

Napisati PHP aplikaciju koja implementira logiku proste potezno-strateške igre.

Igrač vlada ima svojim gradom, koji imaju svoje resurse i unapređenja.

Postoje tri resursa: *drvo*, *kamen* i *zlato*.

Nova unapređenja se mogu kupiti za zlato, jedno po potezu, dok se resursi prikupljaju u rudnicima, od kojih svaki proizvodi jedan određen tip resursa.

Gradovi imaju i odbranu, sa bročano izraženom snagom. Jedan od načina da se odbrana pojača je izgradnja zidina (jedno od unapređenja).

Grad može da unajmi i nove radnike, koji će u svakom potezu odabrati jedno radno mesto na kome će raditi taj dan.

Slede detaljna uputstva za izradu zadatka:

- Interfejs **Potez** predstavlja unapređenja u kojima će nešto da se dešava u svakom potezu igre, kroz poziv *potez(Grad \$grad)* metode.
- Interfejs **RadnoMesto** predstavlja lokacije na koje Radnik može da dođe i doprinese gradu kroz metodu *radi(Radnik \$r, Grad \$grad)*.
- Klasa **Grad** ima asocijativni niz zaliha gde su ključevi resursi a vrednosti kolicina, niz unapređenja i red čekanja (niz) za izgradnju unapređenja. Klasa implementira sledeće:
 - Konstruktor prima ime grada i početno stanje resursa, a odbrana uvek počinje od nule.
 - Metoda *dodajUProizvodnju(Unapredjenje \$unapredjenje)* ubacuje novo unapređenje u red čekanja.
 - Metoda *nadjiRadnoMesto()* nasumično bira iz niza unapredjenja jedno unapredjenje, ukoliko to izabrano unapredjenje implementira interfejs RadnoMesto onda ga vraća kao rezultat, u suprotnom vraća null.
Hint: Pogledati metodu rand() iz PHPa
 - Metoda *naplati(\$resurs, \$svota)* pokušava da oduzme traženu svotu traženog resursa i vraća true ako je to uspešno, false ako nije bilo dovoljno tog resursa.
 - Metoda *dodaj(\$resurs, \$svota)* dodaje datu svotu datog resursa u svoje zalihe.
 - Metoda *odigrajPotez()* prvo prolazi kroz sva postojeća unapređenja i onim koja implementiraju Potez interfejs poziva *potez()* metod.
Hint: proveriti tipa se vrši preko instanceof
Nakon toga, ako postoji išta u redu čekanja proizvodnje, uzima se prvi element i pokušava naplatiti njegova cena u zlatu. Ako grad uspešno plati to unapređenje, ono se uklanja iz reda čekanja, prebacuje u niz unapređenja, i poziva mu se završeno metoda.
 - Metoda *pojacakOdbranu(\$poeni)* dodaje zadati broj poena na stanje odbrane grada.

- Metoda *ispisiStanje()* ispisuje ime grada, koliko kog resursa trenutno ima i koja je snaga odbrane.
- Klasa **Unapredjenje** je apstraktna i ima naziv, cenu, i apstraktnu metodu *zavrsono(Grad \$grad)*, koju će Grad da poziva svaki put kada dovrši izgradnju tog unapređenja.
- Klasa **Radnik** je **Unapredjenje** (relacija “je”) u gradu, koji implementira i Potez interfejs. Radnici mogu unutar poteza da odu na neko od unapređenja koja su i RadnoMesto i tamo pomognu. Karakterise ga snaga. Klasa implementira sledeće:
 - Konstruktor uzima snagu, ime radnika, što se prosleđuje kao naziv unapređenja, dok mu je cena fiksno 50 zlatnika.
 - Metoda *zavrsono(Grad \$grad)* ispisuje poruku koja kaže da je taj radnik stigao u grad (ime radnika i ime grada).
 - Metoda *potez(Grad \$grad)* predstavlja akciju radnika u jednom potezu, gde će on u gradu pokušati pronaći gde može da radi (koristeći *nadjiRadnoMesto()* metod). Ako radnik nadje radno mesto, onda će on poceti da radi na tom radnom mestu tako što poziva metodu *radi(Radnik \$r, Grad \$g)* sa samim sobom i prosledjenim gradom kao parametrima. Ukoliko ne dobije radno mesto treba da ispiše da je u tom gradu nezaposlen.
- Klasa **Rudnik** je **Unapredjenje** u gradu, predstavlja proizvodnju jednog od tri resursa u igri. Takođe je i RadnoMesto (implementira interfejs) gde unajmljeni radnici mogu doći da pomognu, kao i Potez jer se u rudniku može u jednom potezu proizvesti izvesna kolicina resursa. Klasa implementira sledeće:
 - Konstruktor uzima vrstu resursa koju će ovaj rudnik da proizvodi, pamti ga kao atribut, i od toga sastavlja naziv (Rudnik + resurs), dok je cena fiksno 100 zlatnika.
 - Metoda *potez(Grad \$grad)* predstavlja jedan radni dan u rudniku, gde će biti proizvedeno 5 jedinica zadatog resursa. Tu svotu treba dodati gradu i ispisati poruku o toj proizvodnji.
 - Metoda *zavrsono()* treba ispisati poruku da je grad dobio rudnik datog resursa.
 - Metoda *radi(Radnik \$r, Grad \$grad)* će biti pozvana za svakog radnika koji u tom potezu dođe da radi u rudniku. Unutar te metode, proizvesti još zadatog resursa u kolicini kolika je radnikova snaga, slično kao i u regularnoj proizvodnji.

- Klasa **Zidine** je **Unapredjenje** u gradu koje mu pojačava odbranu, koje implementira interfejs RadnoMesto. Klasa implementira sledeće:
 - U konstruktoru se fiksno zadaju naziv "Zidine" i cena 50 zlatnika.
 - Metoda *zavrseno(Grad \$grad)* ispisuje poruku da je izgradnja zidina završena i odmah datom gradu pojačava odbranu za 30 poena.
 - Metoda *radi(Radnik \$r, Grad \$grad)* predstavlja radnika koji je taj dan došao da čuva stražu i održava zidine. On tada pojačava odbranu grada za broj poena kolika je njegova snaga. Ispisati vrednost odbrane.

Radi testiranja u novom fajlu napraviti:

- Novi Grad sa proizvoljnim imenom i po 500 jedinica svakog od tri resursa.
- Dodati u proizvodnju tri nova Radnika sa proizvoljnim imenima i snagom.
- Dodati u proizvodnju tri Rudnika, za svaki od tri moguća resursa.
- Dodati u proizvodnju Zidine.
- U petlji od 10 iteracija uzastopno poziva odigrajPotez i ispisiStanje nad Gradom, kako bismo pratili napredak.