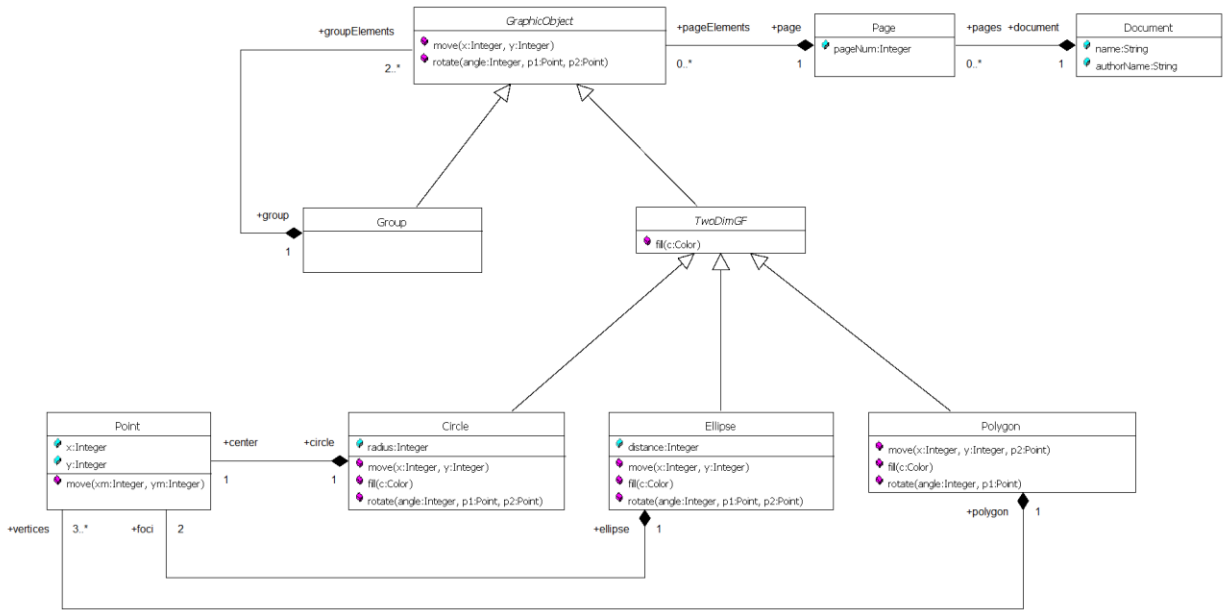
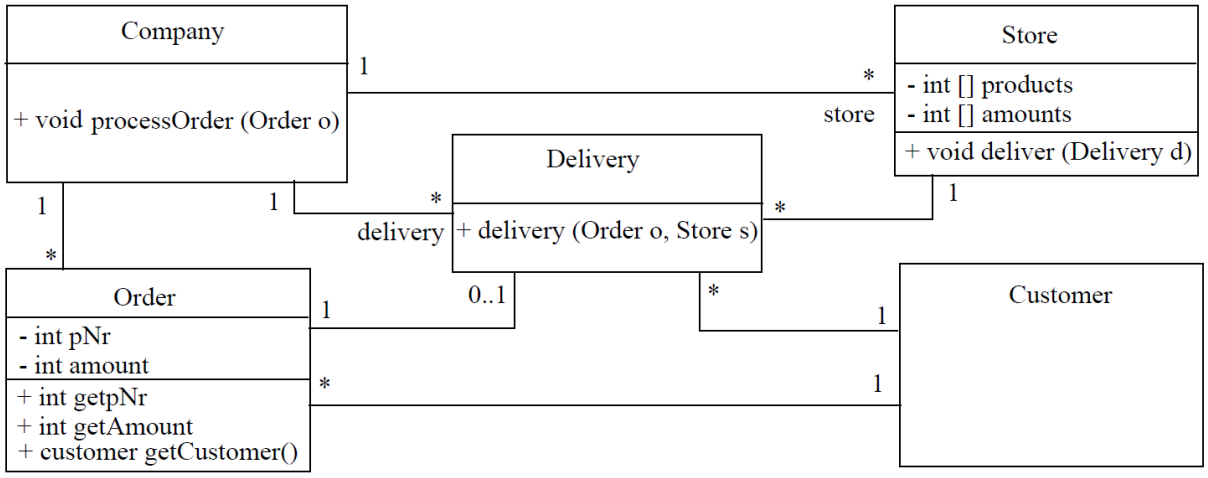
ISS – 21-06-2021

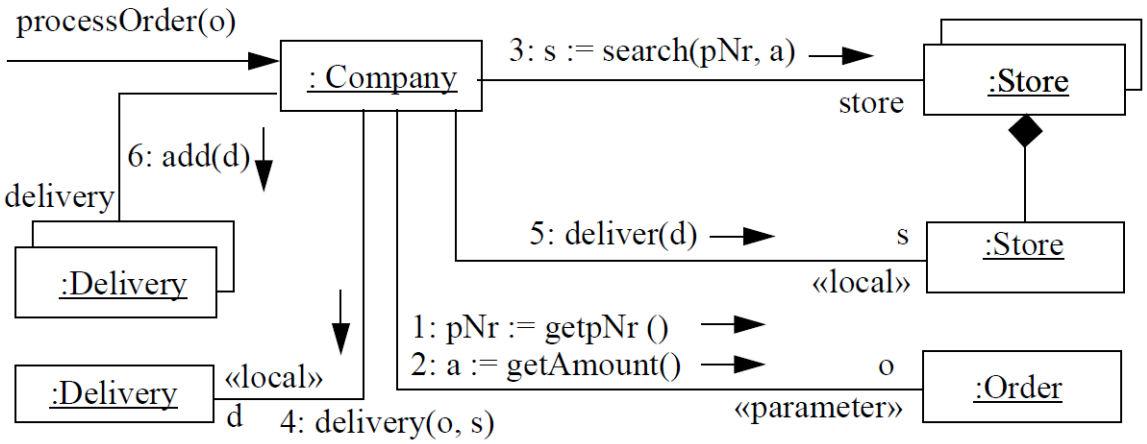
1. A digital document is characterized by name and authorName and contains pages. Each page has a pageNumber which is an integer and contains pageElements which are bidimensional graphical figures (polygon, ellipse and/or circle) and groups. A group contains groupElements which may include other groups and/or bidimensional graphical figures. A polygon is defined by an ordered set of 3 or more vertices. An ellipse is defined by 2 points referred as foci (focus points) and by a distance which is an integer. A circle is defined by a point referred as center and a radius which is an integer. Each point is defined by 2 coordinates refered as x and y and by a color. The value of x and y are integers because it is about pixels. The colors are the fundamental colors: red, green, blue. Each graphical object can be moved by a reference (x:Integer, y:Integer) or can be rotated by an angle having an integer value and by a line defined by 2 points. A bidimensional graphical figure can be filled by a color.
    a. Using the UML, please represent a class diagram complying with the above-mentioned requirements. Please design an extensibile model (the color numbers may be increased; other kind of graphical figures may be added). 3pt
    b. Using the OCL, please specify invariants for ellipses and circles. In case of the ellipse, the two foci must be different, and the distance value must be equal or greater than 2. In case of circle, the radius must be equal or greater than 2. In case of the move operation defined on points, please specify a postcondition ensurig that after aplying this operation, the coordinates of the point will be increased by the reference values and the operation different on points. 2pt



```
context Circle
    inv circle:
        self.radius >=2

context Ellipse
    inv ellipse:
        let fo:Sequence(Point)=self.foci->asSequence
            in fo->first <> fo->last and distance >= 2

context Point::move(xm:Integer, ym:Integer)
    post move:
        self.x = self.x@pre + xm  and   self.y = self.y@pre + ym

context Point
    def different:
        let different(p: Point):Boolean = self.x <> p.x or self.y <> p.y
/* or */
        let different2(p: Point):Boolean = not( self.x = p.x and self.y = p.y)
```

2. In the two figures below, there are represented related elements of two different UML views.
   a) Please name these views and the elements represented in each view. 1pt
   b) Please mention the tuple of related elements from these 2 views and justify if the information of these views represented by diagrams are consistent or not. 2pt



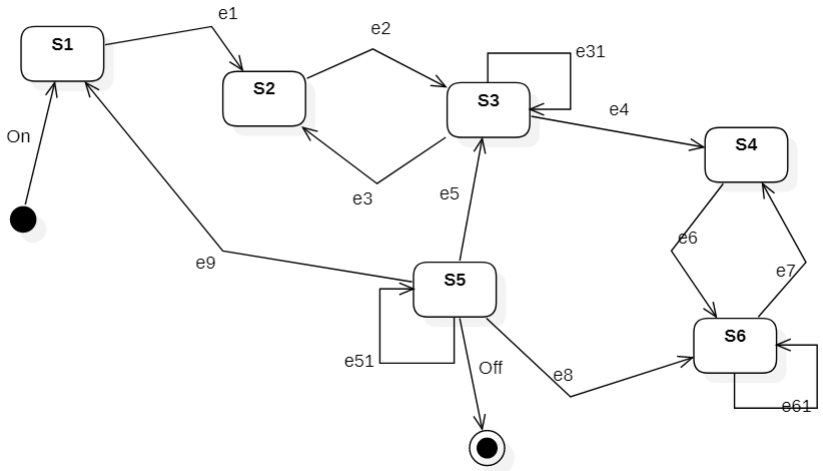


The first figure is a class diagram and the second one a collaboration diagram. In the class diagram the elements represented are classes (Company, Store, Customer, Order, Delivery), attributes (pNr, amount, products, amounts), operations (Company::void processOrder(Order o), Store::void deliver(Delivery d), a.s.o) and bidirectional associations between pairs of classes (Company, Store; Company, Delivery, Company, Order; Store, Delivery; Customer, Delivery; a.s.o.). The second diagram is a collaboration/communication diagram, in which instances of classes represented in the class diagram collaborate by sending and receiving messages to delivery a functionality. In our case the functionality is trigerred by the message processOrder(o) sent to an anonymous instance of the Company class. All instances/objects represented in the diagram are anonymous. The relationships between objects are links (instances of associations represented in the class diagram. Excepting the processOrder(o) message, all the others are ordered by numbers to explicit the sequence in which these will be processed. To increase the level of generality for this diagram, the values of parameters are generically specified by letters. In ours analyse we will consider that these values comply with the types specified in operations' signature. To be consistent, the information represented in both diagrams must fit. (It is about objects' type, links, conformance of messages with operations. The sequence in which the functionality is delivered:

- After receiving the message processOrder(o), the company interogate the order received by the actual parameter, to obtain the values pNr (productNumber) and a (amount) characterizing the order. Messages 1 & 2. In the class diagram the operations corresponding to getters & setters are not mentioned. Let us consider these as implicit since associated attributes are private.
- The next step is to find a store containing the product in the required amount, message 3.
- Once the store was found, the company creates a delivery object complying with the order o and the store s, message 4.
- The next step, described by the message 5 is to send to the apropriate store stereotyped <<local>> to deliver the delivery d, transmitted by parameter.
- Finally, the functionality processOreder(o) is ended after the above-mentioned delivery is added by the company to the delivery, message 6

Therefore, we may conclude that the information presented in the two diagrams is consistent.

3. Let us consider the diagram represented in the figure below. Please enumerate the elements represented in this diagram and justify if the behavior specified is consistent or not mentioning all the rationales supporting your statemet. 1pt



The elements are:

- pseudostates: the Input State and the Output State,
- simple states: S1, S2, …, S6
- transitions: Input State to S1, S5 to Output State, S1 to S2, S2 to S3, S3 to S2 a.s.o.,
- events/triggers: e1, e2, …, e9, e31, e51, e61, On, Off

The behavior description is inconsistent because:

- states S5 and Output State cannot be atteint,
- states S4 and S6 are a trap.