

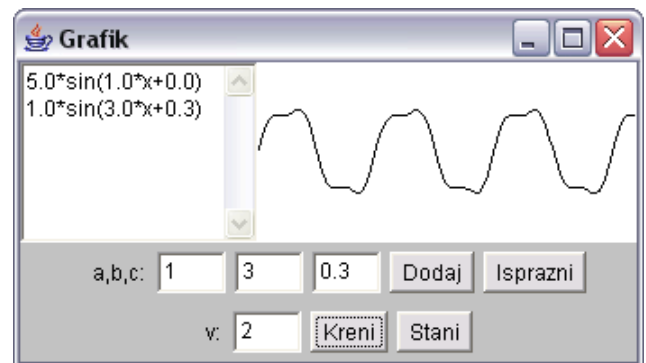
Лабораторијска вежба број 3 из Објектно оријентисаног програмирања II

У сваком задатку:

- Грешке пријављивати изузецима типа класа које садрже текст поруке.

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

- **Функција** предвиђа израчунавање вредности реалне функције $f(x)$ и састављање текстуалног описа функције.
- **Синусоида** је функција која на основу задатих реалних параметара a , b и c израчунава $a \cdot \sin(b \cdot x + c)$. Текстуални облик је $a \cdot \sin(b \cdot x + c)$.
- **Збир функција** је функција која може да садржи произвољан број функција. Ствара се празан после чега се функције додају једна по једна. Могу одједном да се избаце све функције. Вредност је једнака збиру вредности свих садржаних функција за дато x . Текстуални опис садржи текстуалне описе садржаних функција, једна функција по реду.
- Активан **график** је графичка компонента која сваких 10 ms у задатом броју тачака црта задату функцију $f(x+v \cdot t)$, где је t [s] време протекло од почетка цртања, а v параметар који може да се поставља. Приликом стварања цртежа задају се опсеги координата који се пресликају на целу компоненту и који касније не могу да се промене, тако да изглед криве зависи од актуелне величине компоненте. Рад графика може да се заустави, да се настави даље и да се дефинитивно прекине.
- **Програм** на графичкој корисничкој површи према слици може да саставља један збир функција и да управља цртањем те функције. Опсег координата на цртежу дуж обе осе је ± 10 . Цртање се врши у 200 тачака. Промена параметра v има тренутно дејство. У компоненти у горњем левом углу се само приказује текстуални опис функције.

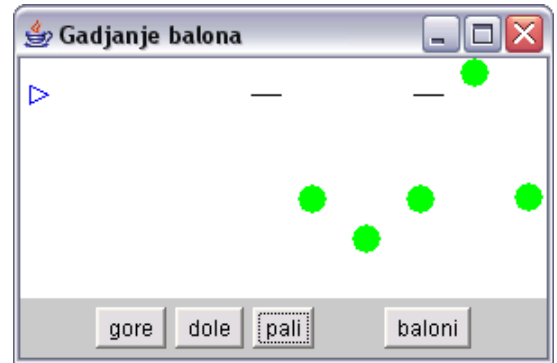


2) Написати на језику *Java* следећи пакет типова:

- Апстрактна **фигура** у равни предвиђа дохватање тренутних координата и исцртавање фигуре на графичкој сцени.
- Активна **сцена** је графичка компонента која може да садржи произвољан број фигура које исцртава сваких 40 ms. Ствара се празна, после чега се фигуре могу појединачно додавати. Из сцене може да се избаци одабрана фигура. Фигуре у сцени могу да се дохватају једна по једна, по редоследу додавања.
- **Стрела** је фигура која се исцртава као водоравна дуж задате боје и дужине. Ствара се са задатим координатама врха после чега се креће задатом реалном брзином удесно. Стрела се избацује из сцене када напусти видљиви део сцене.
- **Лук** је фигура која се исцртава задатом бојом унутар кавдрата задате дужине ивица тако да једно теме показује удесно (\triangleright). Ствара се са задатим координатама средине леве ивице после чега може да се помера за унапред задати корак нагоре или надолу, али само до ивице сцене. Може да испали стрелу задате дужине задатом реалном брзином.
- **Балон** је фигура која се исцртава као попуњен круг задате боје и пречника. Ствара се са задатим координатама центра после чега се креће задатом реалном брзином нагоре. Кад центар балона пређе горњу ивицу сцене, поново се појављује на доњој ивици на истој усправној линији као и пре. Кад балон буде пробушен неком стрелом, промени боју у другу унапред задату боју после

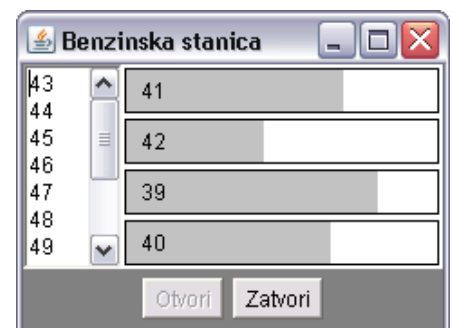
чега се избацује из сцене. Сматра се да је стрела пробушила балон кад врх стреле доспе у круг балона.

- **Игра** на графичкој корисничкој површи према слици поставља један лук плаве боје у тачки (5, 20) у односу на горњи леви угао сцене унутар квадрата дужине ивица $10p$ (*pixel*). Лук може да се помера нагоре и надоле с кораком $10p$. Стреле црне боје дужине $16p$ испуљају се брзином $0,2 p/ms$. На захтев, одједном се додаје 5 зелених балона пречника $16p$ на доњој ивици сцене од десне ивице улево на међусобном размаку од по $30p$. Балони се крећу брзинама између $0,06 p/ms$ и $0,08 p/ms$. Кад буду пробушени, боја им се мења у црвено.



3) 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. Пумпа 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 се 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.

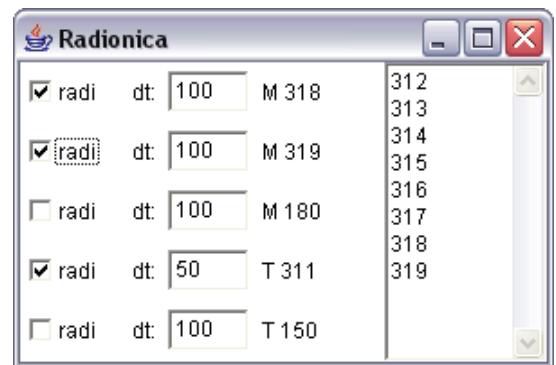


4) Написати на језику *Java* следећи пакет типова:

- **Производ** има јединствен, аутоматски генерисан идентификациони број. Текстуални облик производа садржи само његов идентификациони број.
- **Складиште** производа може да смести задати број производа. Ствара се празно, после чега може да се додаје и узима по један производ. При покушају стављања у пуно складиште односно

узимања из празног складишта нит извршиоца радње се привремено блокира. Текстуални облик складишта састоји се од низа текстуалних облика садржаних производа, један производ по реду.

- Апстрактан **механизам** предвиђа извођење неке радње.
- **Машина** је механизам чија се радња састоји од стварања једног производа и његовог стављања у складиште које је задато приликом стварања машине. Текстуални облик садржи слово **М** и текстуални облик производа који је управо стављен у складиште.
- Покретна **трака** је механизам чија се радња састоји од узимања једног производа из складишта које је задато приликом стварања траке. Текстуални облик садржи слово **Т** и текстуални облик производа који је управо узет из складишта.
- Активан **мотор** понавља радњу једног механизма, који се задаје приликом стварања мотора, у случајним временским интервалима $\Delta t \pm 20\%$ ms (на почетку је $\Delta t = 100$ ms и може да се промени за време живота мотора). Рад мотора може да се привремено заустави, да се настави и да се дефинитивно прекине.
- **Графичко складиште** је складиште које се иницијализује графичким текстуалним простором (TextArea) на коме приказује сваку промену стања складишта.
- **Графички мотор** је мотор који се иницијализује графичком плочом (Panel) коју попуњава компонентама тако да може зауставити и наставити рад мотора, променити временски интервал Δt (промена има ефекта одмах) и приказати текстуални облик придруженог механизма.
- **Радионица** је програм који на графичкој корисничкој површи према слици приказује рад с једним складиштем капацитета 20, три машине и две покретне траке.



НАПОМЕНЕ:

- а) Потребно је решавати искључиво задатак чији се број добије на почетку вежбе.
- б) За израду лабораторијске вежбе, на располагању је **120** минута.
- в) Дозвољено је коришћење оригиналних књига и збирки задатака (не фотокопија) и **Подсетника за АWT**.
- г) 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.
- д) 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.
- ђ) 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.
- е) Текст решења распоредити у *.java датотеке (не стављати цело решење у једну датотеку) на диску L.
- ж) Оцене радова биће објављене на Web-у на адреси: <http://rti.etf.bg.ac.rs/rti/ir2002/index.html>