lehet ugyanazokat használni! Tartalmukat tekintve:

* a .gitignore: Java és NetBeans specifikus mappák, fájlok kihagyása
* a readme.md: teljes nevet tartalmazza

# Konzolon elkészítendő feladat

Egy galériának kell kiállítási tárgyakat tárolnia, ahol egyelőre csak festmények lesznek. Az alábbi UML alapján hozd létre a projektet GaleriaProjekt néven:



*Itt legyen az* ***1. commit***

**A következő commit-ig csak a modell csomag osztályaiban szabad dolgozni!**

A kiállítási tárgyaknak van **létrehozási dátumuk**, ami **dátum** formátumú. **Ha nem adunk meg dátumot**, akkor a **mai dátum** kerül beállításra. **Ha jövőbeli dátumot adunk meg, akkor dobjon saját kivételt**, amit a galéria kezeljen! Minden kiállítási tárgynak van készítője és címe.

A galériába fel tudunk venni kiállítási tárgyakat, a festményeknek lehet egy fájl elérési utat tartalmazó mezőjük, ami alapján meg lehet jeleníteni a képet. Ha létezik a fájl, akkor a megjelenítő metódus írja ki, hogy „megjelenítés folyamatban…”, ha nem létezik, akkor azt, hogy „nem lehet megjeleníteni!”. Tehát a kép megjelenítésével nem kell foglalkozni!

Minden kiállítási tárgy rendezhető készítője és a tárgy címe alapján.

A galéria legyen bejárható, amikor visszaadja tárgyait, nem módosítható (vagy másolat) listában.

*Itt legyen az 2****. commit***

Most a main metódusban a galéria példányosítása után, be kell mutatni a funkcionalitást. A festmények felvétele lehet a galéria konstruktorában vagy itt, mindegy. De a következő adatokkal példányosítsunk:

1. készítő és cím
2. készítő és cím a mai dátummal
3. készítő és cím a *hibás* dátummal
4. készítő és cím fájl eléréssel
5. készítő és cím fájl eléréssel, mai dátummal

A bejárást (foreach) alkalmazva mutassuk meg a cím szerinti rendezést, majd a készítő szerintit is, végül mutassuk be a megjelenítő metódust!

A fájl elérésének teszteléséhez hozzunk létre egy kep.txt fájlt, aminek csak a létezése érdekes, a tartalma nem. Az utolsó kettő objektumnál a „megjelenítés folyamatban…” szöveg jelenjen meg.

*Itt legyen az 3****. commit***

A galéria osztály állapota kerüljön mentésre! Ehhez természetesen vissza lehet nyúlni a szükséges osztályokhoz! A kiírás, visszaolvasás is kerüljön bemutatásra a main-ben!

*Itt legyen a 4****. commit***

# Konzolos feladat értékelése

|  |  |  |
| --- | --- | --- |
| **Feladat** | **Elérhető** | **Kapott** |
| commit-ok száma és szövege megfelelőek | 1 pont |  |
| az 1. commit tartalma UML szerinti | 1 pont |  |
| Nem kötelező dátumot adni (de be lesz állítva) | 2 pont |  |
| Saját kivételt dob | 1 pont |  |
| Kivétel kezelése megtörtént | 1 pont |  |
| Kiállítási tárgyak felvehetőek | 1 pont |  |
| Nem kötelező elérési utat adni (de be lesz állítva) | 2 pont |  |
| megjelenítő metódus kész | 1 pont |  |
| A kiállítási tárgyak rendezhetőek (nem a festmény) | 1 pont |  |
| A galéria bejárható | 1 pont |  |
| Megfelelő az 5 példány, van benne hibás dátum is | 1 pont |  |
| Bejárások megfelelőek | 1 pont |  |
| Mentés / visszaállítás sikeres | 1 pont |  |
| Összesen: | 15 pont |  |

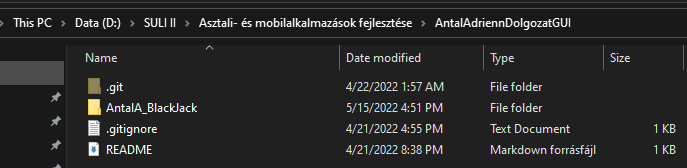
## jegyek:

* elégséges (2) 8-9 pont
* közepes (3) 10-11 pont
* jó (4) 12-13 pont
* jeles (5) 14 ponttól

# Grafikus felületen elkészítendő feladat

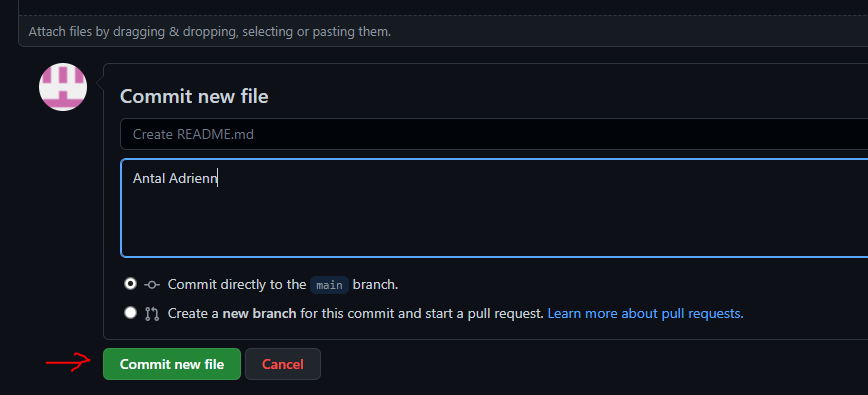
A mellékelt BlackJack NetBeans projektet kell megnyitni és abban elvégezni a következő feladatokat:

1. Nevezd át a projektet, a **Saját** neveddel kezdődjön, minta (ha Nagy Pálnak hívnak): Nagy**P\_**BlackJack. *Itt legyen az* ***1. commit***



1. Add a projekthez .gitignore és a readme.md fájlokat. *Itt legyen az* ***2. commit***

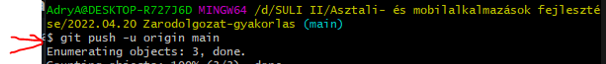
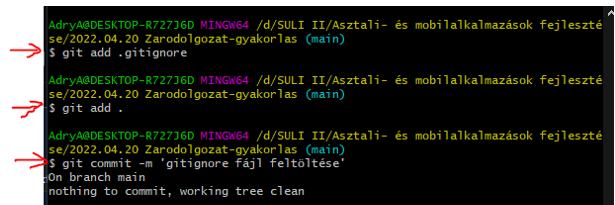
Online repo: readme fájlt megkreálom:

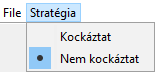


Majd git pull-lal lehúzom.

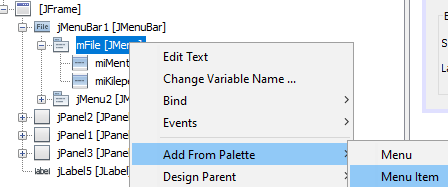
Gitignore.io 🡪 kulcsszavak: Java, Netbeans 🡪 Create 🡪 a megjelenő szöveg fájlba mentése .**gitignore** néven, és az egészet lokális mappába helyezem el

Ezután jöhetnek a köv. parancsok:



1. Adj egy menüt a felülethez a következő minta alapján:  
   , a menüpontok: , a stratégia menü elemei egymást kizáró választást tegyenek lehetővé: . Ez legyen az alapértelmezett kijelölés. *Itt legyen az* ***3. commit***File menü alatti menüpontok kialakítása:

Menüpontra jobb klikk 🡪 Add From Palette 🡪 Menu Item

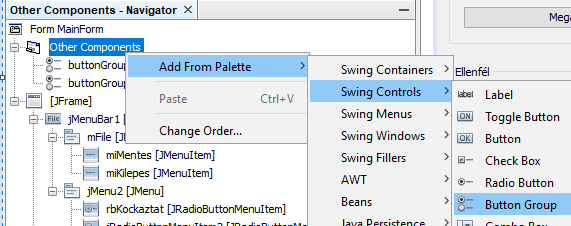


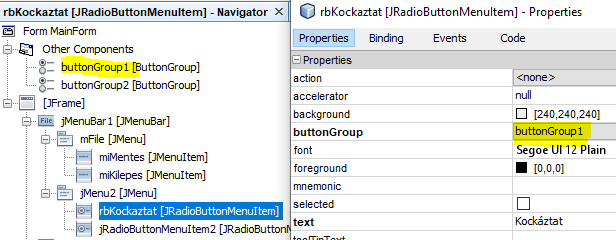
Stratégia menü alatti menüpontok kialakítása:

Menüpontra jobb klikk 🡪 Add From Palette 🡪 Menu Item / Radiobutton

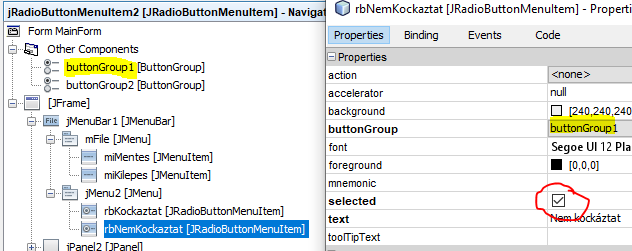


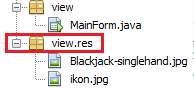
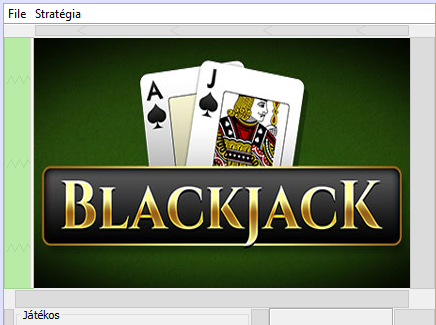
Egymást kizáró választás lehetővé tétele: buttongroup alkalmazása



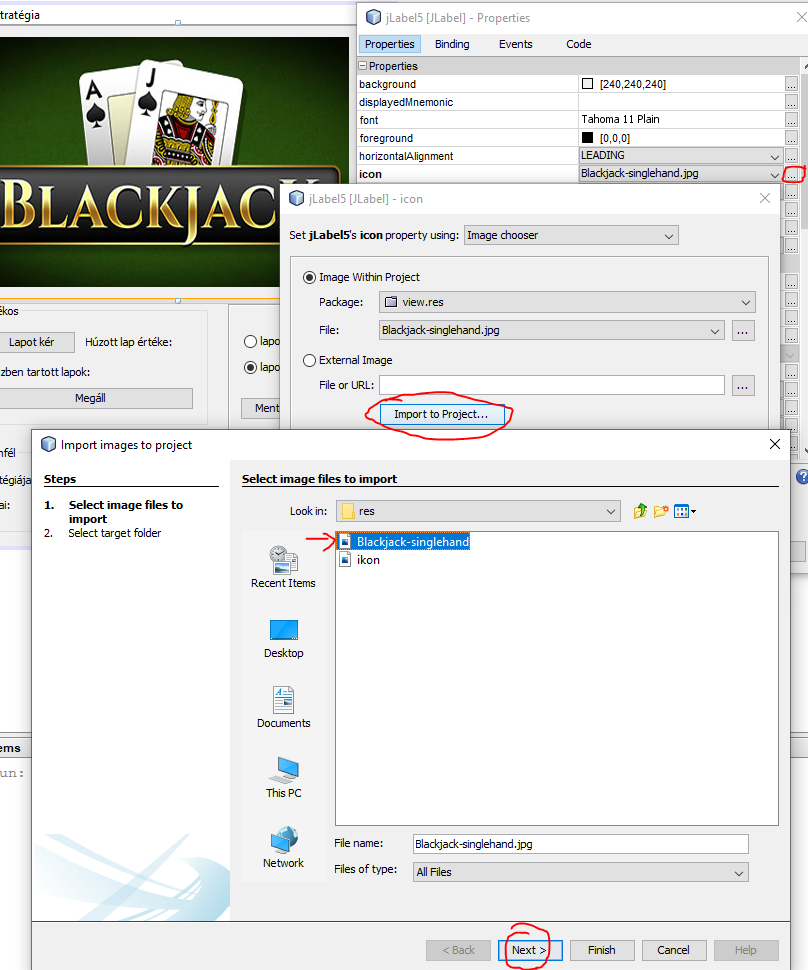


Itt látszik az alapértelmezett kijelölés beállítása is:

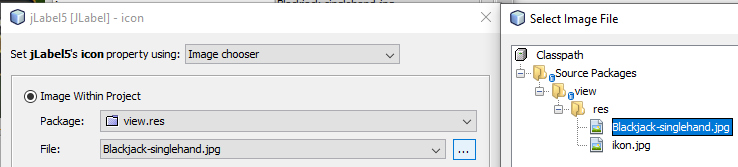


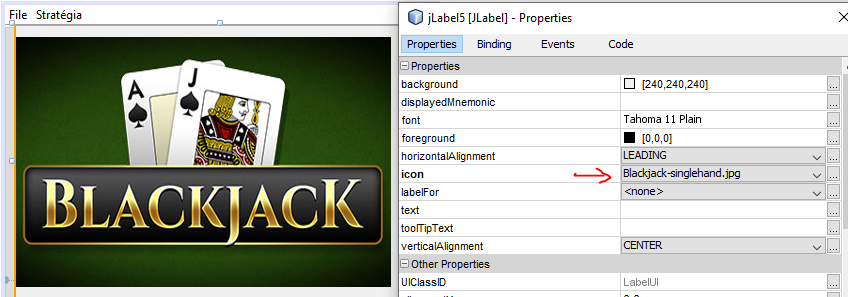
1. A mellékelt képeket helyezd a res mappába, ami a view mappában legyen: 
2. A *Blackjack-singlehand.jpg* képet jelenítsd meg egy Label elemen, közvetlenül a menü alatt:  
   

Kép hozzáadása jLabelhez: 1. importálom: jLabel jobbklikk 🡪 icon … 🡪 Import to project



2. Újra megnyitom az icon felületet, és Image Within Project 🡪 File … 🡪 kijelölöm a használni kívánt képet





1. Állítsd be a form fejlécének szövegét: , illetve a kép alatti rádió gombok közül a „lapok összértéke” legyen kiválasztva, de ez és a menüben lévő választások legyenek egymástól függetlenül beállíthatóak: . *Itt legyen az* ***4. commit***

A bal felső sarokban lévő ikon megváltoztatása:

// a kép elérési útvonala: AntalA\_BlackJack\src\view\res\ikon.jpg

private void setIcon() {

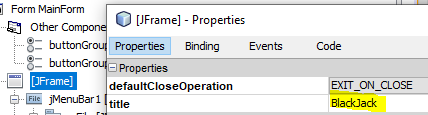
Image icon = Toolkit.getDefaultToolkit().getImage("src\\view\\res\\ikon.jpg");

this.setIconImage(icon);

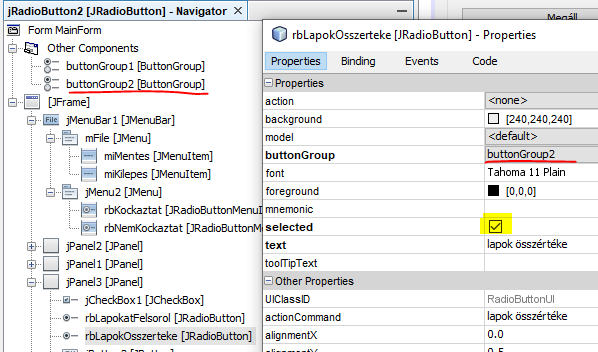
//this.setLayout(null);

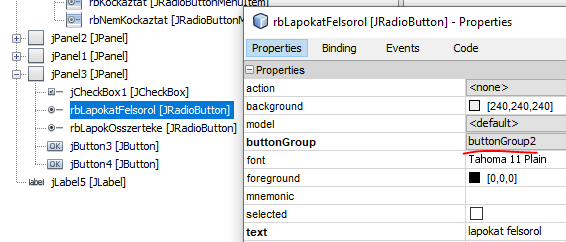
}

A form maga a Jframe, címének beállítása:

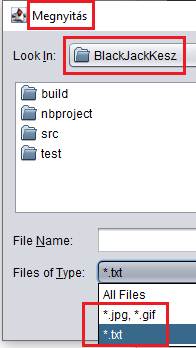


A kép alatti rádiógombokat egy külön buttongrouphoz kell adni:





1. A File menü Mentés eleme és a 6. feladat rádiógombjai alatt található Mentés feliratú gomb nyisson meg egy fájlválasztó ablakot! Ugyanaz a feladtuk, de ne legyen kódduplikálás! A fájlválasztó legyen paraméterezve:
   * + - 1. A címe: Megnyitás
         2. A filter szövege: „\*.jpg, \*.gif” és 2\*.txt”, amiből alapértelmezetten a \*.txt legyen kiválasztva
         3. Az aktuális könyvtár a projekt könyvtára legyen (*név az 1. feladat szerint!!!)*



A *fajlvalasztas* metódus, amelyet a gombokhoz rendelt kattintásesemény hatására hívunk meg:

private void fajlValasztas() {

JFileChooser fc = new JFileChooser();

// a fájlválasztó címének beállítása:

fc.setDialogTitle("Megnyitás");

// a képekre vonatkozó filter beállítása:

FileNameExtensionFilter filterImage = new FileNameExtensionFilter("\*.jpg, \*.gif", "jpg", "gif");

fc.setFileFilter(filterImage);

// a szöveges fájlokra vonatkozó filter beállítása:

FileNameExtensionFilter filterText = new FileNameExtensionFilter("\*.txt", "txt");

fc.setFileFilter(filterText);

// a projektkönyvtár legyen az aktuális könyvtár:

fc.setCurrentDirectory(new File(System.getProperty("user.dir")));

// int result = fc.showOpenDialog(this);

int valasztas = fc.showSaveDialog(this);

//ha történik fájlkiválasztás, akkor az adatait egy változóba mentjük stringként:

if (valasztas == JFileChooser.*APPROVE\_OPTION*) {

File kivFajl = fc.getSelectedFile();

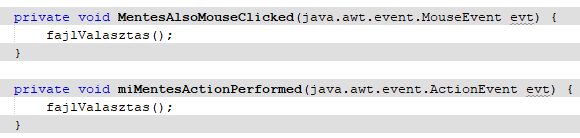
String tartalom = "Fájl neve: " + kivFajl.getPath() + "\nElérése: " + kivFajl.getName();

//a felugro metódus meghívásával a fájl adatait megjelenítjük a felugró ablakban:

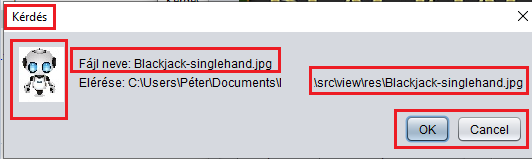
felugro(tartalom);

}

}



1. Ha a 7. feladatban kiválasztunk egy fájlt, akkor a neve és az elérési útja jelenjen meg egy felugró ablakban, ahol a felugró ablak címe, képe (ikon.jpg), szövege, és gombjai az alábbiak:



//felugró ablak

private String felugro(String tartalom) {

//ikon beállítása

Icon ikon = new ImageIcon("src/view/res/ikon.jpg");

//a gomb változóban fogjuk eltárolni a felhasználó által adott választ

int gomb = JOptionPane.showConfirmDialog(rootPane, tartalom, "Kérdés", JOptionPane.*OK\_CANCEL\_OPTION*, JOptionPane.*WARNING\_MESSAGE*, ikon);

String valasz;

//ha az OK gombra kattintunk:

if (gomb == JOptionPane.*OK\_OPTION*) {

valasz = "OK";

//ha a Cancelre kattintunk:

} else {

valasz = "Cancel";

}

return valasz;

}

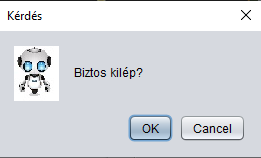
1. Ezt a felugrót egy külön private metódus jelenítse meg, paraméterként átadott szöveggel, visszatérési értéke a felugróban megnyomott gomb legyen. *Itt legyen az* ***5. commit***

Ld. 7-es feladat

1. A kilépési funkciót a File menü Kilépés eleme és a form jobb felső piros X gombja egyaránt megvalósítja. Ugyanaz a feladatuk, ne legyen kódduplikálás és a 9. feladatban megírt saját private metódus jelenítse meg a felugrót, az alábbi specifikáció alapján:

**Kilépés működésének specifikációja:**

Ha nem változtatunk a formon semmit, akkor kérdés nélkül kilép, nincs felugró! Ha a menüben változik a Stratégia vagy a formon a lapok megjelenítése (**tehát bármelyik rádió gomb**), akkor viszont rákérdez a mentésre és csak az OK gomb megnyomásakor lép ki.



Ha változtattunk, akkor onnantól kezdve mindig kérdez, akkor is, ha visszaállítjuk az eredeti állapotot!

private void kilepes(){

//ha változott a form

if(formChanged){

//ha OK-t nyomtunk a felugró ablakon

if((felugro("Biztos kilép?").equals("OK"))) {

System.exit(0);

}

//ha nem változott a form

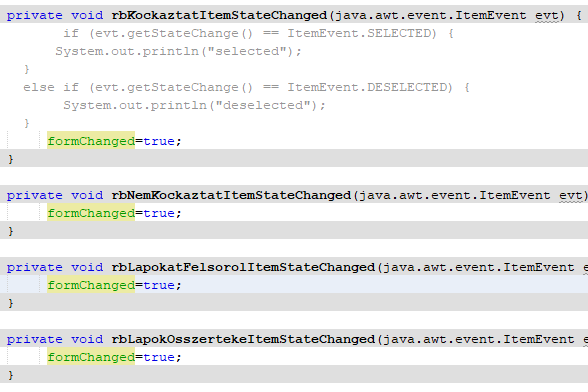
}else{

System.exit(0);

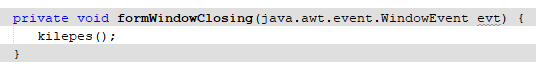
}

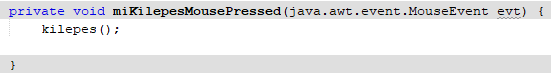
}

Az ***ItemStateChanged*** esemény kiváltódását vagy be nem következését vizsgáljuk a változtatás tényének megállapításához. Amennyiben bekövetkezett az esemény, a formChanged boolean változó értékét igazra állítom.



A kilépés metódus meghívása a megfelelő komponensekre, a megfelelő eseménykezelők alkalmazásával:





<https://www.codejava.net/java-se/swing/preventing-jframe-window-from-closing>

*Itt legyen a* ***6. commit!***

# Grafikus feladat értékelése

|  |  |  |
| --- | --- | --- |
| **Feladat** | **Elérhető** | **Kapott** |
| commit-ok száma és szövege megfelelőek | 1 pont |  |
| a 3, 5, 6 feladat elkészült | 1 pont |  |
| a kép a megfelelő helyről jön | 1 pont |  |
| mentéshez nincs kódduplikálás | 1 pont |  |
| mentés ablak jól paraméterezett | 2 pont |  |
| felugrót függvény kezeli | 1 pont |  |
| kilépés működik, nincs kód duplikáció | 1 pont |  |
| kilépés specifikáció szerinti | 2 pont |  |
| Összesen: | 10 pont |  |

## jegyek:

* elégséges (2) 6 pont
* közepes (3) 7 pont
* jó (4) 8 pont
* jeles (5) 9 ponttól