

Kolokvijum se sastoji od dva dela:

- Prvi deo (30 poena) – sve potrebne klase se nalaze u okviru datog projekata. Potrebno ih je dovršiti na osnovu teksta/komentara u kodu. Takođe je dostupan i UML dijagram.
- Drugi deo (20 poena) – dati zahtevi obuhvataju pravljenje odgovarajućeg grafičkog okruženja.

### Prvi deo

Napisati na jeziku *Java* sledeće tipove, stanja i ponašanja:

#### Paket **Merljivo**

- 1) **Apstraktni tip *IUporedivo*** koji sadrži:
  - a) Javnu apstraktnu metodu **UporedivoJeSa** koja prihvata **Object** i vraća **bulovsku** vrednost. Metod vraća tačnu vrednost ako je objekat tipa **Object**, koji se prosleđuje, istog tipa kao i objekat koji poziva metod.
  - b) Javnu apstraktnu metodu **Uporedi** koja prihvata **Object** i vraća celobrojnu vrednost. Metod proverava da li je prosleđeni objekat veći (vraća -1), manji (vraća 1) ili jednak objektu (vraća 0) koji implementira ovaj metod. Metod vraća izuzetak tipa **Exception** ukoliko objekti nisu uporedivi.
- 2) **Apstraktni tip *IOPorezivo*** koji sadrži:
  - a) Javnu apstraktnu metodu **Porez** koja vraća **realnu** vrednost.

#### Paket **Gradjevine**

- 3) Enumeraciju **Strana** koja može uzimati vrednosti **Levo** i **Desno**.
- 4) Konkretan tip **Adresa** ima sledeća stanja i ponašanje:
  - a) **NazivGrada** i **NazivUlice** koji imaju vidljivost samo u klasi.
  - b) Ceo **broj** koji je po pravilu 0 i ima vidljivost samo u klasi.
  - c) **StranaUlice** tipa **Strana** koja ima vidljivost samo u klasi.
  - d) Javni konstruktor koji prihvata naziv grada i naziv ulice i vrši odgovarajuću inicijalizaciju.
  - e) Javni konstruktor koji se **nadovezuje na prethodni konstruktor (poziva ga)** i inicijalizuje naziv grada, naziv ulice i broj. U zavisnosti od toga da li je vrednost broja parna ili neparna dodeljuje strani ulice vrednost desno odnosno levo.
  - f) Obezbediti metode kojima se mogu dohvatiti vrednosti svih stanja objekta (geteri). Takođe obezbediti metodu kojom se može postaviti vrednost broja.
  - g) **Tekstualnu reprezentaciju** adrese predstaviti u sledećem formatu "naziv\_ulice broj, grad". Ako je broj jednak 0, onda umesto broja stampa BB.
- 5) Apstraktni tip **Gradjevina** koji ima sledeća stanja i ponašanje:
  - a) **Adresu** koja ima vidljivost samo u klasi.
  - b) Ceo broj koji predstavlja **BrojSpratova** i ima vidljivost samo u klasi.
  - c) Apstraktnu metodu **Vrednost** koja računa vrednost građevine (ne treba implementacija).
  - d) Metode za dohvaćanje i postavljanje broja spratova i adresu građevine.

- e) **Tekstualnu reprezentaciju** građevine predstaviti u sledećem formatu „Ovo je građevina tipa (**Kuća/Zgrada/Garaža**) od **BrojSpratova** spratova, ima vrednost **vrednostGrađevine** i nalazi se na adresi „**adresa**“ i u narednom redu adresu. Ukoliko adresa ima broj, dodatno štampati i stranu ulice.
- 6) *Uporedivu i oporezivu* građevinu **Kuca** koja ima sledeća stanja i ponašanja:
- Podatak koji čuva broj kreiranih objekata tipa kuca (promenljiva je zajednička za sve objekte tipa Kuca)
  - Realan broj koji predstavlja broj kvadrata po spratu kuće i ima vidljivost samo u paketu.
  - Realan broj koji predstavlja cenu po kvadratu i ima vidljivost samo u paketu.
  - Javni konstruktor koji prihvata adresu, broj spratova, broj kvadrata po spratu i cenu kvadrata i vrši odgovarajuće inicijalizacije.
  - Javni metod **UporedivoJeSa** koji vraća tačnu vrednost ukoliko je objekat koji se prosleđuje tipa **Kuca**, u suprotnom vraća netačnu vrednost.
  - Javni metod **Uporedi** koji dozvoljava poređenje samo ukoliko se prosledi objekat tipa kuća, u suprotnom izbacuje odgovarajući izuzetak. Poređenje vrši na taj način što vraća 1 ukoliko prva kuća (objekat koji poziva metod) ima više spratova, -1 ukoliko druga kuća ima više spratova i 0 ukoliko kuće imaju jednak broj spratova.
  - Vrednost kuće se računa kao ukupna vrednost svih kvadrata u kući.
  - Porez na kuću predstavlja 5% vrednosti cele kuće.
- 7) *Oporezivu* građevinu **Zgrada** koja ima sledeća stanja i ponašanja:
- Ukupan broj stanova po spratu zgrade koji je vidljiv samo u paketu.
  - Realan broj koji predstavlja broj kvadrata po stanu koji je vidljiv samo u paketu.
  - Realan broj koji predstavlja cenu po kvadratu i ima vidljivost samo u paketu.
  - Konstruktor koji prihvata adresu, broj spratova, broj stanova po spratu, broj kvadrata po stanu i cenu kvadrata i vrši odgovarajuće inicijalizacije.
  - Vrednost zgrade se računa kao ukupna vrednost svih kvadrata u zgradi.
  - Porez na zgradu predstavlja 10% vrednosti cele zgrade.
- 8) Građevinu **Garaža** koja ima sledeće ponašanje:
- Konstruktor koji prihvata adresu i vrši odgovarajuće inicijalizacije. Smatra se da garaža ima nula spratova.
  - Vrednost garaže je konstantna i iznosi 1000 eura.

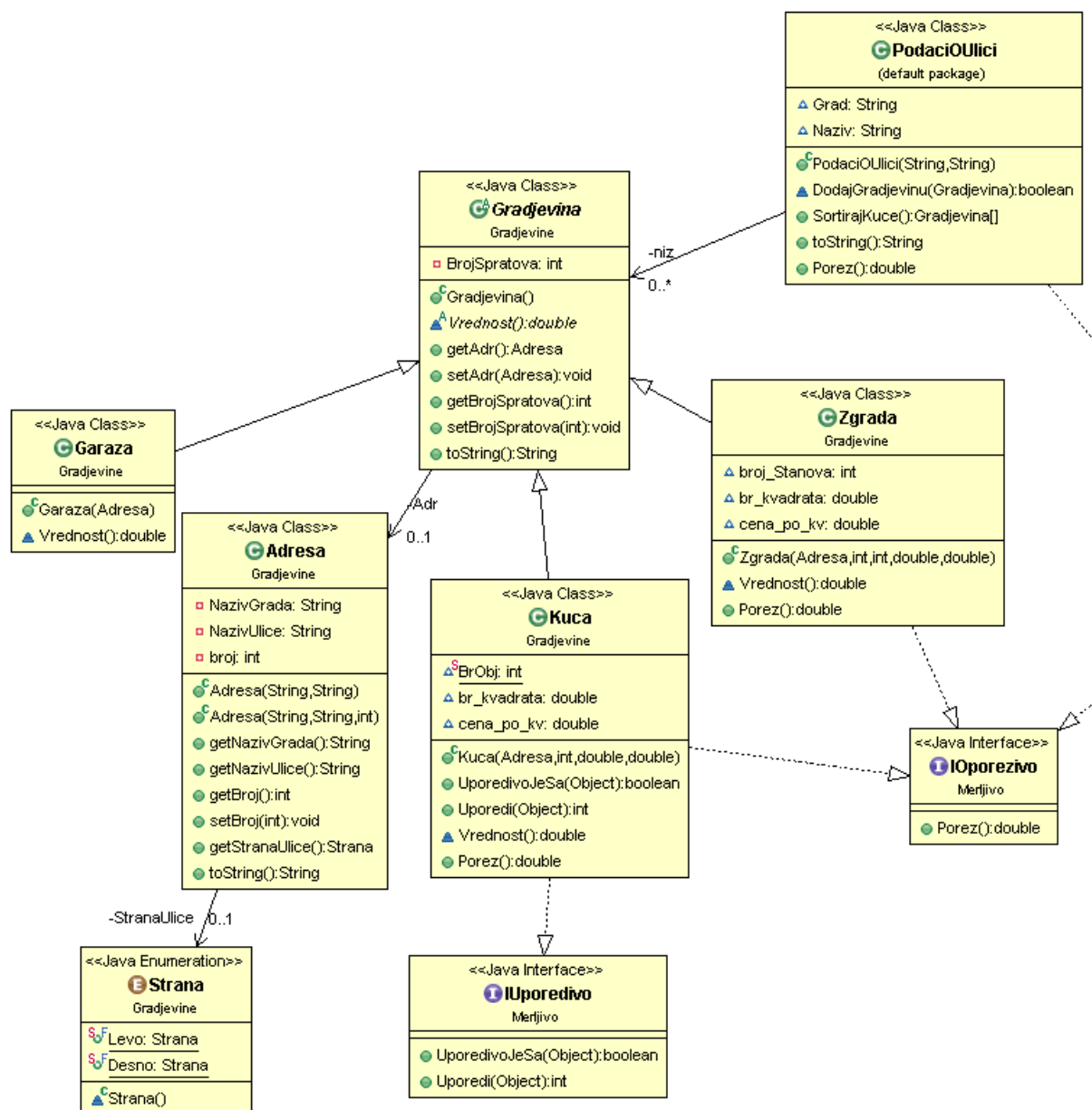
#### Podrazumevani paket

- 9) *Oporezivu* **Ulicu** koja ima sledeća stanja i ponašanje:
- Naziv grada koji je vidljiv samo u paketu.
  - Naziv koji je vidljiv samo u paketu.
  - Privatni niz od 100 građevina.
  - Konstruktor koji prihvata naziv grada i naziv ulice i vrši odgovarajuće inicijalizacije.

- e) Mogućnost dodavanja građevine ako je prosleđena građevina zaista u ovoj ulici. Greška je ukoliko se dodaje građevina sa pogrešnom ulicom.
- f) Metodu za dohvatanje niza svih kuća date ulice. Pri tom taj niz kuća mora biti sortiran po broju spratova od one sa najvećim brojem spratova ka najmanjim.
- g) Omogućiti računanje ukupnog poreza u ulici.
- h) **Tekstualnu reprezentaciju** svih građevina date ulice.

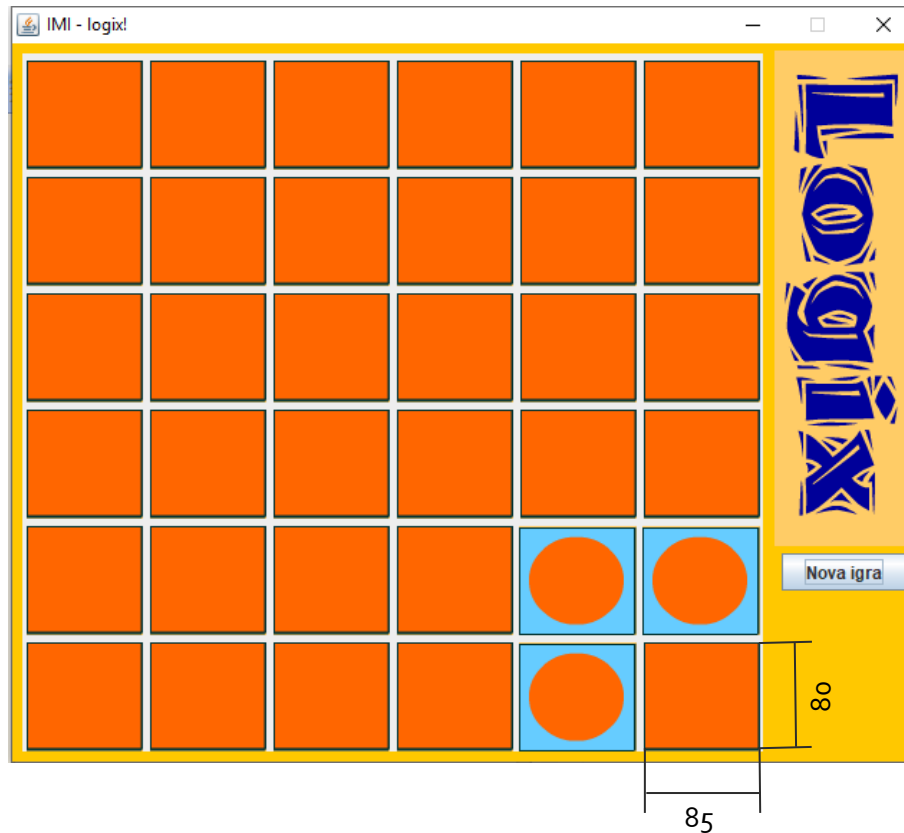
Implementirati **testnu** klasu sa sledećim ponašanjem:

Izvršiti testiranje svega napravljenog kreiranjem 1 instance tipa Ulica, 4 instance tipa Kuca (od kojih su 3 u odgovarajućoj ulici), 2 instance tipa Zgrada (obe su u odgovarajućoj ulici). Smestiti sve što je moguće od građevina u datu ulicu. Odštampati sve građevine u ulici. Odštampati niz svih kuća date ulice sortiran prema broju spratova. Odštampati ukupnu vrednost i porez date ulice.



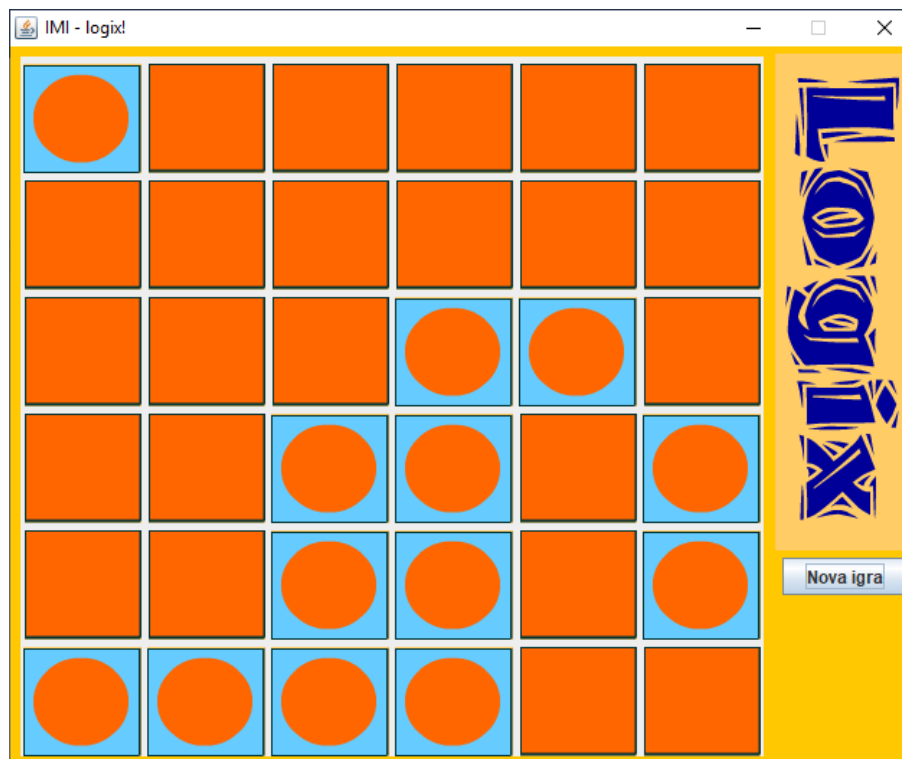
## Drugi deo

Napisati program koji omogućava igranje igrice IMI-Logix. Grafički prikaz početka igre se može videti na slici.

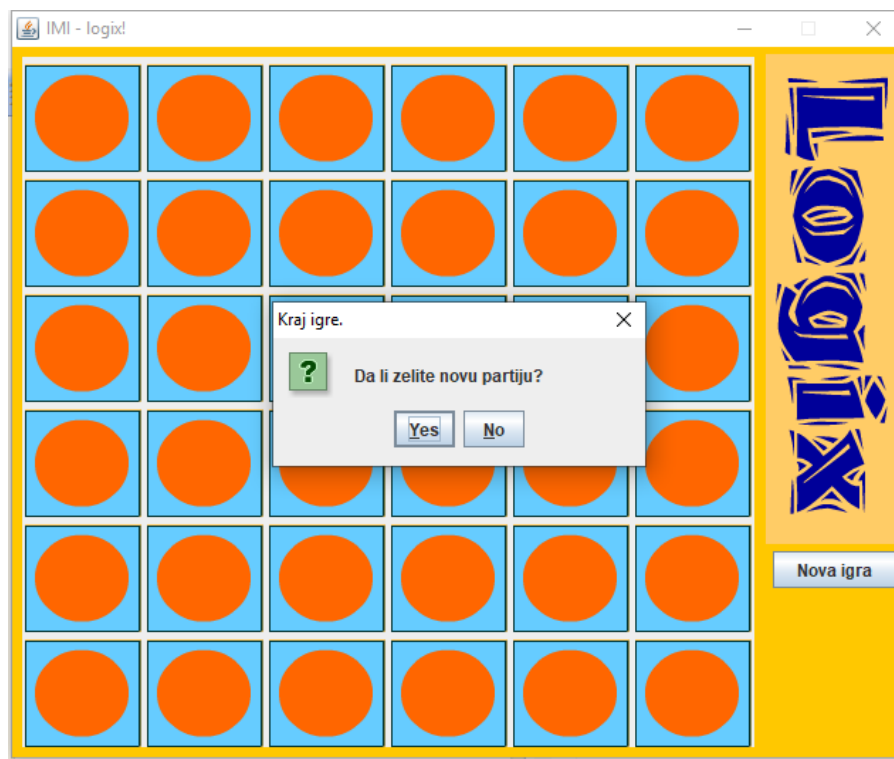


Polja za igranje se sastoje od tabele dimenzija 6x6. Na početku igre 3 slučajno izabrana polja sadrže kružnice, dok su na ostalim poljima kvadrati.

Igra se na taj način što se mišem prevlači preko polja. Prelaskom preko polja menja se slika na polju, kružnica postaje kvadrat, dok kvadrat postaje kružnica. Cilj igrice je da na kraju na svim poljima stoje kružnice.



Kada se na svim poljima nalazi kružnica, igra se završava i imate mogućnost ponovnog igranja.



Klikom na dugme **Nova igra** ona se inicijalizuje na početno stanje.

Pomoć:

<code>komponenta.setIcon(new ImageIcon(Toolkit.getDefaultToolkit().getImage(getClass().getResource("putanjaDoSlike"))));</code>
<code>JOptionPane.showConfirmDialog(this, "Da li zelite novu partiju?", "Kraj igre.", JOptionPane.YES_NO_OPTION);</code>
<code>addMouseListener --- (public void mouseEntered(MouseEvent e))</code>