

# Main.cpp

---

```
#include "henkilo.h"
#include "paivays.h"
#include "noppa.h"
#include "Osoite.h"
#include "Kalenterimerkinta.h"

void testiohjelma() {
    Paivays tanaan;
    tanaan.tulostaPaivaysHuomenna();    //purkaja aktivoituu ennen paluuta mainiin, tuhoaa oli
}

int main() {

    Osoite osoite("Teiskontie", "33540", "Tampere");
    osoite.tulostaTiedot();

    henkilo tyyppi;
    tyyppi.kysyTiedot();
    tyyppi.tulostaHenkilonTiedot();

    Kalenterimerkinta merkinta;
    merkinta.kysyTiedot();
    merkinta.tulostaKalenterimerkinta();

    return 0;
}
```

## henkilo.h

---

```
#pragma once
#include <iostream>
#include <string>
#include "Osoite.h"

// Henkilö - luokan esittely (kirjoitetaan omaan .h-tiedostoon)

class henkilo
{
    // luokan julkinen rajapinta (API), toiminnot, metodit
public:
```

```

//Constructors:
henkilo();
henkilo(std::string aNimi, int aIka);
henkilo(std::string aNimi, int aIka, Osoite aOsoite);

//Destructors:
~henkilo();

void kysyTiedot();
void setNimi(std::string aNimi);
void setIka(int aIka);
int getIka() const;
std::string getNimi() const;
void tulostaHenkilonTiedot() const;
void kasva();

Osoite getOsoite() const;
void setOsoite(Osoite aOsoite);

// Luokan tietojen asetukset, yksityinen osuus (yleensä private)
// Nimiin perustuen kutsuksi vain luokan metodeissa
private:

std::string nimi;
int ika;
Osoite osoite;
};

```

## henkilo.cpp

---

```

#include "henkilo.h"
#include "Osoite.h"
#include <istream>

// Henkilö - luokan määrittely eli toteutus (implementation)
//

henkilo::henkilo() : nimi("---"), ika(0)
{
    std::cout << "Luodaan parametrin rakentaja" << std::endl;
}

henkilo::henkilo(std::string aNimi, int aIka) : nimi(aNimi), ika(aIka)
{
    std::cout << "Luodaan parametrinen henkilo-rakentaja" << std::endl;
}

```

```
henkilo::henkilo(std::string aNimi, int aIka, Osoite aOsoite)
    : nimi (aNimi), ika(aIka), osoite(aOsoite)
{
    std::cout << "henkilo-rakentaja osoite-luokan kanssa aktivoitu" << std::endl;
}

henkilo::~henkilo()
{
    std::cout << "Henkilo-luokan purkaja aktivoitu" << std::endl;
}

void henkilo::kysyTiedot()
{
    std::cout << "Tervetuloa käyttämään ohjelmaa!" << std::endl;
    std::cout << "Anna nimi ";
    std::getline(std::cin, nimi);
    std::cout << "Anna ika: ";
    std::cin >> ika;
    std::cin.ignore(80, '\n');
    osoite.kysyOsoitetiedot();
}

void henkilo::setNimi(std::string aNimi) {
    // asetetaan henkilölle nimi
    nimi = aNimi;
}

void henkilo::setIka(int aIka)
{
    ika = aIka;
}

int henkilo::getIka() const
{
    return ika;
}

std::string henkilo::getNimi() const
{
    return nimi;
}

void henkilo::tulostaHenkilonTiedot() const
{
    std::cout << "Henkilon tiedot" << std::endl;
    std::cout << "Nimi: " << nimi << std::endl;
    std::cout << "Ika: " << ika << std::endl;
    osoite.tulostaTiedot();
}

void henkilo::kasva()
{
    ika++;
}
```

```
Osoite henkilo::getOsoite() const
{
    return Osoite();
}

void henkilo::setOsoite(Osoite aOsoite)
{
    osoite = aOsoite;
}
```

## osoite.h

---

```
#pragma once
#include <iostream>
#include <string>

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

private:
    std::string katuosoite;
    std::string kunta;
    std::string postinumero;
};
```

## Osoite.cpp

---

```
#include "Osoite.h"
```

```
Osoite::Osoite() : kunta("N/A"), postinnumero("N/A"), katuosoite("N/A")
{
    std::cout << "Osoite-luokan parametrin rakentaja aktivoitu" << std::endl;
}

Osoite::Osoite(std::string aKatuosoite, std::string aKunta, std::string aPostinnumero)
    : katuosoite(aKatuosoite), postinnumero(aPostinnumero), kunta(aKunta)
{
    std::cout << "Osoite-luokan parametrinen rakentaja aktivoitu" << std::endl;
}

Osoite::~~Osoite()
{
    std::cout << "Osoite-luokan purkaja aktivoitu" << std::endl;
}

std::string Osoite::getKatuosoite() const
{
    return katuosoite;
}

std::string Osoite::getPostinnumero() const
{
    return postinnumero;
}

std::string Osoite::getKunta() const
{
    return kunta;
}

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

void Osoite::setPostinnumero(std::string aPostinnumero)
{
    postinnumero = aPostinnumero;
}

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

void Osoite::tulostaTiedot() const
{
    std::cout << "Katuosoite on: " << katuosoite << std::endl;
    std::cout << "Postinnumero on: " << postinnumero << std::endl;
    std::cout << "Kunta on: " << kunta << std::endl;
}

void Osoite::kysyOsoitetiedot()
{

```

```
std::cout << "Kerro osoitteesi: ";
std::getline(std::cin, katuosoite);
std::cout << "Kerro postinumerosi: ";
std::getline(std::cin, postinumbero);
std::cout << "Kerro kotikuntasi: ";
std::getline(std::cin, kunta);
std::cout << std::endl;
}
```

## paivays.h

---

```
#pragma once
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <string>
#include <ctime>

class Paivays
{
public:
    // Constructors:
    Paivays();
    Paivays(int aPaiva, int aKuukausi, int aVuosi);

    // Destructors:
    ~Paivays();

    void setPaiva(int aPaiva);
    void setKuukausi(int aKuukausi);
    void setVuosi(int aVuosi);
    int getPaiva() const;
    int getKuukausi() const;
    int getVuosi() const;
    void tulostaPaivaysTanaan() const;
    void tulostaPaivaysHuomenna() const;
    void kysyPaivays();
    void kasvataPaivaysta();
    int tarkistaKuukausi(int aSyote);

private:
    int paiva;
    int kuukausi;
    int vuosi;
    time_t now = time(0);
    tm* ltm = localtime(&now);
};
```

# paivays.cpp

---

```
#include "paivays.h"

Paivays::Paivays() : paiva(0), kuukausi(0), vuosi(2000)
{
    std::cout << "Luodaan Paivays oletusrakentaja" << std::endl;
}

Paivays::Paivays(int aPaiva, int aKuukausi, int aVuosi)
    : paiva(aPaiva), kuukausi(aKuukausi), vuosi(aVuosi)
{
    std::cout << "Luodaan 3-parametrinen rakentaja" << std::endl;
}

Paivays::~Paivays()
{
    std::cout << "Paivays-luokan purkaja aktivoitu" << std::endl;
}

// luokan metodien koodi alkaa t?st?

void Paivays::setPaiva(int aPaiva)
{
    if (aPaiva > 0 && aPaiva <= 31)
        paiva = aPaiva;
    else
        paiva = ltm->tm_mday;
}

void Paivays::setKuukausi(int aKuukausi)
{
    if (aKuukausi > 0 && aKuukausi < 13)
        kuukausi = aKuukausi;
    else
        std::cout << "Sy?tit liian suuren tai pienen kuukauden, tulostetaan nykyinen kuukausi"
        kuukausi = 1 + ltm->tm_mon;
}

void Paivays::setVuosi(int aVuosi)
{
    if (aVuosi > 0)
        vuosi = aVuosi;
    else
        vuosi = 1900 + ltm->tm_year;
}

int Paivays::getPaiva() const
```

```
{
    return paiva;
}

int Paivays::getKuukausi() const
{
    return kuukausi;
}

int Paivays::getVuosi() const
{
    return vuosi;
}

void Paivays::tulostaPaivaysTanaan() const
{
    std::cout << paiva << "." << kuukausi << "." << vuosi << std::endl;
}

void Paivays::tulostaPaivaysHuomenna() const
{
    if (kuukausi == 1 || kuukausi == 3 || kuukausi == 5 || kuukausi == 7 || kuukausi == 8 || k
        if (paiva < 31) {
            std::cout << paiva + 1 << "." << kuukausi << "." << vuosi;
        }
        else {
            std::cout << paiva - 31 + 1 << "." << kuukausi + 1 << "." << vuosi;
        }
    }
    else if (kuukausi == 2) {
        if (paiva < 28) {
            std::cout << paiva + 1 << "." << kuukausi << "." << vuosi;
        }
        else {
            std::cout << paiva - 28 + 1 << "." << kuukausi + 1 << "." << vuosi;
        }
    }
    else {
        if (paiva < 30) {
            std::cout << paiva + 1 << "." << kuukausi << "." << vuosi;
        }
        else {
            std::cout << paiva - 30 + 1 << "." << kuukausi + 1 << "." << vuosi;
        }
    }
}

void Paivays::kysyPaivays()
{
    int pv, kk, vv;
    std::cout << "Anna paivamaara, ensin vuosi: ";
    std::cin >> vv;
    std::cout << "kuukausi: ";
    std::cin >> kk;
    std::cout << "paiva: ";
```



```
std::cin >> pv;
setVuosi(vv);
setKuukausi(kk);
setPaiva(pv);
}

void Paivays::kasvataPaivaysta()
{
    if (paiva < 31) {
        paiva++;
    }
    else {
        paiva = 1;
        kuukausi++;
    }
}

int Paivays::tarkistaKuukausi(int aSyote)
{
    if (kuukausi == 4 || kuukausi == 6 || kuukausi == 9 || kuukausi == 11) {
        return 30;
    }
    else if (kuukausi == 2) { //helmikuu
        if (vuosi % 4 == 0) {
            return 29;
        }
        return 28;
    }
    return 31;
}
```

## kalenterimerkinta.h

---

```
#pragma once
#include <iostream>
#include <string>
#include "paivays.h"

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

    void kysyTiedot();
    std::string getAsia() const;
```

```
std::string getMuistutus() const;
void setAsia(std::string aAsia);
void setMuistutus(std::string aMuistutus);
void tulostaKalenterimerkinta() const;

private:
    std::string asia;
    std::string muistutus;
    Paivays paivays;

};
```

## kalenterimerkinta.cpp

---

```
#include "Kalenterimerkinta.h"

Kalenterimerkinta::Kalenterimerkinta() : asia("---"), muistutus("---")
{
    std::cout << "kalenteri-luokan oletusrakentaja aktivoitu" << std::endl;
}

Kalenterimerkinta::Kalenterimerkinta(std::string aAsia, std::string aMuistutus, Paivays aPaiva
    :asia(aAsia), muistutus(aMuistutus), paivays(aPaivays)
{
    std::cout << "Kalenteri-luokan parametrillinen rakentaja aktivoitu." << std::endl;
}

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

void Kalenterimerkinta::kysyTiedot()
{
    std::cout << "Anna kalenterimerkinta: ";
    std::getline(std::cin, asia);
    std::cout << "Laitetaanko muistutus? (Kylla/Ei)";
    std::getline(std::cin, muistutus);
    paivays.kysyPaivays();
}

std::string Kalenterimerkinta::getAsia() const
{
    return asia;
}
```

```
}

std::string Kalenterimerkinta::getMuistutus() const
{
    return muistutus;
}

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

void Kalenterimerkinta::setMuistutus(std::string aMuistutus)
{
    muistutus = aMuistutus;
}

void Kalenterimerkinta::tulostaKalenterimerkinta() const
{
    std::cout << "Asetit seuraavanlaisen kalenterimerkinnan: " << std::endl;
    std::cout << "Paivays: ";
    paivays.tulostaPaivaysTanaan();
    std::cout << "Asettamasi merkint💎: " << asia << std::endl;
    std::cout << "Muistutus laitettu: " << muistutus << std::endl << std::endl;
}
```