Luento 2 Kotitehtävä

Maksimipistemäärä 10 pistettä

Tee tehtäviä niin, että saat kerättyä yhteensä 10 pistettä.

Tehtävä 1 (3 p)

Jatketaan kirjan Java For Dummies, luvun 7 tehtävien kanssa.

Tämä on kirjan Try It Out tehtävä sivulla 190.

Käytä edellisen kotitehtävän Organization and FoodProduct luokkia.

1. Merkitse luokkien attribuutit (ominaisuudet) privateiksi (1 p)
2. Ota getterit ja setterit käyttöön. (1 p)
3. Setter metodeissa lisää tarvittavat tarkastukset, joilla varmistat, että String arvot eivät ole tyhjiä ja numeeriset arvot eivät ole negatiivisia. (1 p)

Organization set/get 1

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Multimediaohjelmisto, ohjelmisto

Kuvaus luotu automaattisesti

FoodProduct set/get

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto, Multimediaohjelmisto

Kuvaus luotu automaattisesti

Kuva, joka sisältää kohteen teksti, kuvakaappaus, ohjelmisto, näyttö

Kuvaus luotu automaattisesti

Tehtävä 2 (1 p)

Selitä jommankumman esimerkin koodi yksityiskohtaisesti, kerro mitä siinä tapahtuu ja mihin muuttujien arvot milloinkin ”liikkuvat”.

Tässä on selitys FoodProduct-luokan koodista, mukaan lukien, mitä siinä tapahtuu ja milloin muuttujien arvot päivitetään:

1. Luodaan FoodProduct-luokka, joka kuvaa elintarviketuotetta ja sisältää viisi yksityistä attribuuttia: tyyppi, paino, hinta, annostenMaara ja kaloreitaAnnostaKohti.
2. Luodaan FoodProduct-luokan konstruktori, joka ottaa vastaan nämä attribuutit ja asettaa ne luokan sisäisiin muuttujiin. Konstruktori varmistaa, että attribuuttien arvot ovat kelvollisia, koska se käyttää settereitä niiden asettamiseen.
3. setTyyppi, setPaino, setHinta, setAnnostenMaara ja setKaloreitaAnnostaKohti ovat setter-metodeja, jotka tarkistavat, että niiden saamat syötteet ovat kelvollisia. Jos ne ovat kelvollisia, ne päivittävät vastaavan attribuutin arvon. Muussa tapauksessa ne näyttävät virheviestin.
4. getTyyppi, getPaino, getHinta, getAnnostenMaara ja getKaloreitaAnnostaKohti ovat getter-metodeja, jotka palauttavat attribuutin arvon.
5. näytäTiedot-metodi tulostaa elintarviketuotteen tiedot konsoliin, mukaan lukien tyyppi, paino, hinta, annosten määrä ja kalorit annosta kohti.
6. displayInfo-metodi on yksinkertaisesti kutsu näytäTiedot-metodiin. Se on lisätty vain luettavuuden parantamiseksi.

Kokonaisuudessaan tämä koodi mahdollistaa elintarviketuotteiden luomisen, niiden tietojen päivittämisen turvallisesti settereiden avulla ja tietojen näyttämisen käyttämällä näytäTiedot- tai displayInfo-metodeja. Setterit tarkistavat, että syötteet ovat kelvollisia ennen attribuuttien päivittämistä, mikä tekee luokasta turvallisen ja ylläpitävän.

Tehtävä 3 (4 p)

Luo luokka Ajastin. Ajastimella on oliomuuttuja private int aika, parametriton konstruktori (asettaa muuttujan aika alkuarvoksi 0), sekä seuraavat neljä metodia:

1. Metodi public void lisaa() kasvattaa oliomuuttujan aika arvoa viidellä. Arvoa ei kasvateta yli 30:n.
2. Metodi public void vahenna() vähentää oliomuuttujan aika arvoa viidellä. Arvoa ei vähennetä negatiiviseksi.
3. Metodi public int aika() palauttaa oliomuuttujan aika arvon.
4. Metodi public boolean loppu() palauttaa arvon true, mikäli oliomuuttujan arvo on 30, muutoin palautetaan false.

Luo Main luokka erilliseen tiedostoon, jossa on seuraava koodi:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Aika time = new Aika();

while(!time.loppu()) {

System.out.println("Aika ei ole vielä loppu! Aika: " + time.aika());

time.lisaa();

}

System.out.println("Aika loppui! Aika: " + time.aika());

aika.vahenna();

System.out.println("Aika ei ole loppu! Aika: " + time.aika());

}

}

Tämän koodin avulla pystyt testaamaan, toimivatko luomasi metodit.

Kopioi vastaukseesi myös tulosteet.

Tehtävä 4 (Maks. 6 p)

Lue artikkeli sivulta <https://dzone.com/articles/java-getter-and-setter-basics-common-mistakes-and>

Vastaa seuraaviin kysymyksiin käyttäen artikkelia lähteenä. Lue kohtaan 6 asti. Kohta 7 saattaa tuntua jo vähän hankalalta tässä vaiheessa.

1. Mitä ovat getterit ja setterit? (1 p)

ovat kaksi tavanomaista menetelmää, joita käytetään muuttujan arvon hakemiseen ja päivittämiseen.

1. Miksi settereitä ja gettereitä tarvitaan? (1 p)

Näitä käyttämällä voidaan hallita, miten tärkeitä muuttujia käytetään ja päivitetään tarkasti, mikä auttaa varmistamaan turvallisen ja hallitun pääsyn muuttujiin ohjelman rakenteessa

1. Mitä ovat tyypillisimmät sudenkuopat? (1 p)

voivat aiheuttaa ongelmia, kuten liiallista monimutkaisuutta, jos niihin lisätään monimutkaista logiikkaa. Niiden väärinkäyttö tai puutteellinen suojaus voi myös johtaa koodin turvallisuusongelmiin ja odottamattomaan käyttäytymiseen. Siksi niitä tulisi käyttää huolellisesti ja yksinkertaisesti, ja muuttujien tulisi olla asianmukaisesti yksityisiä varmistaaksesi koodin eheyden ja ylläpidettävyyden.

1. Miten niiltä vältytään? (1 p)

Sudenkuopilta gettereiden ja settereiden käytössä Javassa voi välttyä noudattamalla muutamia periaatteita. Käytä yksinkertaisia gettereitä ja settereitä ilman monimutkaista liiketoimintalogiikkaa, merkitse muuttujat yksityisiksi (private), anna selkeät ja kuvaavat nimet metodeille, tarjoa vain tarvittavat getterit ja setterit, testaa ne huolellisesti, ja dokumentoi niiden käyttö ja rajoitukset. Näillä toimenpiteillä varmistat koodin eheyden ja ylläpidettävyyden.

1. Testaa kohtien 1-6 esimerkkejä omilla muuttujilla (1 p)

Tässä esmierkki

public class Auto {

private String merkki;

private int vuosimalli;

// Getter autojen merkille

public String getMerkki() {

return this.merkki;

}

// Setter autojen merkille

public void setMerkki(String merkki) {

this.merkki = merkki;

}

// Getter autojen vuosimallille

public int getVuosimalli() {

return this.vuosimalli;

}

// Setter autojen vuosimallille

public void setVuosimalli(int vuosimalli) {

this.vuosimalli = vuosimalli;

}

public static void main(String[] args) {

Auto auto = new Auto();

// Asetetaan auton merkki

auto.setMerkki("Toyota");

// Asetetaan auton vuosimalli

auto.setVuosimalli(2020);

// Tulostetaan auton tiedot

System.out.println("Auton merkki: " + auto.getMerkki());

System.out.println("Auton vuosimalli: " + auto.getVuosimalli());

}

}

1. Sovella sivuston koodiesimerkkejä Organization tai FoodProduct -esimerkkiin (1 op)

Kuva, joka sisältää kohteen teksti, kuvakaappaus, Multimediaohjelmisto, ohjelmisto

Kuvaus luotu automaattisesti