

**ATK III**

**HARJOITUSTYÖ 3**  
**Säähavaintorekisteri**

Tekijä	Virpi Karhula
Email	virpi.karhula@oulu.fi
Koul.ohj.	TTK/SO 94
Palautettu	25.3.2003

## 1. TYÖN MÄÄRITYS

Harjoitustyössä oli tehtävänä toteuttaa säähavaintorekisteri. Vanhat säähavainnot tuli lukea tiedostosta ohjelman käynnistyessä ja lopettaessa kaikki säähavainnot tuli kirjoittaa takaisin samaan tiedostoon.

Sähavainnoista kirjattiin seuraavat tiedot: päivämäärä, havaintopaikka, lämpötila, tuulen nopeus ja suunta, suhteellinen kosteus ja sademäärä (mm).

Ohjelmalla piti pystyä tulostamaan annetun aikavälin keskilämpötila, keskimääräinen tuulennopeus ja kokonaissademäärä.

Lisäksi toteutettiin:

- kaikkien säähavaintojen tulostus kuvaruudulle
- valitun ajanjakson säähavaintojen tulostus kuvaruudulle
- uuden säähavainnon lisääminen

## 2. OHJELMAN AJOTIEDOT

Päävalikosta käyttäjä voi valita haluaako hän syöttää uusia säähavaintoja tai tulostaa raportteja säähavainnoista.

Uutta säähavaintoa syötettäessä käyttäjältä kysytään seuraavat tiedot: päivämäärä, havaintopaikka, lämpötila, tuulen nopeus ja suunta, suhteellinen kosteus ja sademäärä (mm). Muuttujien päivä, kuukausi, vuosi, suhteellinen kosteus ja sademäärä arvoiksi käyttäjä voi syöttää vain lukuja 0-9. Lämpötila ja tuulen nopeus hyväksyvät lisäksi miinusmerkin ja pisteen. Havaintopaikka ja tuulen suunta –muuttujien lukemia arvoja ei ole rajoitettu.

Raportteja voi tulostaa kaikista säähavainnoista tai rajoitetun ajanjakson säähavainnoista. Rajoitettu ajanjakso kysytään käyttäjältä. Sähavaintoraportissa tulostetaan ensin yksittäiset säähavainnot sekä sen jälkeen kyseisen ajanjakson keskilämpötila ja keskimääräinen tuulennopeus ja kokonaissademäärä.

Tiedostosta luku ja tiedostoon tallennus ovat automaattisia tapahtumia, eivätkä ne vaadi käyttäjän toimenpiteitä. Tiedostosta luku tapahtuu ohjelman käynnistyessä ja tiedostoon kirjoitus ohjelman lopetuksen yhteydessä.

Ohjelman suoritus ei keskeydy käyttäjän antamiin väärin syötetietoihin. Ainoastaan käyttäjän antama numero 4 päävalikossa lopettaa ohjelman suorituksen.

### 3. TOTEUTUS

#### *Luokat*

Ohjelman tietojen käsittely on toteutettu Paiva- ja Saahavainto-luokkien avulla. Saahavainto-luokka perii Paiva-luokan. Luokilla on yksi virtuaalinen metodi tulosta(), jota käytetään kun halutaan tulostaa Paiva- ja Saahavainto-olioiden tietoja näytölle. Paiva-luokalle on toteutettu <, > ja == operaattorit, joiden avulla tarkistetaan, kuuluuko säähavainto annetun aikajakson sisälle. Operaattoreita tarvitaan myös listan järjestämisessä.

Molemmat luokat käyttävät Virhe-luokkaa poikkeusten käsittelyssä.

#### *Tiedoston käsittely*

Tiedosto 'tiedosto.txt' avataan erikseen tiedoston lukua (ifstream infile) ja kirjoittamista varten (ofstream outfile). Ellei tiedostoa ole olemassa, ohjelma ilmoittaa epäonnistuneensa tiedoston avauksessa. Lopettaessaan ohjelma luo tiedoston, ellei sitä ole.

#### *Tiedostosta lukeminen*

Tiedostosta luetut tiedot luetaan ja sijoitetaan Saahavainto-luokan olioihin.

#### *Tietorakenteet*

Saahavainto-oliot sijoitetaan listaan. Listassa oliot järjestetään päivämäärän mukaiseen järjestykseen sort()-komennolla. Listan olioihin viitataan iteraattorin avulla.

#### *Tiedostoon tallentaminen*

Tiedostoon tallentaminen tapahtuu käyttäen hyväksi Saahavainto-luokan <<-operaattorin ylikuormasta. Tiedostoon tallennuksessa käytetään muuttujien erotinmerkkinä #-merkkiä. Yksi Saahavainto-olio tallennetaan yhdelle riville. Tiedoston vanhat tiedot korvautuvat uusilla tiedoilla.

#### *Poikkeuskäsittely*

Käyttäjän syötteiden oikeellisuuden tarkastamisessa käytetään poikkeuskäsittelyä. Jos syötteessä on virheellinen merkki, heitetään poikkeus metodeissa lueLuku() ja lueDouble(). Poikkeukset pyydytetään mahdollisimman alhaisella tasolla heti funktiokutsujen jälkeen. Funktio lueLuku() hyväksyy vain merkit '0' - '9' ja lueDouble() numeroiden lisäksi miinusmerkin '-' ja pisteen '.'. Muut syötteen merkit aiheuttavat poikkeuksen.

Funktioiden toiminta on kommentoitu yksityiskohtaisesti ohjelmakoodissa.

### 3. OHJELMAKOODI

Ohjelmatiedostot:

Paaohjelma.cpp	paaohjelma
Paiva.h	Paiva-luokan määrittely
	Virhe-luokan määrittely ja toteutus
Paiva.cpp	Paiva-luokan toteutus
Saahavainto.h	Saahavainto-luokan määrittely
Saahavainto.cpp	Saahavainto-luokan toteutus
tiedosto.txt	tiedosto tietojen tallennusta varten

```
/*
 * Atk III, harjoitustyö 3
 * Paaohjelma.cpp
 *
 * (c) Virpi Karhula 2003
 */

#include <iostream>
#include <list>
#include <fstream>
#include <string>

#include "Paiva.h"
#include "Saahavainto.h"

using namespace std;
using namespace saa;

int naytaPaavalikko(void);
void lueTiedostoListaan(ifstream* infile, list<Saahavainto>* lista);
void kirjoitaTiedostoon(ofstream* outfile, list<Saahavainto>* lista);
void tulostaKeskiarvot(list<Saahavainto>* lista);
void tulostaLista(list<Saahavainto>* lista);
int lueLuku(void);
double lueDouble(void);
void tulostaAikavali(list<Saahavainto>* lista);

int main(void){

    list<Saahavainto> lista; //Muodostetaan lista, johon
                                //sijoitetaan Saahavainto-olioita
    list<Saahavainto>::iterator it; //Iteraattori listan
                                    //läpikäymistä varten

    ifstream infile; // Tiedosto lukemista varten
    ofstream outfile; // Tiedosto kirjoittamista varten

    int valinta = 0; // pitää olla !=4
    Saahavainto saa; // apumuuttuja havaintojen lisäämisessä
    int luku = -1;
    bool lisays = false;

    // TIEDOSTON AVAUS
    int avausliput = ios_base::in | ios_base::out; //korvaa
    //int avausliput = ios_base::in | ios_base::out | ios_base::app; //lisää
    loppuun
    char* nimi = "tiedosto.txt";
    infile.open(nimi, avausliput);
```

```

if(!infile.is_open()) cout << "\nTiedoston avaus epäonnistui.";
else cout << "\nTiedoston avaus onnistui";
lueTiedostoListaan(&infile, &lista);
infile.close();

// PÄÄVALIKKO
while ((valinta = naytaPaavalikko()) != 4) {
    switch(valinta){
        case 1:
            // Säähavainnon kysyminen käyttäjältä
            lisays = saa.lueSaahavainto();
            if(lisays){
                cout << "\nSäähavainto lisätty rekisteriin:";
                saa.tulosta();
                lista.push_back(saa);
            }
            break;
        case 2:
            // Koko listan tulostus
            tulostaLista(&lista);
            if(lista.size() != 0)
                tulostaKeskiarvot(&lista);
            else
                cout << "\nRekisterissä ei ole yhtään säähavaintoa.";
            break;
        case 3:
            // Valittujen tietojen tulostus
            tulostaAikavali(&lista);
            break;
        case 4:
            break;
        default:
            break;
    }
}

// TIEDOSTOON KIRJOITTAMINEN
outfile.open(nimi);
lista.sort();
kirjoitaTiedostoon(&outfile, &lista);
outfile.close();

lista.clear();

//system("PAUSE");

return 0;
}

/*
 * Funktio näyttää päävalikon ja lukee käyttäjän syötteen.
 * Positiiviset kokonaisluvut ovat hyväksytyjä syötteitä.
 * Muut syötearvot aiheuttavat poikkeuksen (Virhe).
 */
int naytaPaavalikko(void){
    int valinta = 0;
    cout << "\n\nSÄÄHAVAINTOREKISTERI";
    cout << "\n1 = Lisää uusi säähavainto";
    cout << "\n2 = Tulosta kaikki säähavainnot";
    cout << "\n3 = Tulosta valitut säähavainnot";
    cout << "\n4 = Lopetus";
    cout << "\nValitse: ";

```

```

        try{
            valinta = lueLuku();
        } catch(Virhe& err){
            cout << endl << err.annaTeksti() << endl;
        }
        return valinta;
    }

/*
 * Funktio lukee tiedoston sisällön listaan.
 * Sijoitetaan tiedoston rivin sisältö Saahavainto-olioon,
 * joka sijoitetaan listaan.
 */
void lueTiedostoListaan(ifstream* infile, list<Saahavainto>* lista){
    Saahavainto s;
    char apu[80];
    int paiva, kk, vuosi, kusteus, sadem;
    double lampo, nopeus;
    if((*infile).is_open()){
        //(*infile).seekg(ios_base::beg); //turha
        //File.seekg(ios::beg); //get back to the beginning of the file
        //http://www.cpp-home.com/loobian/tutorials/file_io/3.php
        while((*infile).peek() != EOF){
            (*infile).getline(apu, 80, '#');
            sscanf(apu, "%i", &paiva); //numero
            s.asetapaiva(paiva);
            (*infile).getline(apu, 80, '#');
            sscanf(apu, "%i", &kk); //numero
            s.asetakk(kk);
            (*infile).getline(apu, 80, '#');
            sscanf(apu, "%i", &vuosi); //numero
            s.asetavuosi(vuosi);
            (*infile).getline(apu, 80, '#'); //merkkimuuttuja
            s.asetapaikka(apu);
            (*infile).getline(apu, 80, '#');
            sscanf(apu, "%lf", &lampo); //double
            s.asetalampotila(lampo);
            (*infile).getline(apu, 80, '#');
            sscanf(apu, "%lf", &nopeus); //double
            s.asetanopeus(nopeus);
            (*infile).getline(apu, 80, '#'); //merkkimuuttuja
            s.asetasuunta(apu);
            (*infile).getline(apu, 80, '#');
            sscanf(apu, "%i", &kusteus); //numero
            s.asetakusteus(kusteus);
            (*infile).getline(apu, 80, '\n');
            sscanf(apu, "%i", &sadem); //numero
            s.asetasademaara(sadem);

            (*lista).push_back(s);
            //s.tulosta();
        }
    }
    //cout << "Tiedoston luku loppui";
}

/*
 * Funktio kirjoittaa listan sisältämät tiedot tiedostoon.
 * Kirjoittamisessa käytetään ylikuormattua operaattoria <<.
 * Tiedot on järjestetty päiväyksen mukaan.
 */
void kirjoitaTiedostoon(ofstream* outfile, list<Saahavainto>* lista){
    list<Saahavainto>::iterator it;

```

```

        if((*outfile).bad()){
            cout << "Tiedostoa ei voi avata" << endl;
        } else {
            int koko = (*lista).size();
            (*lista).sort();
            it = (*lista).begin();
            for (int i = 0; i < koko; i++){
                //cout << endl << *it;    // <<operaattorin ylikuormaus toimii!
                (*outfile) << *it;        //kirjoitus tiedostoon
                it++;
            }
            (*outfile).flush(); //tarkistetaan, että kaikki on kirjoitettu
        }
    }

/*
 * Funktio tulostaa listan sisällön näytölle.
 * Tulostamisessa käytetään Saahavainto::tulosta()-metodia.
 * Listassa edetään alkiosta toiseen iteraattorin avulla.
 */
void tulostaLista(list<Saahavainto>* lista){
    list<Saahavainto>::iterator it;
    (*lista).sort();
    it = (*lista).begin();
    for (int i = 0; i < (*lista).size(); i++){
        (*it).tulosta();
        it++;
    }
}

/*
 * Funktio tulostaa näytölle lämpötilan ja tuulen nopeuden keskiarvot
 * sekä kokonaissademäärän.
 */
void tulostaKeskiarvot(list<Saahavainto>* lista){
    list<Saahavainto>::iterator it;
    it = (*lista).begin();
    double ka_lampo = 0.0;
    double ka_nopeus = 0.0;
    double kok_sade = 0;
    for (int i = 0; i < (*lista).size(); i++){
        ka_lampo += (*it).annaLampotila();
        ka_nopeus += (*it).annaNopeus();
        kok_sade += (*it).annaSademaara();
        it++;
    }
    int lkm = 0;
    if ((lkm = (*lista).size()) != 0){
        ka_lampo = ka_lampo / lkm;
        ka_nopeus = ka_nopeus / lkm;
        cout.setf(ios_base::fixed);    //muotoillaan double tulostumaan
        cout.precision(2);              //kahdella desimaalilla
        cout << "\n\nKeskimääräinen lämpötila on " << ka_lampo;
        cout << "\nKeskimääräinen tuulen nopeus on " << ka_nopeus;
        cout.precision(0);
        cout << "\nKokonaissademäärä on " << kok_sade;
    } else
        cout << "\n\nListassa ei ole yhtään säähavaintoa.";
}

/*
 * Funktio tulostaa käyttäjän antaman aikavälin tiedot.
 * Kaikki aikaehdon täyttävät säähavainnot tulostetaan.
 */

```

```

* Lopuksi tulostetaan myös keskiarvot lämpötilalle
* ja tuulen nopeudelle sekä kokonaissademäärä.
*/
void tulostaAikavali(list<Saahavainto>* lista){
    list<Saahavainto>::iterator it;
    Paiva* p1 = new Paiva();
    Paiva* p2 = new Paiva();

    cout << "\nRajoitetut aikavälin tietojen tarkastelu.";
    cout << "\nAnna aloituspäivämäärä. ";
    p1->luePvm();
    p1->tulosta();

    if (p1->annaPvm() == 0) {
        cout << "\nPäivämäärän luku epäonnistui.\n";
    } else {
        cout << "\nAnna lopetuspäivämäärä. ";
        p2->luePvm();
        if (p2->annaPvm() == 0) {
            cout << "\nPäivämäärän luku epäonnistui.\n";
        }
        p2->tulosta();
    }

    if ((p1->annaPvm() == 0) || (p2->annaPvm() == 0)){
        cout << "\nAikavälin tietojen lukeminen epäonnistui.";
    } else if ((*p1) > (*p2)) {
        cout << "\nAntamasi lopetuspäivämäärä oli ennen "\
            "aloituspäivämäärää.";
    } else {
        (*lista).sort();
        it = (*lista).begin();
        Paiva apu;
        double ka_lampo = 0.0;
        double ka_nopeus = 0.0;
        double kok_sade = 0;
        int lkm = 0;

        for (int i = 0; i < (*lista).size(); i++){
            apu.asetapP((*it).annaPvm());
            apu.asetakK((*it).annaKk());
            apu.asetavV((*it).annaVv());

            if (((apu < (*p2)) || (apu == (*p2))) && \
                ((apu > (*p1)) || (apu == (*p1)))) {

                (*it).tulosta();
                ka_lampo += (*it).annaLampotila();
                ka_nopeus += (*it).annaNopeus();
                kok_sade += (*it).annaSademaara();
                lkm++;
            } else
                cout << "\nEi ehtoa täyttävää havaintoa";
            it++;
        }
        if (lkm != 0){
            ka_lampo = ka_lampo / lkm;
            ka_nopeus = ka_nopeus / lkm;
            cout.setf(ios_base::fixed); //muotoillaan double tulostumaan
            cout.precision(2); //kahdella desimaalilla
            cout << "\n\nKeskimääräinen lämpötila on " << ka_lampo;
            cout << "\nKeskimääräinen tuulen nopeus on " << ka_nopeus;
            cout.precision(0);
        }
    }
}

```



```

        cout << "\nKokonaissademäärä on " << kok_sade;
    } else
        cout << "\n\nAikavälillä ei ollut yhtään säähavaintoa.";
    }
    delete p1;
    delete p2;
}

/*
 * Funktio lukee positiivisen kokonaisluvun.
 * Onnistuessaan palauttaa positiivisen kokonaisluvun.
 * Jos syötteenä on muuta kuin merkit 0-9, tulkitaan syöte virheelliseksi.
 * Virhetilanteessa palautusarvo on -1.
 * Virhetilanteessa funktio heittää poikkeuksen (luokka Virhe).
 */
int lueLuku(){
    int luku = -1;
    char ch[80];
    bool oikein = true;
    //cout << "Anna kokonaisluku";
    cin >> ch;
    for (int i = 0; ch[i] != '\0'; i++) {
        if (ch[i] == '0' || ch[i] == '1' || ch[i] == '2' || \
            ch[i] == '3' || ch[i] == '4' || ch[i] == '5' || \
            ch[i] == '6' || ch[i] == '7' || ch[i] == '8' || \
            ch[i] == '9'){
            //cout << "if " << ch[i];
        } else {
            //cout << "else: " << ch[i];
            oikein = false;
            throw Virhe("Väärä syöttötieto");
            break;
        }
    }
    if (oikein){
        luku = atoi(ch);
        return luku;
    } else
        return (-1);
}

/*
 * Funktio lukee liukuluvun.
 * Onnistuessaan palauttaa liukuluvun.
 * Jos syötteenä on muuta kuin miinusmerkki '-', piste '.' tai
 * merkit 0-9, tulkitaan luku epäonnistuneeksi.
 * Virhetilanteessa funktio heittää poikkeuksen (luokka Virhe).
 */
double lueDouble(){
    double luku = -1;
    char ch[80];
    //cout << "Anna liukuluku";
    cin >> ch;
    for (int i = 0; ch[i] != '\0'; i++) {
        if (ch[i] == '0' || ch[i] == '1' || ch[i] == '2' || \
            ch[i] == '3' || ch[i] == '4' || ch[i] == '5' || \
            ch[i] == '6' || ch[i] == '7' || ch[i] == '8' || \
            ch[i] == '9' || ch[i] == '-' || ch[i] == '.'){
        } else {
            throw Virhe("Väärä syöttötieto");
            break;
        }
    }
}

```

```
    luku = atof(ch);  
    return luku;  
}
```

```

/*
 * Atk III, harjoitustyö 3
 * Paiva.h
 *
 * (c) Virpi Karhula 2003
 */

#ifndef PAIVA_H_INCLUDED
#define PAIVA_H_INCLUDED

using namespace std;

namespace saa {

    class Paiva {
    public:
        Paiva();
        Paiva(unsigned int pvm, unsigned int kk, unsigned int vv);
        Paiva(const Paiva&);
        ~Paiva();

        //Metodit
        friend bool operator<(const Paiva&, const Paiva&);
        friend bool operator>(const Paiva&, const Paiva&);
        friend bool operator==(const Paiva&, const Paiva&);

        inline int annaPvm() const {return pvm;};
        inline int annaKk() const {return kk;};
        inline int annaVv() const {return vv;};
        inline void asetaPp(int pvm) {this->pvm = pvm;}
        inline void asetaKk(int kk) {this->kk = kk;}
        inline void asetaVv(int vv) {this->vv = vv;}
        virtual void tulosta(void) const;
        void luePvm(void);

    protected:
        unsigned int pvm;
        unsigned int kk;
        unsigned int vv;    //>1900
    };

    // Luokka virheenkäsittelyä varten
    class Virhe {
    public:
        inline Virhe(string teksti) {this->teksti = teksti; }
        inline string annaTeksti() {return teksti;}
    private:
        string teksti;
    };
}    //namespace

#endif

```

```

/*
 * Atk III, harjoitustyö 3
 * Paiva.cpp
 *
 * (c) Virpi Karhula 2003
 */

#include <iostream>
#include <string>

#include "Paiva.h"

extern int lueLuku();

namespace saa {

    //Konstruktorit
    Paiva::Paiva(){
        pvm = 0;
        kk = 0;
        vv = 0;
    }
    Paiva:: Paiva( unsigned int pvm, unsigned int kk, unsigned int vv){
        this->pvm = pvm;
        this->kk = kk;
        this->vv = vv;
    }
    //Kopionmuodostin
    /* //myös tämä kopionmuodostin toimii ok
    Paiva::Paiva(const Paiva& p){
        pvm = p.pvm;
        kk = p.kk;
        vv = p.vv;
    }
    */
    //Kopionmuodostin, parametrivälitys
    Paiva::Paiva(const Paiva& p) : pvm(p.pvm), kk(p.kk), vv(p.vv){}

    //Hajotin
    Paiva::~~Paiva(){}

    bool operator<(const Paiva& p1, const Paiva& p2){
        bool vertailu = false;
        if(p1.vv < p2.vv) vertailu = true;
        if(p1.vv == p2.vv){
            if(p1.kk < p2.kk) vertailu = true;
            if(p1.kk == p2.kk){
                if(p1.pvm < p2.pvm) vertailu = true;
            }
        }
        return vertailu;
    }

    bool operator>(const Paiva& p1, const Paiva& p2){
        bool vertailu = false;
        if(p1.vv > p2.vv) vertailu = true;
        if(p1.vv == p2.vv){
            if(p1.kk > p2.kk) vertailu = true;
            if(p1.kk == p2.kk){
                if(p1.pvm > p2.pvm) vertailu = true;
            }
        }
        return vertailu;
    }
}

```

```

}

bool operator==(const Paiva& p1, const Paiva& p2){
    bool vertailu = false;
    if((p1.vv == p2.vv) && (p1.kk == p2.kk) && (p1.pvm == p2.pvm)){
        vertailu = true;
    }
    return vertailu;
}

//// METODIT ////

/*
 * Funktio tulostaa päivämäärän näytölle.
 */
void Paiva::tulosta() const{
    cout << "Päivämäärä: " << pvm << "." << kk << "." << vv;
}

/*
 * Funktio lukee käyttäjältä päivämäärän tiedot.
 * Oikein annettujen syötteiden jälkeen asettaa
 * arvot parametrina tuodun pointterin osoittamaan
 * Paiva-luokan ilmentymään.
 */
void Paiva::luePvm(){
    int pp;                //unsigned pois, koska voi pal. -1
    int kk;                //unsigned pois, koska voi pal. -1
    int vv;    //>1900    //unsigned pois, koska voi pal. -1
    bool oikein = true;

    try {
        cout << endl << "Anna päivä: ";
        pp = lueLuku();
        if(pp <= 0 || pp > 31) oikein = false;
    } catch (Virhe& err) {
        cout << endl << err.annaTeksti() << endl;
        oikein = false;
    }

    if (oikein) {
        try {
            cout << "Anna kuukausi: ";
            kk = lueLuku();
            if(kk <= 0 || kk > 12) oikein = false;
        } catch (Virhe& err) {
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }

    if (oikein) {
        try {
            cout << "Anna vuosi (neljällä numerolla): ";
            vv = lueLuku();
            if(vv <= 1900) oikein = false;
        } catch (Virhe& err) {
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }

    if(oikein){
        asetaPp(pp);
        asetaKk(kk);
        asetaVv(vv);
    }
}

```

```
        }  
    }  
} //namespace
```

```

/*
 * Atk III, harjoitustyö 3
 * Saahavainto.h
 *
 * (c) Virpi Karhula 2003
 */

#ifndef SAAHAVAINTO_H_INCLUDED
#define SAAHAVAINTO_H_INCLUDED

#include <string>

using namespace std;

namespace saa {

class Saahavainto : public Paiva {
public:
    //Konstruktorit
    Saahavainto();
    Saahavainto( \
        Paiva p, \
        string paikka, double lampotila, \
        double tuulenNopeus, string tuulenSuunta, \
        unsigned int kosteus, unsigned int sademaara);
    Saahavainto( \
        unsigned int pvm, unsigned int kk, unsigned int vv, \
        string paikka, double lampotila, \
        double tuulenNopeus, string tuulenSuunta, \
        unsigned int kosteus, unsigned int sademaara);
    //Kopionmuodostin
    Saahavainto(const Saahavainto& s);
    //Destruktori
    ~Saahavainto();

    // Metodit
    Saahavainto& operator=(const Saahavainto& s);
    friend ostream& operator<<(ostream& oo, const Saahavainto& s);

    void sijoita(unsigned int pvm, unsigned int kk, \
        unsigned int vv, string paikka, double lampotila, \
        double tuulenNopeus, string tuulenSuunta, \
        unsigned int kosteus, unsigned int sademaara);
    bool lueSaahavainto();
    void tulosta(void) const; //virtual
    inline void asetaPaikka(string paikka)
        {this->paikka = paikka;}
    inline void asetaLampotila(int lampotila)
        {this->lampotila = lampotila;}
    inline void asetaNopeus(double tuulenNopeus)
        {this->tuulenNopeus = tuulenNopeus;}
    inline void asetaSuunta(string tuulenSuunta)
        {this->tuulenSuunta = tuulenSuunta;}
    inline void asetaKosteus(unsigned int kosteus)
        {this->kosteus = kosteus;}
    inline void asetaSademaara(double sademaara)
        {this->sademaara = sademaara;}
    inline double annaLampotila() const {return lampotila;}
    inline double annaNopeus() const {return tuulenNopeus;}
    inline int annaSademaara() const { return sademaara;}

private:
    string paikka;

```

```
        double lampotila;
        double tuulenNopeus;
        string tuulenSuunta;
        unsigned int kosteus;
        unsigned int sademaara;
    };
} //namespace

#endif
```



```

/*
 * Atk III, harjoitustyö 3
 * Saahavainto.cpp
 *
 * (c) Virpi Karhula 2003
 */

#include <iostream>

#include "Paiva.h"
#include "Saahavainto.h"

extern int lueLuku();
extern int lueNegatLuku();
extern double lueDouble();

namespace saa {

    //Konstruktorit
    Saahavainto::Saahavainto() {
        pvm = 0;
        kk = 0;
        vv = 0;
        paikka = "";
        lampotila = 0;
        tuulenNopeus = 0;
        tuulenSuunta = "";
        kosteus = 0;
        sademaara = 0;
    }
    Saahavainto::Saahavainto( \
        Paiva p, \
        string paikka, double lampotila, \
        double tuulenNopeus, string tuulenSuunta, \
        unsigned int kosteus, unsigned int sademaara){
        pvm = p.annaPvm();
        kk = p.annaKk();
        vv = p.annaVv();
        this->paikka = paikka;
        this->lampotila = lampotila;
        this->tuulenNopeus = tuulenNopeus;
        this->tuulenSuunta = tuulenSuunta;
        this->kosteus = kosteus;
        this->sademaara = sademaara;
    }
    Saahavainto::Saahavainto( \
        unsigned int pvm, unsigned int kk, unsigned int vv, \
        string paikka, double lampotila, \
        double tuulenNopeus, string tuulenSuunta, \
        unsigned int kosteus, unsigned int sademaara){
        this->pvm = pvm;
        this->kk = kk;
        this->vv = vv;
        this->paikka = paikka;
        this->lampotila = lampotila;
        this->tuulenNopeus = tuulenNopeus;
        this->tuulenSuunta = tuulenSuunta;
        this->kosteus = kosteus;
        this->sademaara = sademaara;
    }
    //Kopionmuodostin
    Saahavainto::Saahavainto(const Saahavainto& s){
        pvm = s.pvm;

```

```

        kk = s.kk;
        vv = s.vv;
        paikka = s.paikka;
        lampotila = s.lampotila;
        tuulenNopeus = s.tuulenNopeus;
        tuulenSuunta = s.tuulenSuunta;
        kosteus = s.kosteus;
        sademaara = s.sademaara;
    }
    //Destruktori
    Saahavainto::~Saahavainto() {}

    //// METODIT ////

    /*
     * Metodi sijoittaa Saahavainto-olioon kaikki private-muuttujien tiedot.
     * Olio on luotava ennen kuin se voi kutsua tätä metodiaan.
     */
    void Saahavainto::sijoita(unsigned int pvm, unsigned int kk, \
        unsigned int vv, string paikka, double lampotila, \
        double tuulenNopeus, string tuulenSuunta, \
        unsigned int kosteus, unsigned int sademaara){
        this->pvm = pvm;
        this->kk = kk;
        this->vv = vv;
        this->paikka = paikka;
        this->lampotila = lampotila;
        this->tuulenNopeus = tuulenNopeus;
        this->tuulenSuunta = tuulenSuunta;
        this->kosteus = kosteus;
        this->sademaara = sademaara;
    }

    void Saahavainto::tulosta() const{
        cout << endl << endl;
        Paiva::tulosta(); //yliluokan metodin kutsu
        cout << endl << "Havaintopaikka: " << paikka;
        cout << "\t" << "Lämpötila: " << lampotila << " oC";
        cout << endl << "Tuulen nopeus: " << tuulenNopeus << " m/s";
        cout << "\t" << "Tuulen suunta: " << tuulenSuunta;
        cout << endl << "Suhteellinen kosteus: " << kosteus << " %";
        cout << "\t" << "Sademäärä: " << sademaara << " mm";
    }

    Saahavainto& Saahavainto::operator=(const Saahavainto& s){
        pvm = s.pvm;
        kk = s.kk;
        vv = s.vv;
        paikka = s.paikka;
        lampotila = s.lampotila;
        tuulenNopeus = s.tuulenNopeus;
        tuulenSuunta = s.tuulenSuunta;
        kosteus = s.kosteus;
        sademaara = s.sademaara;

        return *this;
    }

    // Käytetään tiedostoon kirjoittamisessa (muotoilu).
    // # -merkkiä käytetään tietojen erotinmerkkinä.
    ostream& operator<<(ostream& oo, const Saahavainto& s) {
        oo << s.pvm << "#" \
            << s.kk << "#" \

```

```

        << s.vv << "#" \
        << s.paikka << "#" \
        << s.lampotila << "#" \
        << s.tuulenNopeus << "#" \
        << s.tuulenSuunta << "#" \
        << s.kosteus << "#" \
        << s.sademaara << "\n" ;
    return oo;
}

/*
 * Metodi lukee käyttäjältä kaikki säähavainnon tiedot.
 * Kokonaislukujen lukemiseen käytetään metodeja:
 * lueLuku() (hyväksyy vain positiiviset kokonaisluvut)
 * lueNegatLuku() (hyväksyy negatiiviset ja positiiviset kokonaisluvut)
 * Ko.metodit heittävät muuta kuin numeerista tietoa
 * syötettäessä poikkeuksen (luokka Virhe), joka
 * otetaan kiinni tässä metodissa.
 * Merkkitiedon syötön oikeellisuutta ei tarkisteta
 * millään tavalla.
 * Jos kaikki syötteet ovat merkistöltään hyväksytyjä,
 * sijoitetaan syötetyt tiedot kutsuvaan Saahavainto-olioon.
 * Jos yksikin syötetieto on väärä, ohjelma keskeytyy
 * eikä jatka seuraavien syötetietojen kysymistä.
 */
bool Saahavainto::lueSaahavainto(){
    int pvm;          //unsigned pois, koska voi pal. -1
    int kk;           //unsigned pois, koska voi pal. -1
    int vv;           //>1900 //unsigned pois, koska voi pal. -1
    string paikka;
    double lampotila;
    double tuulenNopeus;
    string tuulenSuunta;
    int kosteus;       //unsigned
    int sademaara;     //unsigned
    bool oikein = true;

    cout << endl << "Anna säähavainnon tiedot";
    try{
        cout << endl << "Anna päivä: ";
        pvm = lueLuku();
        if(pvm <= 0 || pvm > 31) {
            cout << "\nArvoalue oli väärä.";
            oikein = false;
        }
    } catch (Virhe& err){
        cout << endl << err.annaTeksti() << endl;
        oikein = false;
    }
    if (oikein) {
        try {
            cout << "Anna kuukausi: ";
            kk = lueLuku();
            if(kk <= 0 || kk > 12) {
                cout << "\nArvoalue oli väärä.";
                oikein = false;
            }
        } catch (Virhe& err){
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }
    if (oikein) {

```

```

        try{
            cout << "Anna vuosi (neljällä numerolla): ";
            vv = lueLuku();
            if(vv <= 1900) {
                oikein = false;
                cout << "\nAnna myös vuosisata." << endl;
            }
        } catch(Virhe& err){
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }
    if (oikein) {
        cout << "Anna havaintopaikka: ";
        cin >> paikka;
    }
    if (oikein) {
        try{
            cout << "Anna lämpötila: ";
            lampotila = lueDouble();
        } catch(Virhe& err){
            // Huom. poikkeuskäsittely ainoa mahdollisuus havaita virhe
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }
    if (oikein) {
        try{
            cout << "Anna tuulen nopeus (m/s): ";
            tuulenNopeus = lueDouble();
        } catch(Virhe& err){
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }
    if (oikein) {
        cout << "Anna tuulen suunta (pohjoinen, etelä, itä, "\
            "länsi, luode, koillinen, lounas tai kaakko)";
        cin >> tuulenSuunta;
    }
    if (oikein) {
        try{
            cout << "Anna suhteellinen kosteus: ";
            kosteus = lueLuku();
        } catch(Virhe& err){
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }
    if (oikein) {
        try{
            cout << "Anna sademäärä: ";
            sademaara = lueLuku();
        } catch(Virhe& err){
            cout << endl << err.annaTeksti() << endl;
            oikein = false;
        }
    }

    if (oikein) {
        sijoita(pvm, kk, vv, paikka, lampotila, \
            tuulenNopeus, tuulenSuunta, kosteus, sademaara);
    }

```

```
        }  
        return oikein;  
    }  
}  
//namespace
```

## 4. TESTIAJO

### 4.1 Normaaliajo

tiedosto.txt (ennen ohjelman suoritusta)

```
1#1#2000#Oulu#-10#2#pohjoinen#12#2
2#1#2000#Kuusamo#-13#12#etel,,#13#2
3#1#2000#Utsjoki#-20#2#luode#12#0
5#5#2001#Hamina#15#3#etel,,#13#1
6#6#2001#Oulu#23#13#lounas#12#13
1#1#2002#Helsinki#-2#3#kaakko#12#12
3#2#2002#Marjaniemi#-15#12#l,,nsi#12#0
```

-----

Tiedoston avaus onnistui

S—HAVAINTOREKISTERI

1 = Lisää uusi säähavainto  
2 = Tulosta kaikki säähavainnot  
3 = Tulosta valitut säähavainnot  
4 = Lopetus  
Valitse: 2

Päivämäärä: 1.1.2000

Havaintopaikka: Oulu Lämpötila: -10 oC  
Tuulen nopeus: 2 m/s Tuulen suunta: pohjoinen  
Suhteellinen kosteus: 12 % Sademäärä: 2 mm

Päivämäärä: 2.1.2000

Havaintopaikka: Kuusamo Lämpötila: -13 oC  
Tuulen nopeus: 12 m/s Tuulen suunta: etelä  
Suhteellinen kosteus: 13 % Sademäärä: 2 mm

Päivämäärä: 3.1.2000

Havaintopaikka: Utsjoki Lämpötila: -20 oC  
Tuulen nopeus: 2 m/s Tuulen suunta: luode  
Suhteellinen kosteus: 12 % Sademäärä: 0 mm

Päivämäärä: 5.5.2001

Havaintopaikka: Hamina Lämpötila: 15 oC  
Tuulen nopeus: 3 m/s Tuulen suunta: etelä  
Suhteellinen kosteus: 13 % Sademäärä: 1 mm

Päivämäärä: 6.6.2001

Havaintopaikka: Oulu Lämpötila: 23 oC  
Tuulen nopeus: 13 m/s Tuulen suunta: lounas  
Suhteellinen kosteus: 12 % Sademäärä: 13 mm

Päivämäärä: 1.1.2002

Havaintopaikka: Helsinki Lämpötila: -2 oC  
Tuulen nopeus: 3 m/s Tuulen suunta: kaakko  
Suhteellinen kosteus: 12 % Sademäärä: 12 mm

Päivämäärä: 3.2.2002

Havaintopaikka: Marjaniemi Lämpötila: -15 oC  
Tuulen nopeus: 12 m/s Tuulen suunta: länsi  
Suhteellinen kosteus: 12 % Sademäärä: 0 mm

Keskimääräinen lämpötila on -3.14

Keskimääräinen tuulen nopeus on 6.71

KokonaissademΣΣrΣ on 30

S—HAVAINTOREKISTERI

1 = LisΣΣ uusi sΣΣhavainto  
2 = Tulosta kaikki sΣΣhavainnot  
3 = Tulosta valitut sΣΣhavainnot  
4 = Lopetus  
Valitse: 1

Anna sΣΣhavainnon tiedot

Anna pΣivΣ: 25  
Anna kuukausi: 3  
Anna vuosi (neljΣllΣ numerolla): 2003  
Anna havaintopaikka: Oulu  
Anna lΣmp+tila: -3  
Anna tuulen nopeus (m/s): 5  
Anna tuulen suunta (pohjoinen, etelΣ, itΣ, lΣnsi, luode, koillinen, lounas tai ka  
akko)pohjoinen  
Anna suhteellinen kosteus: 10  
Anna sademΣΣrΣ: 0

sΣΣhavainto lisΣtety rekisteriin:

PΣivΣmΣΣrΣ: 25.3.2003  
Havaintopaikka: Oulu LΣmp+tila: -3 oC  
Tuulen nopeus: 5 m/s Tuulen suunta: pohjoinen  
Suhteellinen kosteus: 10 % SademΣΣrΣ: 0 mm

S—HAVAINTOREKISTERI

1 = LisΣΣ uusi sΣΣhavainto  
2 = Tulosta kaikki sΣΣhavainnot  
3 = Tulosta valitut sΣΣhavainnot  
4 = Lopetus  
Valitse: 3

Rajoitetut aikavΣlin tietojen tarkastelu.

Anna aloituspΣivΣmΣΣrΣ.  
Anna pΣivΣ: 2  
Anna kuukausi: 1  
Anna vuosi (neljΣllΣ numerolla): 2000  
PΣivΣmΣΣrΣ: 2.1.2000  
Anna lopetuspΣivΣmΣΣrΣ.  
Anna pΣivΣ: 31  
Anna kuukausi: 12  
Anna vuosi (neljΣllΣ numerolla): 2001  
PΣivΣmΣΣrΣ: 31.12.2001  
Ei ehtoa tΣyttΣvΣΣ havaintoa

PΣivΣmΣΣrΣ: 2.1.2000  
Havaintopaikka: Kuusamo LΣmp+tila: -13 oC  
Tuulen nopeus: 12 m/s Tuulen suunta: etelä  
Suhteellinen kosteus: 13 % SademΣΣrΣ: 2 mm

PΣivΣmΣΣrΣ: 3.1.2000  
Havaintopaikka: Utsjoki LΣmp+tila: -20 oC  
Tuulen nopeus: 2 m/s Tuulen suunta: luode  
Suhteellinen kosteus: 12 % SademΣΣrΣ: 0 mm

PΣivΣmΣΣrΣ: 5.5.2001  
Havaintopaikka: Hamina LΣmp+tila: 15 oC  
Tuulen nopeus: 3 m/s Tuulen suunta: etelä  
Suhteellinen kosteus: 13 % SademΣΣrΣ: 1 mm

```
Päivämäärä: 6.6.2001
Havaintopaikka: Oulu    Lämpötila: 23 oC
Tuulen nopeus: 13 m/s   Tuulen suunta: lounas
Suhteellinen kosteus: 12 %   Sademäärä: 13 mm
Ei ehtoa täyttävää havaintoa
Ei ehtoa täyttävää havaintoa
Ei ehtoa täyttävää havaintoa
```

```
Keskimääräinen lämpötila on 1.25
Keskimääräinen tuulen nopeus on 7.50
Kokonaissademäärä on 16
```

```
S—HAVAINTOREKISTERI
1 = Lisää uusi säähavainto
2 = Tulosta kaikki säähavainnot
3 = Tulosta valitut säähavainnot
4 = Lopetus
Valitse: 4
Press any key to continue
```

Tiedosto.txt (ohjelman suorituksen jälkeen)

```
1#1#2000#Oulu#-10#2#pohjoinen#12#2
2#1#2000#Kuusamo#-13#12#etel,,#13#2
3#1#2000#Utsjoki#-20#2#luode#12#0
5#5#2001#Hamina#15#3#etel,,#13#1
6#6#2001#Oulu#23#13#lounas#12#13
1#1#2002#Helsinki#-2#3#kaakko#12#12
3#2#2002#Marjaniemi#-15#12#l,,nsi#12#0
25#3#2003#Oulu#-3#5#pohjoinen#10#0
```

## 4.2 Virhetilanteita säähavainnon syöttämisessä

Tiedoston avaus onnistui

```
S—HAVAINTOREKISTERI
1 = Lisää uusi säähavainto
2 = Tulosta kaikki säähavainnot
3 = Tulosta valitut säähavainnot
4 = Lopetus
Valitse: 1
```

Anna säähavainnon tiedot  
Anna päivä: e

Väärä syöte+tieto

```
S—HAVAINTOREKISTERI
1 = Lisää uusi säähavainto
2 = Tulosta kaikki säähavainnot
3 = Tulosta valitut säähavainnot
4 = Lopetus
Valitse: 1
```

Anna säähavainnon tiedot  
Anna päivä: 32

Arvoalue oli väärä.

```
S—HAVAINTOREKISTERI
```



1 = LisEE uusi sEEhavainto  
2 = Tulosta kaikki sEEhavainnot  
3 = Tulosta valitut sEEhavainnot  
4 = Lopetus  
Valitse: 1

Anna sEEhavainnon tiedot  
Anna pEivE: 1  
Anna kuukausi: 1  
Anna vuosi (neljElle numerolla): 98

Anna my+s vuosisata.

S—HAVAINTOREKISTERI  
1 = LisEE uusi sEEhavainto  
2 = Tulosta kaikki sEEhavainnot  
3 = Tulosta valitut sEEhavainnot  
4 = Lopetus  
Valitse: 1

Anna sEEhavainnon tiedot  
Anna pEivE: 1  
Anna kuukausi: 1  
Anna vuosi (neljElle numerolla): 1998  
Anna havaintopaikka: Tammisaari  
Anna lEmp+tila: -2  
Anna tuulen nopeus (m/s): we

VSErE sy+tt+tieto

S—HAVAINTOREKISTERI  
1 = LisEE uusi sEEhavainto  
2 = Tulosta kaikki sEEhavainnot  
3 = Tulosta valitut sEEhavainnot  
4 = Lopetus  
Valitse: 1

Anna sEEhavainnon tiedot  
Anna pEivE: 1  
Anna kuukausi: 1  
Anna vuosi (neljElle numerolla): 1998  
Anna havaintopaikka: Tammisaari  
Anna lEmp+tila: -2  
Anna tuulen nopeus (m/s): 3  
Anna tuulen suunta (pohjoinen, etelE, itE, lEnsi, luode, koillinen, lounas tai ka  
akko)etelä  
Anna suhteellinen kosteus: 12  
Anna sademSErE: -4

VSErE sy+tt+tieto

S—HAVAINTOREKISTERI  
1 = LisEE uusi sEEhavainto  
2 = Tulosta kaikki sEEhavainnot  
3 = Tulosta valitut sEEhavainnot  
4 = Lopetus  
Valitse:

### 4.3 Virhetilanteita aikavälin syöttämisessä

S—HAVAINTOREKISTERI

1 = Lisää uusi sähavainto  
2 = Tulosta kaikki sähavainnot  
3 = Tulosta valitut sähavainnot  
4 = Lopetus  
Valitse: 3

Rajoitetut aikavälin tietojen tarkastelu.

Anna aloituspäivämäärä.

Anna päivä: 3

Anna kuukausi: e

Vuorokauden syöttötieto

Päivämäärä: 0.0.0

Päivämäärän luku epäonnistui.

Aikavälin tietojen lukeminen epäonnistui.

S—HAVAINTOREKISTERI

1 = Lisää uusi sähavainto  
2 = Tulosta kaikki sähavainnot  
3 = Tulosta valitut sähavainnot  
4 = Lopetus  
Valitse: 3

Rajoitetut aikavälin tietojen tarkastelu.

Anna aloituspäivämäärä.

Anna päivä: 1

Anna kuukausi: 13

Päivämäärä: 0.0.0

Päivämäärän luku epäonnistui.

Aikavälin tietojen lukeminen epäonnistui.

S—HAVAINTOREKISTERI

1 = Lisää uusi sähavainto  
2 = Tulosta kaikki sähavainnot  
3 = Tulosta valitut sähavainnot  
4 = Lopetus  
Valitse: 3

Rajoitetut aikavälin tietojen tarkastelu.

Anna aloituspäivämäärä.

Anna päivä: 1

Anna kuukausi: 1

Anna vuosi (neljällä numerolla): 2000

Päivämäärä: 1.1.2000

Anna lopetuspäivämäärä.

Anna päivä: 3

Anna kuukausi: 2

Anna vuosi (neljällä numerolla): 1999

Päivämäärä: 3.2.1999

Antamasi lopetuspäivämäärä oli ennen aloituspäivämäärää.

S—HAVAINTOREKISTERI

1 = Lisää uusi sähavainto  
2 = Tulosta kaikki sähavainnot  
3 = Tulosta valitut sähavainnot  
4 = Lopetus  
Valitse: 3

```

Rajoitetut aikavälin tietojen tarkastelu.
Anna aloitusPäivämäärä.
Anna päivä: 1
Anna kuukausi: 5
Anna vuosi (neljellä numerolla): 2003
Päivämäärä: 1.5.2003
Anna lopetusPäivämäärä.
Anna päivä: 30
Anna kuukausi: 7
Anna vuosi (neljellä numerolla): 2003
Päivämäärä: 30.7.2003
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa
Ei ehtoa tyyttävää havaintoa

```

Aikavälillä ei ollut yhtään sähäaintoa.

```

S—HAVAINTOREKISTERI
1 = Lisää uusi sähäainto
2 = Tulosta kaikki sähäainnot
3 = Tulosta valitut sähäainnot
4 = Lopetus
Valitse:

```

#### 4.4 Väärät syöttötiedot päävalikossa

```

S—HAVAINTOREKISTERI
1 = Lisää uusi sähäainto
2 = Tulosta kaikki sähäainnot
3 = Tulosta valitut sähäainnot
4 = Lopetus
Valitse:
5

```

```

S—HAVAINTOREKISTERI
1 = Lisää uusi sähäainto
2 = Tulosta kaikki sähäainnot
3 = Tulosta valitut sähäainnot
4 = Lopetus
Valitse: r

```

Väärä syöttötieto

```

S—HAVAINTOREKISTERI
1 = Lisää uusi sähäainto
2 = Tulosta kaikki sähäainnot
3 = Tulosta valitut sähäainnot
4 = Lopetus
Valitse: 4
Press any key to continue

```