

## Oblikovni obrasci u programiranju

### 4. laboratorijska vježba

1. U program je potrebno dodati novi alat za masovno odgrupiranje grupa objekata. Uporaba je zamišljena na sljedeći način: kada korisnik odabere taj alat, svaki klik na lijevu tipku miša pronaći će objekt koji bi u alatu za selekciju time bio odabran. Ako je taj objekt grupa, razgrupirat će ga. Ako je bilo koji od sastavnih dijelova bio grupa - i to će razgrupirati, i tako redom, dok ne ostanu samo jednostavni elementi.
  - (a) navedite koji su sve oblikovni obrasci izravno povezani s rješenjem ovog zadatka, nacrtajte njihove strukturne dijagrame i povežite ih sa sudionicima u svojem rješenju,
  - (b) napišite traženu implementaciju zadanog alata.
2. Potrebno je omogućiti promjenu boje grafičkih objekata. Zamišljeno je da se ta promjena boje odvija u stanju za selekciju na sljedeći način:
  - odabirom novog stanja u dnu prozora pojavila bi se paleta mogućih akcija za to stanje; u slučaju stanja za selekciju ta bi paleta sadržavala dostupne akcije promjene boje,
  - korisnik bi u stanju za selekciju označio željene objekte i zatim bi klikom na boju promijenio boju odabranih objekata.

Navedite koji oblikovni obrazac sudjeluje u iscrtavanju grafičkih objekata, nacrtajte njegov strukturni dijagram te ga povežite sa sudionicima u svojem rješenju. Skicirajte u kodu najvažnije promjene koje je potrebno napraviti da bi se omogućila promjena boje na opisani način.

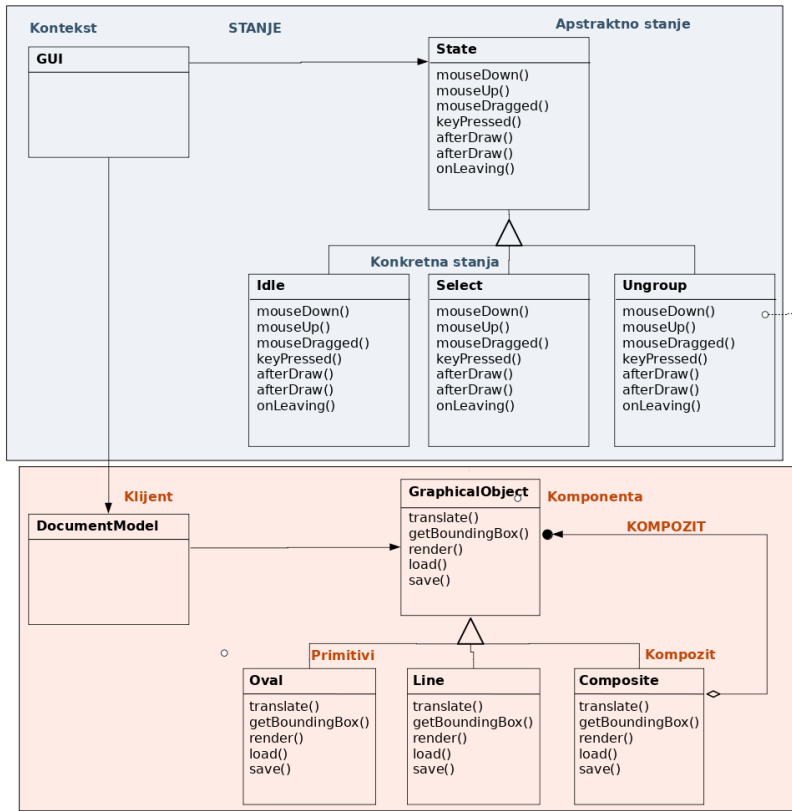
Skicirajte promjene u kodu potrebne za prikaz palete akcija prilikom promjene stanja.

3. Proširite program iz četvrte laboratorijske vježbe tako da omogućite umetanje teksta u crtež. Akcija dodavanja u crtež treba unijeti tekst "Proba".

Zbog jednostavnosti, nije potrebno implementirati editiranje teksta. Program treba podržati proizvoljni tekst u operacijama za učitavanje i pohranjivanje crteža.

Navedite koji obrazac koristimo za dodavanje novih objekata u sliku, nacrtajte njegov strukturni dijagram te ga povežite sa sudionicima svog rješenja. Pojasniti s kojim je oblikovnim načelima usklađeno Vaše rješenje.

Skicirajte dijelove koda potrebne za ispravno snimanje i učitavanje teksta.



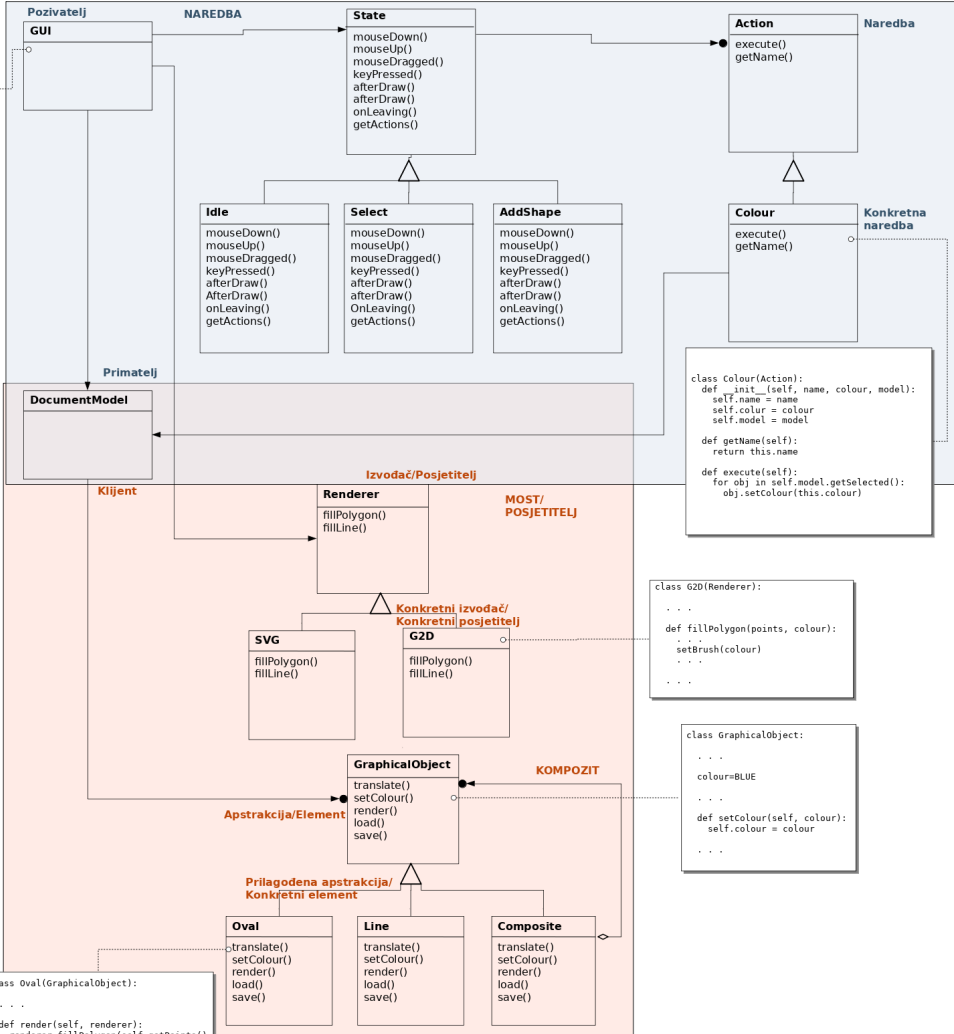
```

class Ungroup(State):
    ...
    def mouseUp(self, mousePoint):
        selectedObject = self.model.getSelectedObjects()[0]
        SOChildren = self.getCompositeChildren(selectedObject)
        self.model.removeGraphicalObject(selectedObject)
        for obj in SOChildren:
            self.model.addGraphicalObject(obj)
    ...
    def getCompositeChildren(self, object):
        if isinstance(object, Composite):
            descendants = []
            for obj in object.children:
                descendants.extend(self.getCompositeChildren(obj))
        else:
            return [object]
    ...
  
```

```

class GUI:
    ...
    actionToolbar
    model
    currentState
    ...
    def toSelectState(self):
        currentState.onLeaving()
        currentState = Select(self.model)
        actionToolbar = self.addAction()
    ...
    for stateAction in currentState.getActions():
        action = Action(stateAction.getActionName(), self)
        self.actionToolbar.addAction(action)
        action.triggered.connect(stateAction.execute())
    ...

```



```

class Colour(Action):
    def __init__(self, name, colour, model):
        self.name = name
        self.colour = colour
        self.model = model
    def getName(self):
        return this.name
    def execute(self):
        for obj in self.model.getSelected():
            obj.setColour(this.colour)

```

```

class G2D(Renderer):
    ...
    def fillPolygon(points, colour):
        ...
        setBrush(colour)
    ...

```

```

class GraphicalObject:
    ...
    colour=BLUE
    ...
    def setColour(self, colour):
        self.colour = colour
    ...

```

```

class Oval(GraphicalObject):
    ...
    def render(self, renderer):
        renderer.fillPolygon(self.getPoints(),
            self.colour)
    ...

```

