Automating the preheating of a car

Benefits of preheating and design of a preheater for automated preheating of a car

Bachelor’s thesis
Valkeakoski, Automation Engineering
Spring 2019
Henry Roos
ABSTRACT

The purpose of this thesis project was to examine the benefits of preheating a car and to create a program to automate the preheating process. This could be done with a compact and inexpensive computer, for example Raspberry Pi.

This project was commissioned by HAMK university of Applied Sciences. The topic was chosen based on real life experience dealing with the preheating of my own car and my personal interest in computer software.

The thesis project consisted of two parts, figuring out the overall benefits of preheating based and judging whether the whole process is even worth it, the second part was to design and create a program that could be used to automate the process. The main material used in figuring out the benefits of the preheating was derived from a bachelor’s thesis done by Jan Rautalin, a Bachelor of Engineering student from Metropolia University of applied sciences.

The main results were that preheating is very beneficial to a car, it as multiple benefits such as reduced pollution and increased safety and comfort make preheating worth while. The second result was that after completing the program and testing it, it turned out to be a completely viable option for automating the preheating process.

The conclusion of the thesis project was that preheating is beneficial to a personal vehicle and that the program created works and can be used to automate the process. Next step in developing the program would be to integrate it to a remote controlled socket and figure out a way for the program to get accurate temperature readings regardless of the location of the device.

Keywords: Java, preheating, tomcat
Pages: 39 pages
LIST OF ABBREVIATIONS

API: Application programmin interface
CSS: Cascading style sheets
DOM: Document object model
GUI: Graphical user interface
HC: hydrocarbon
HTML: Hypertext markup language
IDE: Integrated development environment
JVM: Java virtual machine
kW: kilowatt
PC: Personal computer
URL: Uniform resource locator
W: Watt
XSS: Cross-site scripting
## CONTENTS

1 INTRODUCTION .................................................................................................................. 1

2 PREHEATING OF A CAR ..................................................................................................... 2
   2.1 Different types of preheaters ....................................................................................... 2

3 ELECTRICAL PREHEATING ............................................................................................... 3
   3.1 Coolant heaters .......................................................................................................... 3
   3.2 Block heaters ............................................................................................................. 4
   3.3 Inline heaters ............................................................................................................. 5
   3.4 Radiation heaters ....................................................................................................... 7
   3.5 Electrical interior heaters .......................................................................................... 9

4 FUEL POWERED HEATERS ............................................................................................. 10

5 BENEFITS OF PREHEATING ........................................................................................... 12
   5.1 Improved fuel efficiency ........................................................................................... 12
   5.2 Reduced pollutions .................................................................................................... 14

6 ANALYSIS OF THE STATISTICS ON PREHEATERS ....................................................... 16
   6.1 Improved fuel efficiency ........................................................................................... 16
   6.2 Improved emissions ................................................................................................... 18
   6.3 Summary of analysis ................................................................................................ 18

7 JAVA ................................................................................................................................... 19
   7.1 Concurrent computing ............................................................................................... 19
   7.2 Object oriented programming ................................................................................... 20
   7.3 Write once, run anywhere ......................................................................................... 20
   7.4 Java in this project .................................................................................................... 20

8 APACHE TOMCAT ........................................................................................................... 21
   8.1 Introduction ................................................................................................................. 21
   8.2 Servlet ....................................................................................................................... 22
   8.3 Web server ................................................................................................................ 22
   8.4 Role of Apache Tomcat in this project ...................................................................... 22

9 ECLIPSE ............................................................................................................................. 23
   9.1 Introduction ................................................................................................................. 23
   9.2 Integrated development environment (IDE) ............................................................... 23

10 JSOUP ................................................................................................................................. 24
   10.1 Introduction ............................................................................................................... 24
   10.2 Role of Jsoup in this project .................................................................................... 24
1 INTRODUCTION

The purpose of this thesis project was to examine what are the benefits of preheating a car and how one could create a program used to automate the preheating process to make it easier and also to conserve energy and maximise the benefits resulted by preheating without wasting energy. This program could be used with some small and cheap computer, for example raspberry pi, to control the preheating process.

This project was commissioned by HAMK University of Applied Sciences. The topic was chosen based on real life experiences dealing with the preheating of my own car, wanting to find a way to ease the process of setting the timers, figuring out the upcoming weather and calculating the optimal time needed for preheating. Another reason for the topic was my personal interest in computer software and learning how to code.

The thesis has 2 main parts, figuring out based on pre-existing data and studies the overall benefits of preheating and judging whether the whole process is even worth it, the second part was to design and create a program that could be used the automate the process. The main material used in figuring out the benefits of the preheating was a thesis done by Jan Rautalin, a Bachelor of Engineering student from Metropolia University of applied sciences.

The main results were that preheating is very beneficial to a car, the overall energy saved by preheating is not enough to cover the cost of the energy used in preheating but the other benefits such as reduced pollution and increased safety and comfort make preheating worth while. The second result was that after completing the program and testing it, it turned out to be a completely viable option of automating the preheating process, there are quite of few of remotely controlled electrical sockets on the market and one could use the program to control one of such sockets to achieve the desired automated result.

The conclusion of the thesis was that preheating is beneficial to ones car and that the program created works and could be used to automate the process. Next step to developing the program would be to integrate it to a pre-existing remote controlled socket with the help of raspberry pi or some similar small cheap computer to make it a complete working system. Also figuring out a way for the software to know the location to get accurate temperature readings in multiple locations or to integrate a temperature meter to the program.
2 PREHEATING OF A CAR

Electrical preheating of a car is a way of heating a car’s engine and the interior during cold weather in order to achieve multiple different positive results such as saving gas, reducing pollution and increasing the safety of the driver.

Preheating is mainly done by heating the car’s coolant or oil. The most efficient way to heat the cars engine is to heat the coolant, this can be done by electrical heaters that are powered by the houses main electrical outlet or in some cases heaters powered by diesel fuel. These preheaters are not installed into the vehicles by the manufacturer due to the fact that in most of the countries preheating is not necessary because of the climate. Because of this it is the importer of the car who is ultimately responsible for the installation or in some cases the owner of the car themself. These are installed into the cars cooling system or if that is not possible there are radiation heaters available that can heat the cars oil. The most effective method is heating the coolant, due to this the radiation heaters that heat the oil are only used if the coolant heaters are incompatible with the particular model of a car. Electrical preheating is very common in Finnish cars almost all new cars are already fitted with one.

2.1 Different types of preheaters

Preheaters can be divided into two categories based on their powersource. This is shown in Table 1.

Table 1. Different types of preheaters.

<table>
<thead>
<tr>
<th>Electrical preheaters</th>
<th>Radiation heater</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liquid heaters</td>
<td></td>
</tr>
<tr>
<td>• Block heater</td>
<td>• For motor oil or Coolant</td>
</tr>
<tr>
<td>• Inline heater</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fuel powered heaters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Factory installed</td>
<td>Retrofitted</td>
</tr>
<tr>
<td>• No user interface, only not used for preheating only used to heat up the cars engine while driving during cold weather</td>
<td>• With user interface, used for preheating</td>
</tr>
<tr>
<td>• With user interface, used for preheating.</td>
<td></td>
</tr>
</tbody>
</table>
3 ELECTRICAL PREHEATING

The most common preheater is an electrical preheater. These are most commonly found in gasoline powered cars and they are retrofitted to the cars after they are imported to Finland. Electrical preheaters are slower than fuel powered ones but they do not drain the car’s battery so they can be used more freely. Electrical preheaters were the main focus of this thesis due to the fact that they are the ones that lack the complex user controls that ease their use. Electrical preheaters do not usually have any controls built in them they are always on if they are connected to a socket which means that if you want to preheat your car you have to manually plug in your car to start the eating or set up somekind of a timer that turns the heater on by itself. Electrical preheaters have two main categories, the ones that heat the coolant of the car or the ones that heat the motor oil.

3.1 Coolant heaters

The most common electrical preheaters are the ones that heat up the coolant of the car. This is the most effective way to heat the whole engine of the car due to the fact that the coolants’ purpose in the car normally is to cool down the engine which means that the coolant is spread all over the engine, now that process is just reversed. Coolant heaters are split in to three different types: a block heater an inline heater and a radiation heater.
3.2 Block heaters

A Block heater is an external heating element which is connected to the engine block and it heats up the coolant. When the coolant heats up it will slowly start to circulate in the engine slowly heating it up. Block heaters are getting more uncommon in new cars due to the engines not having any place to install them. People often use the name block heater to refer to their preheater even though these days they are very rarely used anymore. Block heaters are effective way of heating the coolant, they are quite powerful (500-600w). (Vähäsarja, 2018)

Figure 1. Block heater (SafeStart 100, 2013)
3.3 Inline heaters

Inline heaters are a common option in cars these days, they do the same job as block heaters but they are connected to the engine differently. Their heating power is similar to block heaters which makes them very effective way of heating up the engine. Inline heaters are connected to the coolant hose that runs to the engine block. They heat the coolant in the hose and the heated coolants starts to slowly move in the engine heating it up. The issue with inline heaters is that not all car engine types work with them. Some engine types are constructed in a way that there is an electrically controlled heating system that controls the flow of the coolant using several thermostats, valves and pumps. This means that when the engine of the car is not running the coolant the inline heater is heating will not be able to run freely inside the engine preventing the heating and can cause damage to the heater due to overheating. (Tallirenki, 2008)

Figure 2. Inline heater(SafeStart PTC, 2017)
Figure three presents the change of temperatures in different parts of the engine when heating with an inline heater. The differently coloured graphs represent measurements done with different outside temperatures. The numbers at the end of the each lines represent the final temperature reached at the end of the test. (Rautalin, 2013, p.24)

Figure 3. The development of temperatures in different parts of the engine as a function of time when using a 500W inline heater. (Rautalin, 2013, p.24)
3.4 Radiation heaters

Radiation heaters are used when neither of the previously presented methods are available. They are not as effective and they are less powerfull which means that they require longer operation time. Radiation heaters have the power of approximately 300W. They are installed to the outer side of the engine. They don't have any access inside the engine their function is to be in tight contact with the outer side of the engine block and transfer the heat by radiating it inside the engine. The heating element is molded to fit the outer side of the engine perfectly to help the heat transfer better. Radiation heaters are usually used to heat up the motor oil in the engine. Due to their efficiency and power it takes longer to heat a car using them compared to the inline heater and the block heater. Radiation heaters can be installed either into the engine block or the oil pan.

Figure 4. Radiation heater(SafeStart 800, 2017)
Figure 5. The development of temperatures in different parts of the engine as a function of time when using a 300W radiation heater. (Rautalin, 2013, p.25)

Figure five presents the change of temperatures in different parts of the engine when heating with a radiation heater. The differently coloured graphs represent measurements done with different outside temperatures. The numbers at the end of each line represent the final temperature reached at the end of the test. (Rautalin, 2013, p.25)
3.5 **Electrical Interior heaters**

Many of the Electrical powered engine heaters are also used with a electrical interior heater that is installed inside the car to heat the interior. The goal of this is different from the engine heater, this does not directly reduce pollution or increase fuel efficiency but what it does it warms up the air inside the car melting all the snow and ice from the window improving the safety of the car and also making it significantly more comfortable to use. Electrical interior heaters do save fuel and reduce pollutions in an indirect way, when the interior of the car is warm and the windshield is not frozen the drive does not have to keep their car running on idle before starting to drive. Electrical interior heaters are commonly just a heating element with a fan with the power of approximately 1400W, they are connected to a electrical socket installed inside the car that is powered the same way as the electrical engine preheaters. The problem with having both engine preheater and the interior heater on at the same time is that their combined power can be too much for the fuses to handle so one should be aware of that before using them at the same time.

![Electrical Interior heater](image)

**Figure 6. Interior heater (Termini II, 2017)**
Fuel powered heaters can be factory installed or retrofitted. They are very different from the electrical heaters. Fuel heaters use the car's own fuel supplies to generate heat and they spread it to the engine and the interior of the car taking power from the car's own batteries. They are vastly more powerful compared to electrical heaters (4kW) which means that the required heating time using them is only 30 to 60 minutes even in cold weather. The fact that fuel powered heaters use the car's battery while the car isn't running means that when using them one has to be wary not to drain their battery too much, the recommendation is to drive the same amount of time as you used the preheater for.

Fuel powered heaters use approximately 100W of power to run the heater, water pump, and the car's own interior heaters' fans. Not all fuel powered heaters are used for preheating; some of them are used for heating the interior of the car and the engine while driving because specially modern diesel-powered engines do not produce enough excess heat to heat up the interior of the car, these heaters are not used for preheating the car. The control of the heater happens typically by an installed timer or with a radio-controlled controller or in some cases with a smartphone application. (Rautalin, 2013, p.7)

Fuel powered heaters can be used in environments where external electrical power sources are not available, the one restriction is thought that they cannot be used indoors if the ventilation is not properly done due to the exhausts from the heater. Another restriction is if there is flammable material under the car such as grass. (Rautalin, 2013, p.8)

Figure 7. Fuel powered heater (Webasto vesilämmittimet, N.D)
Figure 8. The development of temperatures in different parts of the engine as a function of time when using a 5 kW fuel heater. (Rautalin, 2013, p. 25)
5 BENEFITS OF PREHEATING

The main benefits of preheating the engine of a car are improving the fuel efficiency and reducing the pollutions of the car. When an engine is cold it doesn’t run as smooth as a warm engine resulting in slightly increased fuel consumption and also the burning of the fuel does not happen as cleanly compared to a warm engine resulting in increased pollutions.

5.1 Improved fuel efficiency

![Figure 9. Fuel consumption reduction by preheating 0-4 km trip (L/100 km)](Rautalin, 2013, p. 29)
5.2 Reduced pollutions

One of the big benefits from preheating are the reduced hydrocarbon (HC) emissions produced by the engine. When the cars engine is warm the fuel burns better resulting in reduced hydrocarbon emissions and also reduced particle pollution. Reduced particle pollution is mainly significant in gas powered cars due to most new diesel cars having particle filters installed in them.
Figure 11. Reduction of carbohydrate emissions by preheating. (g/km) (Rautalin, 2013, p.32)

Figure 12. Reduction of carbohydrate emissions by preheating. (%) (Rautalin, 2013, p.32)
Figure 13. Reduction of particle emissions by preheating. (g/km) (Rautalin, 2013, p.34)

Figure 14. Reduction of particle emissions by preheating. (%) (Rautalin, 2013, p.34)
6 ANALYSIS OF THE STATISTICS ON PREHEATERS

6.1 Improved fuel efficiency

Based on the statistics preheating the engine of a car saves some fuel during the first 4 kilometers of driving. When comparing fuel consumption in temperatures below zero degrees Celsius between a cold and a preheated car it was seen that a preheated gasoline powered car saves approximately 5% of fuel for diesel 11% was saved and for ethanol (E85) powered cars 9% was saved. In temperatures above 0 degrees Celsius the fuel savings were nonexistent for gasoline power cars, for diesel they were 9% and for ethanol (E85) they were 4%. Preheated hybrid cars gasoline consumption was reduced by 12% above below zero degrees Celsius and at zero degrees Celsius approximately 8% when compared to a cold car. Even though there is some fuel saved, the overall fuel savings are quite small specially considering that they are only during the first 4 kilometers of driving. (Rautalin, 2013, p.35)

Figure 15. Fuel consumption difference between a warm and a preheated engine during 4 km trip in temperature of -7 °C

Figure 15. Fuel consumption difference between a warm and a preheated engine during 4 km trip. (Rautalin, 2013, p.35)
As we see in figure 15 the fuel consumption savings with preheating are miniscule compared to the savings gotten when the car has been driven until the engine is hot, this is due to the cars engine having significantly more substantial heating power compared to the small electrical or fuel powered preheaters.

Figure 16. The ratio of energy used in preheating to the achieved energy savings on a 4 km trip at temperature of -20 °C.

![Bar chart](image)

Figure 16. The ratio of energy savings achieved to the energy consumption of preheating. (Rautalin, 2013, p.36)

As seen in the figure 16 when taking in to account the energy used in the preheating and comparing to the actual fuel savings at -20 degrees celsius the actual savings in fuel are largely overshadowed by the required energy in preheating meaning that it is not worth preheating your car just for the fuel savings.
6.2 Improved emissions

Based on the measurements presented on the graphs it is seen that preheating has a large effect on the emissions done by the car in the first four kilometers of driving. Specially the carbohydrates that are very unhealthy have been reduced significantly. The carbohydrate emissions in below zero degrees celsius condition were reduced by 56 % for gasoline power cars, 74 % for diesel powered, 42 % for E85 and 57 % for hydrid cars.

Particle emissions were reduced much more significantly in gasoline powered cars because diesel powered cars already have a built in particle filters. Preheating reduced the particle emissions of gasoline powered cars by 63 % in below zero degrees celsius temperatures.

6.3 Summary of analysis

Based on the information and graphs it can be seen that preheating a car is beneficial to it in many ways, the fuel conservation is not enough to warrant preheating unless you are able to get the electricity for the preheating for free, which many times can be the case when you are preheating your car at work or school for example. The biggest benefit for preheating was the reduced emissions, when heated the car pollutes significantly less compared to starting it cold meaning an overall reduction in pollution.

One important aspect not measured by the graphs is the safety and comfort of the car after its preheated, heating the inside of the car means that the driver has improved vision of their surroundings.

Based on the information it doesn’t matter much what kind of a preheating solution one has in their car, as long as its run the proper amount of time depending on the outside temperature, this is the important point that a automated preheater can solve.
Java is a programming language developed by Sun Microsystems. It’s a part of a technology tree which consists of multiple different developing platforms once of them being the hardware independent object based programmin language with its runtime invironments and classes. Java platform is used in billions or computers anything between small compact inexpensive computers like Raspberry pi or even more compact computers designed for some very simple tasks including mobile phones all the way to expensive supercomputers. (Gosling 2015, 1).

Java is a programmin language that has been around for long time, it was created in 1995 and still to this day it is used and the most important benefits java offers is the ease of coding and the fact that the programs created can run on many different platforms regards of the hardware used reducing the need to modify the software for every single platform. Java programmin language is related to the C++ language but it is organized quite differently. (Gosling 2015, 1).

The java programming language is typed in a way that it clearly differentiates between compile errors making the overall coding experience more pleasant and the eventual debugging smoother. (Gosling 2015, 1).

Java was originally developed by Ryan Gosling at Sun Microsystems and then bought by Oracle. The lastest version of Java is supported currently by Oracle.

7.1 Concurrent computing

Concurrent computing is computing in which several different computations are executed during different time periods at the same time. A concurrent system is something where a computation can advance without having to wait for other computations to finish. The advantages of concurrent computing are multiple, for example it allows the program to complete more tasks in a given time period. It also increases the responsiveness of the system due to the program not having to wait for operations to finish.
7.2 Object oriented programming

The java programming language is class based and object-oriented language.

An object-based Java application is a Java application whose design is based on declaring classes, creating objects from them, and designing interactions between these objects. (Jeff Friesen, 2015.)

A class is a template for manufacturing objects. You declare a class by specifying the class keyword followed by a non-reserved identifier that names it. A pair of matching open and close brace characters ( { and }) follow and delimit the class's body. A Java application is implemented by one or more classes. Small applications can be accommodated by a single class, but larger applications often require multiple classes. In that case one of the classes is designated as the main class and contains the main() entry-point method. (Jeff Friesen, 2015.)

7.3 Write once, run anywhere

“write once, run anywhere” is a slogan created by Sun microsystems to illustrate the benefits of java language. It means that once you have written the code once you can then run it on any device with Java virtual machine (JVM), without having to change the code.

This means a programmer can develop code on a PC and can expect it to run on Java enabled cell phones, as well as on routers and mainframes equipped with Java, without any adjustments. This is intended to save software developers the effort of writing a different version of their software for each platform or operating system they intend to deploy on (Langey, 2002.)

7.4 Java in this project

Java was the main programming language used to make to software in the Thesis, it was chosen due to the ease of use and for the fact that it can be used on many different devices reducing the need for modifications to the software.
8 APACHE TOMCAT

8.1 Introduction

The Apache Tomcat software is an open source implementation of the Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket technologies. The Java Servlet, JavaServer Pages, Java Expression Language and Java WebSocket specifications are developed under the Java Community Process. (Apache Tomcat n.d.)

Apache Tomcat software powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. (Apache Tomcat n.d.)

Figure 17. JSP container (Wikipedia)
8.2 **Servlet**

Servlets are modules of Java code that run in a server application (hence the name "Servlets", similar to "Applets" on the client side) to answer client requests. Servlets are not tied to a specific client-server protocol but they are most commonly used with HTTP and the word "Servlet" is often used in the meaning of "HTTP Servlet" (Zeiger, 1999.)

Servlets make use of the Java standard extension classes in the packages javax.servlet (the basic Servlet framework) and javax.servlet.http (extensions of the Servlet framework for Servlets that answer HTTP requests). Since Servlets are written in the highly portable Java language and follow a standard framework, they provide a means to create sophisticated server extensions in a server and operating system independent way (Zeiger, 1999.)

Typical uses for HTTP Servlets include: Processing and/or storing data submitted by an HTML form, providing dynamic content, e.g. returning the results of a database query to the client, Managing state information on top of the stateless HTTP, e.g. for an online shopping cart system which manages shopping carts for many concurrent customers and maps every request to the right customer (Zeiger, 1999.)

8.3 **Web server**

A web server program is software that runs on the web site hosting Server computer. Its main purpose is serving web pages; which means it waits for requests from web browsers (also known as clients) and responds by sending the required data back. (What is web server – a computer OR a program? n.d.)

8.4 **Role of Apache Tomcat in this project**

Apache Tomcat was used to create a user interface to the preheater program. It connects the web browser and the java program together, and acts as link between them, passing the information inbetween. This was done to make the use of the program practical, it allows the users to easily set the time when they want to have their car heated, and the program does the rest.
9  ECLIPSE

9.1  Introduction

Eclipse is an integrated development environment (IDE) used in computer programming, and its the most widely used java IDE (White, 2014.)

9.2  Integrated development environment (IDE)

An integrated development environment (IDE) is a software suite that consolidates the basic tools developers need to write and test software. Typically, an IDE contains a code editor, a compiler or interpreter and a debugger that the developer accesses through a single graphical user interface (GUI). An IDE may be a standalone application, or it may be included as part of one or more existing and compatible applications. (Rouse, 2016.)

An IDE's toolbar looks much like a word processor's toolbar. The tools in the toolbar facilitate color-coding, source-code formatting, error diagnostics, and reporting and intelligent code completion. The interface allows the developer to compile and execute code incrementally and manage changes to source code in a uniform manner. IDEs are typically designed to integrate with third-party version control libraries, like GitHub or Apache Subversion. (Rouse, 2016.)
10 JSOUP

10.1 Introduction

Jsoup is a Java library that is used with HTML websites to parse and extract data from them. All it needs is a URL and after that it can extract all the data in the website and make it easy for the user to read through it to find some specific thing they were looking for. (Hedley, n.d.)

Jsoup is designed to work on all different types of websites and different HTML varieties found on the internet. (Hedley, n.d.)

10.2 Role of Jsoup in this project

Basically jsoup was used in the program to extract HTML data from the website and transform it into a format that the java program can understand, and to find the one specific data wanted and eliminate the rest.

HTML parsing is basically: taking in HTML code and extracting relevant information like the title of the page, paragraphs in the page, headings in the page, links, bold text etc.
11 PREHEATER PROGRAM

11.1 Preheater web application

The preheating automation is done by a web application which lets the user specify a date and a time when they wish to start using their car and after that the program does the rest. The program apache tomcat is used to create the web application, Apache tomcat connects the java program and the user interface together exchanging information between them. The web application is created when tomcat has been started and after that the website is ready to accept the users chosen preheating times.

Figure 18. Preheater web application chart.
The way the web application works is that when tomcat is started it calls the initialize function which starts the web application and also starts the actual preheater program which controls the whole preheating process. The preheater program is instanced so there can only by one of them at any given point in time. After the preheater program has started it turns on the measurer class that begins to monitor the outside temperature by taking it from a website. Preheater program also turns on the storage handler class that is used to store the starting and stopping times. After a user has chosen a time when they want their car to be ready tomcat program will send the time to the storage handler and storage handler will save it to the disk into a storage.dat file.

**Set time when you want to start using your car**

| Select day | Select hour | Select minute | Set |

**Stored starting times:**

No times

Temperature from web : -20 and preheating time is 4 hours and 23 minutes

Figure 19. Preheater web application interface.

This is what the preheating programs web interface looks like when opened, it allows the user to choose a date when they want their car to be warm, it shows the already stored starting times and it also shows the outside temperature and the approximated preheating time based on the current temperature. All the date here is handled by Apache tomcat, it takes the outside temperature the calculated preheating time and the stored times from the program and after the user has specified a time when they want their car to be heated it sends that data to the program.

**Set time when you want to start using your car**

| Select day | Select hour | Select minute | Set |

**Starting times:**

<table>
<thead>
<tr>
<th>Saturday</th>
<th>Select hour</th>
<th>Select minute</th>
<th>Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>5:58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monday</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuesday</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wednesday</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thursday</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Friday</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Temperature from web : 6 and preheating time is 0 hours and 0 minutes

Figure 20. Preheater web application interface.
The user can select the weekday they want their car to be heated, the list changes based on the current day showing the current date on the top.

Set time when you want to start using your car

Select day Select hour Select minute Set

Stored start

2019-03-30 17:00

Temperature from

and preheating time is 0 hours and 0 minutes

Figure 21. Preheater web application interface.

Selecting the hour.
Set time when you want to start using your car

<table>
<thead>
<tr>
<th>Select day</th>
<th>Select hour</th>
<th>Select minute</th>
<th>Set</th>
</tr>
</thead>
</table>

**Stored starting times:**

2019-03-30 17:58

Temperature from web: 6 and 40 is 0 hours and 0 minutes

Figure 21. Preheater web application interface.

Selecting the minute with 10 minutes accuracy, higher accuracy is not needed.

Thanks for Tuesday at 09:10 Your preheating will be ready at Tue Apr 02 09:10:00 EEST 2019 In 2 days 13 hours and 13 minutes

**Stored starting times:**

2019-04-01 02:10
2019-04-02 01:10
2019-04-02 09:10

Figure 22. Preheater web application interface.

After the user has set the specified time the program will show the user exactly when their car will be ready for usage and how long till that time. The stored times are shown below that.
11.2 Preheater class

The preheater program is comprised of 6 different classes the class preheater is the main class where most things are happening.

```java
import java.io.IOException;
import java.text.ParseException;
import java.util.Calendar;
import java.util.Iterator;
import java.util.Vector;

import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
```

Figure 23. Preheater class imported premade classes

These are the imported premade classes from the library that the class needs to use. java.io.IOException tells the program that something went wrong with the input or outputs and it send back information on exactly what went wrong.

java.text.ParseException Informs if the text wasn’t parseable.

java.util.Calendar Includes time and date handling tools.

java.util.iterator Eases the usage of tables.

java.util.vector Expands tables based on the amount of information coming.

org.apache.logging.log4j.LogManager Creates a logger for writing information into logfile.

org.apache.logging.log4j.Logger Is created by logmanager for writing information into the logfile (info,debug,warning,error,critical). after 4M of logs are collected the older ones are removed.

After all the required classes are imported the variables are defined.

The preheater class is a thread, after it has been created it starts a loop. It checks whether the preheating has to be started or stopped based on the times from the storage, this continues indefinitely.
The way the preheater class figures out if preheating needs to be started or stopped works by the class asking the measurer class that was is the current outside temperature, after that it asks the motivator class what is the needed preheating time for the specific temperature and then it figures out when the preheating has to be started. If the time before the preheating has to be started is below 0 seconds it starts the preheating. The program is constantly monitoring if the preheating needs to be stopped/started. After the preheating is completed the program goes through the old saved times and removes them.

Method Summary:

addTime (java.util.Calendar cal): Adds starting time to the storage
getRecords(): Get all stored starting times in list
calculatePreheatingTimeInMinutes(int temp): Calculates the preheating time in minutes according to the given outdoor temperature
getInstance(): Get PreHeater instance
getTemperature(): Get outdoor temperature
run(): Working loop for the PreHeater

```java
// Check if preheating state should be changed
boolean heatingOn = this.heatingOn;

// Current needed heating time
int temp = this.measurer.getTemperature();
int preheatingTime = Motivator.calculatePreheatingTimeInMinutes(temp, this.temperature);

// Get earliest heating starting time
String earliestRecord = this.storage.getFirstRecord();
// Calculate heating start for the first one
firstCal.add(Calendar.MINUTE, -preheatingTime);
log.debug( "pre: " + firstCal.get(Calendar.HOUR).toString());

// Calculate minutes to heating start
long msToHeatingStart = firstCal.getTimeInMillis() - System.currentTimeMillis();
int minsToHeatingStart = (int)(msToHeatingStart/1000/60);

// Are we too late for some reason?
if( !this.heatingOn && minsToHeatingStart < 0 )
{
    log.error( "Heating start for " + earliestRecord + " was not started in time");
}

// Do we need to start heating now?
if( minsToHeatingStart <= 0 )
{
    // Heating needed now
    heatingOn = true;
}
else
{
    // Heating not needed now
    heatingOn = false;
}
```

Figure 24. Preheater class methods.
11.3 DateUtil class

Imported classes:

```java
import java.util.Calendar;
import java.util.Iterator;
import java.util.Locale;
import java.util.Map;
import java.util.Vector;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
```

Locale: determines the format of the date.

Map: returns a map containing all names of the calendar field in the given style and locale of their corresponding field value.

The purpose of the class is to deal with the format of the dates and to determine the exact time when the preheating time is wanted, the data the user inserts into the website has to be translated into a format that the program can understand. The DateUtil class gives the preheater webinterface the available days/hours/minutes that you can set the timer for. It also gives them in the correct order based on the date.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static java.util.Vector</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static java.util.Vector</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static java.util.Calendar</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static java.lang.String</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static java.lang.String</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static java.util.Vector</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Figure 25. DateUtil class method summary.
11.4 Measurer class

Imported Classes:

```java
import java.io.IOException;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;
```

Jsoup: Jsoup is a html parser, it is used to find the temperature in "Pirkkala airport weather station" from the html code.

The purpose of the measurer class is to find out the current outside temperature for the program to use in calculating the required preheating time. The way the class currently gets the temperature is that it goes to a website that has live weather data from weather stations all over Finland, and it uses the Jsoup html parser to find the data from the website. Currently it finds the weather in “pirkkala airport weather station” which is the closest weather station to the current location of the program, this could be changed into basically any location. Jsoup finds the table row that starts with pirkkala, from that it checks the right column and from there it is able to find the current temperature. The temperature is checked once every 10 minutes.

This class could be changed to get the temperature reading from a outside probe but for the purposes of this program to check its functionality it gets the temperature from the internet.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getTemperature()</td>
</tr>
<tr>
<td>Static void main(java.lang.String[] args)</td>
</tr>
<tr>
<td>void run()</td>
</tr>
</tbody>
</table>

Figure 26. Measurer class method summary
11.5 Motivator class

The Motivator class is just used to calculate the preheating time needed in minutes based on the outdoor temperature. The basis for the calculation is a chart from the website “motiva” that gives rough preheating times needed in certain temperatures, this chart was transformed into a mathematical formula that the program uses for its calculation.

![Motiva preheating chart](image1)

**Figure 27. Motiva preheating chart. (Motiva.fi)**

<table>
<thead>
<tr>
<th>Outdoor temperature °C</th>
<th>Motiva/min</th>
<th>Temperature difference °C</th>
<th>Linear/min</th>
<th>Error/min</th>
<th>Formula1/min</th>
<th>Error/min</th>
<th>Formula2/min</th>
<th>Error/min</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>30</td>
<td>-5</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-5</td>
<td>60</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-10</td>
<td>70</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-15</td>
<td>80</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-20</td>
<td>90</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-25</td>
<td>100</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-30</td>
<td>110</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>-35</td>
<td>120</td>
<td>-15</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>30</td>
</tr>
</tbody>
</table>

![Motiva preheating chart transformed into excel](image2)

**Figure 28. Motiva preheating chart transformed into excel**
The preheating chart from motiva was transformed into excel where it was drawn into a line, from there a mathematical equation was figured out that most closely represents the Motiva line. Drawing just a linear line wouldn’t have been sufficiently accurate.

The Motiva chart for preheating in different temperatures for electrical and fuel powered preheaters, for the purposes of this program only electrical ones were considered because fuel powered preheaters use their own internal timer system.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int calculatePreheatingTimeInMinutes(int temperature)</td>
</tr>
<tr>
<td>Calculates preheating time in minutes according to given outdoor temperature</td>
</tr>
<tr>
<td>static int calculatePreheatingTimeInMinutes(int temperature, boolean testing)</td>
</tr>
<tr>
<td>Calculates preheating time in minutes according to given outdoor temperature</td>
</tr>
</tbody>
</table>

Figure 29. Motivator class method summary.

```
// Calculate estimated preheating time according to Motiva
time = 30 + Math.pow( tempDiff, 2 )/3 + tempDiff*5;
```

The formula used to calculate the preheating needed.

11.6 PreHeaterinputervlet class

Imported classes:

```java
import java.io.*;
import java.util.Calendar;
import java.util.Iterator;
import java.util.Vector;
import javax.servlet.*;
import javax.servlet.http.*;
import org.apache.logging.log4j.LogManager;
import org.apache.logging.log4j.Logger;
```

javax.servlet: Offers the servlet interface.
This class is used to produce the website that controls the preheater program. The browser requests the preheater page from the server and the server forwards the request to the appropriate servlet then the servlet starts the preheater program. The servlet transfers the data inputted by the user to the preheater software. The page offers the user to set the time they want their car to be ready, it also has the stored times that are not expired. It also gives the outside temperature and the calculated preheating time based on the outside temperature. The servlet stores the set time and gives it to preheater program.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void doGet(HttpServletRequest request, HttpServletResponse response)</td>
</tr>
<tr>
<td>Serve HTTP GET For showing starting time selections</td>
</tr>
<tr>
<td>void doPost(HttpServletRequest request, HttpServletResponse res)</td>
</tr>
<tr>
<td>Serve HTTP POST For storing selected starting time</td>
</tr>
<tr>
<td>void init(ServletConfig servletConfig)</td>
</tr>
<tr>
<td>void showRecords(java.io.PrintWriter out)</td>
</tr>
<tr>
<td>Output stored starting time records</td>
</tr>
</tbody>
</table>

Figure 30. Summary of PreHeaterinputServlet class methods

### 11.7 StorageHandler class

Imported Classes:

```java
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.TreeSet;
import java.util.Vector;
```

BufferedReader is used to read data from the disk, efficient reading of characters and line. 
FileReader is a convenience class for reading character files 
PrintWriter is used to write to the disk.
The StorageHandler class is used to store and read starting times set by the user in the website created by Tomcat. When the Tomcat servlet gives the storage handler a time it checks if the time is non existing, if it is it adds it to the disk. Any class can ask about the times saved on the disk to see if there are any upcoming times. The Preheater class uses the stored times to start the preheating process. The Tomcat servlet uses the saved times to present to the user the current saved upcoming preheating times stored. After a time has expired the storagehandler deletes it from the memory to avoid wasting disk space.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| boolean addTime(java.util.Calendar cal)  
  Add starting time to storage |
| void deleteAllRecords()  
  Delete all records Used only for testing |
| boolean deleteRecord(java.lang.String record)  
  Delete record |
| boolean deleteRecords(java.util.Iterator records)  
  Delete list of records |
| static java.util.Calendar deserializeTime(java.lang.String time)  
  Deserialize time from storage |
| java.lang.String getFirstRecord()  
  Get oldest record |
| static StorageHandler getInstance()  
  Get StorageHandler instance |
| java.util.Vector getRecords()  
  Get list of stored records |

Figure 31. Summary of StorageHandler class methods
12 CONCLUSION

The main objective of this thesis project was to figure out how beneficial preheating a car is, and how to automate the process to make it easier and more efficient.

The conclusion of this thesis project was that preheating has multiple different benefits to a car, including reduced fuel usage, safety and reduced pollutions. Based on this information preheating is something that is worth conducting on a personal vehicle. The program created to help automate the process of the preheating to make it more efficient and conserve energy while maintaining maximal benefits was found to be working very well. The was program able to automate a large part of the preheating removing the need for the end user to figure out the exact timings of their preheating and with a control system preheating could be turned on remotely. The program can run on many different devices without any need for modifications because it was coded with Java. The software works as planned, it is easy to run on any machine and it achieves what it was supposed to achieve.

The next steps in developing the program would be to fit the software with a compact computer like Raspberry pi for example, make it control a remotely controlled power socket to turn the heating on and off and add a thermometer to it to get precise temperature readings from outside.
REFERENCES

Apache Tomcat n.d Accessed 24.3.2017
http://tomcat.apache.org/


Accessed 14.5.2019

Accessed 14.5.2019

Accessed 14.5.2019

Friesen, J. 2015 Java 101: Classes and objects in Java Accessed 20.3.2017


Hedley, J. n.d. soup: Java HTML Parser Accessed 27.4.2017
https://jsoup.org/

http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html

https://web.archive.org/web/20130211022952/http://www.hs.fi/artikkeli/Lohko%C3%A4mmitin+ja+k%C3%A4ynnistys/1135235123914 Accessed 31.5.2019
Langley, N. 2002 Write once, run anywhere? Accessed 20.3.2017
http://www.computerweekly.com/feature/Write-once-run-anywhere


http://searchsoftwarequality.techtarget.com/definition/integrated-development-environment


What is web server – a computer OR a program? n.d Accessed 30.3.2017
http://www.webdevelopersnotes.com/what-is-web-server

Accessed 14.5.2019

Vähisarja, S. 2018. "Valtaosa ei tiedä, mikä lämmitin omassa autossa on" – säteilylämmittin vaatii lohkolämmittintä pidemmän käyttöajan
https://yle.fi/uutiset/3-10029329 Accessed 31.5.2019