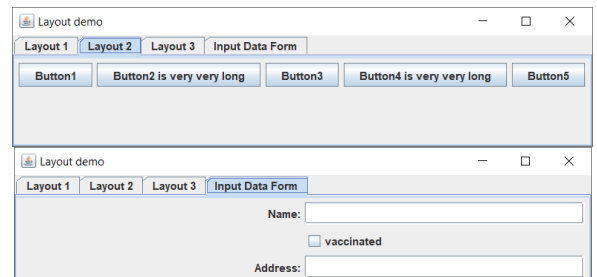
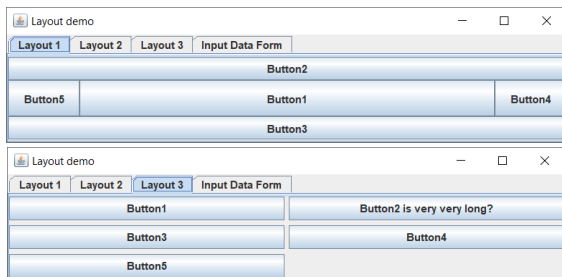


Zadaci za vježbu iz teme 14 (Swing GUI. Obrada dugotrajnih poslova.)

1. Klasa `LayoutDemo` koja je prozor aplikacije `JFrame`, a ujedno i klasa aplikacije, ima konstruktor u kojem se na prozor dodaje kartični kontejner `JTabbedPane` koji sadrži 4 kartice. Na kartice se dodaju paneli (vidi priloženi kod) čije klase trebate napisati. Izgled panela i grafičke komponente koje trebate dodati na panele uz primjenu odgovarajućih upravitelja raspoređivanja (layout manager-a) prikazane su na slikama. Provjerite kako se ponašaju komponente prilikom promjene veličine prozora.

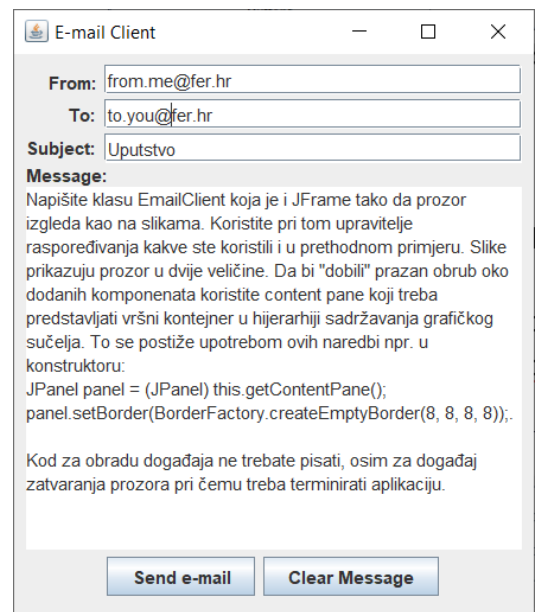
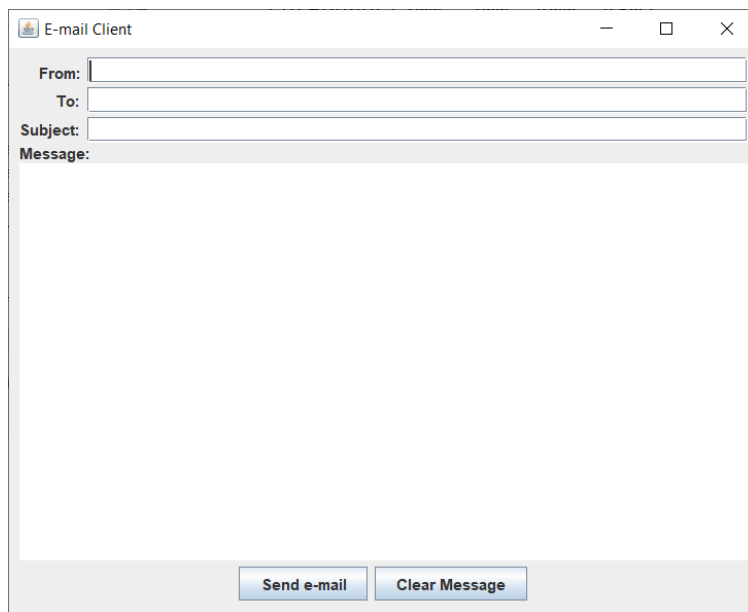
```
public LayoutDemo() {  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    JTabbedPane tp = new JTabbedPane();  
    tp.addTab("Layout 1", new PanelWithLayout1());  
    tp.addTab("Layout 2", new PanelWithLayout2());  
    tp.addTab("Layout 3", new PanelWithLayout3());  
    tp.addTab("Input Data Form", new InputDataForm());  
    add(tp);  
    pack();  
}
```

```
public static void main(String[] args) {  
    SwingUtilities.invokeLater(new Runnable() {  
        @Override  
        public void run() {  
            new LayoutDemo().setVisible(true);  
        }  
    });  
}
```

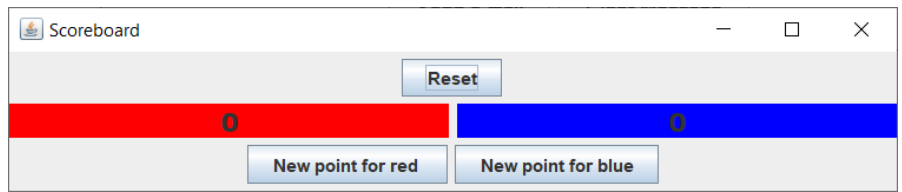
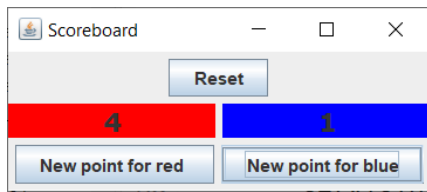


Za izgradnju posljednja dva panela (dvije donje slike) koristite isti layout manager, a na posljednjem panelu kod labela koristite desno pozicioniranje teksta (`SwingConstants.RIGHT`). Kod za obradu događaja ne treba pisati.

2.

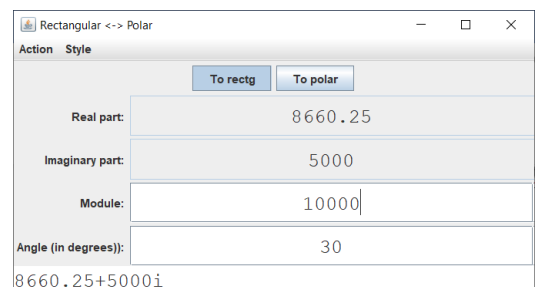
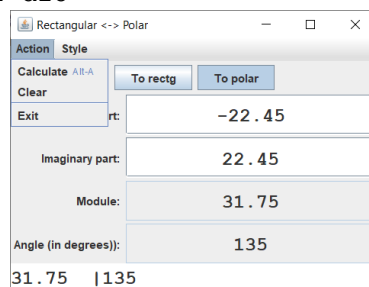
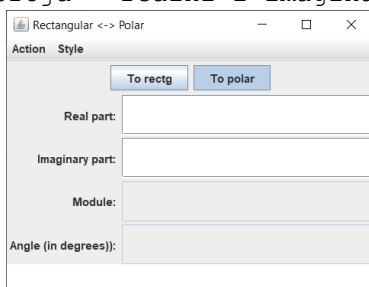


3. Napišite klasu `Scoreboard` koja je i `JFrame` prozor i aplikacija. U konstruktoru klase napišite kod kojim ćete postići izgled GUI-a kakav je prikazan na slici (prikazan je prozor u dvije veličine). U konstruktoru klase spojite "Reset" gumb s osluškivačem kojeg implementirajte pomoću lambda izraza. Prilikom pritiska na taj gumb, bodovi crvenih i plavih se postavljaju na 0. Za obradu događaja pritiska na gumbe "New point for red" i "New point for blue" trebate koristiti jedan jedini osluškivački objekt. Klasu tog objekta implementirajte kao inner klasu od klase prozora. Kod pritiska na "New point for red" uvećava se vrijednost koja je prikazana u crvenoj labeli za 1, a pritiskom na "New point for blue" uvećava se vrijednost koja je prikazana u plavoj labeli za 1.

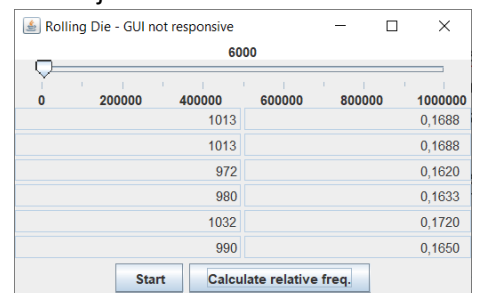
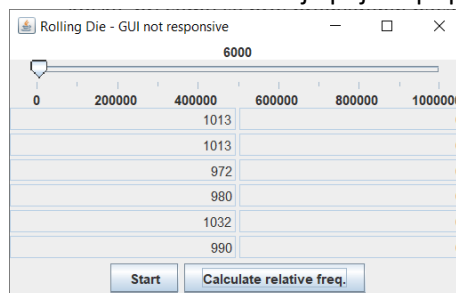
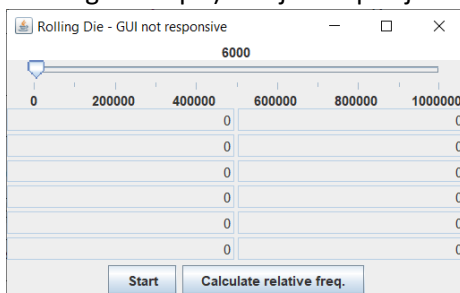


4. Napišite klasu `ComplexNumbersConverter` koja je i `JFrame` i aplikacija. Aplikacija omogućava da se za poznati zapis kompleksnog broja u pravokutnim koordinatama izračunaju polarne koordinate i obratno. U konstruktoru klase napišite kod kojim ćete postići izgled GUI-a kakav je prikazan na slici (prikazan je inicijalni izgled prozora i prozor u dvije veličine). Pri vrhu su dva gumba za prebacivanje `JToggleButton` pomoću kojih se sučelje iz moda za upis pravokutnih koordinata prebacuje u mod za upis polarnih koordinata i obratno. Ova dva gumba je potrebno staviti u grupu `ButtonGroup` kako bi ispravno radili. Kad je pritisnut gumb "To polar" tada se mogu upisati samo pravokutne koordinate, a polarne se izračunavaju pritiskom na naredbu "Calculate" (naredba ima i kraticu `Alt-A`) koja se nalazi u meni-u "Action" i obratno. Po pritisku na "Calculate" se ispisuje i cjeloviti zapis broja u donjem polju za rezultat koje je tipa `JTextArea`. U "Action" meni-u se još nalazi i naredba "Clear" čijim pokretanjem se brišu sve vrijednosti upisane u poljima "Real part", "Imaginary part", "Module", "Angle" i donjem polju za rezultat. Nakon naredbe "Clear" je u meni-u postavljen separator te slijedi naredba "Exit" kojom se terminira rad aplikacije. U meni-u "Style" se nalaze naredbe "BOLD" i "PLAIN" kojima se prikaz brojeva (`Font`) na sučelju mijenja u BOLD odnosno vraća u PLAIN stil (vidi se na srednjoj slici BOLD stil). Za proračun pravokutnih u polarne koordinate i obratno trebate u posebnoj klasi `Calculator` napisati dvije statičke metode s potpisom:

```
public static double[] toPolar(double real, double imag) // vraća dva realna
broja - moduli i kut u stupnjevima
public static double[] toRectg(double module, double angle) // vraća dva realna
broja - realni i imaginarni dio
```



5. Napišite klasu `RollingDieGUIUnresponsive` koja je i `JFrame` i aplikacija tako da postignete izgled sučelja koji je prikazan na slikama. Aplikacija omogućava simulaciju bacanja kocke, pri čemu se broj predviđenih bacanja definira klizačem, a bacanja započinju pritiskom na gumb "Start". Broj predviđenih bacanja se može u grubo očitati s klizača, a precizno je zapisan u bijeloj labeli iznad klizača. Tijekom izvođenja bacanja se mora moći pritiskom na gumb "Calculate relative freq." izračunavati relativna frekvencija pojave pojedinih vrijednosti kocke i pratiti njezina konvergencija prema 0,16666. U prvom stupcu nalaze se `JTextField`-ovi (onemogućen upis) u koje se upisuje broj pojave pojedine vrijednosti kocke od 1-6 tijekom bacanja, a u drugom stupcu su `JTextField`-ovi (onemogućen upis) u koju se upisuju izračunate relativne frekvencije pojave pripadnih vrijednosti.



Obrada događaja:

- Prilikom pomicanja klizača ažurira se vrijednost predviđenog broja bacanja koja je upisana u bijelu labelu na vrhu prozora.
- Obrada događaja pritiska na gumb "Start" obuhvaća sljedeće: gumb "Start" i klizač onemogućeni su za korištenje, brojači vrijednosti kocke i izračunate frekvencije postavljaju se na 0, čita se broj predviđenih

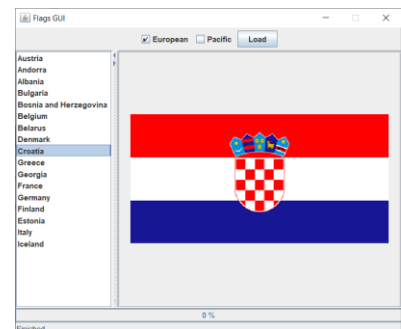
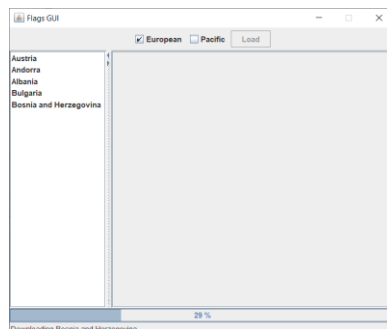
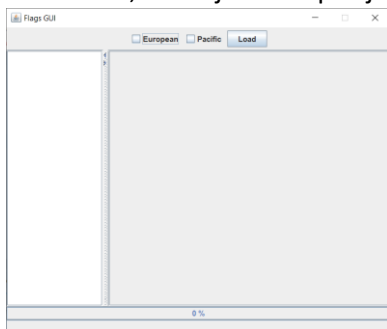
bacanja, poziva se metoda u kojoj je zapisan kod dugotrajnog posla (bacanja) - u iteracijama (bacanjima) se koristi generator slučajnih brojeva te se ažuriraju brojevi pojave vrijednosti kocke upisivanjem u prvi stupac. Na kraju se ponovo omogućava korištenje gumba "Start" i klizača. Važno je napomenuti da se dugotrajan posao izvodi u Event Dispatch Thread-u (EDT) pa je zbog toga tijekom bacanja blokirano grafičko sučelje (sučelje ne odgovara na akcije korisnika).

- Prilikom pritiska na gumb "Calculate relative freq.", ako je dosadašnji broj bacanja bio 0 ne radi se ništa, inače se u `JTextField`-ove u drugom stupcu upisuju omjeri broja upisanog u pripadni `JTextField` iz prvog stupca i do tog trenutka ukupnog broja bacanja.

6. Potrebno je napisati klasu `RollingDieGUIResponsive` kojom se stvara isto grafičko sučelje kao i u prethodnom zadatku. Promjena je u obradi događaja pritiska na gumb "Start". Izvođenje metode dugotrajnog posla (bacanja) se „povjerava“ novostvorenom objektu iz klase `Thread`. Tijekom izvođenja tog thread-a, samo se aktivnosti djelovanja nad komponentama grafičkog sučelja - ažuriranje vrijednosti `JTextField`-ova prvog stupca te ponovno aktiviranje gumba "Start" i klizača prepušta EDT dretvi. To se radi korištenjem metode `SwingUtilities.invokeLater()` za ažuriranje te metode `SwingUtilities.invokeLater()` za reaktiviranje gumba i klizača. Na taj način je EDT dretva oslobođena izvođenja dugotrajnog posla bacanja te GUI može odgovarati na aktivnosti korisnika tijekom bacanja.

7. Potrebno je napisati klasu `RollingDieSwingWorker` kojim se stvara isto grafičko sučelje kao i u prethodnom zadatku. Također, potrebno je definirati klasu `RollingTask` koja je `SwingWorker`. Tu klasu definirajte kao inner klasu od klase `RollingDieSwingWorker`, a preko konstruktora te klase inicijalizirajte atribut `int noOfRolling` na vrijednost predviđenih bacanja koja je upisana u labeli iznad klizača. Posao bacanja kocke ugradite u metodu `doInBackground` koja neka prilikom svakog bacanja objavi posljednje stanje ishoda bacanja u formi `Integer []` veličine 6. Prije objavljivanja možete usporediti izvođenje bacanja uz pomoć metode `Thread.sleep(millisec)`. Metoda ne vraća nikakvu vrijednost. U metodi `process` koja kao argument prima listu stanja ishoda bacanja u proteklom periodu ažurirajte prikaz u `JTextField`-ovima u prvom stupcu, ali samo koristeći posljednje stanje ishoda, a prethodne informacije u listi odbacite. Po završetku posla definiranog u metodi `doInBackground` potrebno je omogućiti korištenje gumba "Start" i klizača. Ispitajte u sklopu kojih dretvi se izvode metode `doInBackground`, `process` i `done` (koristite `SwingUtilities.isEventDispatchThread()` i `Thread.currentThread()` metode).

8. Napišite klasu `FlagsAgain` koja je `JFrame`, ali i klasa osluškivača za `JList` komponentu. U konstruktor klase ugradite naredbe kojima ćete postići izgled grafičkog sučelja prikazanog na slikama. Lijeva slika prikazuje GUI kod pokretanja. Srednja slika je situacija nakon što je odabrana samo European grupa zastava za preuzimanja te je pritisnut Load gumb i preuzimanje je započelo. Lijevo je prikazano sučelje kad smo u `JList`-i odabrali zastavu Croatia za prikaz. Na gornjem dijelu su smješteni gumbi. Na desnoj strani dijeljenog panela nalazi se labela u kojoj se prikazuje slika. U donjem dijelu je smješten pokazivač napredovanja, a ispod je polje za tekst u kojem je onemogućen unos teksta, a u kojem se ispisuju poruke o trenutnom statusu.



Definirajte i klasu `worker` koji je `SwingWorker` kao inner klasu od klase `FlagsAgain`. `doInBackground` ispituje status checkbox-ova (checkbox omogućava da se odabere geo grupa zemalja čije zastave se žele preuzeti). Ako ni jedan box nije checked metoda vraća string "Nothing to load" koji se ispisuje u liniji statusa. Inače, započinje s preuzimanjem slika zastava (URL za preuzimanje slike poznat vam je iz vježbe 10). Ako preuzimanje nije moguće, `doInBackground` prestaje s radom i vraća string "Problems with network" koji se ispisuje u poruci na liniji statusa. Ako je sve u redu preuzima se slika po slika, a naziv zemlje čija se slika zastave trenutno preuzima objavljuje se, a

također se postavlja trenutna vrijednost napredovanja. Objavljen naziv zemlje se prikazuje na liniji statusa npr. kao "Downloading Croatia". Po završetku preuzimanja metoda vraća string "Finished" koji se prikazuje na liniji statusa. Zatim se omogućava upotreba gumba "Load", pokazivač napredovanja se postavlja na vrijednost 0 te se selektira prva vrijednost (država) u listi. Obrada događaja:

- Po pritisku na "Load" gumb – onemogućava se pritiskanje gumba, prazni se sadržaj liste, kreira se worker objekt i na njega se spaja osluškivač promjene svojstva preko kojeg se pokreće pokazivač napredovanja i na kraju se pokreće worker.
- Kad se promijeni vrijednost u listi treba ažurirati sliku u labeli za prikaz slike.

Rješenje zadataka dostupna su na:

<https://github.com/FER-OOP/Lectures/tree/master/Exercises/Homework-14>