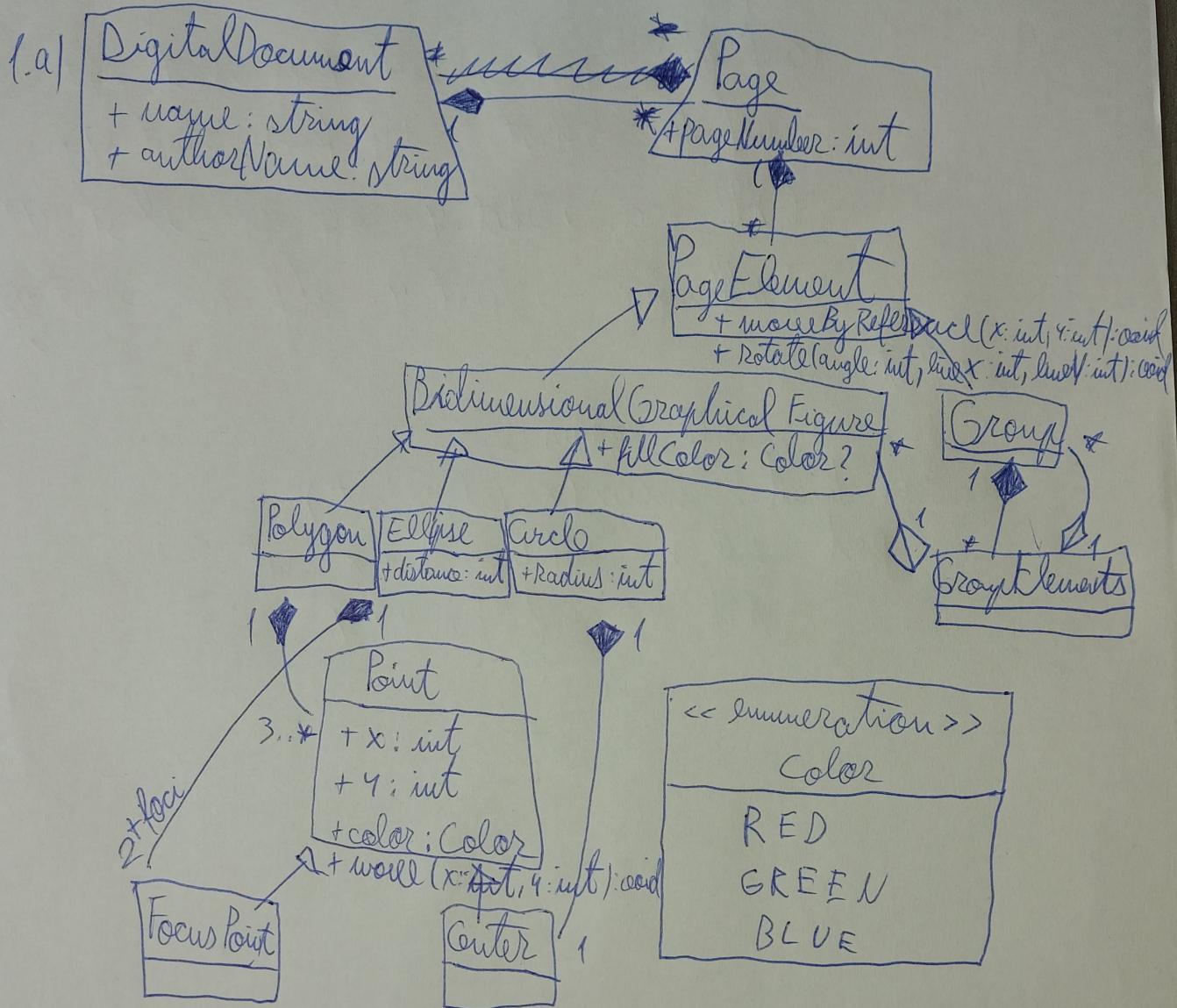


Crācium /an - Elanin

922

CR



`Color?` means the type is nullable. The colors can be extended by adding values to the enum class and other graphical figures may be added through inheritance. The `2` functions in `PageElement` will be realised through the move from `Point`.

Gracium Ivan-Flavien 9228c

1.6) context Ellipse

rule: (self. foci \rightarrow first. $x \leftrightarrow$ self. foci \rightarrow last. x) or
(self. foci \rightarrow first. $y \leftrightarrow$ self. foci \rightarrow last. y) and
(self. foci \rightarrow first. $x -$ self. foci \rightarrow last. x) *
(self. foci \rightarrow first. $x -$ self. foci \rightarrow last. x) +
(self. foci \rightarrow first. $y -$ self. foci \rightarrow last. y) *
(self. foci \rightarrow first. $y -$ self. foci \rightarrow last. y))
>=

context circle

rule: self. radius ≥ 2

Context Point :: move (x:int, r:int)

post: self. $x =$ self & pre. $x + x$ and
self. $y =$ self & pre. $y + y$

Practical Pan-Elavum 922 (b)

Q1 The UML diagram shown in the first figure is a class diagram, thus we are considering a structural view, because in this kind of view only the structure of the model is explained. In a class diagram for each class we have 3 sections:

- upper section which the name of the class is mentioned
- middle section which contains the attributes of the class
- bottom section which contains the methods/functions

Also apart from classes we have attributes and relationships. The second diagram is a collaboration diagram, so it is a behavioral ~~diagram~~ view. This type of diagram has objects, relationships and messages (an arrow that points from one object to another)

Bracium Icam - Flavien 922 FR

2. 6) The ~~classes from~~ objects from the collaboration diagram are of the classes from the class diagram. There is one Company object on which processOrder is called first. However it had to be numbered with a 1. to disambiguate the operation order. This is actually the entry point of the diagram and all that follow are in that order.

Thus there are 2 Store objects (Store → : Store), 1 Company (Company → : Company), 2 Delivery (Delivery → : Delivery) and (order → : Order). The search method from the collab diagram is missing (from the Store object) and also the add method is missing from Delivery. The rest of the methods in the collab diagram are also present in the class diagram (related elements). So to conclude no, the 2 diagrams are not consistent with each other because of methods missing from the class diagram. As last remark, the customer class is not used by the collaboration diagram which adds to the inconsistency.

Gracim Iam - Flawin

922 Br

3. This is a State Transition Diagram.

Concrete states: s_1, s_2, s_3, s_4, s_5

Pseudostates: input state, output state

Events: e_1 (from s_1 to s_2)

e_2 (from s_2 to s_3)

e_3 (from s_3 to s_2)

e_4 (from s_3 to s_4)

e_5 ($s_5 \rightarrow s_3$)

e_6 ($s_4 \rightarrow s_6$)

e_7 ($s_6 \rightarrow s_4$)

e_8 ($s_5 \rightarrow s_6$)

e_9 ($s_5 \rightarrow s_1$)

e_{10} ($s_3 \rightarrow s_3$)

e_{11} ($s_5 \rightarrow s_5$)

e_{12} ($s_6 \rightarrow s_6$)

on (input $\rightarrow s_1$)

off ($s_5 \rightarrow$ output)

Events transition the states to another state.

The diagram is not consistent because from input
it's possible to ~~not~~ reach output $s_5 \xrightarrow{e_5} s_3 \xrightarrow{e_3} s_2 \xrightarrow{e_2} s_1 \xrightarrow{e_1} s_2 \xrightarrow{e_4} s_4 \xrightarrow{e_6} s_6$

because only from s_5 we can reach output but all
can't reach s_5 from any state.