



Punainen – Kielletty, ei saa käyttää

Tuotos tulee luoda ilman tekoälyn apua. Opiskelijan tulee käyttää vain omia tietojaan, ymmärrystään ja taitojaan. Tekoälyn käyttäminen on kielletty perustellusta syystä ja tulkitaan vilpiksi.

Harjoitus 7

Luo Visual Studio Code:lla C# Konsoli sovellus. Nimeä projekti “**Harjoitus_V_Etunimi_Sukunimi**”. Palauta koko projekti ZIP tiedostona Moodleen. Ennen Projektin projektin paketoitua, aja Visual Studio Codessa Clean Toiminto (dotnet clean). Clean toiminto poistaa käännös aikaiset tiedostot ja pienentää ZIP paketin kokoa. Muista laadukas kommentointi tarvittaessa, se helpottaa sinua ja minua! Kommentoi vähintään tehtävä. Esimerkki alla.

```
// Tehtävä 1.  
0 references  
static void PrintMyName(string name)  
{  
    Console.WriteLine("My name is " + name);  
}
```

Tehtävissä sinulla on vapaus miettiä sopivat muuttuja tyypit, sekä muuttujien ja metodien nimet. Huomioi hyvät koodaus käytänteet.

Mikäli haluat tehtävistä palautetta, palautathan tehtävät tehtäväkohtaiseen määrä aikaan mennessä. Opintojakson tehtäviä voi palauttaa opintojakson päättymiseen asti, mutta tehtäväkohtaisen määräajan jälkeisiin palautuksiin ei tule palautetta.

Tehtävien arviointi

Tehtävistä toteutettu oikein	Arvosana
1	1
1	2
2	3

3	4
4	5

Toteuta ohjelma, jossa mallinnetaan työntekijöitä eri rooleissa (esim. ohjelmoija ja siivooja). Kaikki työntekijät jakavat yhteiset ominaisuudet, mutta vain osa voi tehdä töitä etänä. Harjoittelet abstrakteja luokkia, rajapintoja, ylikirjoittamista ja polymorfismia.

1. Toteuta Abstrakti luokka työntekijä (**Employee**)
 - a. Ominaisuudet
 - i. Nimi (**name**)
 - ii. Paikka (**position**)
 - b. Metodit
 - i. Abstrakti metodi. **Work()**
 - ii. Toteuta metodi: **PrintInfo()**, joka tulostaa työntekijän nimen ja paikan
2. Määrittele rajapinta **IRemoteWorkable**
 - a. Metodit
 - i. **StartRemoteWork()**
 - ii. **CanWorkRemotely()**, joka palauttaa bool paluuarvon.
3. Toteuta luokka Ohjelmoija (**SoftwareDeveloper**), joka perii työntekijä luokan, ja toteuttaa **IRemoteWorkable** rajapinnan
 - a. Ominaisuudet
 - i. Ohjelmointikieli (**programmingLanguage**)
 - ii. onkoKannettava (**hasLaptop**)
 - b. Metodit
 - i. Toteuta rakentaja metodi, joilla voidaan asettaa kaikki jäsenmuuttujien arvot
 - ii. Toteuta **Work()** metodi, metodi tulostaa "Nyt koodataan x", korvaa x ohjelmoijan ohjelmointikielellä

- iii. Toteuta **StartRemoteWork()** metodi, metodi tulostaa "Etätyö käyntiin"
 - iv. Toteuta **CanWorkRemotely()** metodi, Palauttaa true, jos hasLaptop == true
4. Toteuta luokka siivooja (**Janitor**), joka perii työntekijä luokan
- a. Ominaisuudet
 - i. vastuualue (**areaAssigned**)
 - b. Metodit
 - i. Toteuta rakentaja metodi, joilla voidaan asettaa kaikki jäsenmuuttujien arvot
 - ii. Toteuta **Work()** metodi, joka tulostaa "Siivotaan aluetta x", korvaa x vastuualue jäsenmuuttujan arvolla.
5. Toteuta Main metodissa seuraavat asiat
- a. Luo Ohjelmoija ja siivooja luokista oliot
 - b. Tulosta kummankin työntekijän tiedot
 - c. Kutsu **Work()** metodia molemmilla olioilla.
 - d. Tarkista voiko työntekijä tehdä etätyötä:
 - i. Jos kyllä, kutsu **StartRemoteWork()** metodia
 - ii. Jos ei, tulosta etätyö ei mahdollista.