Projektni zadatak iz predmeta Testiranje softvera, jun/jul 2019.

Vrsta zadatka: Testiranje aplikacije

Opis sistema:

Program koji se posmatra predstavlja jednostavnu implementaciju video igre. Potrebno je osmisliti testove i implementirati ih koristeći JUnit alat za testiranje. Koristiti i metode crne i bele kutije. Sve pronađene bugove zavesti na način prikazan u templejtu za bug. Nakon završenog testiranja, napraviti izveštaj o testiranju na osnovu templejta izveštaja. Specifikacija sistema koji se testira je sledeća:

1. Klasa Igrac ima atribute za čuvanje podataka o nazivu igrača, zdravlju, energiji, stanju, snazi, inteligenciji, oružju i odeći koju nosi, kao i magijama koje poznaje. Stanja u kojima igrač može da se nalazi su: agresivno, defanzivno i pasivno stanje. Oružje, odeća i magije igrača su realizovane kao liste (ArrayList) objekata Oruzje, Odeca i Magija. Svi članovi su privatni, i za njih je potrebno obezbediti metode za pristup. Klasa ima jedan konstruktor koji prima po jedan argument za svaki od atributa klase.

2. Klasa Oruzje nije još implementirana. Oružje ima atribute za čuvanje podataka o težini oružja, potrebnoj snazi za upotrebu oružja, kao i za štetu koju nanosi. Takođe ima geter metode za ove atribute. Potrebno je kreirati stub za ovu klasu (moguće je po želji koristiti i dodatne radne okvire, poput Mockito).

3. Klasa Odeca nije još implementirana. Odeća ima atribute za čuvanja podataka o težini odeće i odbrambenoj vrednosti odeće. Takođe ima i geter metode za ove atribute. Potrebno je kreirati stub za ovu klasu (moguće je po želji koristiti i dodatne radne okvire, poput Mockito).

4. Klasa Magija nije još implementirana. Magija ima atribute za čuvanje podataka o nazivu magije, potrebnoj inteligenciji za upotrebu magije, šteti koju nanosi, kao i energiji potrebnoj za upotrebu magije.Takođe ima i geter metode za ove atribute. Potrebno je kreirati stub za ovu klasu(moguće je po želji koristiti i dodatne radne okvire, poput Mockito).

5. U klasi Igrac postoji toString() metod koji vraća string u sledećem formatu:

(Ime, Zdravlje/Energija, STR:Snaga, INT:Inteligencija)

6. U klasi Igrac postoji javni metod koji se zove napadnilgraca koji kao argumente prima oružje koje se koristi (kao redni broj ili indeks u listi) i objekat tipa Igrac, koji predstavlja metu napada. Ova metoda vraća nanesenu štetu kao realan broj. Šteta se računa na sledeći način:

- Ukoliko igrač nema više od 20 energije, šteta se postavlja na 0.
- Ukoliko igrač ima više od 20 energije, potroši 21 bod energije.
- Određuje se šteta koju će oružje naneti. Ukoliko igrač nema dovoljno snage da koristi oružje, šteta oružja je prepolovljena.
- Osnovna formula za računanje izlazne štete je:

  $steta = stetaOruzja + (snaga * 2)$
- Zavisno od stanja u kojem se igrač nalazi, dobijena šteta se množi sa:
  - 0.8 za defanzivno stanje,
  - 1 za pasivno stanje,
  - 1.2 za agresivno stanje.

7. U klasi Igrac postoji javni metod koji se zove odbraniSe koji kao argument prima štetu koju bi igrač trebao da primi od napada. Ukoliko se ovoj metodi prosledi negativan broj kao argument, ona vraća IllegalArgumentException. Metoda vraća štetu koju bi igrač trebao da primi, nakon što se uračunaju odbrambene vrednosti odeće koju igrač nosi. Prvo je potrebno izračunati maksimalnu težinu opreme koje igrač može da nosi. To se računa kao:

$$maxTezina = snagaIgraca * 3.$$

Ukoliko je ukupna težina opreme igrača manja od izračunatog maksimuma, formula za izračunavanje konačne štete koju igrač prima je:

$$steta = PocetnaSteta / (UkupnaOdbrambenaVrednost * K)$$

Vrednost koeficijenta K zavisi od stanja u kojem se igrač nalazi i iznosi:

- 0.3 za defanzivno stanje,
- 0.2 za pasivno stanje,
- 0.1 za agresivno stanje.

Ukoliko je ukupna težina opreme igrača veća od izračunatog maksimuma, šteta koju igrač prima se računakao:

$$steta = PocetnaSteta * 1.5 / (UkupnaOdbrambenaVrednost * 0.9 * K)$$

Vrednosti K u ovom slučaju su:

- 0.25 za defanzivno stanje,
- 0.15 za pasivno stanje,
- 0.08 za agresivno stanje.

8. U klasi Igrac postoji javni metod upotrebiMagiju koji kao argument prima magiju koja se upotrebljava, u vidu rednog broja u listi magija koje igrač poznaje, kao i objekat tipa Igrac, koji predstavlja metu magije. Metod vraća nanesenu štetu. Ukoliko je igračeva inteligencija manja od inteligencije potrebne za upotrebu odabrane magije, igračevo zdravlje će se umanjiti za 10% i izgubiće svu energiju pre nego što se izračuna konačna šteta koja se nanosi. Ukoliko igrač ima dovoljno energije da upotrebi magiju, njegova energija će se umanjiti za vrednost koja je potrebna za upotrebu magije. Ukoliko igrač nema dovoljno energije da upotrebi magiju, energija će mu biti smanjenja na 0, i izgubiće onoliko bodova zdravlja, koliko je potrebno da se upotrebi magija. Ukoliko nema ni dovoljno zdravlja da pokrije cenu upotrebe magije, umesto toga će šteta magije biti postavljena na 0. Nakon ovoga, računa se konačno nanesena šteta magije po sledećoj formuli:

$$steta = stetaMagije * (inteligencijaIgraca * 0.1 + (inteligencijaIgraca - potrebnaInteligencija) + (inteligencijaIgraca - inteligencijaMete))$$

9. U klasi Igrac postoji javni metod odmoriSe koji služi za oporavljanje energije igrača. Ovaj metod ne prima nikakve argumente, a kao povratnu vrednost vraća količinu energije koja je oporavljena. U jednom pozivu, može najviše da se oporavi 50 bodova energije. Igrač ne može da ima više od 100 bodova energije. Ukoliko igrač ima više od 75 energije pri pozivu, neće biti oporavljena energija, i metod vraća 0.