



Antti Kruuti e2101587, Miika Rinta-Korkeamäki e1900516

Java harjoitustyö

Selostuksen palautuspäivä: 16.12.2022

Työryhmän jäsenet: Antti Kruuti, Miika Rinta-Korkeamäki

Tekniikka
2022

SISÄLLYS

1	MENOT-LUOKKA.....	3
2	MENOLASKURI-LUOKKA.....	4
3	GUI-LUOKKA.....	7
4	GUIMETHOD.....	8

1 MENOT-LUOKKA

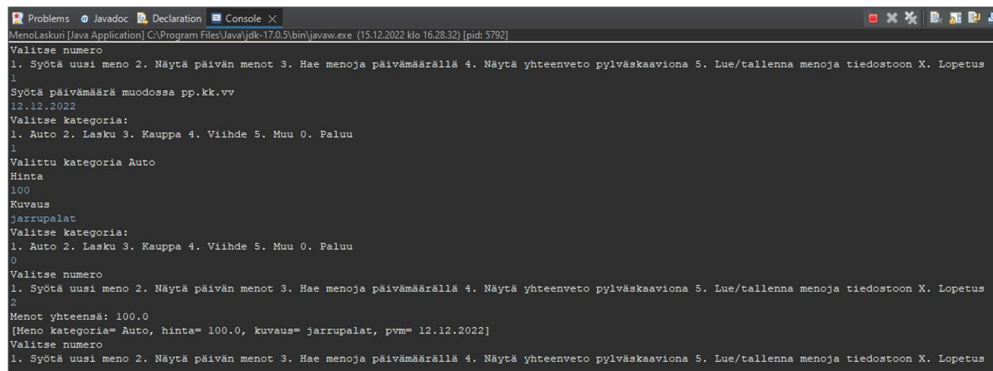
Menot-luokkaan lisättiin attribuutit, getterit, setterit, konstruktorit sekä toString-metodi. Attribuutit ovat string *kategoria*, double *hint*a, string *kuvaus* sekä string *pvm*. Tämän luokan tarkoituksena on alustaa attribuutit sekä toString-metodi, joita seuraavissa luvuissa käsiteltävä Menolaskuri- ja GUI-luokka ottavat käyttöönsä perimällä Menot-luokan.

2 MENOLASKURI-LUOKKA

Menolaskuri perii aiemmin mainitulta Menot-luokalta attribuutit, getterit & setterit sekä toString -metodin. Tämä luokka sisältää ohjelman toiminnallisuuden niin, että sitä voidaan ajaa terminaalissa. Ohjelmassa voidaan valita erilaisia vaihtoehtoja. Erilaiset vaihtoehdot ovat tehty metodien sisälle, joita kutsutaan main-metodissa switch-casen avulla.

Ensimmäisenä metodina on uusiMaksu(), jossa voi alustaa uuden menon. Käyttäjältä kysytään ensin menon päivämäärä scanner-toteutuksen avulla. Sen jälkeen tulee switch-casella toteutettu kategorian valinta, jossa valittu kategoria viedään string-muodossa metodiin uusiMeno().

Metodissa uusiMeno() käyttäjältä kysytään menon hinnan suuruus sekä kuvaus, jotka ovat myöskin toteutettu scannerin avulla. Kun käyttäjältä on kysytty kaikki tiedot, meno lisätään arraylistiin ja ohjelma palaa takaisin kategorian valintaan. Seuraavassa kuvassa on esimerkki metodin ajosta terminaalissa (Kuva 1).



```

Menolaskuri [Java Application] C:\Program Files\Java\jdk-17.0.5\bin\javaw.exe (15.12.2022 klo 16:28:32) [pid: 5792]
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus
1
Syötä päivämäärä muodossa pp.kk.vv
12.12.2022
Valitse kategoria:
1. Auto 2. Lasku 3. Kauppa 4. Viihde 5. Muu 0. Paluu
1
Valittu kategoria Auto
Hinta
100
Kuvaus
jarrupalat
Valitse kategoria:
1. Auto 2. Lasku 3. Kauppa 4. Viihde 5. Muu 0. Paluu
0
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus
2
Menot yhteensä: 100.0
[Meno kategoria= Auto, hinta= 100.0, kuvaus= jarrupalat, pvm= 12.12.2022]
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus

```

Kuva 1. Esimerkki uusiMaksu- & uusiMeno-metodien ajosta

Seuraavana metodina käsitellään metodia laskeMenot(). Kyseinen metodi kertoo nimessään tarkoituksensa hyvin. LaskeMenot-metodi siis tulostaa kyseisellä istunnolla luodut menot terminaaliin ja laskee kaikkien luotujen menojen kulut yhteen. Toteutus tälle on tehty for-loopilla, jossa se käy kaikki arrayt läpi menolista-arraylististä ja poimii jokaisesta arraysta hinnan. Tässä tapauksessa pystyimme hyödyntämään Menot-luokan luomaa getteriä getHinta, joka palauttaa hinnan. Hinnat sijoitetaan menot-nimiseen attribuuettiin, joka tulostetaan for-loopin jälkeen (Kuva 2)

```
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenvedo pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus
2
Menot yhteensä: 150.0
(Meno kategoria= Auto, hinta= 100.0, kuvaus= jarrupalat, pvm= 12.12.2022, Meno kategoria= Kauppa, hinta= 50.0, kuvaus= ruoka, pvm= 12.12.2022)
```

Kuva 2. Esimerkki laskeMenot-metodin ajosta

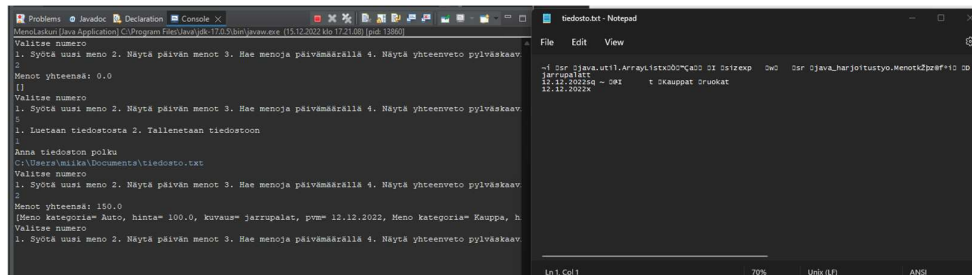
Yksi vaihtoehto on myös hakea menoja tietyllä päivämäärällä. Tämän toiminnallisuuden toteuttaa paivaMaaraMenot-metodi. Ohjelma kysyy tässä käyttäjältä päivämäärän scannerin avulla ja annettu päivämäärä viedään stringiin nimeltä etsi, joka tallentaa päivämäärän. Kuten äskeisessä metodissa, for-loopin avulla kaikki arrayt käydään läpi, mutta tällä kertaa niistä etsitään annettua päivämäärää contains-metodin avulla. For-loopin sisään on rakennettu if-lause, jotta contains on mahdollista toteuttaa sekä ilmoittaa mahdollisesta virheellisestä syötteestä (Kuva 3).

```
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenvedo pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus
3
Etsi menoja päivämäärällä
Syötä päivämäärä muodossa pp.kk.vvvv
12.12.2022
Meno kategoria= Auto, hinta= 100.0, kuvaus= jarrupalat, pvm= 12.12.2022
Meno kategoria= Kauppa, hinta= 50.0, kuvaus= ruoka, pvm= 12.12.2022
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenvedo pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus
3
Etsi menoja päivämäärällä
Syötä päivämäärä muodossa pp.kk.vvvv
11.12.2022
Ei menoja haetulla päivämäärällä
Ei menoja haetulla päivämäärällä
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenvedo pylväskaaviona 5. Lue/tallenna menoja tiedostoon X. Lopetus
```

Kuva 3. Esimerkki paivaMaaraMenot-metodin ajosta

Viimeisenä metodina Menolaskuri-luokassa on tiedosto-niminen metodi. Jos käyttäjä valitsee kyseisen vaihtoehdon, ohjelma kysyy, luetaanko tiedostosta menoja vai tallennetaanko tiedostoon menoja. Tämä toteutus on tehty scannerin ja switch-casen avulla. Mikäli käyttäjä valitsee tiedostoon lukemisen, ohjelma vie lueTiedosto-nimiseen metodiin ja toisessa tapauksessa tallennaTiedosto-nimiseen metodiin.

LueTiedosto-metodissa käyttäjältä kysytään polku tekstitiedostoon, josta ohjelma yrittää lukea mahdollista tallennettua arraylistia läpi. Lukeminen on tehty try-catch-metodilla, jossa se ensin yrittää lukea tiedostoa, mutta sen epäonnistuessa catch-osa antaa virheviestin käyttäjälle. Kun metodin ajaminen onnistuu, se ottaa arrayt ohjelmaan tiedostosta, ja ne ovat tallennettuina sinne (Kuva 4).



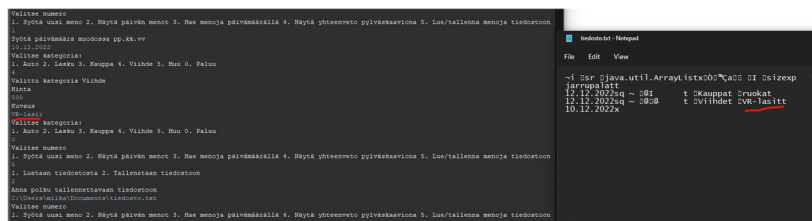
```

Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav
2
Menot yhteensä: 0.0
()
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav
5
1. Luetaan tiedostosta 2. Tallennetaan tiedostoon
1
Anna tiedoston polku
C:\Users\mikka\Documents\tiedosto.txt
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav
2
Menot yhteensä: 150.0
[Meno kategoria= Auto, hinta= 100.0, kuvaus= jarrupalat, pv= 12.12.2022, Meno kategoria= Kauppa, h
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav

```

Kuva 4. Esimerkki lueTiedosto-metodin ajosta

TallennaTiedosto-metodi puolestaan tallentaa ohjelmaan tallennetut arrayt teksti-tiedostoon. Tässäkin tapauksessa käyttäjä antaa polun tiedostoon, mihin haluaa tallentaa ohjelman sisältämät arrayt. Samaan tapaan kuin lueTiedosto-metodissa, arrayt viedään try-lauseeseen, jossa ohjelma yrittää tallentaa arraylistia tiedostoon ja muussa tapauksessa antaa virheilmoituksen käyttäjälle catch-osassa (Kuva 5).



```

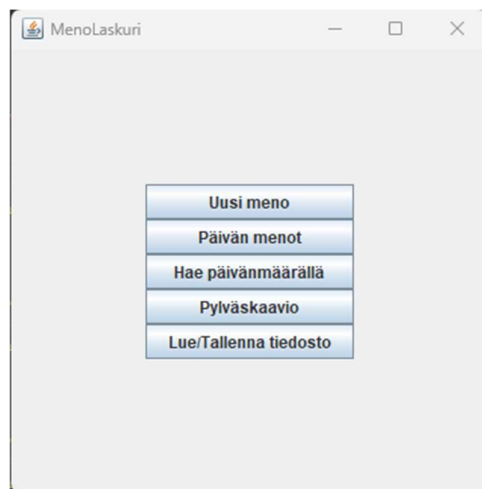
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav
2
Syötä päivämäärä muodossa pp.kk.vv
01.01.2023
Valitse kategoria:
1. Auto 2. Kauppa 3. Kauppa 4. Viikko 5. Muu 6. Paluu
4
Valitse kategoria: Viikko
Hinta
100.0
Kuvaus
jarrupalat
Valitse kategoria:
1. Auto 2. Kauppa 3. Kauppa 4. Viikko 5. Muu 6. Paluu
6
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav
5
1. Luetaan tiedostosta 2. Tallennetaan tiedostoon
1
Anna polku tallennettavaan tiedostoon
C:\Users\mikka\Documents\tiedosto.txt
Valitse numero
1. Syötä uusi meno 2. Näytä päivän menot 3. Hae menoja päivämäärällä 4. Näytä yhteenveto pyöräkaav

```

Kuva 5. Esimerkki tallennaTiedosto-metodin ajosta

3 GUI-LUOKKA

GUI-luokka sisältää vaihtoehtoisen ajoympäristön terminaalissa ajettavalle kululaskurille, joka on toteutettu graafiseen käyttöliittymään. Tässä tapauksessa toteutimme graafisen käyttöliittymän JPanel-kontilla sekä JButton-napeilla. GUI:ta ajassa ruudulle aukeaa ikkuna (JFrame) (Kuva 6), jossa on samat vaihtoehdot kuin aiemmin tarkastellussa Menolaskuri-luokassa. Käydään seuraavaksi läpi näiden vaihtoehtojen toteutusta graafisessa käyttöliittymässä.



Kuva 6. Menolaskuri graafisessa käyttöliittymässä

Luokan ainoana metodina toimii GUI-metodi, joka sisältää yllä olevan kuvan napit sekä ikkunan. Tässä metodissa napit on määritelty olioiksi JButtonin avulla ja alussa on myös alustettu ikkunalle JFramesta olio. Jokaiseen nappiin on sitten lisätty ActionListener, joka kutsuu valitun napin metodin suoritettavaksi GuiMethod-metodista.

Ikkuna alustetaan JPanel-kontilla, jolla määritellään valintaikkunan koko sekä lisätään napit paneeliin kiinni. Lopuksi koodissa ollaan määritelty ikkunan ominaisuudet ja määritelty ikkuna näkyväksi käyttäjälle, kun ohjelmaa ajetaan.

4 GUIMETHOD

GuiMethod luokka perii GUI luokan. GuiMethod luokka sisältää metodeja joita käytetään, kun ohjelmaa ajetaan graafisessa käyttöliittymässä.

Ensimmäisenä metodina on createFrameUusi (Kuva 7) metodi, jolla luodaan uusi meno ja tallennetaan se arraylistaan. Metodia kutsuessa avautuu uusi JFrame, mikä sisältää tekstikenttiä sekä nappuloita. Tekstikentillä on JLabelit, jotka kertovat mitä kenttään pitää syöttää. Napit ovat myös nimetty, joten käyttäjä tietää minkä tyyppisen menon hän luo. Painamalla “Auto” nappia ohjelma asettaa kategoriaksi “Auto” sekä, ottaa tekstikentistä tiedot jonka perusteella uusi olio luodaan, ja tallennetaan arraylistaan. Katteoria määrittyy siis napin nimellä.



Lisää uusi meno

Päivänmäärä muodossa pp.kk.vvvv

Summa. Sentit pisteellä

Kuvaus

Auto

Lasku

Kauppa

Viihde

Muu

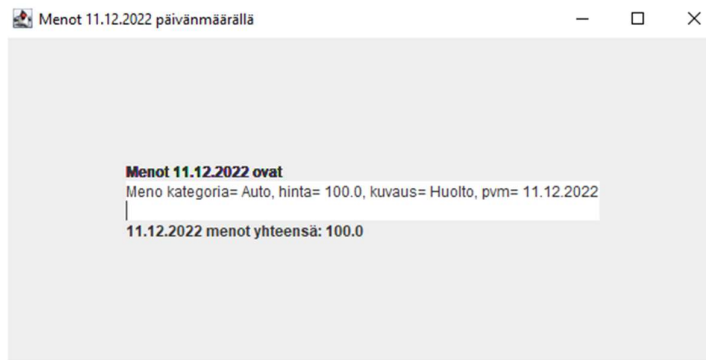
Kuva 7 Uuden menon luominen

Seuraavassa metodissa `createFrameMenot` lasketaan kyseisen istunnon menot yhteensä. Kun metodia kutsutaan uusi `JFrame` (Kuva 8) avautuu, ja ikkunassa on kaksi `JLabelia` sekä yksi tekstikenttä. Metodi käy `for`-loopilla läpi arraylistan ja kutsuu `toString` metodia. Kaikki istunnon menot tulostuvat tekstikenttään, sekä `JLabeliin` tulostuu menojen yhteenlaskettu hinta.



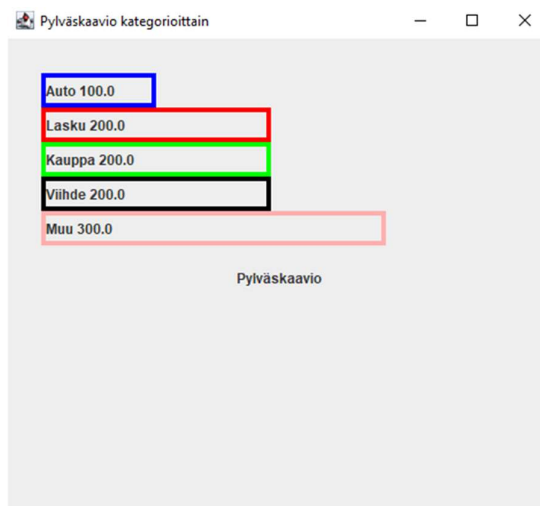
Kuva 8 Istunnon menot yhteensä

Seuraavassa metodissa `createFrameHaepvm`, etsitään kaikki tietyn päivänmäärän menot, ja lasketaan ne yhteen. Kun metodia kutsutaan uusi `JFrame` avautuu, mikä sisältää yhden `JLabelin`, tekstikentän ja napin. `JLabel` ohjeistaa syöttämään päivänmäärän, tekstikenttään syötetään haluttu päivänmäärä. Tekstikentän päivänmäärä otetaan string muuttujaksi jota käytetään `if` lausekkeessa. `For` loop käy arraylistan läpi ja `if` lauseke katsoo, sisältääkö arraylistassa oleva olio, kyseistä päivänmäärää, jos sisältää looppia ottaa kyseiseltä oliolta hinnan ja lisää sen menot muuttujaan. Ohjelma näyttää kaikki kyseisen päivän menot tekstikentässä sekä menot yhteensä `JLabelissä` sen alapuolella.



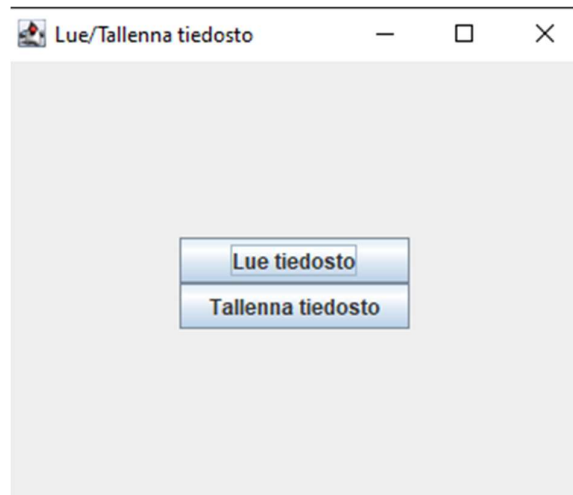
Kuva 9 Haetut menot päivänmäärällä.

createFrameKaavio metodilla luodaan pylväskaavio kaikista menoista kategorioit-
tain. Nappia painamalla uusi JFrame avautuu jossa pylväät ovat. For loopilla
käydään arraylist läpi ja otetaan jokaisen kategorian yhteenlaskettu summa. Pylväät
ovat rakennettu siten että JLabelit on määritelty x-, y-akselin sekä pituuden. Kate-
gorian yhteenlaskettu summa määrittelee JLabelin leveyden. JLabeleille on
asetettuborder joka muodostaa pylvään.



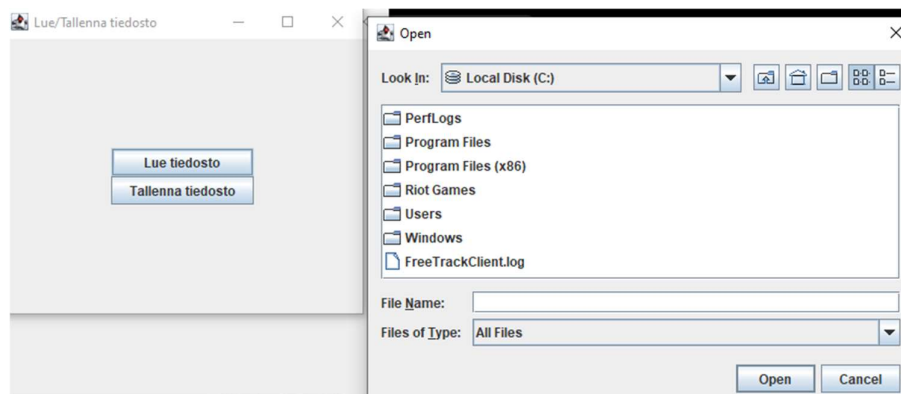
Kuva 10 Pylväskaavio

createFrameTiedosto sisältää kaksi metodia readTiedosto, sekä writeTiedosto. createFrameTiedosto metodia kutsumalla uusi JFrame aukeaa, joka sisältää kaksi nappia



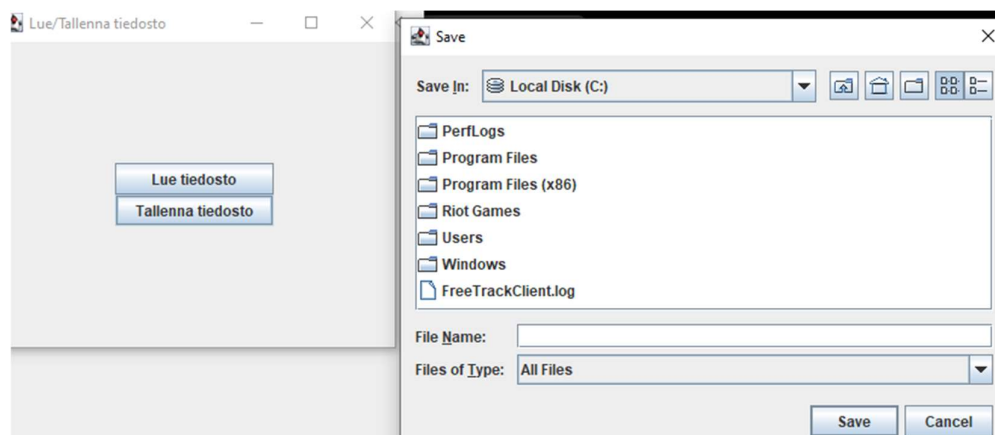
Kuva 11 Lue/Tallenna tiedosto

Lue tiedosto nappi kutsuu metodia readTiedosto, joka avaa file explorerin käyttämällä JFileChooser luokkaa. Avautuvaksi ikkunaksi on asetettu C:\\ asema. Käyttäjää voi file explorerin avulla paikantaa tiedoston, jonka haluaa lukea ohjelmaan. Polku tiedostoon tallennetaan Fileen, josta ObjectInputStream lukee tiedoston ja tallentaa sen arraylistaan.



Kuva 12 Lue tiedosto

Tallenna tiedosto kutsuu metodia saveFile, joka avaa file explorerin käyttämällä JFileChooser luokkaa. Avautuvaksi ikkunaksi on asetettu C:\\ asema. Käyttäjä voi file explorerin avulla valita paikan mihin haluaa tiedoston tallentaa. FileOutputStream, sekä ObjectOutputStream tallentavat arraylistan käyttäjän valitsemaan tiedostoon



Kuva 13 Tallenna tiedosto