

## **1. autorite nimed**

Nikita Filin, Andrei Tšistjakov

## **2. projekti põhjalik kirjeldus, kus on kirjas programmi eesmärk ja selgitus programmi üldisest tööst, vajadusel lühike kasutusjuhis**

Projekti põhiidee oli luua elektrooniline päevik, mis võimaldab õpetajal lisada klasse, lisada nendesse klassidesse õpilasi, teha nende hinnetega erinevaid toiminguid ning salvestada ka kogu klassi hinnete kokkuvõtte faili. Õpilased saavad omakorda vaadata oma hindeid, teada saada keskmise hinde ja kirjutada need faili. Selles programmis saab olla ainult üks õpetaja, kes õpetab erinevates klassides mitut ainet (nagu tavaliselt põhikoolis).

## **3. iga klassi kohta eraldi selle eesmärk ja olulisemad meetodid**

**Õpilane** - klass määratleb õpilase, samuti on olemas meetodid, mis õpilasel peavad olema elektroonilise päeviku kasutamisel.

vaataHindeid - meetod võimaldab õpilasel saada konkreetse aine kõik hinded.

hinnedFaili - meetod võimaldab õpilasel salvestada kõik oma hinded kõigis ainetes eraldi faili.

õpilaseKeskmineHinne - meetod võimaldab õpilasel teada saada oma keskmise hinde teatud õppeaines.

**Õpetaja** - klassis määratletakse õpetaja, samuti on olemas meetodid, mis õpetajal peaksid olema elektroonilise päeviku kasutamisel.

moodustaGrupp - meetod võimaldab luua klasse. Tasub tähele panna, et iga aine jaoks luuakse eraldi klass samade õpilastega (kuid õpilase objektid on erinevad).

kuvaNimekiri - meetod võimaldab kuvada iga klassi õpilaste nimekirja.

lisaHinne - meetod võimaldab lisada õpilasele hinde. Õpetaja valib õpilase ja ka aine, kuhu hinne lisatakse.

aineKeskmineHinne - meetod võimaldab arvutada kogu klassi keskmise hinde konkreetse õppeaine kohta.

tahvliJuurde - meetod genereerib juhuslikult klassist õpilase arvu, kes peaks tahvli juurde minema.

hinnedFaili - meetod võimaldab õpetajal salvestada kõik õpilaste hinded kõigis ainetes eraldi faili.

**Klass** - see klass võimaldab määrata peaklassi, kuhu kuuluvad õpilased. Klassil on ainult get meetodid.

**Main** - põhiklass, kus rollide valik toimub (õpetaja või õpilane).

õpetajaRoll - meetod kuvab õpetaja jaoks menüü, mis hiljem töötab teiste klasside ja meetodite abil.

õpilaseRoll - meetod kuvab õpilasele menüü, kus küsitakse tema perekonnanime ja klassi, mis edaspidi töötab teiste klasside ja meetodite abil.

#### **4. projekti tegemise protsessi kirjeldus**

Algselt tulime välja ainult projekti ideega, kuid meil pole õrna aimugi, kuidas seda ellu viia. Peale väikest mõtlemist joonistasime paberile rakenduse põhiloogika ja võtsime välja, milliseid klasse ja meetodeid see sisaldama peaks. Seejärel jagasime rollid ära ja igaüks hakkas omas tempos töötama. Aeg-ajalt saime kokku ja arutasime, kes mida tegi. Rollide osas tegi Nikita selle osa, mis vastutab õpetaja eest, ja Andrei õpilase eest. Viimane samm oli meie töö üheks tervikuks ühendamise, kuid ka seda polnud raske teha, kuna põhimeetodis tegeles õpilase klassiga Andrei ja õpetaja klassiga Nikita.

#### **5. iga rühmaliikme panus**

Andrei - Õpilase klass, õpilaseRoll meetod, Klass klass (13 tundi).

Nikita - Õpetaja klass, õpetajaRoll meetod, main meetod (16 tundi).

#### **6. tegemise mured**

Mure oli selles, et oli suht raske teha nii, et kõik koodi jupid töotaksid õigesti ja kooskõlas. Samuti oli raske mõelda välja kuidas saab samu õpilasi erinevatesse ainetesse jagada.

#### **7. hinnang oma töö lõpptulemusele**

Üldiselt saime oma eesmärgiga hakkama, programm töötab hästi, vigu ei anna, kuid tulevikus tahame ka seda paremaks teha. Esiteks võiks lisada võimalust, et oleks rohkem, kui üks õpetaja. Teiseks oleks tore, kui kõik info salvestaks pärast programmi lõppu, seega võiks lisada salvestus võimalust kas faili kujul või isegi kasutades andmebaasi. Ja lõpuks seda koodi võiks pisut ilusamaks ja lühemaks teha.

## 8. selgitus ja/või näited, kuidas programmi osi eraldi ja programmi tervikuna testisite ehk kuidas veendusite, et programm töötab korrektselt.

Esiteks katsetasime igat meetodid eraldi, kui programm oli valmis, siis katsetasime läbi kõike erinevaid võimalusi, mida kasutaja sisestada saab, kui mingi asi ei töötanud, siis lahendasime probleemi ja tegime kõik katsed uuesti.

### Väike osa programmi tööst:

```
Hetkel ei ole ühtegi õpetajat, seega sisenen õpetajana
Sisestage oma eesnimi
Egil
Sisestage oma perenimi
Karu
Esmalt peate lisama õpilasi
Sisestage klass:
1a
Mitu ainet te soovite lisada selle klassi jaoks?
2
Sisestage aine nimi:
matemaatika
Sisestage aine nimi:
keemia
Sisestage õpilase eesnimi
Lõpetamiseks sisestage 'q'
Nikita
Sisestage õpilase perenimi
Pilin
Sisestage õpilase eesnimi
Lõpetamiseks sisestage 'q'
Andrei
Sisestage õpilase perenimi
Tšistjakov
Sisestage õpilase eesnimi
Lõpetamiseks sisestage 'q'
q
Valige tegevus:
1. Lisa uus klass
2. Lisa hinne
3. Kuva klassi nimekiri
4. Vali suvalist õpilast
5. Vaata aine keskmist hinnet
6. Hinded faili
Väljumiseks sisestage 'q'
2
Valige klass kuhu soovite hinnet panna
1a
Valige aine kuhu soovite hinnet panna
matemaatika
Valige õpilast kellele soovite hinnet panna (perenimi)
Lõpetamiseks sisestage 'q'
Pilin
```

```
Fillin
Sisestage hinne
5
Valige õpilast kellele soovite hinnet panna (perenimi)
Lõpetamiseks sisestage 'q'
Tõstjakov
Sisestage hinne
4
Valige õpilast kellele soovite hinnet panna (perenimi)
Lõpetamiseks sisestage 'q'
0
Valige tegevus:
1. Lisa uus klass
2. Lisa hinne
3. Kuva klassi nimekiri
4. Vali suvaline õpilane
5. Vaata aine keskmist hinnet
6. Hinded faili
Väljumiseks sisestage 'q'
0
Valige sisenemise viis:
1. Õpetaja
2. Õpilane
2
Sisestage oma perenimi
Fillin
Sisestage oma klass
1a
Valige tegevus:
1. Vaata hindeid
2. Hinded faili
3. Vaata keskmist hinnet
Väljumiseks sisestage 'q'
1
Sisesta mis tunni hindeid soovite vaadata
matemaatika
5
Valige tegevus:
1. Vaata hindeid
2. Hinded faili
3. Vaata keskmist hinnet
Väljumiseks sisestage 'q'
0
```