23

```
    }
  }
}
```

5   The following code fragment describes the functions that implement the behaviour of a graph. Note again that a user does not need to implement these functions. To describe any graph, the user simply needs to provide a two dimensional string array listing the nodes and arcs of the graph.

```
10    String[][] nodes // two dimensional array of listing the nodes and arcs of the graph
      String   start_node // the label of the start node
      String   next_node // the label of the next node
      Node     previous_node // the process of the last node of the graph
15    Node        begin_node // the process identifier of the start node   Graph parent_graph // the graph of which this is a subgraph
      int      state // the current state of the graph


      public void run(Engine engine, Graph parent_graph, Node previous_node,
20                    String next_node) {

          this.parent_graph = parent_graph;
          this.previous_node = previous_node;
          this.next_node = next_node;
25
          start(engine,arc_data);
      }


      public void run(Engine engine, Object input) {
30        start(engine,input);
```