### Objektno orijentisano programiranje – kolokvijum

# Zadatak: Šah

Napisati Java konzolnu aplikaciju kojom se simulira igra šah. U ovoj verziji šaha moguće je koristiti dva tipa figura – skakač i kralj. Svaku figuru karakteriše boja – bela ili crna i moguće ih je pomerati u skladu sa pravilima pomeranja tih figura u šahu. Figuru dodatno karakteriše i pozicija na šahovskoj tabli ( $n \times n$ ). U jednom potezu, kralj se može pomeriti na neko od susednih polja (uključujući i dijagonalna), dok se skakač pomera dva polja napred i jedno u stranu. Šah igraju dva igrača (šahista) koji mogu biti čovek ili računar. Čoveka dodatno karakteriše broj poena koji se menja u zavisnosti da li pobeđuje ili gubi partije, dok računar dodatno karakteriše težina – lak ili težak. Partija šaha se igra na šahovskoj tabli i na početku dvojica šahista dobijaju figure – jedan bele, drugi crne. Šahisti naizmenično povlače poteze sa svojim figurama i pobednik je onaj igrač koji pojede protivničkog kralja.

Potrebno je implementirati sledeće klase i metode:

- (6) prepisati kompletan UML dijagram klasa
- (1) implementirati sve getere i setere za svaki atribut u svakoj klasi i implementirati *toString* metodu u svakoj klasi kojom se ispisuju svi atributi klase

## - Pozicija

- (1) equals(Object object) pozicija je jednaka drugoj poziciji, ukoliko je oznaka reda iz prve pozicije jednaka oznaci reda u drugoj poziciji i ukoliko je oznaka kolone iz prve pozicije jednaka oznaci kolone u drugoj poziciji
- (1) isValidnaPozicija() vraća true ako se pozicija nalazi u okvirima šahovske table (šahovska tabla ima SahovskaTabla.BR\_REDOVA\_KOLONA redova i kolona, a oznake kreću od 0)

### SahosvkaTabla

- (0,5) dodajFiguru(Figura figura) dodaje novu figuru u odgovarajuću listu figura u zavisnosti od njene boje (bele ili crne figure)
- (1) vratiFiguruNaPolju(Pozicija pozicija) vraća figuru koja se na tabli nalazi na prosleđenoj poziciji, odnosno null u slučaju da pozicija nije validna, ili se na njoj ne nalazi nijedna figura
- (1) ukloniSveFigure() uklanja sve figure sa table menja im poziciju na (-1, -1)
- (2) *odstampajTablu()* štampa u tekstualnu datoteku "šahovskaTabla.txt" sve figure na tabli (njihov naziv, boju i poziciju, kao i dimenzije table broj redova i kolona)

### - Figura (abstract)

- (3) vratiMogucePozicije() apstraktna metoda klasa koja je implementira vraća listu svih validnih pozicija na koje je moguće da ta figura pređe u jednom potezu u skladu sa pravilima kretanja za tu figuru. U ovoj metodi se ne vodi računa o tome da li se na nekom od mogućih polja nalazi druga figura, već samo da li je u skladu sa pravilima kretanja te figure.
- (1) pomeriFiguru(Pozicija pozicija) menja poziciju figure samo ako je na prosleđenu poziciju moguće preći (ako se nalazi u listi mogućih pozicija). Vraća true ukoliko je došlo do promene pozicije, false u suprotnom.

### Sahista (abstract)

- (5) odigrajPotez(SahovskaTabla, Figura, Pozicija, Sahista) – pomera figuru na prosleđenu poziciju, na prosleđenoj tabli. Da bi do poteza došlo, figura mora biti iste boje kao boja sa kojom igra šahista (tj. biti u odgovarajućoj listi) i mora biti na tabli (ne sme biti pojedena). Takođe, nije dozvoljeno preći na prosleđenu poziciju ukoliko se na njoj nalazi figura iste boje. Ukoliko su ovi preduslovi ispunjeni, dolazi do pomeranja figure (pozivom odgovarajuće metode u klasi Figura). Ukoliko je potez validan (metoda vrati true), potez je odgiran. Ukoliko se na novoj poziciji nalazi protivnička figura, ona postaje pojedena (menja joj se pozicija na (-1, -1). Ukoliko je pojeden protivnički kralj, igra je završena, i šahista koji je odigrao potez je pobednik – poziva se metoda upisiPobedu(Sahista) i prosleđuje se protivnički šahista. Tom prilikom se uklanjanju sve figure sa šahovske table.

### Covek

- (1) upisiPobedu(Sahista) – ažurira poene na sledeći način: ukoliko je pobedio računar, na trenutni broj poena dodaje se 5, ako je računar lak, odnosno 10, ako je računar težak. Ukoliko je pobedio drugog čoveka, na broj poena se dodaje 10, ako je drugi čovek imao pre partije više bodova od njega, odnosno 5 u suprotnom. Pobeđenom čoveku (koji je prosleđen kao argument metode) se umanjuje broj poena za 3, pri čemu broj poena ne sme biti manji od 0.

### Racunar

- (1) *upisiPobedu(Sahista)* – ako je pobeđeni šahista čovek, tom čoveku se umanjuje broj poena za 3, pri čemu broj poena ne sme biti manji od 0.

### GlavnaKlasa

(0,5) main metoda – kreirati po jednog belog i crnog kralja i skakača i jednu šahovsku tablu.
Dodati kreirane figure na tablu i kreirati jednog čoveka (bele figure) i jedan računar (crne figure). Naizmenično odigrati nekoliko poteza i simulirati partiju šaha.

