**Introduction**

Javan Bridge-suunnittelumalli on rakenteellinen (Structural) suunnittelumalli. Sen idea on irrottaa abstraktio sen toteutuksesta, niin että kumpikin puoli voi kehittyä itsenäisesti. Eli toisinsanoen ei sido tiettyyn toteutukseen, vaan siinä on viittaus rajapintaan, joita konkretiset toteutukset tarjoavat.

**New Functionality**

Esimerkissäni olen käyttänyt Refactoring.guru:n suunnittelumallin esimerkkiä. Olen lisännyt esimerkkiin SmartTv-luokan, joka implementoi Device-rajapintaa. Sen lisäksi luokassa on setApp()- sekä getApp()-metodit.

Kaukosäätimenä toimii SmartRemote-luokka, joka laajentaa Remote-rajapintaa toteuttavaa BasicRemote-luokkaa.

Kaukosäätimellä voi avata applikaatioita älytelevisiossa, kuten nykyisin kaikissa moderneissa televisioissa. Tämä uuden toiminnallisuuden idea tuli arjesta – nykyään TV:ssä käytetään enemmän sovelluksia kuin perinteisiä kanavia.

Linkki suunnittelumallin esimerkkiin:   
https://refactoring.guru/design-patterns/bridge/java/example

**Implementation**

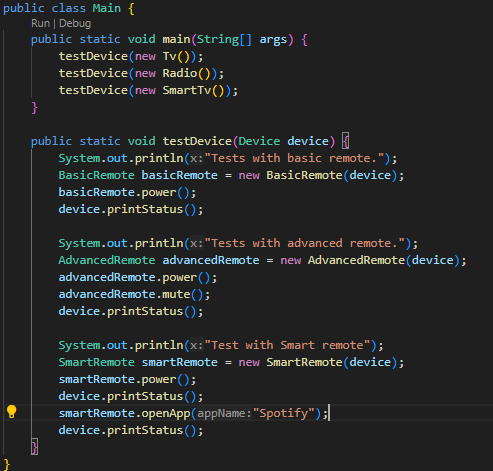
A screen shot of a computer program

AI-generated content may be incorrect.**A screenshot of a computer program

AI-generated content may be incorrect.**

**Verification**

Testasin uuden toiminnallisuuden luomalla SmartTv-olion ja ohjaamalla sitä SmartRemote-kaukosäätimellä. Alla esimerkkikoodi ja tuloste, joka vahvistaa applikaation vaihdon.

****

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Conclusion**

Tässä tehtävässä laajensin Bridge-suunnittelumallin alkuperäistä esimerkkiä lisäämällä SmartTV- sekä SmartRemote luokkia. Uuden toiminallisuuden myötä on mahdollista simuloida modernia käyttötilannetta.

Kokonaisuudessaan uusi toiminnallisuus osoittaa, kuinka Bridge-malli mahdollistaa helposti laajennettavan ja ylläpidettävän rakenteen, johon voidaan lisätä uusia laitteita ja uusia ohjaimia ilman merkittäviä muutoksia koodissa.