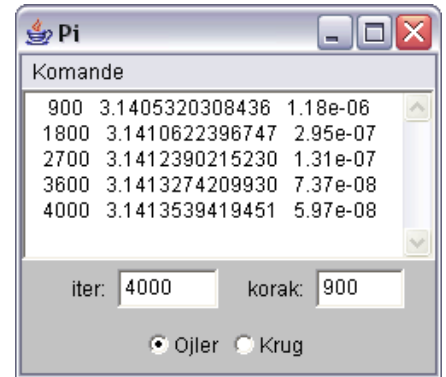


Лабораторијска вежба број 3 из Објектно оријентисаног програмирања II

1) Написати на језику *Java* следећи пакет типова:

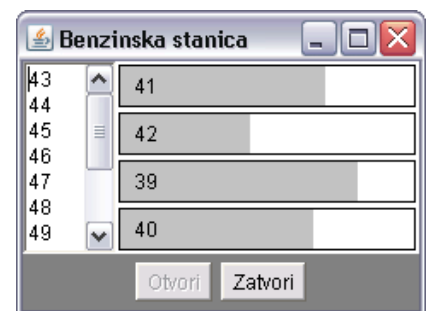
- **Приказивач** на придруженој вишередној текстуалној компоненти графичке корисничке површи може да прикаже један текстуални (*String*) податак. Може да се постави садржај текста на нову вредност и да се на постојећи садржај додаје још текста.
- Активан апстрактан **актер** одређује вредност броја π у задатом броју итерација. После сваког задатог броја итерација, придруженом приказивачу се додаје један ред текста који садржи укупан број итерација, тренутну приближну вредност и апсолутну вредност разлике у односу на претходну итерацију. Рад актера може да се прекине и превремено.
- **Ојлер** је актер који за рачунање броја π користи приложену Ојлерову формулу.
- **Круг** је актер који вредност броја π одређује тако што генерише тачке са случајним координатама у опсегу од 0 до 1 и броји колико њих пада унутар четвртине круга, са центром у координатном почетку и јединичним полупречником, у првом квадранту. Ако од укупно n тачака m пада унутар тог исечка, узима се да је $\pi \approx 4m/n$.
- **Програм** коришћењем графичке корисничке површи према слици приказује рад претходних класа. У менију *Komande* налазе се команде *Kreni*, *Prekini* и *Zavrsti*.

$$\pi \approx \sqrt{6 \sum_{k=1}^n \frac{1}{k^2}}$$



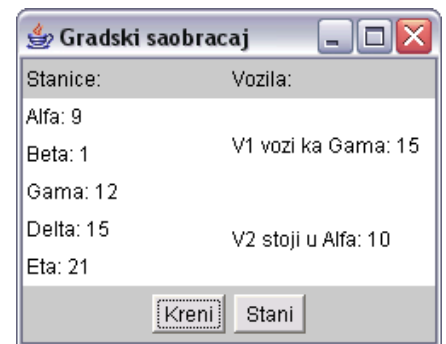
2) Написати на језику *Java* следећи пакет типова:

- **Аутомобил** има јединствен, аутоматски генерисан целобројан идентификатор, задат капацитет резервоара и тренутну количину горива. Сви подаци могу да се дохвате. Почетна количина горива је случајна вредност између 10% и 30% капацитета резервоара. У аутомобил може да се сипа задата количина горива. Грешка је ако се резервоар препуни (тада се резервоар напуни и пријави грешка).
- **Приказивач** предвиђа приказивање података о задатом аутомобилу.
- Активан **аутопут** има задату бензинску станицу. У случајним временским интервалима од 0,5 s до 1 s ствара по један аутомобил капацитета резервоара 50 l који додаје тој станици. Може да се прекине рад аутопута када се прекида и рад његове станице.
- Активна **пумпа** се ствара за задату бензинску станицу. Пумпи може да се придружи задати приказивач у произвољном тренутку. Пумпа циклички дохвата по један аутомобил из реда своје станице и сипа му потребну количину горива до пуног резервоара, брзином од 1 l на сваких 100 ms. Завршетак сипања дојави бензинској станици. Стање аутомобила приказује на придруженом приказивачу после сваке промене количине горива.
- Бензинска **станица** има четири пумпе и ред за чекање за највише 20 аутомобила. Станица може да се отвори и затвори, може да се прекине њен рад, да јој се дода задати аутомобил, да се извади један аутомобил из њеног реда и да јој се дојави завршетак пуњења једног аутомобила. Ако је станица затворена или је ред пун, додавање аутомобила се занемари. Ако је ред празан, при узимању се сачека да се појави неки аутомобил. Приликом затварања, прекида се чекање возила у реду и чека се да се заврши сипање горива које је у току. Приликом прекидања прекида се и рад свих пумпи станице.
- **Графички приказивач** је приказивач и компонента типа *Canvas* на којој се, према слици, исцртава светлосива трака чија је дужина пропорционална количини горива у придруженом аутомобилу. На почетном делу те траке исцртава се идентификатор аутомобила.
- **Графичка станица** је станица која садржи графичку плочу (*Panel*) која може да се дохвати. Уз леву ивицу, према слици, приказује идентификаторе аутомобила који чекају у реду за чекање после сваке промене садржаја реда. Преостали део плоче попуњава са по једним графичким приказивачем за сваку садржану бензинску пумпу.
- **Програм** на графичкој корисничкој површи према слици приказује стање једне бензинске станице. Не сме да се отвори отворена, односно затвори затворена станица.



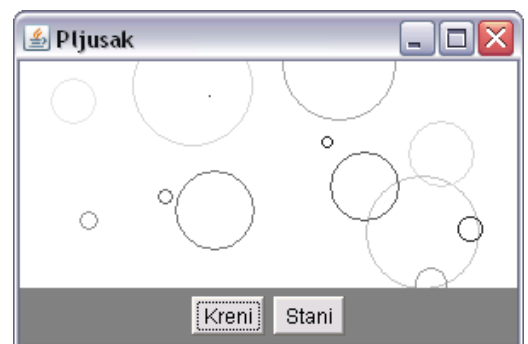
3) Написати на језику *Java* следећи пакет типова:

- Апстрактан активан **актер** понавља одређену апстрактну радњу. При свакој промени стања приказује свој текстуални опис на задатој графичкој компоненти типа натписа (Label). Рад актера може да се заустави, да се настави даље и да се дефинитивно прекине.
- **Станица** је актер који генерише путнике и има своје име. У станицу у случајним временским интервалима између задате најкраће и најдуже вредности стиже по један путник. Може да се дохвати име и број путника у станици и да се смањи број путника за задати број. Текстуални опис станице садржи име станице и број путника који чекају.
- **Возило** је актер који има своју ознаку и капацитет. Креће се наизменично од почетка до краја и назад дуж линије градског саобраћаја. Вожња до наредне станице траје случајно време између задатог најкраћег и најдужег времена. Задржавање у станици траје задато фиксно време. Тада прво из возила излази случајан број путника од 0 до 100% и после улази највећи могући број путника од оних који чекају у станици. Текстуални опис возила садржи ознаку возила, назив станице ка којој се возило креће или у којој се налази и број путника у возилу.
- **Линија** градског саобраћаја садржи извршан број станица и возила које ствара на основу задатог низа имена станица, заједничког најмањег и најдужег времена између долазака путника, низа ознака возила, заједничког капацитета, времена задржавања у станицама и најкраћег и најдужег времена вожње између две станице. Својим актерима придружује по један графички натпис које распоређује на задате две графичке плоче (Panel), за станице на једну и за возила на другу. Може да се дохвати број станица и станица са задатим редним бројем и да се управља радом свих актера у линији (крени, стани, прекини).
- **Програм** на графичкој корисничкој површи симулира рад линије градског саобраћаја помоћу 5 станица у које путници стижу после сваких 0,5 до 1 s и два возила капацитета 40 путника, временом задржавања од 1 s и временом вожње између две станице од 2 до 4 s.



4) Написати на језику *Java* следећи пакет типова:

- **Кап** кише има задате целобројне координате (x,y) и реалну тежину q .
- **Приказива** ствар може да се исцртава на задатом графичком контексту (Graphics).
- Активан приказив **талас** се на задатој бари (видети ниже) исцртава у облику концентричних кругова са задатим центром (x,y) и корацима промене полупречника Δr и промене боје Δb . Талас је у почетку црн и временом бледи до беле боје, после чега нестане из баре. Сваких 40 ms полупречник исцртавањег круга се повећава за Δr , а све три компоненте боје (црвена, зелена и плава) се повећавају за Δb (почев од 0 док не постану 1). Нит таласа може да се прекине.
- **Површ** је графичка компонента (Canvas). Може да се дохвати ширина и висина површи и да се региструје пад капи на површ.
- Активна површина **баре** је површ која може да садржи произвољан број таласа које ствара приликом пада капи. Центар таласа се поклапа са местом пада капи. Корак промене полупречника таласа је $\Delta r = 0,05 \cdot q$, а боје $\Delta b = 0,05/q$, где је q тежина капи. Бара сваких 40 ms исцртава "живе" таласе (који имају траг), а одстрањује таласе који више нису. Нит баре може прекине и тада се прекидају и нити свих затечених таласа.
- Активан **облак** у случајним временским интервалима од задате најмање до задате највеће вредности пушта једну кап на задату површ. Створене капи имају случајне координате унутар габарита површи и случајну тежину између 2 и 8. Нит облака може да се привремено заустави, настави даље и да се трајно прекине.
- **Програм** на графичкој корисничкој површи према слици приказује понашање једне баре и облака. Време између два пуштања капи је од 200 до 500 ms.



НАПОМЕНЕ:

- Потребно је решавати искључиво задатак чији се број добије на почетку вежбе.
- За израду лабораторијске вежбе, на располагању је **120** минута.
- Дозвољено је коришћење оригиналних књига и збирки задатака (не фотокопија) и **Подсетника за АWT**.
- Није дозвољено коришћење унапред припремљених решења у било којем облику. Студент који користи унапред припремљена решења, биће удаљен уз анулирање поена на свим лабораторијским вежбама.
- У току израде лабораторијске вежбе, дежурни може студентима да постаља питања у вези њихових решења, што може утицати на број освојених поена на лабораторијској вежби.
- Студент може бити позван на накнадну одбрану рада, која може да утиче на број поена. Непојављивање студента на одбрани или показивање вишег степена неразумевања сопственог решења повлачи анулирање поена на свим лабораторијским вежбама.
- Текст решења распоредити у *.java датотеке (не стављати цело решење у једну датотеку) на диску L.
- Оцене радова биће објављене на *Web*-у на адреси: home.etf.rs/~kraus/ (одреднице: *настава* | <име предмета> | *оцене* | *колквијуми*).