

APSTRAKTNI TIPOVI

2016/17

APSTRAKTNI TIPOVI

U apstraktne tipove spadaju:

- Apstraktne klase
- Interfejsi

APSTRAKTNI METODI

Pretpostavite da pred sobom imate jednu oop aplikaciju za vektorsko crtanje. Neki od standardnih grafičkih objekata koje bi aplikacija trebala da nudi su: krugovi, pravougaonici, linije, Bezierove krive.

Ovi objekti imaju neka zajednička:

- stanja određena pozicijom, orijentacijom, bojom konture, bojom unutrašnjeg dela objekta itd. i
- ponašanja, kao što su pomeranje na novu poziciju (`moveTo()`), rotacija, `resize()`, iscrtavanje (`draw()`)

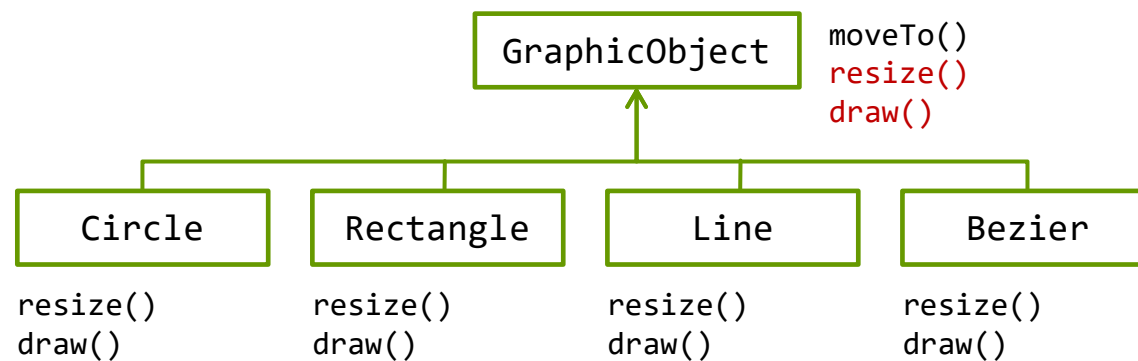
Neka od navedenih stanja i ponašanja su

- Ista za sve grafičke objekte – pozicija, boja konture i `moveTo()`, a neka zahtevaju
- potpuno različite implementacije, kao što su `resize()` i `draw()` .

Za svaki grafički objekat koji želimo da koristimo u aplikaciji, moramo da imamo definisane metode `resize()` i `draw()`, ali se ne može dati opšta implementacija ova dva metoda.

APSTRAKTNI METODI

- Dakle, bilo bi logično definisati jednu roditeljsku klasu `GraphicObject`, a zatim sve različite tipove grafičkih objekata izvesti iz nje



Problem : klasa `GraphicObject` bi morala da ima predviđene metode `resize()` i `draw()`, ali nije moguće definisati implementaciju tih metoda (bar ne na jednostavan, a opšti način, koji važi za sve vrste grafičkih objekata).

Rešenje: U objektnim jezicima je moguće metode ostaviti i bez implementacije. Takvi metodi se nazivaju **apstraktnim** i u Javi se deklariraju na sledeći način

```
abstract void metod_name(arguments);
```

apstraktni metodi nemaju implementaciju,
tj. definisano telo

APSTRAKTNE I KONKRETNE KLASSE

- Klase u kojima postoje metodi koji su apstraktni se nazivaju **apstraktnim klasama** i pri njihovom deklarisanju je neophodno, kao i kod metoda, potrebno apstraktnost naglasiti.

```
abstract class GraphicObject {  
    int x, y;  
    ...  
    void moveTo(int newX, int newY) {  
        ...  
    }  
    abstract void draw();  
    abstract void resize();  
}
```

- **Apstraktna klasa se NE MOŽE INSTANCIRATI.**
- Klase koje se mogu instancirati se nazivaju **konkretnim**.

```
public abstract class absLine {  
    int i;  
    public void a(){}  
    public abstract int b();  
}
```

APSTRAKTNA KLASA - KARAKTERISTIKE

- objedinjuje zajedničke karakteristike većeg broja klasa
- deklarise se upotrebom ključne reči `abstract`
- ne može biti instancirana, ali se može proširivati
- ako u nekoj klasi postoji apstraktna metoda tada i sama klasa mora da bude definisana kao apstraktna, obrnuto ne mora da važi, tj.

klasa se može definisati kao apstraktna, a da pri tome nema ni jedan apstraktan metod

- ukoliko podklasa neke apstraktne klase nema implementirane sve nasledene apstraktne metode onda i sama podklasa mora biti proglašena apstraktnom
- Sama apstraktna klasa se ne može instancirati, ali:
 - se mogu instancirati konkretne deca-klase
 - mogu postojati **reference tipa apstaktne klase** koje ukazuju na objekte konkretnih klasa koje su u izvedene iz apstraktne

```
Graphics g = new Graphics();
```

```
Circle c = new Circle();
```

```
Graphics g = c; ili Graphics g = new Rectangle();
```