

Selitän tämän vaihe vaiheelta, jotta voimme yhdessä ymmärtää, mitä tässä tapahtuu:

## Vaihe 1: Luonti

java

 Copy code

```
TekstiTiedostonKasittelija kasittelija = new TekstiTiedostonKasittelija();
Kayttoliittyma kayttoliittyma = new Kayttoliittyma(kasittelija);
```

Tässä **TekstiTiedostonKasittelija**-olio luodaan, ja sitten tämä olio annetaan **Kayttoliittyma**-olion luomisessa. Tässä vaiheessa tiedämme, että **TekstiTiedostonKasittelija** toteuttaa **TiedostonKasittelija**-rajapinnan. Se tarkoittaa, että **TekstiTiedostonKasittelija**-oliolla on kaikki **TiedostonKasittelija**-rajapinnassa määritellyt metodit.

## Vaihe 2: Kayttoliittyma-luokan konstruktori

java

 Copy code

```
public Kayttoliittyma(TiedostonKasittelija kasittelija) { this.tiedostonKasittelija =
kasittelija; }
```

Tässä **Kayttoliittyma**-luokan konstruktori ottaa vastaan **TiedostonKasittelija**-tyyppisen olion. Tärkeää on, että vaikka konkreettinen olio (**kasittelija**), joka annetaan konstruktorille, on **TekstiTiedostonKasittelija**, se käsitellään **TiedostonKasittelija**-rajapinnan tyyppinä. Tämä tarkoittaa, että **Kayttoliittyma**-luokan sisällä oleva **tiedostonKasittelija**-muuttuja tietää vain **TiedostonKasittelija**-rajapinnassa määritellyt metodit. Se ei tunne **TekstiTiedostonKasittelija**-luokan sisäistä toteutusta.

## Vaihe 3: Metodien kutsuminen Kayttoliittyma-luokassa

java

 Copy code

```
public void tallennaTiedosto(String tiedostonNimi, String sisalto) {
    tiedostonKasittelija.avaaTiedosto(tiedostonNimi);
    tiedostonKasittelija.kirjoitaTiedostoon(sisalto); tiedostonKasittelija.suljeTiedosto();
}
```

Tässä **tallennaTiedosto**-metodissa kutsutaan **tiedostonKasittelija**-muuttujan metodeja. Nyt on tärkeää huomata, että vaikka **tiedostonKasittelija**-muuttujaan on tallennettu **TekstiTiedostonKasittelija**-olio, nämä metodikutsut eivät ole suoraan sidottuja **TekstiTiedostonKasittelija**-luokkaan. Ne ovat sidottuja **TiedostonKasittelija**-rajapintaan. Tämä tarkoittaa, että jos päätät luoda uuden luokan, esimerkiksi **XmlTiedostonKasittelija**, joka myös toteuttaa **TiedostonKasittelija**-rajapinnan, voit muuttaa **Kayttoliittyma**-luokan käyttämään sitä muuttamatta **Kayttoliittyma**-luokan koodia.

## Johtopäätös

Vaikka konkreettisesti **Kayttoliittyma**-luokka saa **TekstiTiedostonKasittelija**-olion konstruktorissaan, se käsittelee sitä **TiedostonKasittelija**-rajapinnan kautta. Tämä eristää **Kayttoliittyma**-luokan tiedoston käsittelyn toteutuksesta ja mahdollistaa, että toteutusta voidaan vaihtaa ilman **Kayttoliittyma**-luokan muuttamista. Tämä on rajapinta-ohjelmoinnin ydin ja yksi sen suurimmista eduista: se tarjoaa joustavuuden vaihtaa toteutuksia ilman, että tarvitsee muuttaa niitä luokkia, jotka käyttävät näitä toteutuksia.