WOJSKOWA AKADEMIA TECHNICZNA

im. Jarosława Dąbrowskiego

PROGRAMOWANIE

WSPÓŁBIEŻNE



SPRAWOZDANIE Z

PROJEKTU

Temat : Symulacja ruchu na stoku narciarskim

Prowadzący: dr inż. Jarosław Rulka

Wykonał: Mikołaj Szabrański

Grupa: WCY19IY1S1

Data wykonania: 18.06.2021

1. Treść zadania

Podstawowe wymagania:

1. liczba procesów sekwencyjnych powinna być dobrana z wyczuciem tak, aby zachować czytelność interfejsu i jednocześnie umożliwić zobrazowanie reprezentatywnych przykładów,
2. kod źródłowy programu musi być tak skonstruowany, aby można było „swobodnie” modyfikować liczbę procesów sekwencyjnych (za wyjątkiem zadań o ściśle określonej liczbie procesów),
3. graficzne zobrazowanie działania procesów współbieżnych,
4. odczyt domyślnych danych wejściowych ze sformatowanego, tekstowego pliku danych (xml, properties, inne),
5. możliwość modyfikacji danych wejściowych poprzez GUI.

Problem do rozwiązania: symulacja ruchu na stoku narciarskim.

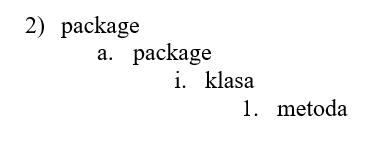
1. Założenia:
   1. 3 trasy zjazdowe obsługiwane przez 3 wyciągi:
      1. wyciąg ze stacji bazowej na szczyt;
      2. wyciąg ze stacji bazowej do stacji w połowie stoku;
      3. wyciąg ze stacji w połowie stoku na szczyt;
   2. Każdy wyciąg ma określoną pojemność i częstotliwość kursowania;
   3. Po upływie pewnego czasu dany wyciąg jest zamykany w celu prac serwisowych.
2. Opis problemu
3. Przyjęte założenia
   1. Narciarze wjeżdżają „na górę” i zjeżdżają na nartach. Nikt nie wraca kolejką górską na dół.
   2. Jeśli liczba osób w kolejce jest nie mniejsza od pojemności wyciągu to wsiada do niego dokładnie tyle osób ile sugeruje pojemność. Jeśli osób w kolejce jest mniej niż może jechać wyciągiem maksymalnie to w takim przypadku jadą one wszystkie.
4. Współdzielone zasoby
5. Wyróżnione punkty synchronizacji
   1. Metoda add() w klasie Queue
   2. Metoda subNumOfPpl (int sub) w klasie Queue
6. Wykaz obiektów synchronizacji
   1. Queue.numOfPpl – liczba osób w kolejce do wyciągu
   2. Queue.current – liczba osób jadących obecnie kolejką linową
7. Wykaz procesów sekwencyjnych

Ropeway.run():

* 1. wsiadanie do wyciągu
  2. rozpoczęcie ruchu wyciągu po odczekaniu ustalonego czasu
  3. jazda na „górę”
  4. zatrzymanie się wyciągu
  5. wysiadanie i odczekanie
  6. powrót na „dół”

1. Struktura projektu

Legenda:



Struktura:

1. com
   1. sample
      1. Controller

Metody:

* + - 1. fromFile – odzczyt wartości danych z pliku
      2. initialize – ustawienie wartości początkowych progressBar’ów oraz utworzenie i uruchomienie wątków
      3. bazaSzczyt – akcja przycisku inkrementująca liczbę osób w kolejce do wyciągu ze stacji bazowej na szczyt
      4. bazaSrodek - akcja przycisku inkrementująca liczbę osób w kolejce do wyciągu ze stacji bazowej do stacji w połowie stoku
      5. srodekSzczyt - akcja przycisku inkrementująca liczbę osób w kolejce do wyciągu ze stacji w połowie stoku na szczyt
    1. Main

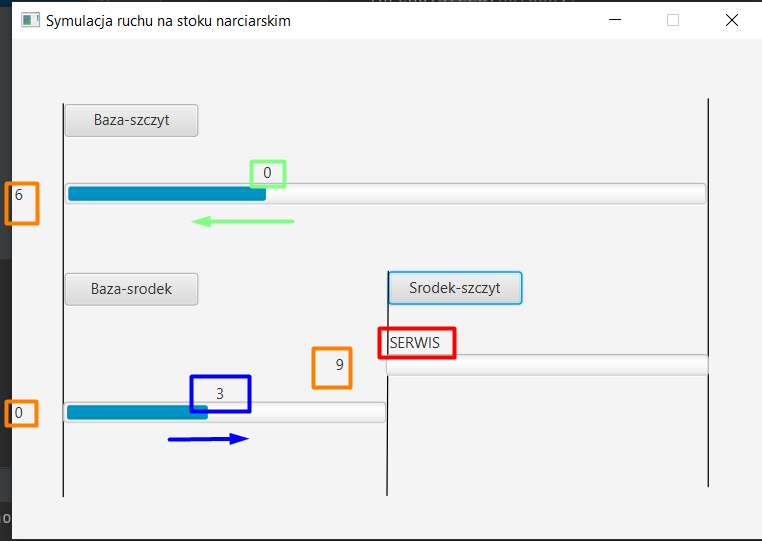
Metody:

* + - 1. start – utworzenie okna GUI
      2. main – launch(args)
    1. sample.fxml - wygląd początkowy GUI
  1. Stok
     1. Queue

Metody

* + - 1. update – aktualizacja liczby osób w kolejce w GUI
      2. update2 - aktualizacja liczby osób w wyciągu w GUI
      3. service – wyświetlenie informacji o serwisie
      4. add – inkrementacja liczby osób w kolejce
      5. subNumOfPpl – pomniejszenie liczby osób w kolejce po tym jak część z nich wsiadła do wyciągu
    1. Ropeway
       1. run – odpowiada za poruszanie się kolejek oraz ich oczekiwanie i przerwę na serwis

1. GUI



Liczba osób w kolejce do wyciągu.

Komunikat o serwisowaniu wyciągu.

Nad wyciągiem wyświetla się liczba osób które nią jadą:

Jeśli kolejka jedzie w górę („w prawo”) to jest to liczba wynikająca z zadania, a jeśli w dół („w lewo”) to jest to 0 zgodnie z przyjętym założeniem.

1. Listing programu