

Solution 3: Of objects and features

ETH Zurich

1 Classes vs. objects

Solution

There is no unique solution. Sample answers:

- 1.1 **Definition and description of the concept "Class"**. A class is a description of a set of possible run-time objects to which the same features are applicable.

A class does several things:

- It provides a description of how objects will behave at runtime in response to feature calls
- It provides an interface for the programmer to interact with the definition of objects during development
- It is a source of new objects in a running system

As such, a class can be interpreted as "a form of template" (where the term class is associated to a collection of related entities - objects that are considered to be instances of the same class defining their properties and operations), "a form of structure" (offering a restricted interface to clients), and "a form of type" (allowing to declare entities with a type that is a class).

Definition and description of the concept "Object". An object is a software machine allowing programs to access and modify a collection of data. An object can be manipulated through a set of operations, defined in its generating class. Objects only exist at run-time.

- 1.2 A class can be looked at as the blueprint of a machine, while the object is the actual machine built according to the blueprint.

2 Categorizing features

Solution

Queries:

- actual_time
- duration
- speedup
- is_time_running

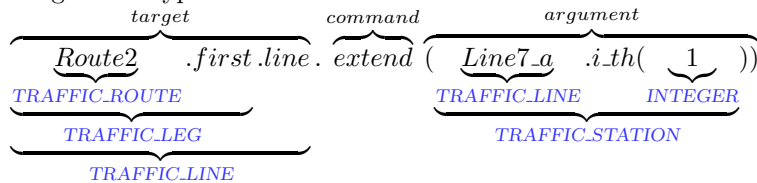
Commands:

- pause
- reset
- set_speedup
- start
- resume
- set

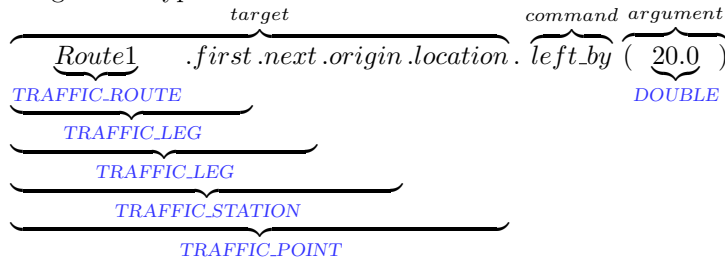
3 Feature reading

Solution

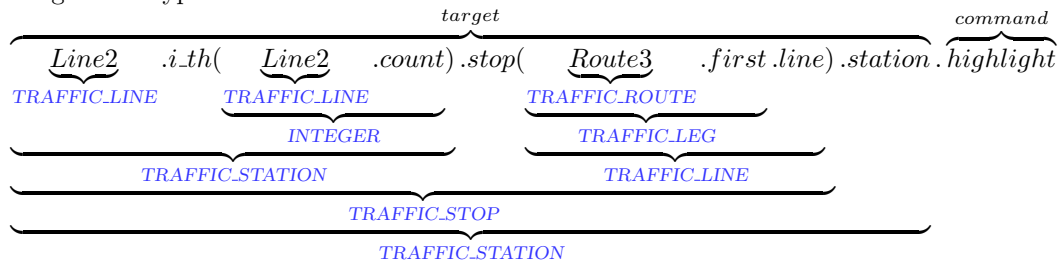
1. Target is of type **TRAFFIC_LINE**.



2. Target is of type **TRAFFIC_POINT**.



3. Target is of type **TRAFFIC_STATION**.



4 Writing more feature calls

Solution

Listing 1: Class **PLANNER**

```

1 indexing
  description: "Planner class (Assignment 3)"
3 date: "$Date$"
  revision: "$Revision$"
5
    
```

```
class
7  PLANNER

9  inherit

11  TOURISM

13  feature -- Explore Paris

15  explore_on_click is
    -- Explore Paris!
17  do
    Paris.display
19  -- Paris.display must be the first line (loads and displays Paris map)

21  Line1.remove_all_segments
    Line1.extend (Station_concorde)
23  Line3.remove_all_segments
    Line3.extend (Station_concorde)
25  Line7_a.remove_all_segments
    Line7_a.extend (Station_concorde)
27  Line8.remove_all_segments
    Line8.extend (Station_concorde)
29  Line2.remove_all_segments
    Line2.extend (Line3.terminal_1)
31  Line2.extend (Line7_a.terminal_1)
    Line2.extend (Line1.terminal_1)
33  Line2.extend (Line8.terminal_1)
    Line2.extend (Line2.terminal_1)
35  end

37 end
```