Interfejsi i nasleđivanje

Interfejs

- Samo definisanje šta se od klasa koje ga implementiraju očekuje da rade
- Kada implementacija metode nije unapred poznata
- Sve metode su podrazumevano **public**
- Sva polja su konstante podrazumevano (nepromenljve)

```
public interface Vozac{
        void skreni();
        void stani();
}
```

Primer – interfejs Comparable

- Ovo je ugrađeni interfejs
- Kada ga klasa implementira, to znači da je ona uporediva
- Mora da implementira metod compareTo
- Zadatak: napraviti dve klase Kvadrat i Krug koje implementiraju ovaj interfejs
- Porediti ih po površini

Interfejs kao tip podataka

- Ako su Kvadrat i Krug implementirali interfejs Comparable, objekti ovih klasa se mogu se referisati kao klase tipa Comparable

```
public Test{
        Comparable c1;
        Comparable c2;

        public Test(Kvadrat kv, Krug kr){
                c1=kv;
                c2=kr;
        }

        int test(){ return c1.compareTo(c2); }
}
```

Nasleđivanje interfejsa

- Jedan interfejs može naslediti drugi, i tako neograničeno, npr.:

```
public interface ZivoBice{ void zivi(); }
public interface Zivotinja extends ZivoBice{ void kreciSe(); }
public interface Biljka extends ZivoBice{ void vrsiFotosintezu(); }
public interface Vodozemac extends Zivotinja{ void plivaj(); }

public class Zaba implements Vodozemac{ …}
```

Implementacija interfejsa

- Jedna klasa može implementirati više od jednog interfejsa
- Ovo nadomešćuje nemogućnost višestrukog nasleđivanja

```
class Kvadrat implements Figura, Comparable{
        //impelmentacije metoda iz Figura i Comparable
}
```

Interfejs tipovi u kolekciji

- Ranije smo videli da interfejsi mogu “sakriti” pravi tip (klasu) nekog polja
- Ovo isto se može primeniti i u kolekciji, npr:

```
Comparable[] comps = new Comparable[3];
comps[0]=new Kvadrat(10,10,20);
comps[1]=new Krug(30,20,40);
…
```

Nastavak Zadatka 1 (sa figurama)

- Da li se nešto sa današnjeg časa može iskoristiti radi poboljšavanja programa koji iscrtava figure?
- Koji je metod zajednički za sve figure?
- Da li je implementacija tog metoda unapred poznata?
- Šta se radi ako implementacija metoda nije unapred poznata?

Nastavak Zadatka 1

- Napraviti novu verziju koda, novi paket koji ostvaruje poboljšanja pomenuta na prethodnom slajdu.
- Trebaće vam 1 interfejs sa metodom zajedničkom za sve figure
- Šta ako hoćemo da zapamtimo sve figure u objektu Platno?
- Da li ćemo imati poseban niz za Kvadrate, poseban za Tacku itd?

Nastavak Zadatka 1

- Da li se može iskoristiti samo jedan niz?
  - Uradite to.
- Da li može postojati i alternativno platno, npr. napredno platno koje pri prikazu boji unutrašnjost figura?
- Da li u tom slučaju i platno treba da bude interfejs?
- Uradite i to sa alternativnim platnom?
  - Dakle imaćemo NaprednoPlatno i ObicnoPlatno (oba implementiraju Platno)