
Advanced Programming

Continuous Assessment Laboratory CAL9

Final Assignment – May 2014

Children's Playground

We want to simulate the behaviour of some **children** (threads) in a playground. There is a **swing** with three sits, a **slide** (it can only be used by a child at a time), and a **carrousel** (merry-go-round) with five seats. Children in the playground change continuously between these games at random. When a child goes to the swing, the child looks if there are seats available and if so, (s)he swings for some time, otherwise, if all seats are busy, the child waits until a seat is free. In the slide, children go up the ladder, get ready and slide. If the slide is busy, children keep waiting in a queue to use it.

When the park closes, children finish playing in the last attraction (swing, slide, carrousel) or waiting in the queue and will not go to any other attraction.

A **monitor** will be checking that no children older than 7 years old use the slide.

A **supervisor** will be in charge of checking and closing the playground.

We want to create a concurrent Java application that simulates this children's playground (queues, times, synchronisation, etc.) that can be controlled from other distributed Java applications from the same or other computers. To do so, we need to represent a model with the following components:

1. Children (threads) identified by an integer from 11 to 60 (it must be known where they are at all times).
2. Slide (shared resource) in which only a child can be using it.
3. Swing (shared resource) in which three children can use it at the same time.
4. Carrousel (shared resource) which will only start when there are 5 children on the carousel. It will go round and round for 5 seconds. Then children will leave the carousel and the cycle starts again.

There should be two queues for each attraction, one showing children using it and another with children waiting. The system is composed of three programs (main classes) that will communicate between them using sockets or RMI.

The module **Playground** ("server") must provide a GUI showing the status of all 50 children. Children's queues must be implemented using the **ArrayList** class. It must include two buttons to **stop** and **resume** the system.

A module **Monitor** ("client") must also have a GUI with a button *Check Slide* which will be used to check if the child is more than 7 years old. If so, the child must get off the slide (without sliding). The

child's age will be shown in a text field. This module will communicate with the Playground module via RMI.

A module **Supervisor** ("client") will be in charge of showing the number of children in each queue at some point in time. A GUI will show three text fields showing the data as well as two buttons **Refresh** and **Close** to refresh the information and close the playground respectively. Supervisor will communicate with the server (Playground) through sockets.

Rules:

1. The main program will create 50 children with their id and an age between 3 and 12 years, which will be calculated as 3 plus the units of the identification number. For instance, the child with id 13 will be 6 years old (3+3), the child with id 40 will be 3 years old (3+0) etc. A new child will be created every 0.2 seconds.
2. Each child (thread) will wait a random time between 0.2 and 2 seconds to choose the attraction (randomly). The child will ride on that attraction or queue if it is busy. Each child does this cycle until the playground is closed, and then the process ends.
3. Each child spends between 0.2 and 2 seconds in the swing. In the slide, it takes 1.2 sec to go up the slide, and 0.5 to go down the slide. In the carousel, each child must wait until all five seats are occupied. Then, the carousel will go round and round for 5 sec before children get off.
4. When the system is on hold, the number of children in each attraction will be shown.

Monitor module:

5. The monitor will access via RMI the method **checkAge()** which returns the child's age and forcing the child to quit the slide if necessary. The interface must be called **InterfaceMonitor**.

Supervisor:

6. The supervisor will communicate with the server via sockets using the port 2222 to obtain the data or close the playground.
-

Submission

1. *This is an individual assignment to be uploaded to Blackboard. It is composed of a report in PDF and a netbeans Project using ZIP.*
2. *Deadline: May 22nd, midnight.*
3. *All programs must work as requested.*
4. *The report will include a design of the application, classes, attributes, etc. It must also contain the code properly explained with no more than 20 pages. The quality of the report will be considered as part of your marks.*
5. *A piece of advice is to use your own computer in the demonstration of this assignment.*