DOMAĆI ZADATAK

Klasa **Aplikacija** sadrzi graficki korisnicki interfejs igre „Nim“ sa dva velika Panela:

- Prvi panel (izbor igraca, tezine, broja stubova i zetona na odgovarajucim stubovima)
- Drugi panel (igra – biranje i igranje poteza za slucaj da igra covek)

Prvi panel:

- Omogucava izbor igraca (covek vs covek , covek vs masina, masina vs masina)
- Za slucaj igranja masine (bota), otvaraju se dodatne opcije , biranje Bota (jednostavnog , jednostavnog sa alfa/beta odsecanjem i takmicarskog), i biranje tezine (dubine razvijanja stabla mogucih poteza)
- Biranje broja stubova koje otvara odgovarajuc broj polja za unos zetona za odgovarajuce stubove
- Ispisuje greske i obavestava korisnika o tome sta je neispravno uneto (nedovoljno podataka, nemoguca vrednost, slovo umesto broja, zaboravljen izbor kod radio button-a...)
- Dugme za zapocinjanje igre koje vodi do drugog panela na kom se nalazi konkretna igra

Drugi panel:

- Omogucava unos broja zetona koje igrac zeli da skine za izabranog stuba
- Obavestava korisnika prilikom greske o kakvoj je gresci rec i na kom stubu (prilikom unosa slova umesto broja, unosa broja na vise stubova, prilikom unosa nemoguce vrednosti...)
- Ispisuje koji je igrac u datom trenutku na potezu i koji je igrac pobedio na zavrsetku igre
- Sadrzi dva dugmeta za svakog od igraca, koja su dozvoljena ili zabranjena u zavisnosti od toga koji je igrac na potezu

Klasa **Algoritmi** sadrzi metode za proveru mogucnosti igranja odgovarajuceg poteza.

boolean promena(int[] stari, int[] novi, int cap);

Proverava da li je doslo do promene broja zetona na vise stubova u istom potezu , vraca false ukoliko jeste. Prihvata trenutno stanje zetona na stubovima , moguce stanje zetona na stubovima I broj stubova.

boolean valid(int noviBroj, int stariBroj, int pos, int[] stari);

Proverava da li uneti broj zetona koje korisnik zeli da skine odgovarajuc (da li je manji od dvostruke vrednosti prethodnog poteza, da li je >0 I da li je manji od trenutnog broja zetona na izabranom stubu).

boolean provera(int[] novi, int cap, int pos, int value);

Proverava da li je ce broj zetona na izabranom stubu da se razlikuje od ostalih stubova za izabranu vrednost zetona.

Klase **Cvor** I **CvorAB** sadrze potrebne promenljive koje opisuju stanje trenutnog cvora.

Vrednost izracunatu za izbor poteza, trenutno stanje zetona na stubovima, broj stubova, prethodno odigran potez, nivo(na kojoj je dubini cvor), pokazivac na svog oca u stablu I listu svoje dece.

Cvorovi iz klase CvorAB sadrze i dodatne promenljive alfa i beta i ovu klasu koriste botovi koji koriste alfa/beta odsecanje.

Klase **Bot**, **BotAB** i **Pro** su klase za igraca masinu (bota).

Svaka od ovih klasa sadrzi metode:

boolean daLiMoze(int pos, int value, int stariBroj, int[] zetoni, int brojStubova);

Ova metoda se poziva za svaki na svakom stubu , vrsi se provera sta sve moze da bude sledeci potencijalni potez. Ukoliko metoda vrati true kreaira se novi cvoj rekurzivno sa odgovarajucim vrednostima polja i sa novim stanjem zetona na stubovima.

Int racunajFjuMAX(LinkedList<Cvor> deca);

Int racunajFjuMIN(LinkedList<Cvor> deca);

Racunaju vrednost cvora u zavisnosti od toga da li je taj cvor MIN ili MAX.

racunajFjuMax/racunajFjuMIN trazi cvor koji ima najvecu/najmanju vrednost od svoje dece i tu vrednost postavlja za svoju.

Void formirajStablo(int[] zetoni, int stariBroj, int brojStubova, int nivo);

Poziva metodu formirajCvor(..) koja se rekurzivno poziva i time se formira kompletno stablo igre.

Void formirajCvor(Cvor koren, int[] zetoni, int stariBroj, int brojStubova, int nivo);

Kreira novi cvor i setuje ogovarajuce parametre. U sebi sadrzi proveru da li je dostigao maksimalnu dubinu , ukoliko jeste poziva metodu za racunanje vrednosti cvora, ukoliko nije proverava sve moguce kombinacije koristeci metodu daLiMoze i kreira svoju decu rekurzivnim pozivanjem metode formirajCvor.

U istoj metodi BotAB i Pro proveravaju da li je alfa > beta i ukoliko jeste prekinu kreiranje svoje dece i time izvrse alfa/beta odsecanje.

Int racunajFunkciju(int[] zetoni, int brojStubova);

Ovu metodu koriste Bot i BotAB. Radi se XOR broja zetona na svim stubovima i povratna vrednost je vrednost tog cvora.

Int racunajFunkciju(int[] zetoni, int brojStubova, int sum, int nivo);

Ovu metodu koristi Pro za racunanje vrednosti lista.

Ukoliko u odgovarajucem listu nije zavrsetak igre povratna vrednost je maxBrojZetona (55 za 10 stubova) – (trenutno ukupan broj zetona).

Time je obezbedjeno da ukoliko Bot ne vidi u svom stablu zavrsetak u kom pobedjuje izabere potez u kom ce da skine najveci broj zetona i time zavrsi partiju u sto manje poteza.

Ukoliko je zavrsetak partije u listu ukoliko je taj list MIN rezultat ce biti 1000 – nivo (1000 da bi MAX iznad izabrao taj cvor pored ostalih mogucnosti , a –nivo da se ukoliko u stablu postoji vise zavrsetaka igre izabere ona najbliza korenu).

Analogno tome ukoliko je cvor MAX vraca -1000 + nivo;