

Päivä 6: C#- ja ASP.NET- ohjelmointia

MIMMIT KOODAA -BOOTCAMP 2020
JANI JÄRVINEN, INTERTECHNO TRAINING OY



Ohjelmalogiikkaa

Keskeisiä käsitteitä

- Muuttuja (variable)
 - Nimetty tietojen tallennuspaikka ohjelmakoodissa
- Vakiot (constants)
 - Muuttujan erikoistapaus, muuttuja jonka arvo voidaan asettaa vain kerran
- Lause (statement)
 - Yksi ohjelmointikielen käsky, päättyy useissa ohjelmointikielissä puolipisteeseen
 - Voi aiheuttaa jopa tuhansien konekielisten komentojen suorittamisen
- Tila (state)
 - Ohjelman käyttämien muuttujien arvo tietyllä ajanhetkellä

Ohjelman suoritus

- Pääsääntö: koodia suoritetaan ylhäältä alas
- Ellei suorituksen kulkua voisi muuttaa, ohjelma tekisi lähtökohtaisesti aina samoja asioita
- Kolme keskeisintä tapaa muuttaa sovelluksen suoritusta ovat:
 - Ehtolause (conditional statement)
 - Silmukka (loop)
 - Alirutiinin (subroutine), eli vaikkapa funktion kutsu (function call)
- Esimerkiksi C-perheeseen kuuluvissa ohjelmointikielissä
 - Ehtolause: "if ..."
 - Silmukka: esim. "for ..." tai "while ..."
 - Kutsu: "metodi()"

Ehtolauseet

- Englanniksi *conditional statements*
- Tunnistaa monissa ohjelmointikielissä komennosta "if"
- If-sanan jälkeen on jokin ehto (condition), jonka tuloksena täytyy olla totuusarvo – eli kyllä/ei
- Tottuusarvoja kutsutaan usein myös Boolean-arvoiksi
- Esimerkiksi
 - `if (luku == 4) { ... }`
- Ehtolauseen jälkeen tuleva lause tai lohko (block) suoritetaan, jos ehto täyttyy
- Ehtolause voi sisältää myös Else-osan, joka suoritetaan, jos ehto ei täyty

Silmukat

- Englanniksi *loop*
- Mahdollistavat yhden tai useamman lauseen suorittamisen useita kertoja
- Silmukoita on tyypillisesti ohjelmointikielissä muutamia erilaisia
 - Tietyn, ennalta tiedetyn lukumäärän mukaisesti: "for"
 - Jonkin taulukon, listan tai kokoelman alkioden käsittely: "foreach"
 - Niin kauan kuin annettu ehto täyttyy: "while"
- Esimerkiksi
 - `for (int index = 0; index < 10; index++) { ... }`
 - `while (i < 10) { ... }`

Esimerkki C#-koodista

```
public static void Main()
{
    // muuttujan määrittely
    int counter;

    // sijoituslause (assignment), eli muuttujan arvon asettaminen
    counter = 1;

    // taulukon (array) määrittely
    int[] radii = { 15, 32, 108, 74, 9 }; const double pi = 3.14159;

    // foreach-silmukka
    foreach (int radius in radii)
    {
        // muuttujan määrittely
        double circumference = pi * (2 * radius);

        Console.WriteLine("Radius of circle #{0} is {1}. Circumference = {2:N2}",
            counter, radius, circumference);

        // laskurin arvon kasvatus
        counter++;
    }
}
```

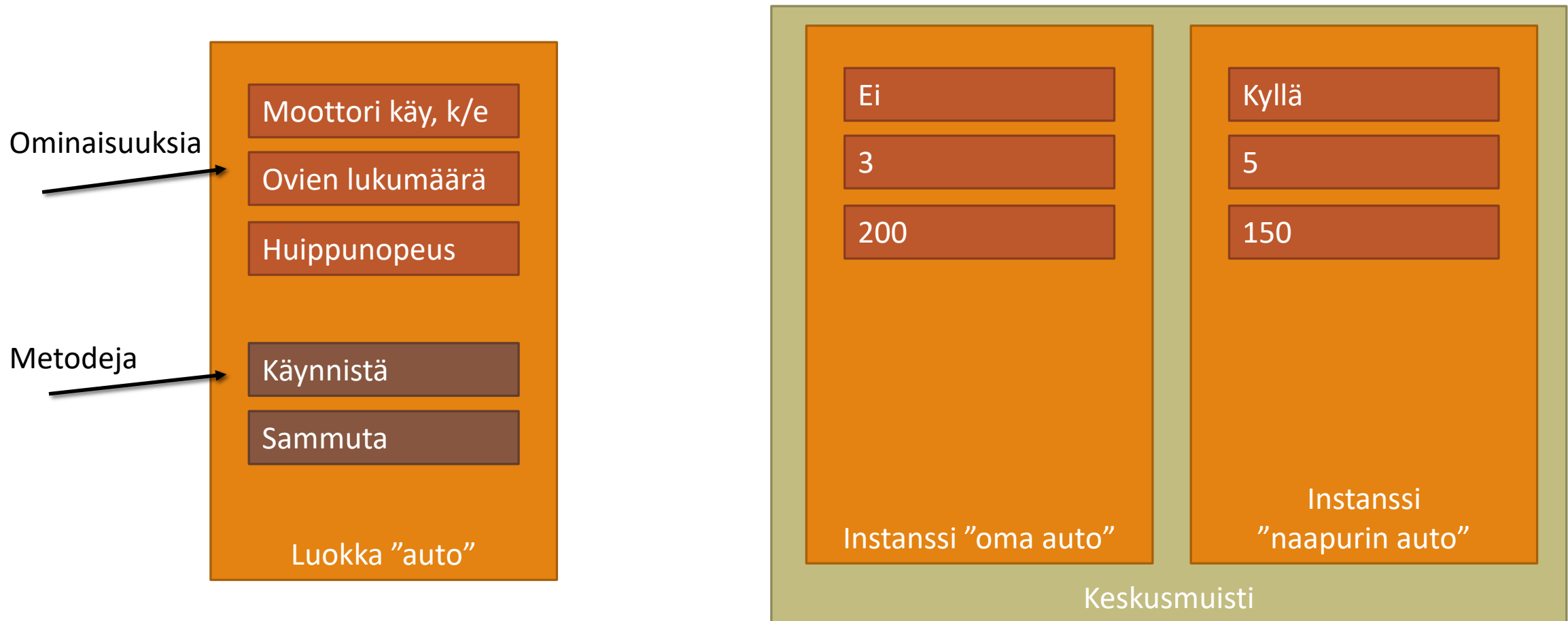
Olio-ohjelmoinnin käsitteitä

- Olio-ohjelmointi on keskeinen osa nykyaikaista sovelluskehitystä, myös .NET-alustalla
- Tärkeitä käsitteitä
 - Luokka, instanssi
 - Metodi, ominaisuus, kenttä
 - Periytyvyys, *inheritance*
 - Näkyvyysmääreet
 - Rajapinnat, *interface*
- C#-kielen huomioita
 - Ei moniperintää (multiple inheritance) kuten esim. C++:ssa

Luokat ja oliot .NET-ympäristössä

- C#-kielessä ja .NET-ohjelmoinnissa yleensäkin kaikki oliot periytyvät System.Objectista
- Oliota ei tarvitse määritellä yhdellä kertaa
 - Partial classes ja partial methods
- C#-esimerkki
 - `public class Customer {`
 ...
 }
 - `Customer cust = new Customer();`
- This-avainsana
 - Viittaa ”tähän olioon”

Luokka ja instanssi eli olio



Web-ohjelmointia

HTTP, REST ja JSON pähkinänkuoressa

- Lyhenteet

- HTTP = HyperText Transfer Protocol
- REST = Representational State Transfer
- JSON = JavaScript Object Notation

- REST (Roy Fielding 2000, HTTP/1.1 1999)

- Arkkitehtuuri- ja ajattelutapa web-palveluiden suunnitteluun ja toteutukseen

- JSON (Douglas Crockford 2001)

- Tekstipohjainen olioiden serialisointi-syntaksi samaan tapaan kuin XML, mutta käyttäen JavaScript-kielen kanssa yhteensopivaa mallia

ASP.NET MVC:n arkkitehtuuri

- MVC = Model, View, Controller
- Controller-luokka ottaa vastaan pyyntöjä ja käsittelee ne
 - Päättää, mikä näkymä selaimelle palautetaan
 - Kontrollerin julkiset metodit ovat nimeltään toimintometodeja (action methods)
- View eli näkymä muodostaa tiedoista HTML:ää eli "näyttää" halutut tiedot käyttäjän selaimessa
- Näkymä (view) voi olla myös koneellisesti käsiteltävää tietoa, kuten JSON:a
- Model eli malli tarjoaa tietomallin, esim. olio joka vastaa SQL-tietokannan taulurakennetta



Moderni Web API -suunnittelu

- RESTful-ajattelutapa
 - HTTP-protokollan hyödyntäminen, verbit ja metodit (GET, POST, PUT, PATCH ja DELETE)
 - Selkeät URL-osoitteet
 - Salaus (SSL/TLS)
- Olioiden/resurssien noutaminen
 - CRUD-operaatiot (Create, Read, Update, Delete)
 - Monipuolisten hakuehtojen yhdistäminen, esim. suodatus (filtering) ja lajittelu (sorting)
- Muita näkökohtia
 - Versiointi
 - Suorituskyky ja skaalautuvuus
 - Pilvipalvelut
 - Dokumentaatio

Harjoituksia

1. Tutustu "dotnet new" -komentoon.
 1. Mitä kaikkea sillä voi tehdä?
 2. Mistä löydät listan komennon tukemista sovellusmalleista (templates)?
2. Toteuta ASP.NET-tekniikkaa käyttäen MVC-sovellus, jonka käyttöliittymää muokkaat omiin tarpeisiin
 1. Muokkaa etusivun tekstejä haluamallasi tavalla
 2. Lisää kokonaan uusi sivu, jonka linkität yläpalkkiin
3. Kirjoita Web API -kontrolleri, joka palauttaa erilaisia tietotyyppejä
 - Yksittäisen luvun
 - Merkkijonoja (taulukon)
 - Listan kehittyneempiä olioita, esim. itse tekemäsi luokan instansseja