



## **Ohjelmoinnin syventävät tekniikat**

Harjoitustyö

Niko Kallio

---

## Sisältö

1	Kuvaus.....	3
2	Kuvakaappauksia .....	4
3	Ohjelmakoodit(C++).....	9
4	Ohjelmakoodit(Java).....	14

## 1 Kuvaus

Tein sovelluksen, joka ylläpitää dataa myytävänä olevista autoista.

Ylläpidettävään dataan kuuluu auton

- Valmistaja
- Malli
- Mittarilukema
- Hinta

Data on säilötyinä tehtävänannossa ehdotettuihin tietorakenteisiin eli vector - tietorakenteessa (C++) ja ArrayList -tietorakenteessa (Java).

Kummassakin versiossa pääohjelma toimii silmukalla, jossa valikkona käytetty switchiä.

Autoja voi lisätä, poistaa ja etsiä sekä järjestää merkkikohtaisesti.

Toteutin myös syötteen tarkistamiset enkä enää ainakaan itse saa sovelluksia kaatumaan.

## 2 Kuvakaappauksia

C++

```
Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 1
Valmistaja -> lada
Malli -> niva
Mittarilukema -> 400000
Hinta -> 3200
Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 3
Valmistaja: CITROEN Malli: Berlingo Mittarilukema: 200000 km Hinta: 2000 EUR

Valmistaja: MERCEDES-BENZ Malli: W124 Mittarilukema: 1000000 km Hinta: 900 EUR

Valmistaja: AUDI Malli: S2 Mittarilukema: 400000 km Hinta: 6000 EUR

Valmistaja: LADA Malli: niva Mittarilukema: 400000 km Hinta: 3200 EUR

Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 2
Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 3
Valmistaja: AUDI Malli: S2 Mittarilukema: 400000 km Hinta: 6000 EUR

Valmistaja: CITROEN Malli: Berlingo Mittarilukema: 200000 km Hinta: 2000 EUR

Valmistaja: LADA Malli: niva Mittarilukema: 400000 km Hinta: 3200 EUR

Valmistaja: MERCEDES-BENZ Malli: W124 Mittarilukema: 1000000 km Hinta: 900 EUR
```

```
Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 4
Hae autoja valmistajan mukaan: audi
Valmistaja: AUDI Malli: S2 Mittarilukema: 400000 km Hinta: 6000 EUR

Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 5
[1.] Valmistaja: AUDI Malli: S2 Mittarilukema: 400000 km Hinta: 6000 EUR
[2.] Valmistaja: CITROEN Malli: Berlingo Mittarilukema: 200000 km Hinta: 2000 EUR
[3.] Valmistaja: LADA Malli: niva Mittarilukema: 400000 km Hinta: 3200 EUR
[4.] Valmistaja: MERCEDES-BENZ Malli: W124 Mittarilukema: 1000000 km Hinta: 900 EUR
Anna poistettavan auton numero. Peruuta poistaminen syöttämällä 0. 2
Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> 3
Valmistaja: AUDI Malli: S2 Mittarilukema: 400000 km Hinta: 6000 EUR

Valmistaja: LADA Malli: niva Mittarilukema: 400000 km Hinta: 3200 EUR

Valmistaja: MERCEDES-BENZ Malli: W124 Mittarilukema: 1000000 km Hinta: 900 EUR
```

## Java

```

- '1' to add new car-
  '2' to remove car-
  '3' to print out all cars-
  '4' to arrange cars-
  '5' to find by manufacturer-
  '0' to stop-

1
give cars manufacturer:
Toyota
give cars model:
hilux
give cars mileage:
650012
give price for the car:
7640
Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
3
Valmistaja: AUDI
Malli: S2
Mittarilukema: 312444
Hinta: 8000
Valmistaja: BMW
Malli: M3
Mittarilukema: 271353
Hinta: 6000
Valmistaja: PEUGEOT
Malli: 206
Mittarilukema: 650000
Hinta: 800
Valmistaja: SUZUKI
Malli: Swift
Mittarilukema: 42020
Hinta: 1000
Valmistaja: FIAT
Malli: Punto
Mittarilukema: 230000
Hinta: 950
Valmistaja: Toyota
Malli: hilux
Mittarilukema: 650012
Hinta: 7640
Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
4
Arranged.
Arranged.
Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
3
Valmistaja: AUDI
Malli: S2
Mittarilukema: 312444
Hinta: 8000
Valmistaja: BMW
Malli: M3
Mittarilukema: 271353
Hinta: 6000
Valmistaja: FIAT
Malli: Punto
Mittarilukema: 230000
Hinta: 950
Valmistaja: PEUGEOT
Malli: 206
Mittarilukema: 650000
Hinta: 800
Valmistaja: SUZUKI
Malli: Swift
Mittarilukema: 42020
Hinta: 1000
Valmistaja: Toyota
Malli: hilux
Mittarilukema: 650012
Hinta: 7640

```

```

Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
2
[1.]
Valmistaja: AUDI
Malli: S2
Mittarilukema: 312444
Hinta: 8000
[2.]
Valmistaja: BMW
Malli: M3
Mittarilukema: 271353
Hinta: 6000
[3.]
Valmistaja: FIAT
Malli: Punto
Mittarilukema: 230000
Hinta: 950
[4.]
Valmistaja: PEUGEOT
Malli: 206
Mittarilukema: 650000
Hinta: 800
[5.]
Valmistaja: SUZUKI
Malli: Swift
Mittarilukema: 42020
Hinta: 1000
[6.]
Valmistaja: Toyota
Malli: hilux
Mittarilukema: 650012
Hinta: 7640
Give number of a car you want to delete:
5
-----
Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
3
Valmistaja: AUDI
Malli: S2
Mittarilukema: 312444
Hinta: 8000
Valmistaja: FIAT
Malli: Punto
Mittarilukema: 230000
Hinta: 950
Valmistaja: PEUGEOT
Malli: 206
Mittarilukema: 650000
Hinta: 800
Valmistaja: Toyota
Malli: hilux
Mittarilukema: 650012
Hinta: 7640
Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
5
Find car by manufacturer:
give cars manufacturer: |
fiat
Valmistaja: FIAT
Malli: Punto
Mittarilukema: 230000
Hinta: 950
Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)
2
[1.]
Valmistaja: AUDI
Malli: S2
Mittarilukema: 312444
Hinta: 8000
[2.]
Valmistaja: BMW
Malli: M3
Mittarilukema: 271353
Hinta: 6000
[3.]
Valmistaja: FIAT
Malli: Punto
Mittarilukema: 230000
Hinta: 950
[4.]
Valmistaja: PEUGEOT
Malli: 206
Mittarilukema: 650000
Hinta: 800
[5.]
Valmistaja: Toyota
Malli: hilux
Mittarilukema: 650012
Hinta: 7640
Give number of a car you want to delete:
2

```

## Muutama virheellinen syöte.

```

'1' to add new car-
'2' to remove car-
'3' to print out all cars-
'4' to arrange cars-
'5' to find by manufacturer-
'0' to stop-

hsdfghs
Invalid number
9
Can't find that option..
Valinta: 1, 2, 3, 4, 5 tai 0 (lopetta)
653
Can't find that option..
Valinta: 1, 2, 3, 4, 5 tai 0 (lopetta)
-123
Can't find that option..
Valinta: 1, 2, 3, 4, 5 tai 0 (lopetta)
ijhouyh9g8
Invalid number

Valinta: 1, 2, 3, 4, 5 tai 0 (lopetta)
1
give cars manufacturer:
Joku
give cars model:
Romu
give cars mileage:
ihan pirusti
Invalid number
2234edqdsa2
Invalid number
123123
give price for the car:
kgåke
Invalid number
32r2f3e2
Invalid number
12312

Give number of a car you want to delete:
wetggwdf
Invalid number
1234512
No car found on that spot!

Give number of a car you want to delete:
-1
No car found on that spot!

```

```

Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> eDASD
Invalid input. Try again: 9
You are back in Menu
[1]Lisää Auto
[2]Järjestä aakkosjärjestykseen
[3]Tulosta
[4]Etsi
[5]Poista
[0]Poistu
-> GGGGDQWD
Invalid input. Try again: 1
Valmistaja -> volvo
Malli -> 740
Mittarilukema -> afjeqdfq
Invalid input. Try again: dtgw2341dsq
Invalid input. Try again: 523103
Hinta -> waefqwe13321s
Invalid input. Try again: 1350

```

```

[1.] Valmistaja: CITROEN Malli: Berlingo Mittarilukema: 200000 km Hinta: 2000 EUR
[2.] Valmistaja: MERCEDES-BENZ Malli: W124 Mittarilukema: 1000000 km Hinta: 900 EUR
[3.] Valmistaja: AUDI Malli: S2 Mittarilukema: 400000 km Hinta: 6000 EUR
[4.] Valmistaja: VOLVO Malli: 740 Mittarilukema: 523103 km Hinta: 1350 EUR
Anna poistettavan auton numero. Peruuta poistaminen syöttämällä 0. 78
Invalid input. Try again: AFASRFDQW
Invalid input. Try again: 21424
Invalid input. Try again: -12
Invalid input. Try again: -2
Invalid input. Try again: g2r3
Invalid input. Try again: 456
Invalid input. Try again: 2
Menu

```



### 3 Ohjelmakoodit (C++)

#### main.cpp

```
#include<iostream>
#include<vector>
#include<memory>
#include<algorithm>
#include"car.h"
using namespace std;
vector<Car> car;
void joutainPohjalle()
{
    car.push_back(Car{ "CITROEN", "Berlingo", 200000, 2000 });
    car.push_back(Car{ "MERCEDES-BENZ", "W124", 1000000, 900 });
    car.push_back(Car{ "AUDI", "S2", 400000, 6000 });
}
void lisäys(){
    string valmistaja;
    string malli;
    int mittarilukema = 0;
    int hinta;
    cout << "Valmistaja -> "; cin >> valmistaja;
    transform(valmistaja.begin(), valmistaja.end(), valmistaja.begin(),
::toupper);
    cout << "Malli -> "; cin >> malli;
    cout << "Mittarilukema -> ";
    while (!(cin >> mittarilukema)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Invalid input. Try again: ";
    }
    cout << "Hinta -> ";
    while (!(cin >> hinta)) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Invalid input. Try again: ";
    }
    car.push_back(Car{ valmistaja, malli, mittarilukema, hinta });
}

void järjestä()
{
    sort(car.begin(), car.end());
}

void tulosta()
{
    for (const Car& car : car)
    {
        car.tulostaTiedot();
    }
}

void etsi()
{
    string etsi;
    cout << "Hae autoja valmistajan mukaan: ";
    cin.ignore();
    getline(cin, etsi);
    transform(etsi.begin(), etsi.end(), etsi.begin(), ::toupper);
    for (auto k : car) {
        if (etsi == k.getValmistaja()) {
            k.tulostaTiedot();
        }
    }
}
```

```

void poista()
{
    int position;
    int i = 1;
    for (const Car& car : car)
    {
        cout << "[" << i << ".]      ";
        car.tulostaTiedotPoistoon();
        i++;
    }
    cout << "Anna poistettavan auton numero. Peruuta poistaminen syöttämällä 0.
";

    while (!(cin >> position) || position > car.size() || position < 0 ) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Invalid input. Try again: ";
    }

    if (position != 0)
    {
        car.erase(car.begin() + position - 1);
    }
}

int main() {
    setlocale(LC_ALL, "");

    jotainPohjalle();

    while (true) {

        int valinta;
        cout << "Menu\n [1]Lisää Auto \n [2]Järjestä aakkosjärjestykseen \n
[3]Tulosta \n [4]Etsi \n [5]Poista \n [0]Poistu \n -> ";
        while (!(cin >> valinta)) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Invalid input. Try again: ";
        }

        switch (valinta) {
            case 1:
                lisäys();
                break;
            case 2:
                järjestä();
                break;
            case 3:
                tulosta();
                break;
            case 4:
                etsi();
                break;
            case 5:
                poista();
                break;
            case 0:
                return EXIT_SUCCESS;
            default:
                cout << "You are back in ";
        }

    };

}

return EXIT_SUCCESS;
}

```

## car.cpp

```

#include<iostream>
#include<memory>
#include "car.h"
using namespace std;
//staattisen tietojäsenen alustus johonkin käännösyksikköön(cpp())
//voi olla myös main

Car::Car() : mValmistaja("tuntematon"), mMalli("tuntematon"), mMittarilukema(0),
mHinta(0)
{

}

Car::Car(const string& aValmistaja, const string& aMalli, const int&
aMittarilukema, const int& aHinta)
:mValmistaja(aValmistaja), mMalli(aMalli),
mMittarilukema(aMittarilukema), mHinta(aHinta)
{

}

Car::~~Car()
{

}

void Car::setValmistaja(const string& aValmistaja)
{
    mValmistaja = aValmistaja;
}

void Car::setMalli(const string& aMalli)
{
    mMalli = aMalli;
}

void Car::setMittarilukema(int aMittarilukema)
{
    mMittarilukema = aMittarilukema;
}

void Car::setHinta(int aHinta)
{
    mHinta = aHinta;
}

string Car::getValmistaja() const
{
    return mValmistaja;
}

string Car::getMalli () const
{
    return mMalli;
}

int Car::getMittarilukema () const
{
    return mMittarilukema;
}

int Car::getHinta() const
{

```

```

        return mHinta;
    }

    bool Car::operator<(const Car& r) const
    {
        if (mValmistaja != r.mValmistaja) {
            return mValmistaja < r.mValmistaja;
        }

        return mMalli < r.mMalli;
    }

    // overload the operator> (if required)
    bool Car::operator>(const Car& r) const
    {
        if (mValmistaja != r.mValmistaja) {
            return mValmistaja > r.mValmistaja;
        }

        return mMalli > r.mMalli;
    }

    void Car::tulostaTiedotPoistoon()const
    {
        cout << "Valmistaja: " << mValmistaja << " Malli: " << mMalli << "
Mittarilukema: " << mMittarilukema << " km" << " Hinta: " << mHinta << " EUR" <<
endl;
    }

    void Car::tulostaTiedot()const
    {
        cout << "Valmistaja: " << mValmistaja;
        cout << "   Malli: " << mMalli;
        cout << "   Mittarilukema: " << mMittarilukema << " km";
        cout << "   Hinta: " << mHinta << " EUR" << endl << endl;
    }

```

## car.h

```

#pragma once
#include <string>
#include<memory>

using namespace std;

class Car {
public:
    Car();
    Car(const string& aValmistaja, const string& aMalli, const int&
aMittarilukema, const int& aHinta);
    ~Car();

    void setValmistaja(const string& aValmistaja);
    void setMalli(const string& aMalli);
    void setMittarilukema(int aMittarilukema);
    void setHinta(int aHinta);
    string getValmistaja() const;
    string getMalli() const;
    int getMittarilukema() const;
    int getHinta() const;
    virtual void tulostaTiedot() const;
    virtual void tulostaTiedotPoistoon() const;
    bool operator <(const Car& r) const;
    bool operator >(const Car& r) const;

private:
    //Tietojäsenet (data, member variables, instance variables) privaattiksi
    string mValmistaja;
    string mMalli;
    int mMittarilukema;
    int mHinta;
};

```

## 4 Ohjelmakoodit (Java)

### Car.java

```
package com.mycompany.syventavatharjoitus;
```

```
public class Car {
```

```
    private String mValmistaja;
```

```
    private String mMalli;
```

```
    private int mMittarilukema;
```

```
    private int mHinta;
```

```
    public Car( String aValmistaja, String aMalli, int aMittarilukema, int aHinta)
```

```
    {
```

```
        mValmistaja = aValmistaja;
```

```
        mMalli = aMalli;
```

```
        mMittarilukema = aMittarilukema;
```

```
        mHinta = aHinta;
```

```
    }
```

```
    public String getmValmistaja() {
```

```
        return mValmistaja;
```

```
    }
```

```
    public void setmValmistaja(String mValmistaja) {
```

```
        this.mValmistaja = mValmistaja;
```

```
    }
```

```
    public String getmMalli() {
```

```
        return mMalli;
```

```
    }
```

```
    public void setmMalli(String mMalli) {
```

```
        this.mMalli = mMalli;
```

```
    }
```

```
    public int getmMittarilukema() {
```

```
        return mMittarilukema;
```

```
    }
```

```
    public void setmMittarilukema(int mMittarilukema) {
```

```
        this.mMittarilukema = mMittarilukema;
    }

    public int getmHinta() {
        return mHinta;
    }

    public void setmHinta(int mHinta) {
        this.mHinta = mHinta;
    }

    public void tulostaTiedot()
    {
        System.out.println("Valmistaja: " + mValmistaja);
        System.out.println("Malli: " + mMalli);
        System.out.println("Mittarilukema: " + mMittarilukema);
        System.out.println("Hinta: " + mHinta);
    }
}
```

## CarInfo.java

```
package com.mycompany.syventavatharjoitus;

import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
import java.util.Scanner;

public class CarInfo {
    private List<Car> mCars = new ArrayList<Car>();
    private Scanner mScanner = new Scanner( System.in );
    public static void main( String args[] ){
        CarInfo h = new CarInfo();
        h.start();
    }

    public CarInfo(){

    }

    public void start() {

        mCars.add(new Car("AUDI", "S2 ",312444 ,8000 ));
        mCars.add(new Car("BMW", "M3 ",271353 ,6000 ));
        mCars.add(new Car("PEUGEOT", "206 ",650000 ,800 ));
        mCars.add(new Car("SUZUKI", "Swift ",42020 ,1000 ));
        mCars.add(new Car("FIAT", "Punto ",230000 , 950 ));

        int valinta;
        System.out.println(" '1' to add new car- \n '2' to remove car- \n '3' to print out all cars- \n '4' to arrange cars- \n '5' to find by manufacturer- \n '0' to stop- \n ");
        String choice = mScanner.nextLine();
        while (true) {
            try {
                valinta = Integer.parseInt(choice);
                break;
            } catch (Exception e) {
                System.out.println("Invalid number");
            }
            choice = mScanner.nextLine();
        }
    }
}
```



OUTER:

```
while (valinta != 0) {
    switch (valinta) {
        case 1 -> {
            int mittarilukema = 0;
            int hinta = 0;
            Scanner car = new Scanner(System.in);
            System.out.println("give cars manufacturer: ");
            String valmistaja = car.nextLine();
            System.out.println("give cars model: ");
            String malli = car.nextLine();
            System.out.println("give cars mileage: ");
            String lukema = car.nextLine();
            while (true) {
                try {
                    mittarilukema = Integer.parseInt(lukema);
                    break;
                } catch (Exception e) {
                    System.out.println("Invalid number");
                }
                lukema = car.nextLine();
            }

            System.out.println("give price for the car: ");
            String hin = car.nextLine();
            while (true) {
                try {
                    hinta = Integer.parseInt(hin);
                    break;
                } catch (Exception e) {
                    System.out.println("Invalid number");
                }
                hin = car.nextLine();
            }
            mCars.add(new Car(valmistaja, malli, mittarilukema, hinta));
        }
        case 2 -> {
            int i = 0;
            int poistettava;
            for (Car h : mCars) {
                System.out.println "[" + (i+1) + ".]");
                h.tulostaTiedot();
                i++;
            }
            System.out.println("Give number of a car you want to delete: ");

            String delete = mScanner.nextLine();
            while (true) {
```

```

    try {
        poistettava = Integer.parseInt(delete);
        break;
    } catch (Exception e) {
        System.out.println("Invalid number");
    }
    delete = mScanner.nextLine();
}

if( -1 < (poistettava-1) && (poistettava-1) <= mCars.size()){
    mCars.remove((poistettava - 1));
    break;
}
else System.out.println("No car found on that spot! ");
}

case 3 -> {
    for (Car h : mCars) {
        h.tulostaTiedot();
    }
}

case 4 -> {
    Collections.sort(mCars, Comparator.comparing(Car::getmValmistaja));
    System.out.println("Arranged.");
}

case 5 -> {
    System.out.println("Find car by manufacturer: ");
    Scanner etsi = new Scanner(System.in);
    System.out.println("give cars manufacturer: ");
    String etsitty = etsi.nextLine();
    for(Car h : mCars){
        if(etsitty.toUpperCase().equals(h.getmValmistaja().toUpperCase())){
            h.tulostaTiedot();
        }
    }
}

case 0 -> {
    break OUTER;
}

default -> {
    System.out.println("Can't find that option.. ");
}
}

System.out.println("Valinta: 1, 2, 3, 4, 5 tai 0 (lopeta)");
String choice2 = mScanner.nextLine();

```

```
while (true) {  
    try {  
        valinta = Integer.parseInt(choice2);  
        break;  
    } catch (Exception e) {  
        System.out.println("Invalid number");  
    }  
    choice2 = mScanner.nextLine();  
}  
  
}  
  
}
```