

//Juha Savikko

//Ohjelmoinnin viikkoharjoitukset – 3 viikkoharjoitusten ratkaisut

Main.cpp

```
#if 0
#include <iostream>
#include <string>
#include "henkilo.h"
#include "Paivays.h"
#include "Osoite.h"
using namespace std;

int main() {

    Osoite juhanOsoite("Teiskontie", "33540", "Tampere");

    Henkilo juha("Juha", 28, juhanOsoite);
    juha.tervehdi();

    Henkilo ville("Ville", 30, Osoite("Hämeenkatu 1", "33100", "Tampere"));
    ville.tervehdi();

    system("pause");
    return EXIT_SUCCESS;
}

#endif

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

int main() {
    Kalenterimerkinta oppitunti("Ohjelmoinnin luennot", Paivays(30, 3, 2021),
true);
    oppitunti.kysyTiedot();

    oppitunti.tulostaMerkinta();

    system("pause");
    return EXIT_SUCCESS;
}
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

Henkilo.cpp

Henkilo.h

```
#pragma once
#include <iostream>
#include <string>
#include "osoite.h"
using namespace std;

class Henkilo {
public:
    Henkilo();
    Henkilo(string aNimi, int aIka);
    Henkilo(string aNimi, int aIka, Osoite aOsoite);
    ~Henkilo();

    void setNimi(string aNimi);
    void setIka(int aIka);
    void setOsoite(Osoite aOsoite);
    Osoite getOsoite() const;

    string getNimi() const;
    int getIka() const;

    void tervehdi() const;
    void tulostaHenkilonTiedot() const;
    void kasva();

private:
    Osoite mOsoite;
    string mNimi;
    int mIka;
};
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

Osoite.cpp

```
#include "Osoite.h"
#include <iostream>
using namespace std;

Osoite::Osoite(): mKatuosoite("N/A"), mPostinumero("N/A"), mKunta("N/A")
{
    cout << "Osoite -luokan oletusrakentaja (parametrin rakentaja)\n";
}

Osoite::Osoite(string aKatuosoite, string aPostinumero, string aKunta)
:mKatuosoite(aKatuosoite), mPostinumero(aPostinumero), mKunta(aKunta)
{
    cout << "Osoite -luoka 3 parametrinen rakentaja\n";
}

Osoite::~Osoite()
{
    cout << "Osoite -luokan purkaja\n";
}

string Osoite::getKatuosoite() const
{
    return mKatuosoite;
}

string Osoite::getPostinumero() const
{
    return mPostinumero;
}

string Osoite::getKunta() const
{
    return mKunta;
}

void Osoite::setKatuosoite(string aKatuosoite)
{
    mKatuosoite = aKatuosoite;
}

void Osoite::setPostinumero(string aPostinumero)
{
    mPostinumero = aPostinumero;
}

void Osoite::setKunta(string aKunta)
{
    mKunta = aKunta;
}

void Osoite::tulostaTiedot() const
{
    cout << "Katuosoite: " << mKatuosoite << endl;
    cout << "Postinumero: " << mPostinumero << endl;
    cout << "Kunta: " << mKunta << endl;
}
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

Osoite.h

```
#pragma once
#include <string>
#include <iostream>
using namespace std;

class Osoite
{
public:
    Osoite();
    Osoite(string aKatuosoite, string aPostinumero, string aKunta);
    ~Osoite();
    string getKatuosoite() const;
    string getPostinumero() const;
    string getKunta() const;
    void setKatuosoite(string aKatuosoite);
    void setPostinumero(string aPostinumero);
    void setKunta(string aKunta);
    void tulostaTiedot() const;

private:
    string mKatuosoite;
    string mPostinumero;
    string mKunta;
};
```

//Juha Savikko

Paivays.cpp

```
#include <iostream>
#include <string>
#include "Paivays.h"
using namespace std;
Paivays::Paivays() :
    mPaiva(1),mKuukausi(1),mVuosi(2000)
{
    cout << "Paivays -luokan parametrin rakentaja\n";
}
Paivays::Paivays(int aPaiva, int aKuukausi, int aVuosi)
    : mPaiva(aPaiva),mKuukausi(aKuukausi),mVuosi(aVuosi)
{
    cout << "Paivays -luokan 3 parametrinen rakentaja\n";
}
Paivays::~~Paivays()
{
    cout << "Paivays -luokan purkaja " << endl;
}
void Paivays::setPaiva(int aPaiva)
{
    if (aPaiva >= 1 && aPaiva <= tamanKuukaudenPituus()) {
        mPaiva = aPaiva;
    }
    else {
        //bugi jossain!
    }
}
void Paivays::setKuukausi(int aKuukausi)
{
    if (aKuukausi >= 1 && aKuukausi <= 12) {
        mKuukausi = aKuukausi;
    }
}
void Paivays::setVuosi(int aVuosi)
{
    mVuosi = aVuosi;
}
int Paivays::getPaiva() const
{
    return mPaiva;
}
int Paivays::getKuukausi() const
{
    return mKuukausi;
}
int Paivays::getVuosi() const
{
    return mVuosi;
}
void Paivays::tulostaPaivays() const
{
    cout << "Paivamaara: " << mPaiva << "/" << mKuukausi << "/" << mVuosi <<
endl;
}
void Paivays::kysyPaiva()
{
    int syote;
    cout << "Syota vuosi: " << endl;
    cin >> syote;
}
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

```
        setVuosi(syote);

        cout << "Syota kuukausi: " << endl;
        cin >> syote;
        setKuukausi(syote);

        cout << "Syota paiva: " << endl;
        cin >> syote;
        setPaiva(syote);
    }
    void Paivays::kasvata()
    {
        //cout << "\nKasvatetaan paivaa." << endl;
        if (mPaiva < tamanKuukaudenPituus()) {
            mPaiva++;
        }
        else {
            mPaiva = 1;
            if (mKuukausi < 12) {
                mKuukausi++;
            }
            else {
                mKuukausi = 1;
                mVuosi++;
            }
        }
    }
}
int Paivays::tamanKuukaudenPituus() const {
    if (mKuukausi == 4 || mKuukausi == 6 || mKuukausi == 9 || mKuukausi ==
11) {
        return 30;
    }
    else if (mKuukausi == 2) {
        if (mVuosi % 4 == 0) {
            return 29;
        }
        return 28;
    }
    return 31;
}
```

//Juha Savikko

Paivays.h

```
#pragma once
#include <iostream>
#include <string>
using namespace std;

//Määritellään class Päiväys
class Paivays {
public:
    // Rakentajat (constructors) == Metodi, joka alustaa olion
    // Rakentaja on metodi, jolla on luokan nimi ilman paluuarvoa
    // Jos emme kirjoita yhtään rakentajaa, kääntäjä tuottaa ns. Oletus
rakentajan,    // joka on parametrin rakentaja

    // IMPORTANT---- Merkitse kaikki sellaiset metodit "const:ksi, jotka eivät
muuta        // olion sisäistä tilaa, eli muokkaa olion tietojäsenten arvoa.

    Paivays();
    Paivays(int aPaiva, int aKuukausi, int aVuosi);

    // Purkajat (destructor) == metodi, jota kutsutaan, kun olio siivotaan
muistista (vapautuu)
    ~Paivays();

    void setPaiva(int aPaiva);
    void setKuukausi(int aKuukausi);
    void setVuosi(int aVuosi);

    int getPaiva() const;
    int getKuukausi() const;
    int getVuosi() const;

    void tulostaPaivays() const;
    void kysyPaiva();
    void kasvata();

    int tamanKuukaudenPituus() const;
private:

    // luokalla voi olla yksityisiä metodeita, jotka ns. apumetodeita
    // luokan sisäiseen käyttöön. Tällaisia ei voida kutsua muuten kuin
luokan omasta koodista

    int mPaiva;
    int mKuukausi;
    int mVuosi;

};
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

Kalenterimerkinta.cpp

```
#include "Kalenterimerkinta.h"
#include <iostream>
#include <string>
using namespace std;

Kalenterimerkinta::Kalenterimerkinta(): mPaivays(), mAsia(), mMuistutus(false)
{
    cout << "Kalenterimerkinta -luokan parametrin rakentaja\n";
}

Kalenterimerkinta::Kalenterimerkinta(string aAsia, Paivays aPaivays, bool aMuistutus):
    mPaivays(aPaivays), mAsia(aAsia), mMuistutus(aMuistutus)
{
    cout << "Kalenterimerkinta -luokan 3 parametrinen rakentaja" << endl;
}

Kalenterimerkinta::~Kalenterimerkinta()
{
    cout << "Kalenterimerkinta -luokan purkaja" << endl;
}

void Kalenterimerkinta::setAsia(string aAsia)
{
    mAsia = aAsia;
}

void Kalenterimerkinta::setMuistutus(bool aMuistutus)
{
    mMuistutus = aMuistutus;
}

void Kalenterimerkinta::setPaivays(Paivays aPaivays)
{
    aPaivays = mPaivays;
}

string Kalenterimerkinta::getAsia() const
{
    return mAsia;
}

bool Kalenterimerkinta::getMuistutus() const
{
    return mMuistutus;
}

Paivays Kalenterimerkinta::getPaivays() const
{
    return mPaivays;
}

void Kalenterimerkinta::kysyTiedot()
{
    cout << "Muistutuksen asia? ";
    cin >> mAsia;

    mPaivays.kysyPaiva();

    char vastaus;
    cout << "Asetetaanko muistutus (K/E): ";
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

```
        cin >> vastaus;
        if (vastaus == 'k' || vastaus == 'K') {
            mMuistutus = true;
        }
        else {
            mMuistutus = false;
        }
    }
    void Kalenterimerkinta::tulostaMerkinta() const
    {
        cout << "Asia: " << mAsia << endl;
        mPaivays.tulostaPaivays();
        if (mMuistutus) {
            cout << "Muistutus asetettu\n";
        }
        else {
            cout << "Ei muistutusta\n";
        }
    }
}
```

Kalenterimerkinta.h#pragma once

```
#include <iostream>
#include <string>
#include "Paivays.h"
using namespace std;

class Kalenterimerkinta
{
public:
    Kalenterimerkinta();
    Kalenterimerkinta(string aAsia, Paivays aPaivays, bool aMuistutus);
    ~Kalenterimerkinta();

    void setAsia(string aAsia);
    void setMuistutus(bool aMuistutus);
    void setPaivays(Paivays aPaivays);

    string getAsia() const;
    bool getMuistutus() const;
    Paivays getPaivays() const;

    void tulostaMerkinta() const;
    void kysyTiedot();

private:
    Paivays mPaivays;
    string mAsia;
    bool mMuistutus;
};
```

//Ohjelmoinnin edistyneet piirteet 3 viikkoharjoitusten ratkaisut

//Juha Savikko

Kalenterimerkinta.h

```
#pragma once
#include <iostream>
#include <string>
#include "Paivays.h"
using namespace std;

class Kalenterimerkinta
{
public:
    Kalenterimerkinta();
    Kalenterimerkinta(string aAsia, Paivays aPaivays, bool aMuistutus);
    ~Kalenterimerkinta();

    void setAsia(string aAsia);
    void setMuistutus(bool aMuistutus);
    void setPaivays(Paivays aPaivays);

    string getAsia() const;
    bool getMuistutus() const;
    Paivays getPaivays() const;

    void tulostaMerkinta() const;
    void kysyTiedot();

private:
    Paivays mPaivays;
    string mAsia;
    bool mMuistutus;
};
```