

Lisätehtävä - Luokat ja oliot

Tee projektiisi uusi tiedosto nimeltä **product.py**. Tee tähän tiedostoon luokka nimeltä **Product**.

Product –luokassa tulee olla seuraavat jäsenmuuttujat:

- id (int), oletusarvo -1
- name (string), oletusarvo "Nimi puuttuu"
- category (string), oletusarvo "Muut"
- price (float), oletusarvo 0.0
- campaign (boolean), oletusarvo False

Product –luokassa tulee olla metodi **print_info()**, joka tulostaa kaikki tiedot tuotteesta siistinä kokonaisena tekstinä:

Kategoria: Tuotteen nimi (ID) – Hinta pyöristettynä yhteen desimaaliin + € -merkki (jos tarjoustuote, tulostetaan perään: [TARJOUSTUOTE])

eli esim:

Kodinkoneet: Mikroaaltouuni 123X (ID:15) – 99.0€

tai

Puhelimet: Nokia 3310 (ID:35) – 129€ [TARJOUSTUOTE]

Tee uusi Python-tiedosto, joka käyttää **Product**-luokkaa, esim. **store.py**. Tee 5 oliota **Product**-luokasta, ja laita ne listaan, ja käy jokainen tuote silmukassa läpi. Tulosta tuotteet käyttämällä luokassa olevaa **print_info()** -metodia.

Pieni lisätehtävä:

Tee luokkaan myös metodi **get_price()**, joka antaa erikseen tuotteen hinnan. Jos campaign-boolean on True, silloin hinta on 10% pienempi.

Lisätehtävä: Järjestä tekemäsi oliolista Kategorioittain (helpoin tapa on käyttää lambdaa)

Lisätehtävä 2. Tee luokka, joka koostuu muista luokista. Tässä harjoituksessa matkitaan pieniltä osin CodeCombatissa olevaa Hero-luokkaa.

Tee **Hero**-luokka, jolla on

- Name (string)
- Character Class (string) vaihtoehdot: Warrior / Mage / Thief
- Level (int)
- Equipment (lista/dictionary)

Tee lisäksi **Equipment** –luokka:

- Name (string)
- Equipment Type (string) – vaihtoehdot: Weapon / Armor / Accessory
- Minimum Damage (int)
- Maximum Damage (int)
- Armor Rating (int)

Tee Python-ohjelma, joka näyttää yhden Heron kaikki tiedot sekä listan myös varusteista, sekä niiden ominaisuuksista. Herolla voi kerrallaan olla vain yksi ase (Weapon), suoja (Armor) sekä lisäväline (Accessory) käytössään. Voit hyödyntää Coloramaa lopputuloksen värittämiseksi!



Erillinen lisätehtävä 2. Luokkien ja olioiden muuttaminen JSONiksi ei ole aivan niin yksinkertaista kun esimerkiksi dictionaryjen kohdalla. Tämä johtuu siitä, että luokka sisältää paljon muutakin kuin pelkästään dataa, jolloin Pythonin sisäänrakennettu JSON-moduuli ei osaa sitä välttämättä käsitellä.

Yksi tapa muuntaa olioita JSONiksi ja takaisin on hyödyntää lisämoduulia: esim. ***jsonpickle***. Tee Python-ohjelma jolla voi muuntaa aiemmin tehdyn **Product**-luokan JSONiksi, ja toisinpäin!

