

Ensimmäinen uusintatentti 17.5.2019

Lue kukin tehtävänanto huolellisesti ja kokonaisuudessaan ennen kuin aloitat vastaamisen. Kustakin tehtävästä voi saada 0–6 pistettä. Tentin läpipääsyraja on 12 / 24 pistettä. Ratkaisut palautetaan tentti-WETO-järjestelmään (<https://weto-exam.uta.fi/>), jossa on kuvailtu kunkin tehtävän automaattinen testaus. Aksessorit kirjoitetaan samoin kuin harjoituksissa (<https://coursepages.uta.fi/tiea2-1a/kevat-2019/harjoitukset/ratkaisujen-tarkistus/automaattinen-tarkistus/>).

1. Tee Javalla konkreettinen *Avain*-luokka perinteisille metallisille avaimille. Luokalla on kätkeyty *String*-tyyppinen attribuutti *vari* avaimen värille. Väri ei saa olla **null**-arvoinen ja siinä on oltava vähintään kolme merkkiä.

Tee parametrin oletusrakentaja, jossa väriksi asetetaan merkkijono "musta". Määrittele attribuutille julkiset aksessorit ja tee luokalle myös yksiparametrinen rakentaja. Parametrillinen rakentaja ja asettava aksessori heittävät *IllegalArgumentException*-poikkeuksen, jos parametrina saatu tieto on virheellinen.

Korvaa *Object*-luokan *equals*-metodi siten, että siinä vertaillaan avainten värejä. Avaimet katsotaan samoiksi, jos ne ovat samanväriset.

Testaa luokkaa *AvainTesti*-luokassa vähintään siten, että aiheutat poikkeuksen antamalla parametrilliselle rakentajalle tai asettavalle aksessorille virheellisen arvon ja sieppaat poikkeuksen **try-catch**-lauseella. Laajempi testaus on toivottavaa, vaikka WETO testaa-kin luokan toiminnallisuutta varsin kattavasti. Älä jätä omaa testausta tekemättä, sillä sen puuttuminen katsotaan virheeksi.

Luokat on sijoitettava omiin tiedostoihinsa.

2. Peri edellisen tehtävän *Avain*-luokasta konkreettinen *TaikaAvain*-luokka, jolla on kätkeyty **boolean**-tyyppinen attribuutti *kertakayttoinen*. Attribuutin arvo on **true**, jos avainta voi käyttää vain kerran.

Määrittele attribuutille julkiset aksessorit. Tee parametrin oletusrakentaja, jossa taika-avaimen väriksi asetetaan merkkijono "kultainen" ja taika-avaimesta tehdään kertakäyttöinen asettamalla attribuutin arvoksi **true**.

Tee luokalle yksiparametrinen rakentaja, jolla on **boolean**-tyyppinen parametri kertakäyttöisyydestä. Kutsu tässä rakentajassa ylläluokan parametrillisesta rakentajasta siten, että annat kutsussa parametrin arvoksi "kultainen".

Korvaa *Object*-luokan *equals*-metodi. Taika-avaimet ovat samat, jos niiden värity ja käyttöisyydet ovat samat. Tee korvaus ketjuttaen: kutsu metodissa ylläluokassa korvattua *equals*-metodia ja liitä kutsun palauttamaan arvoon luokan omaan attribuuttiin liittyvä vertailutulos.

Testaa luokkaa *TaikaAvainTesti*-luokassa. Älä jätä omaa testausta tekemättä, vaikka WETO testaa luokan toiminnallisuutta, koska testauksen puuttuminen katsotaan virheeksi.

Luokat on sijoitettava omiin tiedostoihinsa. Muista palauttaa WETOon myös ylläluokka *Avain*.

Ensimmäinen uusintatentti 17.5.2019

3. Toteuta Javalla konkreettinen *Lukko*-luokka. Luokalla on kätkeyty *Avain*-tyyppinen attribuutti *avain*. Attribuutti viittaa lukon avaavaan avainolioon.

Tee attribuutille vain lukeva aksessori.

Tee parametrillinen rakentaja, jolle välitetään viite avaimen *Avain*-tyyppisen parametrin kautta. Rakentaja liittyy *avain*-attribuutin parametrin viittaamaan avainolioon. Rakentaja heittää *IllegalArgumentException*-poikkeuksen, jos parametrin arvo on **null** tai parametriin liittyy tavallisen avaimen sijasta taika-avain.

Toteuta *Lukko*-luokassa WETOssa annettu *Avautuva*-rajapinta siten, että *avaudu*-metodin paluuarvo on **true**, jos parametrina saatu viite ja attribuutti liittyvät samaan olioon tai jos parametri liittyy *TaikaAvain*-tyyppiseen olioon. *Lukko* avautuu siten tavallisella avaimella vain, jos kyseessä on avain, jota käytettiin lukon rakentamisessa. Toisaalta mikä tahansa taika-avain avaa lukon. Huomaa, että avainten *equals*-metodilla ei ole käyttöä *avaudu*-metodissa.

Testaa valaisinta *LukkoTesti*-luokassa. Älä jätä omaa testausta tekemättä, vaikka WETO testaa luokan toiminnallisuutta, koska testauksen puuttuminen katsotaan virheeksi.

Luokat on sijoitettava omiin tiedostoihinsa.

Älä muuta *Avautuva*-rajapintaa. WETO käyttää testauksessa automaattisesti alkupeleistä tiedostoa.

Muista palauttaa WETOon myös *Avain*- ja *TaikaAvain*-luokat.

4. Periytä Javan *LinkedList*-luokasta *OmaLista*-luokka ja tee sille metodi, jonka otsikko on:

```
public boolean korvaa(int ekaInd, LinkedList<E> korvaavat)
```

Metodi korvaa listan indeksiarvosta *ekaInd* alkaen *m* kpl listan tietoalkioita *m* kpl parametrina saadun listan tietoalkioilla. Listan indeksiarvoon *ekaInd* liittyvä tietoalkio korvataan parametrilistan ensimmäisellä tietoalkiolla, indeksiarvon *ekaInd* + 1 tietoalkio korvataan parametrin toisella tietoalkiolla ja niin edelleen. Lopuksi indeksiarvon *ekaInd* + *m* - 1 tietoalkio korvataan parametrilistan viimeisellä tietoalkiolla.

Korvaus onnistuu ja paluuarvo on **true**, kun *ekaInd* on positiivinen (≥ 0), *korvaavat*-lista ei ole **null**, *korvaavat*-listalla on vähintään yksi alkio ja korvattavat alkioit eivät lopu kesken eli $ekaInd + m - 1 < n$, missä *n* on listan alkioiden lukumäärä.

Korvaus epäonnistuu, listan sisältö jää entiselleen ja paluuarvo on **false**, jos *ekaInd* < 0, *korvaavat*-lista on **null** tai tyhjä tai $ekaInd + m - 1 \geq n$.

Tyhjään listaan kohdistettu korvaus epäonnistuu aina, koska mitään tehtävää ei ole.

Jos metodin parametrit ovat esimerkiksi 0 ja ["H", "E"] ja listan tietoalkiot ovat ["M", "O", "I"], on listan sisältö metodin suorituksen jälkeen ["H", "E", "I"] ja metodi palauttaa **true**-arvon, koska listan kaksi ensimmäistä alkioita voitiin korvata.

Ensimmäinen uusintatentti 17.5.2019

Jos metodin parametrit ovat toisaalta esimerkiksi 2 ja ["H", "A"] ja listan tietoalkiot ovat ["H", "E", "I"], on listan sisältö metodin suorituksen jälkeen edelleen ["H", "E", "I"] ja metodi palauttaa **false**-arvon, koska "A"-alkion tulisi korvata indeksiarvoon 3 liittyvän alkio, mutta listan viimeisen alkion indeksiarvo on 2.

Omaa testiluokkaa ei tarvitse tehdä. Omien testien tekeminen on kuitenkin toivottavaa, koska WETO ei testaa kaikkea mahdollista.