Työpöytänäkymäsimulaattori

Kurssi: Harjoitustyö

Ohjaaja: Ville Kuparinen

## Ohjelman tekijät:

Mio Mattila 513057, tietojenkäsittelytiede

Markus Saloranta 512790, tietojenkäsittelytiede

Sisällysluettelo:

[Tehtävän kuvaus ja analysointi](#h.1aj3en6of2mh) s.3

[Ratkaisuperiaate](#h.m4rt14kacxgb) s.3

[Ohjelman ja sen osien kuvaaminen](#h.svzvqncimnt9) s.4

[Testausjärjestely](#h.6i9gzitf3plo) s.5

[Kirjallisuusviitteet](#h.4nyt3ispxrwn) s.5

[Liitteet](#h.g7zrsn8louh4) s.5

# Tehtävän kuvaus ja analysointi

1. Tehtävä: Valitsimme harjoitustyöksemme, hetken empimisen jälkeen, ohjelman jonka tulisi ottaa printscreenejä tismalleen siitä mitä monitorissamme näkyy painamalla jotakin näppäimistön näppäintä taikka näppäinyhdistelmällä. Ohjelman tulisi tallentaa kaapatut kuvat tiettyyn käyttäjän valitsemaan paikkaan, halusimme myös että ohjelmalla olisi oma UI josta käyttäjä voisi valita tallennuspaikan ja tarkastella ottamiaan kuvia.
2. Haasteet: Koska työ tehtiin Javan koodikielellä sen toteutus tuli varsin hankalaksi javan tekemisen kohdetta vaativien rajoitusten takia, kun kerran halusimme käyttää pikanäppäimiä joita Java ei tue.

# Ratkaisuperiaate

1. Suunnitelman toteutus: Työ oli tarkoitus tehdä Java-koodikieltä käyttäen ja myöhemmin lisätä pikanäppäimille tuki ilman ikkunaa, kuitenkin vaihdoksen teknisen vaikeuden vuoksi emme pystyneet saamaan vaihdosta ikkunallisesta pikanäppäimestä taustallatoimivaksi globaaliksi pikanäppäimeksi. UI:n toteutus tapahtui Jframen avulla ja kaapattujen kuvien tallennus toimii juuri suunnitelman mukaan eli kuvia voi ottaa monta peräkkäin ja ne eivät korvaa toisiaan kansiossa. Kuvan kaappaus itsessään toteutettiin Javan Robot-luokan avulla ja toteutuksen kannalta näimme oleelliseksi poistaa pikanäppäinikkunan juuri ennen kuvakaappausotosta jolloin ikkuna ei näy kaappauksessa.
2. Rajoitusten kiertäminen: Javan tekemisen kohdetta vaativien rajoitusten vuoksi kuvan ottamiseksi syötettävä näppäin voidaan vain syöttää jo olemassa olevaan ikkunaan. Yritimme implementoida koodiimme C++ -koodia taustaprosessin toteuttamiseksi mutta se osoittautui liian haasteelliseksi.

# Ohjelman ja sen osien kuvaaminen

# Ikkuna-luokka:

Ohjelman pääikkuna perii JFrame-luokan ja sen kautta muita ohjataan muita ohjelman toimintoja. Siinä sijaitsevat näppäimet kuvien katsomiseen, uuden kuvan ottamiseen ja kuvien tallennuspaikan valitsemiseen. Konstruktorin ulkopuolella luokassa ovat getPolku() ja ResizeImage() metodit. Näistä getPolku() on käyttäjän valitseman uuden tallennuspaikan siirtämiseen toisen luokan käyttöön. Jälkimmäistä käytetään kuvan näyttämiseen nappi1:n toiminnossa. Kuvan avaamisessa ja tallennuspaikan sijainnin valitsemisessa käytetään javan JFileChooser-luokkaa.

1. Kuvittajahotkey-luokka:

Tämä luokka aktivoituu ja pääikkuna piilotetaan, kun painetaan näppäintä “Ota kuva”. Ruutuun aukeaa pieni ikkuna, jonka jälkeen käyttäjä voi siirtyä haluamaansa näkymään tietokoneellaan. Kun vaikkapa haluttu nettisivu on otettu esille, käyttäjä nostaa pienen ikkunan takaisin alapalkista ja painaa näppäimistöstä jotakin kirjainnäppäintä, jolloin kuvantallentaminen tapahtuu tämän luokan toimesta. Näppäimen vastaanottava ikkuna menee piiloon kuvanottamisen ajaksi ja palaa takaisin uuden kuvan ottamista varten.

1. TeeTiedosto-luokka:

Luokan avulla pidetään ajan tasalla Data-tekstitiedostoa, johon tallennetaan kuvien numero ja tallennussijainti, jotta sitä ei tarvitse valita jokaisella ohjelman käynnistyskerralla. Konstruktori luo tekstitiedoston ensimmäisellä käynnistyskerralla kansioon, josta ohjelmaa ajetaan. Tekstitiedoston ollessa tyhjä luokka kirjaa sinne ykkösen ja jatkossa kirjaa alkuun numeron ja tämän perään nykyisen tallennussijainnin ja erottaa nämä pilkulla niiden lukemista varten. Luokan kaksi metodia ovat numeroLaskuri() ja polkuVahti(). Ensimmäisellä pidetään kirjaa kuvanumerosta, kun kuva otetaan niin Data-tiedostoon kirjataan tämänhetkinen numero. Jälkimmäinen kirjoittaa tallennuspaikan muistiin, jos käyttäjä vaihtaa sitä.

1. Tekstitiedosto-luokka:

Tässä luokassa hoidetaan Data-tiedoston lukeminen ja otetaan talteen kuvanumero sekä tallennuskansio omiin muuttujiinsa. Metodeista getLuku() siirtää kuvanumeron toiseen luokkaan käyttöä varten, getVanhaPolku() taas käytetään kuvan tallennuspaikan valitsemiseen vertailemalla tiedostoon kirjattua ja Ikkuna-luokan getPolku()-metodin antamaa arvoa.

# Testausjärjestely

1. Käytännön testaus koostui lähinnä ohjelman ajamisesta eclipsellä ja sittemmin virheiden ilmaantuessa consolen virheraportoinnin tarkastelulla ja virheenkorjauksella.

# Kirjallisuusviitteet

1. Javan API josta käyttämämme kutsut ja luokat löytyvät.  
   <https://docs.oracle.com/javase/7/docs/api/>

# Liitteet

1. Alkuperäinen tehtävänanto: <https://moodle.utu.fi/pluginfile.php/222329/mod_resource/content/1/slides.pdf>
2. Stackoverflow:n ohjelmointisivuston sivu joka kuvaa ongelmaamme pikanäppäinten kanssa ja tapoja joilla sen olisi voinut kiertää, käytimme tätä huomattavasti avuksi kun suunnittelimme ohjelmaa vaikkakin emme onnistuneet täysin pikanäppäinten implementoinnissa.  
   <http://stackoverflow.com/questions/901224/listening-for-input-without-focus-in-java>