

# Programowanie obiektowe – zadanie oceniane (4/4)

15-06-2018

Należy wysłać rozwiązanie w 90 minut po rozpoczęciu.  
Liczba punktów do zdobycia: 25.

## 1. Wysyłanie rozwiązania

Należy spakować projekt w archiwum zip i nadać mu nazwę: login.zip

Należy wysłać email:

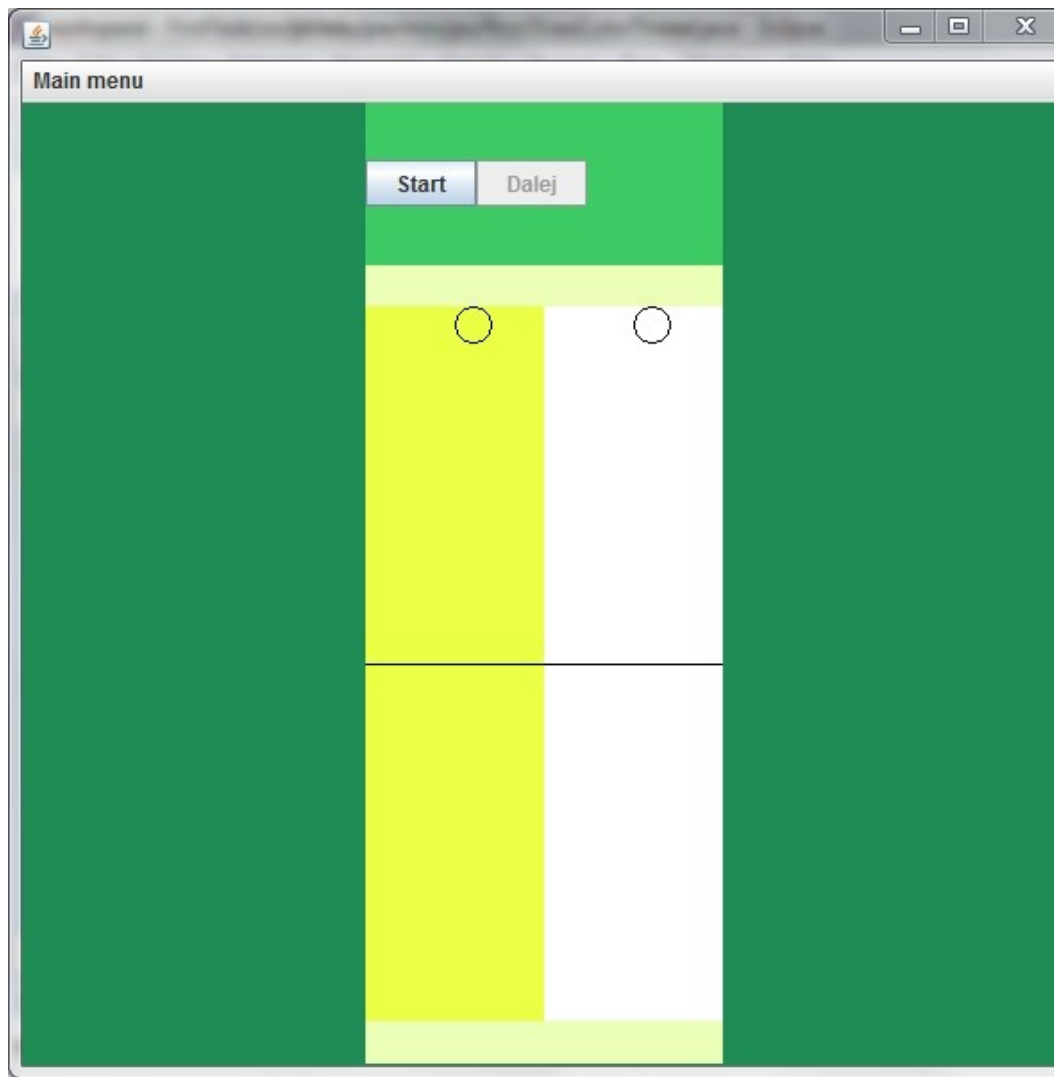
To: [bednarzm@student.mini.pw.edu.pl](mailto:bednarzm@student.mini.pw.edu.pl)

Subject: [PO] Zadanie4/4 2018

Załączniki: login\_studenta.zip

## 2. Opis ogólny

Klient zamówił program, który symuluje wyścigi dwóch kółek.



### 3. Opis szczegółowy

#### Część A (15 pkt).

**Celem tego etapu jest stworzenie ramki (podobnej) do widocznej na zdjęciu.**

Obszar roboczy aplikacji zawiera następujące elementy:

- Tło (kolor zielony)
- Panel główny (pionowy) widoczny w środkowej części, zawierający:
  - ◆ Panel o jasnozielonym tle z dwoma przyciskami: "Start" i "Dalej"
    - Przycisk "Start"
    - Przycisk "Dalej" (o ile nie był przyciśnięty przycisk "Start" uruchamiający wątki, i kółka w ramach części "C" nie dotarły do półmetka) jest dalej nieaktywny.
  - ◆ Panel dolny o limonkowym kolorze posiadający:
    - dwa panele rysujące (wąskie pionowe) umieszczone jeden obok drugiego.

Panel rysujący jest wysoki na 400 pikseli i szeroki na 100. Na górze znajduje się kółko, symbolizujące coś, co będzie startowało w zawodach. Powinien być w stanie (w zależności od ustawionych parametrów) narysować okrąg na ustalonej wysokości po uprzednim odświeżeniu widoku (np. wystawić metodę publiczną `rysuj(int wysokość)`). Kółko jest rysowane na środku toru.

Wątek z części "B" będzie cyklicznie wywoływał tą metodę i tym samym przesuwiał kółko z góry na dół symulując jego ruch. W połowie drogi znajduje się półmetek – pozioma linia, przy której oba kółka będą się zatrzymywały (jeżeli Student podejmie się wykonania części "C"). Zatem w ramach części "A", panel rysujący powinien umożliwiać narysowanie kółka na zadanej wysokości.

W momencie najechania myszką na panel (wskaźnik znajduje się nad jego obszarem roboczym), jego tło staje się białe. W przypadku, gdy mysz opuściła ten rejon (wskaźnik nie znajduje się nad panelem), kolor tła wraca do wartości oryginalnej. Widoczny na zdjęciu prawy panel jest biały, ponieważ znajduje się nad nim wskaźnik myszy.

Podpowiedź: przy nadpisywaniu metody `paint`, należy koniecznie (jako pierwszą linijkę) wstawić ten kod: `super.paintComponent(graphics);` - gdzie `graphics` jest parametrem przekazywanym. Należy też pamiętać o metodzie `revalidate()`, po dodaniu nowych komponentów do ramki (pola tekstowe, przyciski, itp. )

Wymagania:

- Powinna zostać stworzona oddzielna klasa dla panelu rysującego.
- Dwa panele rysujące, to dwie instancje klasy o której wspomniano w w/w punkcie.
- Wspomniana reakcja paneli na najechanie na niego wskaźnikiem myszy

#### Część B (5 pkt).

**Celem niniejszego fragmentu jest oprogramowanie animacji.**

Użytkownik poprzez przyciśnięcie przycisku "Start":

- Powoduje stworzenie dwóch wątków z których każdy zarządza przesuwaniem kółka w dół na swoim panelu (wywołuje metodę na panelu, która rysuje kółka na takiej a takiej wysokości)

- Każdy z dwóch wątków:
  - ◆ Wymusza odmalowanie panelu (wyczyszczenie) – tak żeby poprzednio namalowane kółko było już zamalowane nową wersją obrazu (repaint())
  - ◆ Wykonuje poniższą sekwencję 400 razy (żeby przesunąć kółko piksel po pikselu w dół panelu, który ma wysokość 400 pikseli)
    - Losuje czas na jaki przysnie (0-100 milisekund)  
Jeżeli wylosował powyżej 90-ciu, to do tego czasu dodawane jest jeszcze 300 milisekund. Jeśli wylosował czas 70-90 milisekund, to do tego czasu dodawane jest dodatkowe 100 milisekund. Jeśli wylosował czas poniżej 20-tu, to wstawiana jest wartość 1 milisekundy.
    - Zasypia na skalkulowany okres czasu.
    - Po wybudzeniu przemalowuje panel i rysuje kółko na aktualnej wysokości.
  - ◆ Po zakończeniu pętli wątki kończą swój żywot.

Wymagania:

- Wątek powinien być reprezentowany przez nieanonimową klasę
- Synchronizacja powinna być zastosowana tam, gdzie jest potrzebna (o ile jest potrzebna)

## Część C (5 pkt).

### Oprogramowanie półmetka.

Jeżeli wyżej wymienione kółka (wątki) znajdują się w połowie drogi (200-setna iteracja), zatrzymują się tak, że jeden wątek czeka, aż drugi dojedzie do tego poziomu. Jeżeli wolniejszy wątek dotrze do półmetka, powoduje odblokowanie przycisku "Dalej" i przechodzi w stan wstrzymania na tym poziomie. Przyciśnięcie tego przycisku powoduje zwolnienie wątków z semafora, tak żeby mogły dokończyć pętlę (przesuwać kółka do samego dołu).

Wymagania:

- Wątek powinien być reprezentowany przez nieanonimową klasę
- Synchronizacja powinna być zastosowana tam, gdzie jest potrzebna (o ile jest potrzebna)

### Dodatek dla chętnych (za dodatkowe punkty):

Pokręcanie rolką myszy nad jednym z paneli rysujących, powoduje zmianę średnicy rysowanego kółka w następnych iteracjach. W jedną stronę średnica się powiększa, a w drugą pomniejsza. Oczywiście nie może być większa niż 50 pikseli i mniejsza niż jeden piksel.