

## Olio-ohjelmoinnin harjoitustyö

Ohjelman toiminnallisesta osuudesta löytyy lisätietoa README.md –tiedostosta. Teksti liitetty tähän perään.

[Linkki](https://1drv.ms/f/s!AiU53v4fEAil61Pb1l4_uGuHK57A) Demoon [https://1drv.ms/f/s!AiU53v4fEAil61Pb1l4\\_uGuHK57A](https://1drv.ms/f/s!AiU53v4fEAil61Pb1l4_uGuHK57A)

Sami Kohvakka teki:

Teki Parcelsysteemin ja stuff-kansiossa tiedostot.  
Osan interFaceControllista.  
Osan InterfaceFXML:stä

Oskar Stucki teki:

Osan infterFaceControllista  
Databuilderin  
InterfaceFXML:n pääosin  
Dokumentointi ja uml-kaavio

Molemmat olivat tyytyväisiä tehtävänjakoon ja suunnittelu tehtiin yhdessä keskustellen. Ensin tehtiin alustava uml-kaavio, johon sitten täydennettiin koodin puolella lisätietoja.

Toiminnallisuus:

- Luokat on määritetty, kuten haluttiin (1,2 ja 3 –luokat) joilla kaikilla omat ominaisuudet
- Ohjelmassa on mapview, johon voidaan lisätä Smartpostien sijainteja Comboboxin kautta. Comboboxiin on lajiteltuna kaikki sijainnit ja lisätty niiden perään tarkempia tietoja Smartpostista.
- Smartpost-luokkaan joka sisältää niille halutun datan.
- Käyttäjä voi luoda uusia esineitä, joille voi antaa erilaisia dimensioita.
- Käyttäjä voi valita erilaisista oletusesineistä, joita on luotu 6 erillisen pakettiluokan kautta.
- Käyttäjä voi lähettää paketteja eri smartpostien välillä ja jos esine voidaan lähettää käyttäjän valitsemassa luokassa, niin kartalle piirretään polku pisteiden välille.
- Toiminnallisuus, jossa piirretyt reitit voidaan poistaa kartalta.
- Storage-luokka pitää kirjaa SmartPost-olioista ja toimii myös varastona

Lisäksi:

- Ohjelma on toteutettu käyttäen erilaisia Tabeja.
- Pitää lokia tapahtumista erillisellä loki-tabilla. Loki voidaan tallentaa tiedostona tai tyhjentää kun halutaan.
- Varasto-tabissa nähdään paketit, jotka on lähetetty tai on varastossa tällä hetkellä. Varasto tallennetaan erilliseen tiedostoon, kun ohjelma suljetaan ja se voidaan lukea uudestaan, kun ohjelma avataan.
- Käyttöliittymää on muokattu ”jouluiseksi”

## README:

*# Olio-ohjelmointi\_harkkatyo*

Object based programming assignment

Made by:

Oskar Stucki

Sami Kohvakka

Classes :

Main

- Is used to initialize the program

map

-includes classes Databuilder, geopointadder and smartpost

-Databuilder:

Is used to read smartpost data from xml file and used to initialize smartpost list

-Geopointadder:

Is used to store coordinates of the smartpost locations

-Smartpost:

This class is used to represent one smartpostlocation. It has all the information that was provided by the xml file. (Postcode, city, address, availability, postoffice and coordinates).

Interfacecontroller

-Is used to control the GUI. It has all the functionality of buttons.

-Has

parcel system

-includes classes Deliveryclass, Deliveryclassselector, Firstclass, Secondclass, ThirdClass, Parcel and Storage

-Parcel

This class is essentially the package-class and it is abstract. It stores dimensions of a package, content, source and destination coordinates and postoffices, has variable for broken and fragile content and creates unique ID for each package.

-Deliveryclass

This class extend class parcel and adds deliveryclass to the package. This class is also abstract but it has variables for size limit and distance constraints.

-Deliveryclassselector

This class is used to test weather the users current chosen class can be used for the item.

-Storage

This class has the informations of all packages saved in different classes. It is responsible for saving and reading current package information from files when the program is closed or opened.

-Firstclass

Extend deliveryclass and essentially just sends unique parameters to it. This is used to create new objects in viewcontroller. Has its own size, weightlimits and maximum send distance.

Has also break variable that determines that all fragile packets will break.

Firstclass has the following parameters:

Size: 10 cm x 30 cm x 20 cm Weight: 3 kg

Max distance: 150 km

-Secondclass

Extend deliveryclass and essentially just sends unique parameters to it. This is used to create

new objects in viewcontroller. Has its own size, weightlimits and maximum send distance.

Has also break variable that determines that all fragile packets won't break.

Secondclass has the following parameters:

Size: 5 cm x 30 cm x 20 cm Weight: 2 kg

Max distance: 1500 km

#### -Thirdclass

Extend deliveryclass and essentially just sends unique parameters to it. This is used to create new objects in viewcontroller. Has its own size, weightlimits and maximum send distance.

Has also break variable that determines that all fragile packets will break. This class also has a method that

determines that during wintertime due to extreme weather conditions 10 % of non fragile packets will break.

During other months 5% of non fragile packets will break.

Thirdclass has the following parameters:

Size: 60 cm x 185 cm x 120 cm Weight: 80 kg

Max distance: 1500 km

stuff

-Includes classes Defaultitems and Item

Item:

Is a class to create items. It has dimensions as variables, content as string and weather the content is fragile.

Defaultitems:

Is used to store an arraylist of defaultitems. It also has methods to add defaultitems for which random fragile number can be generated.

view

-Has FXML and css files to create the GUI

