

//Juha Savikko

// Ohjelmoinnin viikkoharjoitukset – 4 viikkoharjoitusten ratkaisut ja tehdyt muutokset

Tehtävät 1 - 4

Main.cpp

```
#include <iostream>
#include <string>
#include <vector>
#include "henkilo.h"
#include "Paivays.h"
#include "Osoite.h"
#include "Kalenterimerkinta.h"
using namespace std;

// Ilman viittausta == arvoparametri ( arvo kopioidaan )
// Viittauksen kanssa == muuttujaparametri ( itse muuttujan viite (osoite) menee
// aliohjelmalle)
// Viittausparametrilla voimme muuttaa itse muuttujaa funktiossa (aliohjelmassa)

void doSomethingArvoparametri(Kalenterimerkinta aKalenterimerkinta) {
    aKalenterimerkinta.setAsia("Kay kaupassa");
    aKalenterimerkinta.tulostaMerkinta();
    return;
}

// 1. Sääntö == Välitä aina oliot funktioille ja metroidelle viitteenä
// arvoparametri johtaa kopiomiseen ja on HIDAS
// 2. Sääntö == jos funktion tai metodin ei ole tarkoitus muuttaa oliota vaan vain
// käyttää
// välitä olioviite const -viitteenä. (metodi tai funktio ei voi muuttaa olion tilaa)
void doSomethingViittausparametri(Kalenterimerkinta& aKalenterimerkinta) {
    aKalenterimerkinta.setAsia("Kay kaupassa");
    cout << "doSomethingViittausparametri ilman constia\n";
    aKalenterimerkinta.tulostaMerkinta();
    return;
}

void doSomethingViittausparametri(const Kalenterimerkinta& aKalenterimerkinta) {
    //aKalenterimerkinta.setAsia("Käy kaupassa");
    cout << "doSomethingViittausparametri constilla\n";
    aKalenterimerkinta.tulostaMerkinta();
    return;
}

int main() {

    Kalenterimerkinta merkinta;
    //Tehtava 1;
    cout << "\nTehtava 1\n";
    cout << "Kopiorakentaja luotu, kutsutaan nyt kopiorakentajaa
kalenterimerkinnälle";
    Kalenterimerkinta kopioMerkinta(merkinta);

    //Tehtava 2;
    cout << "\nTehtava 2\n";
    merkinta.setAsia("Ei kannata kayda kaupassa");
    merkinta.tulostaMerkinta();
    doSomethingArvoparametri(merkinta);
}
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

```
doSomethingViittausparametri(merkinta);
cout << "doSomethingViittausparametrin jalkeen\n";
merkinta.tulostaMerkinta();

//Tehtava 3;
cout << "\nTehtava 3\n";
int choice = 3;
vector<Henkilo*> henkilot;
int ika;
string nimi;
Osoite aOsoite;
string katuosoite;
string postinumero;
string kunta;
cout << "\nValitse valilta 1-3\n";
cout << "1. Lisaa Henkilo\n2. Tulosta henkilot\n3. Lopeta\n";
cin >> choice;
while (choice) {
    if (choice == 1) {
        cout << "Syota henkilon nimi: \n";
        cin >> nimi;
        cout << "Syota henkilon ika: \n";
        cin >> ika;
        cout << "Syota henkilon osoite: \n";
        cin >> katuosoite;
        aOsoite.setKatuosoite(katuosoite);
        cout << "Syota postinumero: \n";
        cin >> postinumero;
        aOsoite.setPostinumero(postinumero);
        cout << "Syota kunta: \n";
        cin >> kunta;
        aOsoite.setKunta(kunta);

        henkilot.push_back(new Henkilo(nimi, ika,
aOsoite));
    }
    if (choice == 2) {
        for (Henkilo* h : henkilot) {
            // tulostetaan koko vektorissa
            h->tulostaHenkilonTiedot();
        }
        cout << "Tulostetaan henkilöiden tiedot\n";
    }
    if (choice == 3) {
        break;
    }
    if (choice == 4) {
        cout << "Syota poistettavan henkilon nimi";
        cin >> nimi;
        for (Henkilo* h : henkilot) {
            // poistetaan henkilö
        }
    }
    cout << "\nValitse valilta 1-3\n";
    cout << "1. Lisaa Henkilo\n2. Tulosta henkilot\n3. Lopeta\n";
    cin >> choice;
}
//dynaaminen henkilö -vektori (koko sen mukaan, montako vektoriin
lisäilläään)
//henkilot.at(0).tulostaHenkilonTiedot();
system("pause");
return EXIT_SUCCESS;
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

}

Henkilo.cpp

```
Henkilo::Henkilo(const Henkilo& aHenkilo)
    :mNimi(aHenkilo.mNimi), mIka(aHenkilo.mIka), mOsoite(aHenkilo.mOsoite)
{
    cout << "Henkilo -luokan kopiorakentaja " << endl;
}
```

Henkilo.h

```
Henkilo(const Henkilo& aHenkilo);
```

Osoite.cpp

Ei muutoksia

Osoite.h

Ei muutoksia

Paivays.cpp

```
Paivays::Paivays(const Paivays& aPaivays)
    :mPaiva(aPaivays.mPaiva), mKuukausi(aPaivays.mKuukausi),
mVuosi(aPaivays.mVuosi)
{
    cout << "Paivays -luokan kopiorakentaja " << endl;
}
```

Paivays.h

```
Paivays(const Paivays& aPaivays);
```

Kalenterimerkinta.cpp

```
Kalenterimerkinta::Kalenterimerkinta(const Kalenterimerkinta& aKalenterimerkinta)
    :mPaivays(aKalenterimerkinta.mPaivays), mAsia(aKalenterimerkinta.mAsia),
mMuistutus(aKalenterimerkinta.mMuistutus)
{
    cout << "Kalenterimerkinta -luokan kopiorakentaja " << endl;
}
```

Kalenterimerkinta.h

```
Kalenterimerkinta( const Kalenterimerkinta& aKalenterimerkinta);
```