

# Specifikacija za drugu kontrolnu tačku

**Rok za predaju:** 15.12.2024. do 23:59h

**Pitanja:** Studenti mogu postavljati tehnička pitanja vezana za zadatak do 08.12. (uključujući i taj dan).

Poeni: KT2 nosi maksimalno 23 poena.

## Zahtevi i funkcionalnosti

### 1. RoomView panel

- Soba (RoomView/RoomPanel) se prikazuje u okviru JTabbedPane komponente.
- Dimenzije sobe unose se u centimetrima, ali se vizuelni prikaz automatski skalira na JPanel tako da soba zauzima skoro ceo vidljivi deo panela.
- Osmisliti pravilo za konverziju dimenzija iz centimetara u piksele i implementirati ga unutar klase Room.

### 2. Modeli i struktura podataka

#### 2.1. Room i elementi sobe

- **Room:**
  - Sadrži dimenzije sobe u centimetrima (veličinu pravougaonika koji je iscrtan na JPanelu).
  - Čuva listu elemenata (npr. nameštaj) koji se nalaze u sobi.
  - Više nije leaf node Composite šablona, sada može da ima decu
- **RoomElement (apstraktna klasa):**
  - Predstavlja osnovni model za sve elemente sobe.
  - Ovo je sada leaf node Composite šablona
  - Ovaj element takođe želimo da pokažemo u JTree komponenti kada se doda na neki Room element (za nazive možete praviti static counter atribut)
  - Sadrži zajedničke osobine: lokaciju, dimenzije, i rotaciju (rotateRatio, vrednost od 0 do 3 za rotaciju u smeru kazaljke na satu).
  - Nasleđuju je konkretne klase elemenata: Vrata, Krevet, Sto, Ormar, Kada/TusKabina, VesMasina, Bojler, Lavabo, WCSolja.

- Konkretna klasa će implementirati **Prototype** interface, oslanjajući se na Prototype design pattern omogućava se kreiranje kopija elemenata (vrata, krevet, sto...).
- Svaka konkretna klasa može sadržati dodatne attribute po potrebi.

## 2.2. Painter klase

- Svaki element sobe ima pridruženu Painter klasu koja omogućava vizuelno iscrtavanje elementa na JPanel-u.
- Model nikako ne sme da ima u sebi Painter atribut (kršenje MVC-a), dok Painter treba u sebi da ima atribut modela koji iscrtava
- Painter takođe implementira transformacije poput rotacije i skaliranja.
- Hijerarhija Painter klasa je ostavljena na studentima da izaberu njima odgovarajuću

## 3. State Design Pattern

Aplikacija koristi **State Design Pattern** za upravljanje različitim načinima rada (stanja). Svaki state ima specifično ponašanje koje korisnik može aktivirati.

### 3.1. Dostupna stanja

#### Edit Room State

- Klikom na prazan RoomView, iskače prozorčić da se unesu dimenzije sobe
- Nakon unošenja dimenzija, potrebno je nacrtati pravougaonik tih proporcija koji maksimalno pokriva vidljivi deo JPanel-a (RoomView-a) i sačuvati odnos za buduće crtanje
- Ne dozvoliti dodavanje ostalih elemenata ukoliko ova akcija nije odradjena

#### Add State

- Dodavanje novih elemenata u sobu:
  - Korisnik unosi dimenzije elementa u centimetrima.
  - Element se skalira i dodaje na sobu.
  - Dostupni elementi: vrata, krevet, sto, ormar, kada/tuš kabina, veš mašina, bojler, lavabo, WC šolja.
    - Slike grafičkih elemenata se nalaze na kraju ovog dokumenta.
  - Ne dozvoliti dodavanje novog elementa ukoliko se preklapa sa nekim već postojećim

## Select State

- Selektovanje elemenata:
  - Klikom na element selektuje se samo taj element.
  - Prevlačenjem pravougaonika selektuju se svi elementi koji se nalaze unutar tog pravougaonika (**selektovanje više elemenata**).

## Resize State

- Omogućava promenu dimenzija selektovanog elementa.
- Dimenzije se menjaju prevlačenjem donjeg desnog ugla elementa.
- Ne računa se implementacija preko prozora da se unesu nove dimenzije
- Omogućiti live prikaz elementa dok mu se menja dimenzija

## Rotate State

- Rotacija selektovanog elementa ili grupe elemenata:
  - Dugme `rotateLeft` okreće element za 90 stepeni ulevo.
  - Dugme `rotateRight` okreće element za 90 stepeni udesno.
  - Rotacija koristi matrice transformacije koje se čuvaju u samom painteru element (podsetnik da model ima `rotateRatio`).
- Ova funkcionalnost ne mora biti implementirana kao State, mogu da budu i obicne akcije na klik koje stoje desno u ToolBar-u

## Copy and Paste States

- Kopiranje i lepljenje elemenata (**Prototype design pattern**):
  - Prilikom kopiranja, selektovani elementi se dodaju u listu kopija unutar sobe.
  - Lepljenje koristi `clone()` metodu za kreiranje novih instanci elemenata, pomerenih za određeni offset (npr. 20 px udesno). To se postiže tako što se prolazi kroz sve selektovane modele, prave se kopije i prave se novi painter-i za njih koji se lepe u `RoomView listOfPainters`.
- Isto kao za rotate, ovo ne mora da bude state, možete na dugme ili na precicu na tastaturi

## Move State

- Pomeranje selektovanih elemenata unutar sobe.
- Ako se element približi ivici sobe na manje od 10 px, aktivira se funkcija "lepljenja" za precizno poravnanje uz ivicu.
- Proveriti presek sa drugim elementima, ukoliko se seče sa nekim drugim elementom u sobi, ponistiti akciju (vratiti na početno stanje)

- Ako nema selektovanih elemenata, moguće je pomerati celu sobu drag-and-drop (move around) metodom ili pomoću `JScrollPane` – po želji jedan od načina.
- Tokom pomeranja selektovanih elemenata, live iscrtavati elemente tokom pomeranja

#### Zoom State

- Zumiranje cele sobe:
  - Zumiranje se takođe radi pomoću matrica transformacije.
  - Skrolovanjem miša povećava ili smanjuje skalu prikaza.
  - Implementirati ograničenja za minimalni i maksimalni nivo zumiranja.

#### Edit State

- Klikom na element otvara se dijalog sa informacijama o dimenzijama i atributima elementa.
- Promene u dijalogu automatski osvežavaju element na `JPanel`-u.

#### Delete state

- Klikom na neki element, on se obriše
- Ukoliko postoji već neka selekcija elemenata, brišu se svi elementi koji su selektovani
- Brisanje treba da brise i u modelu i na `JTree` pored painter-a

## 4. Tehnički zahtevi

### 4.1. Listeners

Za rad sa listener-om za miš koristiti sledeće interfejse da bi se događaju uspešno uhatili:

- `MouseListener`
- `MouseMotionListener`
- `MouseWheelListener`

### 4.2. Transformacije

- Koristiti matrice transformacije (`.transform()`) za implementaciju rotacije i zumiranja.
- Resetovati matricu transformacije nakon svake primene rotacije u `paint()` Metodi da ne bi bilo problema sa crtanjem ostalih painter-a.

- Obezbediti da nakon zumiranja, lokacija klika mišem bude na istom mestu gde je kliknuto (poeni za zoom state se neće priznavati ukoliko bude postojao offset u drugim stejtovima kada se promeni zoom panela, npr selekcija elemenata)

#### 4.3. Arhitektura

- Obavezno je koristiti arhitekturu za rad sa State patternom koju smo zadali na času (tabla puna teksta sa objašnjavaњem šta se odakle zove: Vežbe6 drive; preporučeno odslušati čas)

#### 4.4. Observer pattern

- U svakim State-ovima, kada se uradi bilo kakva promena u modelu, potrebno je da model koji se promenio observerom obavesti RoomView na kome se nalazi da se nacrtaja ponovo, pozivanjem njegove `repaint()` metode. Ukoliko se promene ne dešavaju u modelu, dozvoljeno je direktno zvati iz State klase: `roomView.repaint()`
- Prilikom promene selektovanih tabova, sve što je nacrtano na njima mora da ostane netaknuto (ne želimo da se naš napredak na jednom crtežu izgubi), isto važi ukoliko se šetamo između više Project-a u našem stablu

#### 4.4. Ideja za akcije umesto state

Ukoliko želite da copy paste i rotate stejtove implementirate preko akcija / tastature, moguće je da ćete imati problem pristupanja roomView-u aktivnom i njegovim atributima koji su vam potrebni za implementaciju. Ovo možete rešiti tako što će to ipak i dalje biti State-ovi, ali u State interfejs dodajte metodu `triggerOnAction()` koju će samo ovi stejtovi implementirati. U kodu za akciju nakon klika, potrebne su vam 2 linije:

- Pozivanje medijatora da aktivira to stanje
- Pozivanje medijatora da odmah izvrši `triggerOnAction()` metodu tog state-a da ne morate da čekate ponovni klik (ovde morate osmisлити kako ćete da pošaljete roomView)

#### Legenda grafičkih elemenata

- Načelni prikaz svih grafičkih elemenata koji će biti u sobi.
- Potrebno je poštovati ovu legendu prilikom pravljenja Painter-a za nove modele

