

Opinnäytetyö (AMK)

Tieto- ja viestintäteknikka

2024

Jooa Jaakkola

GitHub Classroomin käyttö ohjelmoinnin opettamisessa



Opinnäytetyö (AMK) | Tiivistelmä

Turun ammattikorkeakoulu

Tieto- ja viestintäteknikka

2024 | 31 sivua

Jooa Jaakkola

GitHub Classroomin käyttö ohjelmoinnin opettamisessa

Ohjelmoinnin opettamisen haaste on käytännöllisten tehtävien luominen, kun opiskelijat eivät vielä tunne ohjelmointisyntaksia. Tällöin ei voi antaa pelkästään projektityötä tehtävänannoksi, mutta yksittäiset syntaksitehtävät eivät myöskään auta oppimaan kovin hyvin. Ohjelmointia oppii parhaiten tekemällä, ja GitHub Classroom mahdollistaa tavan, jolla voidaan luoda tehtäviä arvioitavaksi alan standardityökaluilla. Tämä opinnäytetyö tutki tätä työkalua ja vertaili muita vaihtoehtoisia tapoja opettaa ohjelmointia.

Aluksi tutkittiin GitHub Classroom -työkalua ja vaihtoehtoja sille. Tämän ohella käytiin myös tiiviisti läpi näiden historiaa. Projektityönä toimi esimerkkinä tehty luokkahuone, jolla tarkasteltiin sen toimivuutta ohjelmoinnin opetuksessa. Ensimmäiseksi käytiin läpi luokkahuoneen sekä tehtävien luominen, ja sen jälkeen opiskelijana tehtävien tekeminen. Opiskelijan käyttökokemusta testattiin kahdella eri käyttäjällä. Lopuksi toistettiin sama prosessi toisella työkalulla, jonka toimintaperiaate oli samankaltainen.

Ohjelmoinnin opettamiselle on olemassa useita eritasoisia oppimisen hallintajärjestelmiä. GitHub Classroom toimi tämän opinnäytetyön perusteella parhaiten, sillä vaihtoehtoisissa ei onnistuttu saamaan tehtäviä yhtä helpolla ja tehokkaasti luotua.

Asiasanat:

GitHub, ohjelmointi, versionhallinta, koulutus, pedagogia

Bachelor's Thesis | Abstract

Turku University of Applied Sciences

Information and Communication Technology

2024 | 31 pages

Jooa Jaakkola

Using GitHub Classroom in teaching programming

A challenge of teaching programming is to create practical tasks when programming syntax is not yet grasped by the students. Project work can therefore not be given to students, and individual syntax tasks are generally not very efficient for teaching programming. The best method for learning programming is by doing, and GitHub Classroom provides a method for creating tasks to be assessed using industry standard tools. This thesis aims to explore the potential of this tool by comparing it to other alternative tools for teaching programming.

To achieve the objective of this thesis, a closer look was taken into the GitHub Classroom tool and alternatives to it, including a brief look into their backgrounds for a bit of context. The main focus was on creating a mock classroom setup that would resemble a real programming class. This involved setting up a digital classroom and designing assignments to go with it. After that, the user experience was tested from a student's perspective using various accounts to get a well-rounded view. This process was then repeated with a different educational tool, which functions in a similar way to GitHub Classroom to compare how effective each one is in teaching programming.

There are several levels of learning management systems for teaching programming, but GitHub Classroom worked best according to this thesis. All the alternatives failed to get the tasks and environments created as easily and efficiently.

Keywords:

GitHub, programming, version control, education, pedagogy

Sisältö

Lyhenteet ja sanasto	7
1 Johdanto	8
2 Työkalut ja teknologiat	9
2.1 Git- ja GitHub-hallintajärjestelmä	9
2.2 GitHub Classroom	9
2.3 A+ -oppimisen hallintajärjestelmä	10
2.4 Python	11
3 GitHub Classroomin luominen	12
4 GitHub Classroom opiskelijana	14
5 A+ -ympäristön luominen	17
6 A+ -ympäristö opiskelijana	18
7 GitHub Classroom verrattuna A+ -järjestelmään	19
8 Pohdinta	20
Lähteet	21

Liitteet

Liite 1. GitHub Classroomin käyttö opettajana

Liite 2. Esimerkki GitHub Classroom luokkahuoneesta

Kuvat

Kuva 1. GitHub Classroom opettajan näkymä tehtävänannosta.....	10
Kuva 2. Puukaavio A+ LMS:n toiminnasta ja API-yhteyksistä (Riekkinen, et al., 2024).....	11
Kuva 3. GitHub Classroom tehtävänannon ja testien luominen.....	12
Kuva 4. Automaattisesti luotu ohjelmistokokoonpano tehtävän tarkistukseen. .	14
Kuva 5. GitHub Classroom lähdekoodista haarukoitu opiskelijan versio ja onnistunut testi.....	15
Kuva 6. GitHub näkymä ja ainoa paikka, jossa opiskelija näkee tehtävänannot kootusti.....	16
Kuva 7. A+ -LMS esimerkkisivusto ja sen kääntäminen sekä hakeminen Docker-konteista.....	17
Kuva 8. A+ -ympäristö ja tehtävänannon näkymä.....	18
Kuva 9. Organisaatiotilin luominen ja sen GitHub-osoite.....	23
Kuva 10. Luokkahuoneen luominen organisaatiotilille.....	24
Kuva 11. Luokkahuoneen alustava nimeäminen.....	24
Kuva 12. Luokkahuoneeseen osallistuvien mahdollisten muiden opettajien kutsuminen.....	25
Kuva 13. Luokkahuoneeseen oppilaiden lisääminen olemassa olevasta LMS:tä joko suoraan tai tekstinä.....	26
Kuva 14. Luokkahuoneen tehtävien luominen kolmessa vaiheessa: perustiedot.....	27
Kuva 15. Luokkahuoneen tehtävien luominen kolmessa vaiheessa: aloituskoodi ja käyttöoikeudet.....	27
Kuva 16. Luokkahuoneen tehtävien luominen kolmessa vaiheessa: automaattitestit ja niiden suojaus muutoksilta.....	28
Kuva 17. Luokkahuoneen tehtäväkohtainen näkymä ja oppilaiden tulokset.....	29
Kuva 18. Lähdekoodiarkisto ensimmäisestä tehtävästä: https://github.com/jooa-thesis/git-github-fundamentals-Git-GitHub-Fundamentals/tree/master	30

Kuva 19. Lähdekoodiarkisto toisesta tehtävästä: <https://github.com/jooa-thesis/Conditionals-Loops>.....31

Lyhenteet ja sanasto

IDE	Ohjelmointiympäristö (engl. integrated development environment). Se on ohjelmisto, joka auttaa ohjelmoijaa suunnittelemaan ja toteuttamaan lähdekoodia. Se koostuu vähintään tekstieditorista ja keinosta yhdistää kääntäjä tai tulkki siihen (Wikimedia Foundation, Inc., 2022)
LMS	Oppimisen hallintajärjestelmä (engl. learning management system) on ohjelmisto, joka auttaa opettajia oppimateriaalien dokumentoinnissa ja kurssien hallinnassa.
RST	Merkintäkieli (engl. reStructuredText), joka käyttää symboleja ja avainsanoja dokumentin rakentamiseen, sekä erittäin intuitiivinen työkalu HTML-sivujen kirjoittamiselle (Aplus team, 2021).
Sphinx	Avoimen lähdekoodin työkalu, jolla voidaan pääasiassa luoda dokumentaatioita ohjelmistokehitysprojekteille. A+-työryhmä jalosti tätä omaan järjestelmäänsä ja käyttää pääasiassa RST-tekstin lukemiseen sekä HTML- ja YAML-tiedostojen luomiseen tehtäviä varten. (Aplus team, 2021)
SCM	Ohjelmistokokoonpanon hallintajärjestelmä (engl. software configuration management) on virallinen nimitys Gitin kaltaisille lähdekoodiarkisto-ohjelmistoille.
YAML	Ihmisläheinen merkintäkieli, joka toimii datan jäsentämiseen minkä tahansa ohjelmointikielen hyödynnettäväksi (yaml.org, 2021). Käytännössä yleisimmin käytetty ohjelmistokokoonpanojen asetuksille.

1 Johdanto

Ohjelmoinnin opettaminen ja tehtävien palauttaminen voi olla hankalaa, mikäli haluaisi myös antaa mahdollisimman autenttisesti työelämään suhteutettavan työnkulun. Ohjelmoinnin opettamisen vaikeus näkyy myös opiskelijalla siten, että tehtävät, materiaalit ja aikataulut ovat eri alustoilla. Alan käytetyimmällä versionhallintapalvelu GitHubilla on tähän yksi mahdollinen ratkaisu: GitHub Classroom.

Tämän opinnäytetyön tavoitteena on tutkia tarkemmin GitHub Classroom -työkalua ja sen toimivuutta ohjelmoinnin opetuksessa sekä oppimisessa. Yleensä korkeakouluissa ja toisen asteen koulutuksissa ohjelmoinnin opetukseen käytetään esimerkiksi Moodlea, joka on hyvin yleinen LTI-standardin (1EdTech, 2024) mukainen oppimisen hallintajärjestelmä (LMS). Turun ammattikorkeakoulussa käytetään It'sLearning -oppimisen hallintajärjestelmää. Kumpikaan näistä ei ole suoraan yhteensopiva GitHubin kanssa ja kaikilla opettajilla on eri tyyli opettaa, joten GitHub Classroom voisi helpottaa tehtävien jakamista, arviointia ja ryhmätöitä. Aalto-yliopiston kehittämä, ylläpitämä ja myös Tampereen yliopistossa käytetty A+ -oppimisen hallintajärjestelmä (Riekkinen, et al., 2024) perustuu hieman samaan tekniikkaan kuin GitHub Classroom, joten tutkin työssä myös tätä ja näiden eroja.

Opinnäytetyössä selvitetään aluksi ohjelmointityön yleisiä käsitteitä ja työkaluja laajemmin. Sen jälkeen tutkitaan GitHub Classroom -työkalua tarkemmin luomalla luokkahuone ja ohjelmointitehtäviä, sekä testataan sitä saadakseen mahdollisimman monipuolista käyttökokemustietoa. Samalla verrataan saman kaltaista prosessia A+ -oppimisen hallintajärjestelmällä. Lopuksi esitellään pohdintoja luokkahuoneen toimivuudesta ja hypoteesin paikkansapitävyydestä: voisiko GitHub Classroom olla helpoin, tehokkain ja autenttisin tapa tehdä ohjelmointitehtäviä samanlaisessa ympäristössä kuin työelämän projekteissakin.

2 Työkalut ja teknologiat

2.1 Git- ja GitHub-hallintajärjestelmä

Versionhallinta on olennainen osa modernia ohjelmistokehitystä ja on kehittynyt paljon vuodesta 2005, jolloin Linus Torvalds kehitti oman konfiguraation hallintajärjestelmänsä (SCM) Gitin oltuaan erittäin tyytymätön silloisiin vaihtoehtoihin (The Linux Foundation, 2015). Nykyään avoimen lähdekoodin Git on käytännössä ainoa konfiguraation hallintajärjestelmä ja aktiivisesti yhteisön ylläpitämä.

Git, ja muut konfiguraation hallintajärjestelmät, perustuu siihen, että jokaisella kehittäjällä on oma paikallinen versio lähdekoodista ja lähdekoodia kehitetään eri vaiheissa. Konfiguraation hallintajärjestelmä mittaa ja ohjaa lähdekoodin muutoksia (Wikimedia Foundation, Inc., 2023). Gitin ero muihin konfiguraation hallintajärjestelmiin on sen avoin lähdekoodi ja myös haarautumismalli, joka mahdollistaa hyvän ja tehokkaan tavan tehdä ohjelmistokehitystä. Git mahdollistaa kitkattoman kontekstinvaihdon, roolipohjaisen koodipohjan, ominaisuuksiin perustuvan työnkulun ja kertakäyttöisen kokeilun (Chacon & Long, 2024).

GitHub puolestaan on yksi monista kaupallisista verkkopalveluista, jotka isännöivät Git-organisoituja koodipohjia sekä tarjoaa useita lisäominaisuuksia. GitHub avattiin vuonna 2008, kun Tom Preston-Werner alkoi kehittää harrasteprojektiaan täysipäiväisesti (Preston-Werner, 2008). Nykyään GitHub on suurin lähdekoodin ylläpitopalvelu yli 100 miljoonalla kehittäjällä sekä yli 420 miljoonalla lähdekoodiarkistolla (engl. repository) (Dohmke, 2023).

GitHub perustuu siihen, että rekisteröitynyt käyttäjä luo lähdekoodiarkiston eli käytännössä sivuston lähdekoodillensa ja lataa siitä Gitillä kopion omaan kehitysympäristöönsä. Kloonaus (engl. clone) tapahtuu Gitin saman nimisellä komennolla pääasiassa joko HTTP- tai SSH-protokollalla. Tällöin lähdekoodiarkistossa (eli pilvessä) sekä paikallisesti olevaa lähdekoodia voi helposti jakaa, varmuuskopioida ja jopa isännöidä esimerkiksi verkkosovelluksia.

2.2 GitHub Classroom

GitHub Classroom on GitHubin kehittämä lisäominaisuus, jonka tavoitteena on helpottaa tietotekniikan opettajien työtä ja tarjota opiskelijoille mahdollisimman autenttinen ympäristö opiskella ohjelmistokehitystä. Classroom antaa opettajien ja oppilaitosten ylläpitäjien luoda virtuaalisia luokkahuoneita, joissa jakaa, pisteyttää ja ajoittaa tehtäviä automaattisesti (GitHub, Inc., 2024).

GitHub Classroom julkaistiin vuonna 2015 ja on ollut siitä lähtien aktiivisessa kehityksessä. Merkittävimpiin päivityksiin kuuluu suosituimpien oppimisen hallintajärjestelmien sekä ohjelmointiympäristöjen (IDE) integroiminen, joka ei kuitenkaan tällä hetkellä ole enää ylläpidetty.

Oppimisen hallintajärjestelmät, kuten esimerkiksi Moodle ja Google Classroom, GitHub Classroom integroivat yhdistämällä oppilastiedot, jolloin kurseille ilmoittautuneet lisätään automaattisesti luokkahuoneeseen.

Ohjelmointiympäristö Visual Studio Code integroidaan laajennuksen avulla, joka tarjoaa käyttöliittymän luokkahuoneeseen suoraan ohjelmointiympäristöstä.

GitHub Classroom perustuu siihen, että opettajalla on organisaatiotili tehtävien lähdekoodiarkistoille ja oppilaat saavat niistä luokkahuoneen kautta omat arkistonsa. Tähän Classroom käyttää Gitin ominaisuutta haarukoida (engl. fork) eli käytännössä kopioi lähdekoodiarkistot säilyttäen referenssit alkuperäiseen ja mahdollisuuden päivittää muutokset. Opettajalla on pääsy kootusti jokaisen opiskelijan lähdekoodiarkistoon tehtävänannon sivustolta (Kuva 1).

The screenshot shows the GitHub Classroom interface for an assignment titled "Git & GitHub Fundamentals". At the top, there is a navigation bar with the Classroom logo and GitHub Education link. A yellow banner below the header contains a warning: "Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments." The breadcrumb trail indicates the current location: "Classrooms / jooa-thesis-classroom / Git & GitHub Fundamentals".

The main content area displays the assignment details:

- Assignment Details:** Individual assignment, Active. URL: <https://classroom.github.com/a/VA403KKT>. Buttons for Edit and Download are visible.
- Statistics:**
 - Students total: 4 (4 Rostered, 0 Added students)
 - Accepted assignments: 2 (2 Students)
 - Assignment submissions: 2 (2 Submitted, 0 Not submitted)
 - Passed students: 1 (1/2 Passed)
- Filters:** Search for an assignment, Filter by unlinked accounts, Filter by accepted, Filter by passing, Sort.
- Classroom roster:**
 - Student 1: jooajaakkola@edu.turkuamk.fi (Submitted). Latest commit: last month. 1 commit. Repository icon.
 - Student 2: jooajaa@gmail.com (Submitted). Latest commit: 2 months ago. 1/1 commit. Repository icon.

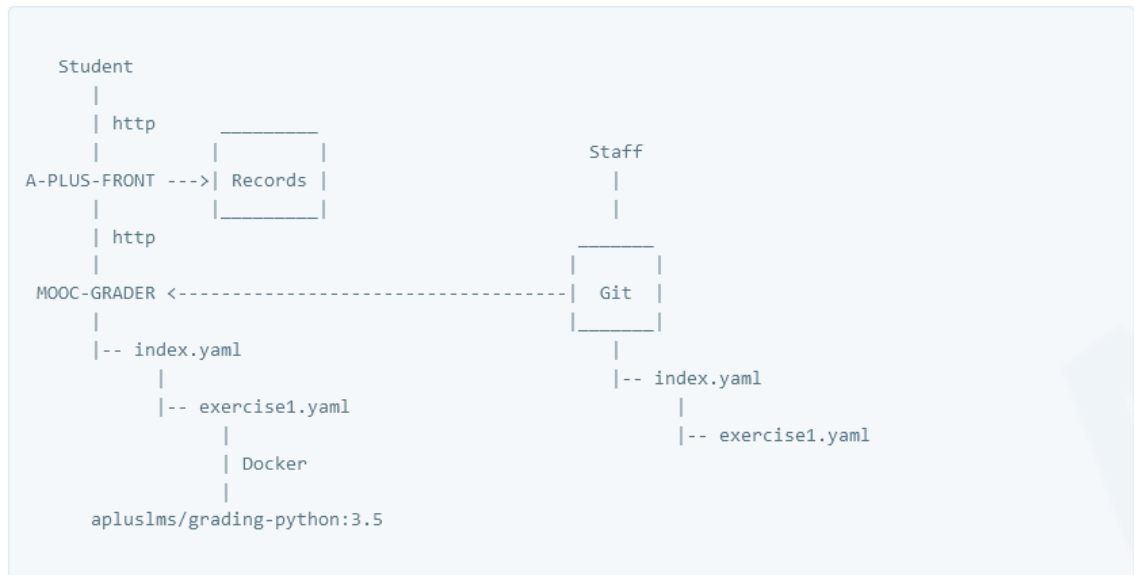
Kuva 1. GitHub Classroom opettajan näkymä tehtävänannosta.

2.3 A+ -oppimisen hallintajärjestelmä

Aalto-yliopiston kehittämä ja ylläpitämä oppimisen hallintajärjestelmä A+ kehitettiin alun perin yliopiston tutkimusprojektissa vuonna 2013 julkaistun toisen tutkimuksen (Karavirta, et al., 2013) pohjalta ja nykyään pääasiassa EDIT-työryhmän ylläpitämä. A+ on pääasiassa ollut käytössä tietotekniikan

opetusmateriaaleissa, mutta teoriassa mahdollista soveltaa myös muuhun opetukseen (Riekkinen , et al., 2020).

A+ perustuu Gitiin ja tehtävien automaattiseen tarkastukseen ohjelmallisesti. Se täydentää tarvittaessa yliopistojen käyttämiä Moodle-ympäristöjä, mutta on käytännössä opettajan paikallisesti isännöimä verkkosovellus (Kuva 2).



Kuva 2. Puukaavio A+ -järjestelmän toiminnasta ja API-yhteyksistä (Riekkinen, et al., 2024).

2.4 Python

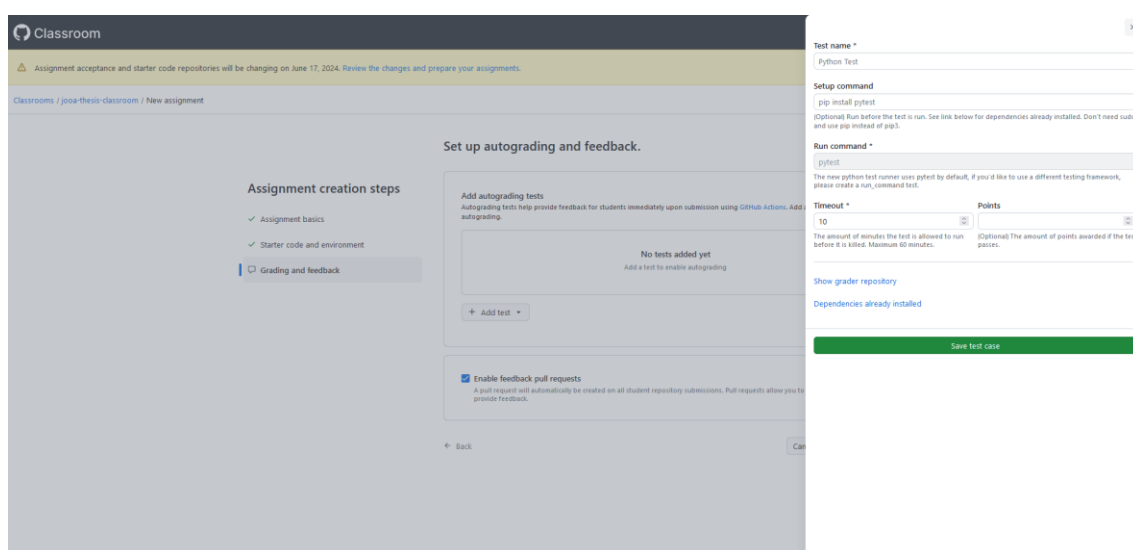
Python on erittäin yleinen ja helposti lähestyttävä korkean tason tulkittava ohjelmointikieli. Tästä syystä Python on myös ohjelmoinnin opetuksessa suuressa suosiossa, vaikka sen käyttökohteet pääasiassa ovatkin muun muassa data-analyysissä, tekoälyssä ja tilastotieteessä. Pythonin filosofia korostaa koodin luettavuutta ja yksi keino, jolla tätä edistetään, on syntaksissa sisennysten tiukka edellytys. Python on myös olio-ohjelmointiin kykenevä ohjelmointikieli, joten tämä monipuolisuus lienee myös merkittävä syy laajaan käyttöön opetuksessa.

3 GitHub Classroomin luominen

GitHub Classroomin luominen itsessään on helppoa, kun seuraa vaiheita modernissa ja selkeässä käyttöliittymässä. Alkuun pitää luoda GitHubiin organisaatiotili, joka hallinnoi luokkahuoneita sekä tehtävien lähdekoodiarkistoja. Luotua uuden luokkahuoneen, voi joko luoda ensimmäisen tehtäväpaketin esimerkistä "Git & GitHub Fundamentals" tai valita lähdekoodiarkiston organisaatiotililtä.

Ainoastaan Google Classroom -oppimisen hallinta järjestelmällä saa suoraan tuotua opiskelijoita GitHub Classroomiin, mutta tämän ominaisuuden laajeneminen on ilmoitusten perusteella kehityksessä ja laajentumassa muihinkin LTI-standardin mukaisiin järjestelmiin. Muuten opiskelijat pitää kutsua luokkahuoneeseen antamalla käyttöliittymään suoraan joko sähköposti tai GitHub-tunnus, tai lähettämällä linkit tehtäviin. GitHub-tunnukset Classroom tunnistaa vasta vastaanottaessa ensimmäisen tehtävän linkin. Luokkahuonetta luodessa olisi hyvä olla siis aina ensimmäisenä tehtävänä yksinkertainen ympäristön testaus, jotta opettajana saa selkeästi kaikki kurssille osallistuvat organisoitua.

Tehtävien kokoaminen olemassa olevista lähdekoodiarkistoista, joko organisaatiotililtä tai muualta GitHubista, on myös yksinkertaista, kun seuraa käyttöliittymän ohjeita. Opiskelijat pääsevät käsiksi omiin kopioihinsa niistä ja voivat jatkaa tehtävän kanssa kuin kyseessä olisi mikä tahansa lähdekoodiarkisto. Tehtävien tarkistaminen onnistuu puolestaan GitHubin Toimet-toiminnolla (engl. actions) ja ohjelmointikielikohtaisilla testityökaluilla (Kuva 3).



Kuva 3. GitHub Classroom tehtävänannon ja testien luominen.

Tehtävät sekä testit on ohjelmitava alusta alkaen ja kiinnitettävä huomiota ainutlaatuisuuteen, mutta niiden sijaitessa GitHubissa on ne tarpeen tullen lähes äärettömästi uudelleenkäytettävissä vain pienillä muutoksilla. Liitteissä on kuvattuna tarkemmin opettajan tehtävistä ennen GitHub Classroom kurssin aloittamista ja sen aikana (Liite 1). Esimerkkinä tehdyt kaksi yksinkertaista ohjelmointitehtävää näyttävät, miten GitHub Classroomin toimintamalli muistuttaa tavallista ohjelmistokehitysprojektia (Liite 2).

4 GitHub Classroom opiskelijana

Opiskelijaksi GitHub Classroomiin voi liittyä tehtävän linkistä, joka luo automaattisesti opiskelijalle oman haarukan siitä lähdekoodiarkistosta. Tällöin opiskelijan GitHub-tunnus yhdistyy myös opettajan luokkahuoneeseen ja opiskelijoiden edistyminen tallennetaan aina työnnon jälkeen (engl. push).

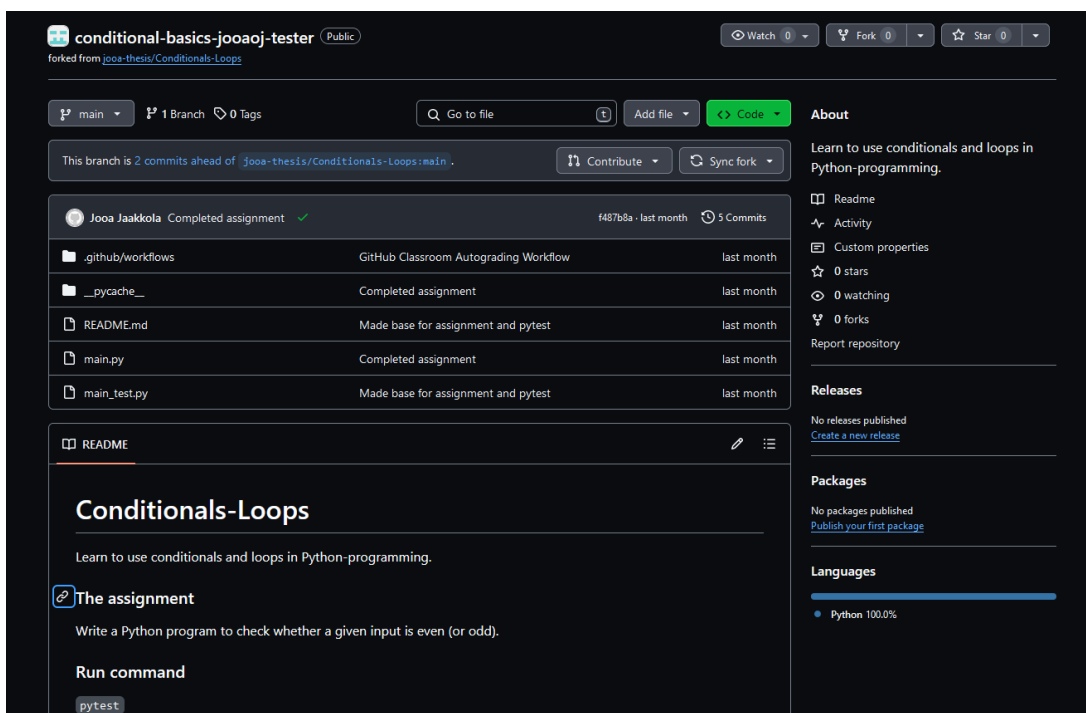
Opiskelija voi käyttää vapaasti Git-haaroja (engl. branch), kuten on hyvän ohjelmointikehityksen mukaista. GitHub Classroom ei kuitenkaan oletusarvoisesti tue tätä, sillä tehtävänannot tarkistetaan jokaisen tallennuksen jälkeen riippumatta haarasta. Tämän opettaja saa kuitenkin halutessaan korjattua muokkaamalla automaattisesti luotua GitHub-toimet kokoonpanoa, joka on yksi YAML-tiedosto (Kuva 4). Tämä vaatii enemmän teknistä tuntemusta ja hieman YAML-kielellä ohjelmointia.

```
1   name: Autograding Tests
2   'on':
3     - push
4     - workflow_dispatch
5     - repository_dispatch
6   permissions:
7     checks: write
8     actions: read
9     contents: read
10  jobs:
11    run-autograding-tests:
12      runs-on: ubuntu-latest
13      if: github.actor != 'github-classroom[bot]'
14      steps:
15        - name: Checkout code
16          uses: actions/checkout@v4
17        - name: evenOrOdd test
18          id: evenorodd-test
19          uses: education/autograding-command-grader@v1
20          with:
21            test-name: evenOrOdd test
22            setup-command: sudo -H pip3 install pytest
23            command: pytest main_test.py
24            timeout: 10
25            max-score: 1
26        - name: Autograding Reporter
27          uses: education/autograding-grading-reporter@v1
28          env:
29            EVENORODD-TEST_RESULTS: "${{steps.evenorodd-test.outputs.result}}"
30          with:
31            runners: evenorodd-test
```

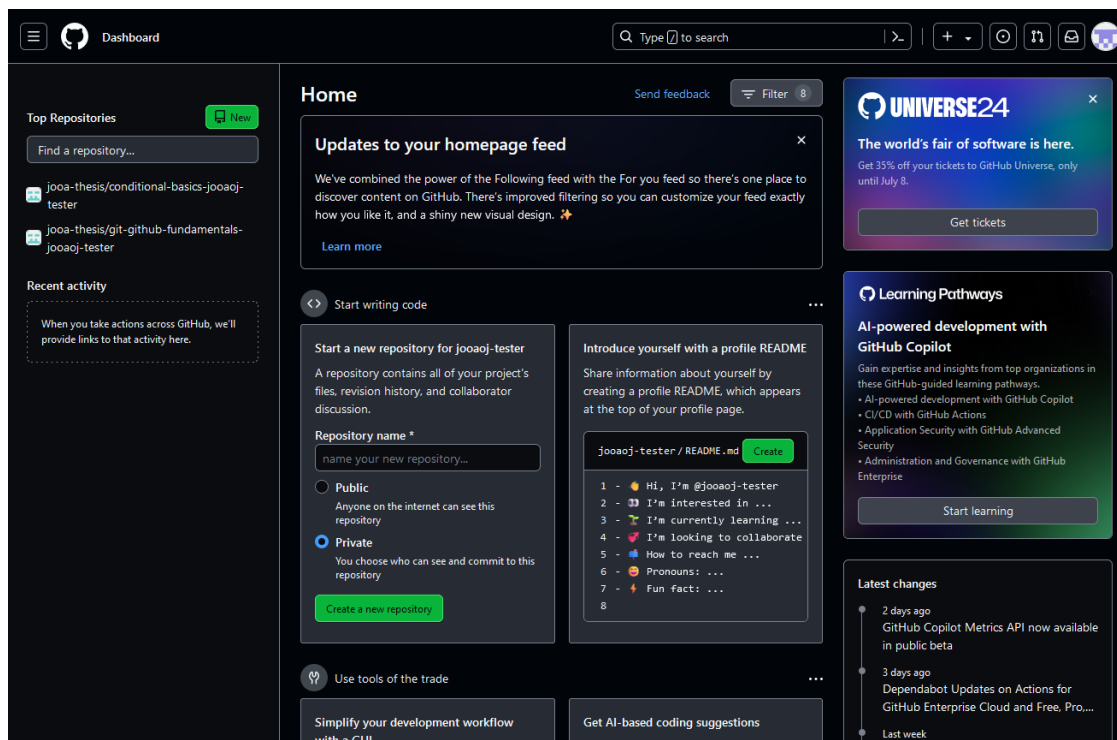
Kuva 4. Automaattisesti luotu ohjelmistokokoonpano tehtävän tarkistukseen.

Opiskelijana GitHub Classroomin käyttökokemus on erittäin intuitiivinen. Se noudattaa linkin hyväksymisen jälkeen täydellisesti GitHubin ohjelmistokehitysprosessia (Kuva 4). Luokkahuonenäkymään opiskelijoilla ei

kuitenkaan ole pääsyä, joten opiskelijat eivät näe mistään kootusti kokonaispisteitä eikä tehtävänantoja. Puute korostuu vielä, mikäli opiskelijalla on paljon muitakin projekteja GitHub-tiillään ja kurssin tehtävänannot katoavat helposti lähdekoodiarkistolistasta (Kuva 5). Tämä käyttökokemuspuute opiskelijakäytössä on syytä ottaa huomioon opettajana. Toinen merkittävä puute on, että uusia tehtäviä ei saa haarukoitua itselleen kuin ainoastaan opettajan lähettämän linkin kautta. GitHub Classroom ei siis yksin riitä opetusympäristöksi vaan olisi suotavaa olla jokin palvelu, kuten esimerkiksi oppimisen hallintajärjestelmä, jossa jaettaisiin tehtävänantojen linkit kootusti.



Kuva 5. GitHub Classroom lähdekoodista haarukoitu opiskelijan versio ja onnistunut testi.

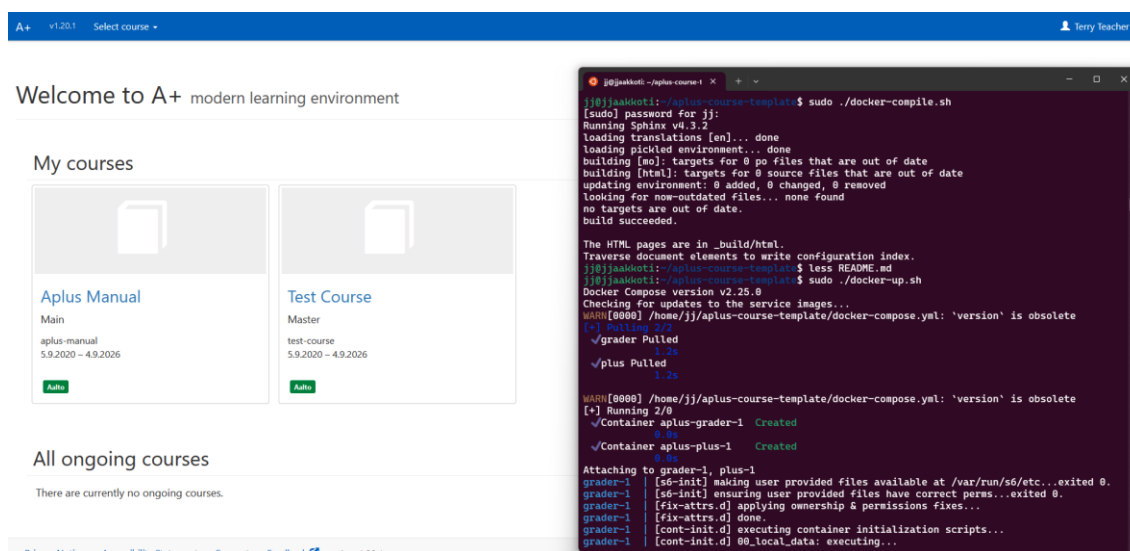


Kuva 6. GitHub näkymä ja ainoa paikka, jossa opiskelija näkee tehtävänannot kootusti.

5 A+ -ympäristön luominen

Aalto-yliopiston A+ -järjestelmän alustaminen eli käytännössä oman verkkosivuston ja palvelimen käynnistäminen valmiista pohjasta on suhteessa GitHub Classroomiin työläämpi prosessi, mutta myös hyvin dokumentoitu ja suoraviivainen. A+ -järjestelmään tehdyt kurssit pitää kuitenkin tallentaa Aalto-yliopiston GitLab-palvelimelle, joten suurin rajoite verrattuna Classroomiin on HAKA-tunnusten välttämättömyys (GitHub-organisaatio A+ LMS, 2023).

Dokumentaation mukainen 30–60 minuutin kestoisen ympäristön asennusprosessi koostuu pääosin työkalujen asennuksesta. Alkuun suositellaan kopioimaan valmis lähdekoodiarkisto pohjaksi, johon alustetaan Gitillä alamoduuli (engl. submodule) RST-pohjaisten kurssimateriaalien kääntämistä varten. Tämän jälkeen käännetään ja käynnistetään Docker-kontti, jolloin verkkosivusto tarjoillaan paikalliselle http-portille (Kuva 6).

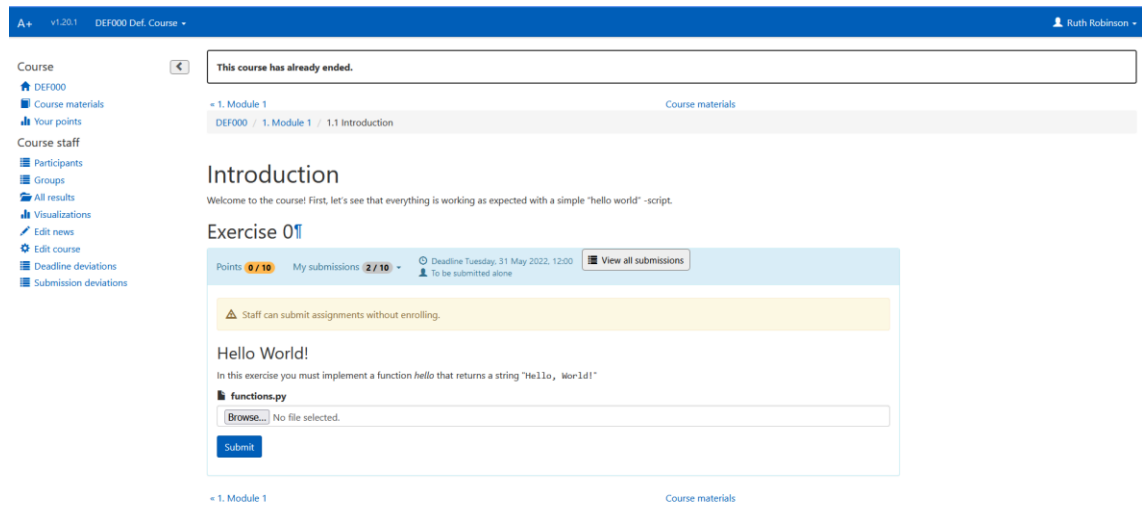


Kuva 7. A+ -järjestelmästä esimerkkisivusto ja sen kääntäminen sekä hakeminen Docker-konteista.

Tehtävänannot A+ -järjestelmässä ovat RST-dokumentteja, jotka renderöidään Sphinx-työkalun avulla HTML-sivuksi palvelimelle. Järjestelmän taustalla on valmiita A+ -työryhmän tekemiä RST- ja Sphinx-direktiivejä tehtävänantojen luomiseen ja mahdollistaa erittäin tarkan, joskin verrattain hyvin teknisen ja työlään, tehtävänantojen koodauksen.

6 A+ -ympäristö opiskelijana

Opiskelijan käyttökokemus A+ -ympäristössä on selkeä, mutta riippuu myös paljon kunkin opettajan tekemästä oppimisympäristöstä. Olettaen sivuston palvelinpuolen (engl. backend) toimivan normaalisti, opiskelijan yksinkertaisesti tarvitsee vain avata linkki tehtävlistasta ja luoda paikallisesti itselleen tehtävänannon mukainen tiedosto. Valmis tiedosto lähetetään A+ -järjestelmään ja erillisellä palvelimella Dockerin kautta ajettava ohjelma tarkistaa sen tehtävänannon mukaisella ohjelmointikielellä. Näkymät opettajan ja opiskelijan näkökulmasta hyvin samanlaiset, joten opettajan on helppo testata tehtäviä (Kuva 7).



Kuva 8. A+ -ympäristö ja tehtävänannon näkymä.

7 GitHub Classroom verrattuna A+ -järjestelmään

GitHub Classroomilla ja A+ -järjestelmällä on hyvin samankaltainen perusajatus ennen kaikkea ohjelmointitehtävien organisointiin. Vaikka ne ovatkin täysin erilaiset ympäristön luomisen suhteen, molemmat kuitenkin perustuvat mahdollisimman automaattiseen tehtävien tarkistukseen opettajan työn helpottamiseksi. Sen lisäksi, molemmilla opiskelijan on helppo organisoida ohjelmistokehitystään Gitin avulla, vaikka A+ -järjestelmällä se pitääkin tehdä irrallaan oppimisen hallintajärjestelmästä.

GitHub Classroom on opettajan kannalta suotavampi, sillä kurssin ja tehtävien luominen on todella yksinkertaista seuraten vain GitHub-käyttöliittymää. A+ on sen sijaan opettajalle hieman vaikeampi asentaa, mutta tarjoaa opiskelijalle perinteisemmän tavan tarkkailla ja palauttaa tehtäviä. Koska A+ on käytännössä oma verkkosivustonsa sekä palvelimensa, se on myös epävakaampi riippuessaan useasta erillisestä palvelusta ja ohjelmistosta. Hyvin jäsenelty ja ohjeistettu GitHub Classroom luokkahuone sekä tehtävät mahdollistavat kuitenkin opiskelijalle erittäin suotuisan tavan opiskella ohjelmointia.

8 Pohdinta

Ohjelmoinnin opettaminen mahdollisimman intuitiivisesti on haaste, mutta GitHub Classroomissa on mahdollista käyttää intuitiivisia ja toimivia opetusratkaisuja. Se käyttää hyödyksi tietotekniikan alalla yleisesti käytettyjä versionhallintatyökaluja sekä mahdollistaa tehtävien tekemisen ja jakamisen saumattomasti.

Tässä opinnäytetyössä tutkittiin GitHub Classroomin toimintaa vertaamalla oppimisympäristön luomista tällä työkalulla sekä toisella hieman samankaltaisella työkalulla, Aalto yliopiston A+ -oppimisen hallintajärjestelmällä. GitHub Classroom osoittautui paremmaksi kuin A+ pääasiassa sen helppokäyttöisen käyttöliittymän ansiosta. A+ on erittäin monipuolinen ohjelmointitehtävien jakamiseen ja arviointiin, mutta vaatii huomattavasti enemmän työtä ja teknistä osaamista. Tällä Aalto-yliopiston kehittämällä järjestelmällä pitää käytännössä ohjelmoida RST-merkintäkielellä alusta kokonainen verkkosivusto kurssia varten. GitHub Classroomilla verkkosivustona sekä palvelimena toimii GitHub, ja opettajan tehtävä on ainoastaan luoda yksittäiset tehtävät sekä testit organisaatiolilleen. Muita hyviä vaihtoehtoja Classroomille ei juurikaan ole lukuun ottamatta tiedostojen jakamista suoraan oppimisen hallintajärjestelmässä ja erillistä manuaalista ohjelmistotestausta.

Jos on olemassa Git-lähdekoodiarkistoja ohjelmointitehtävistä, voidaan GitHub Classroom ottaa käyttöön erittäin helposti. Tästä syystä se soveltuisi erittäin hyvin ohjelmoinnin tai muiden vastaavien opetustehtävien jakamiseen ja arvioimiseen. Myös opiskelijan näkökulmasta tämä opetusympäristö on toimiva, sillä tehtäviä voidaan suorittaa samalla tavalla kuin mitä tahansa ohjelmistokehitysprojektia.

Lähteet

1EdTech, 2024. *Learning Tools Interoperability (LTI)*. [Online]

Available at: <https://www.1edtech.org/standards/lti>

[Haettu 08 02 2024].

Aplus team, 2021. *RST Guide*. [Online]

Available at: https://plus.cs.aalto.fi/aplus-manual/master/rst_guide/get_started/

[Haettu 28 03 2024].

Chacon, S. & Long, J., 2024. *About*. [Online]

Available at: <https://git-scm.com/about>

[Haettu 28 02 2024].

Dohmke, T., 2023. *100 million developers and counting*. [Online]

Available at: <https://github.blog/2023-01-25-100-million-developers-and-counting/>

[Haettu 28 02 2024].

GitHub, Inc., 2024. *About GitHub Classroom*. [Online]

Available at: <https://docs.github.com/en/education/manage-coursework-with-github-classroom/get-started-with-github-classroom/about-github-classroom>

[Haettu 08 02 2024].

GitHub, Inc., 2024. *About GitHub Classroom*. [Online]

Available at: <https://docs.github.com/en/education/manage-coursework-with-github-classroom/get-started-with-github-classroom/about-github-classroom>

[Haettu 28 02 2024].

GitHub, Inc., 2024. *Organizations and teams documentation*. [Online]

Available at: <https://docs.github.com/en/organizations>

[Haettu 6 2 2024].

GitHub-organisaatio A+ LMS, 2023. *aplu-course-template*. [Online]

Available at: <https://github.com/apluslms/aplus-course-template?tab=readme-ov-file>

[Haettu 28 03 2024].

Karavirta, V., Ihantola, P. & Koskinen, T., 2013. *Service-Oriented Approach to Improve Interoperability of E-Learning Systems*. Beijing, China, s.n., pp. 341-345.

Preston-Werner, T., 2008. *tom.preston-werner.com*. [Online]

Available at: <https://tom.preston-werner.com/2008/10/18/how-i-turned-down-300k>

[Haettu 27 02 2024].

Riekkinen, M., Sarolahti, P. & Rytönen, A., 2020. *A+ LMS*. [Online]
Available at: <https://apluslms.github.io/about/authors/>
[Haettu 13 03 2024].

Riekkinen, M., Sarolahti, P. & Avoimen lähdekoodin yhteistyökijät, 2024. *A+ LMS*. [Online]
Available at: <https://apluslms.github.io/>
[Haettu 13 03 2024].

Riekkinen, M., Sarolahti, P. & Rytönen, A., 2024. *A+ LMS*. [Online]
Available at: <https://apluslms.github.io/architecture/>
[Haettu 11 04 2024].

The Linux Foundation, 2015. *10 Years of Git: An Interview with Git Creator Linus Torvalds*. [Online]
Available at: <https://www.linuxfoundation.org/blog/blog/10-years-of-git-an-interview-with-git-creator-linus-torvalds>
[Haettu 15 02 2024].

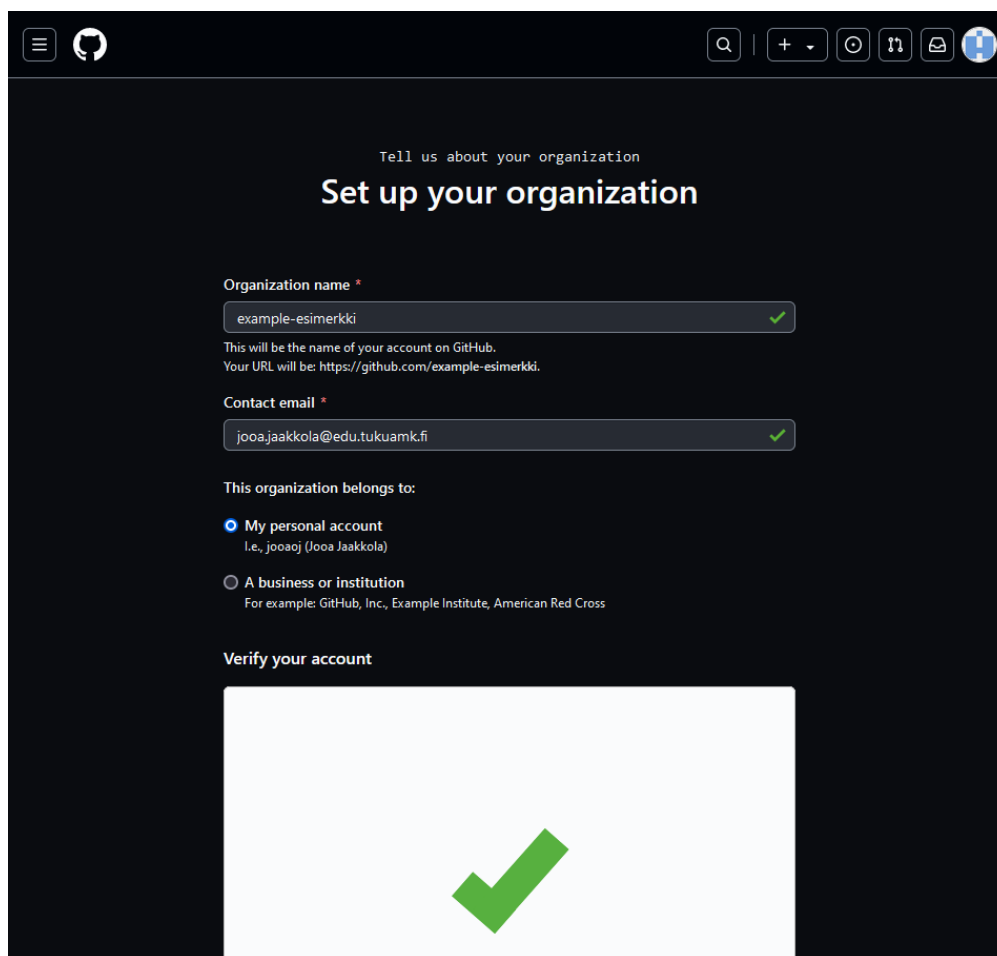
Wikimedia Foundation, Inc., 2022. *Integrated development environment*. [Online]
Available at: https://en.wikipedia.org/wiki/Integrated_development_environment
[Haettu 08 02 2024].

Wikimedia Foundation, Inc., 2023. *Software configuration management*. [Online]
Available at: https://en.wikipedia.org/wiki/Software_configuration_management
[Haettu 28 02 2024].

yaml.org, 2021. *YAML 1.2*. [Online]
Available at: <https://yaml.org/>
[Haettu 28 05 2024].

GitHub Classroomin käyttö opettajana

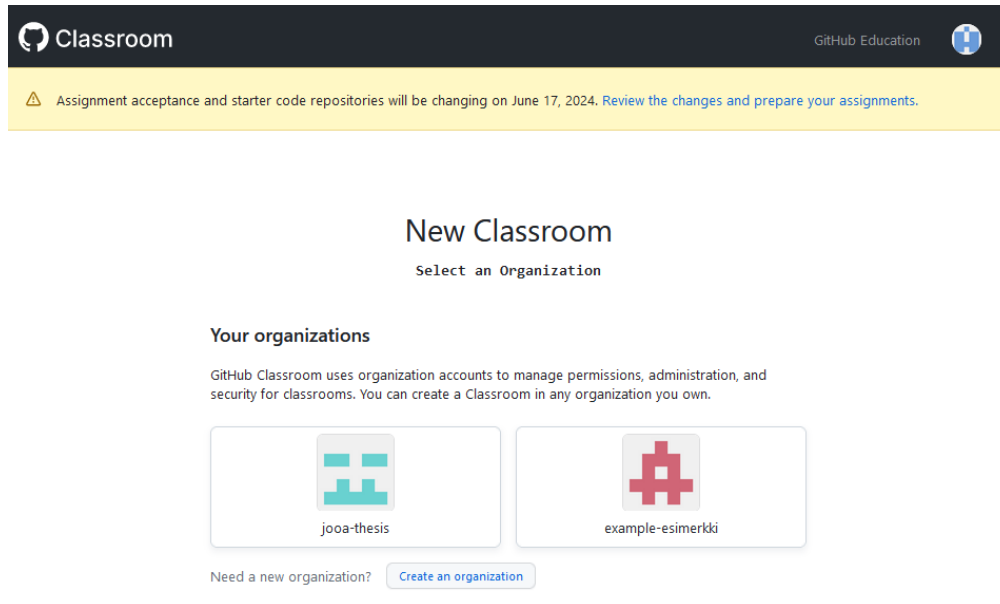
Luokkahuoneen tekeminen aloitetaan GitHub-organisaatiotilin luomisella, mikäli sellaista ei entuudestaan ole, mutta se on hyvin suoraviivainen rekisteröintikysely (kuva 8).

The image shows a screenshot of the GitHub 'Set up your organization' form. The form is titled 'Set up your organization' and includes the following fields and options:

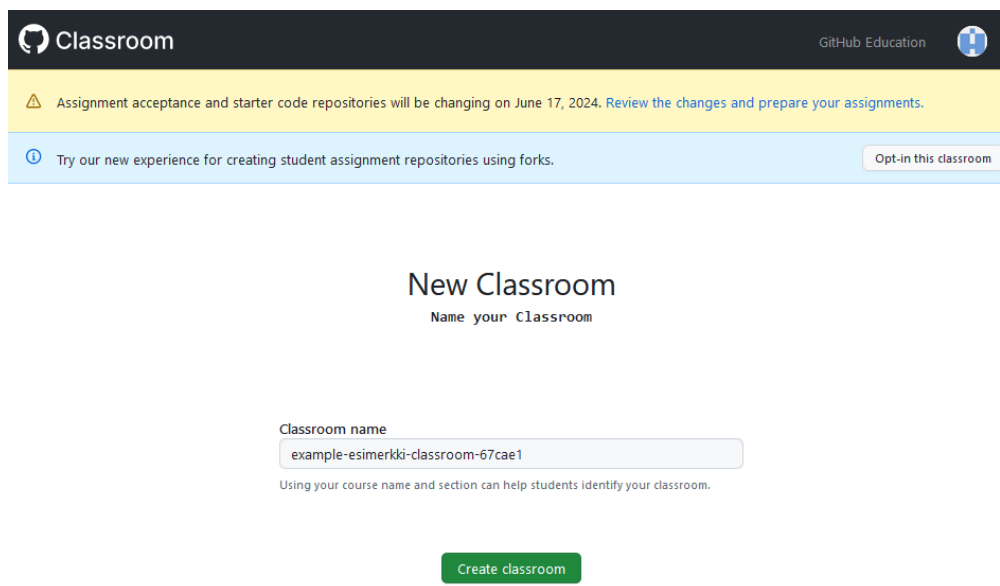
- Organization name ***: A text input field containing 'example-esimerkki' with a green checkmark to its right. Below the field, it says: 'This will be the name of your account on GitHub. Your URL will be: https://github.com/example-esimerkki.'
- Contact email ***: A text input field containing 'jooa.jaakkola@edu.tukuamk.fi' with a green checkmark to its right.
- This organization belongs to:**
 - My personal account**
i.e., jooaj (Jooa Jaakkola)
 - A business or institution**
For example: GitHub, Inc., Example Institute, American Red Cross
- Verify your account**: A large white box containing a large green checkmark, indicating successful verification.

Kuva 9. Organisaatiotilin luominen ja sen GitHub-osoite.

Organisaatiotilin rekisteröinnin jälkeen luokkahuone luodaan sen pohjalle kahden napinpainalluksen ja nimeämisen jälkeen (kuva 9 ja kuva 10).



Kuva 10. Luokkahuoneen luominen organisaatiotilille.

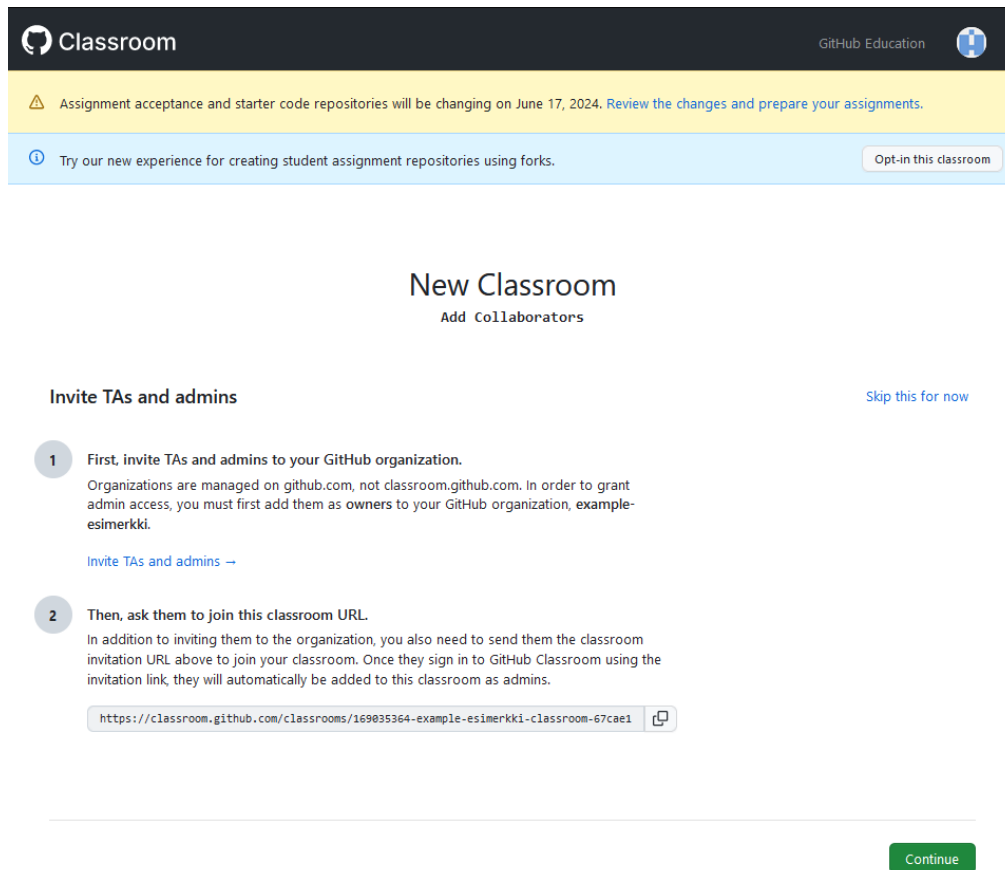


Kuva 11. Luokkahuoneen alustava nimeäminen.

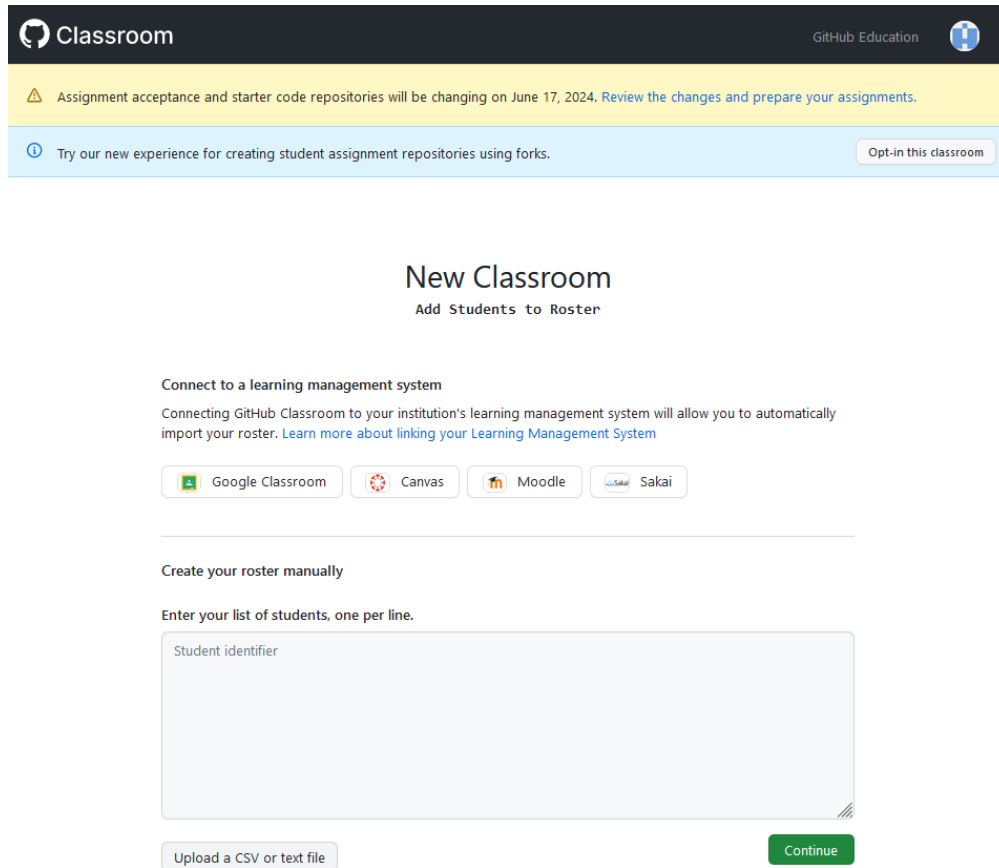
Kun luokkahuone on yhdistetty organisaatiotiliin sekä nimetty, voidaan siihen kutsua ja liittää yhteistyötä tekevät opettajat sekä oppilaat. Opettajat kutsutaan suoraan GitHub-tunnuksilla tai linkillä, joka sisältää ylläpito-oikeudet. Oppilaat sen sijaan tuodaan joko suoraan tietyistä oppimisen hallintajärjestelmistä tai listana muualta (kuva 11 ja 12). Oppilaat ja niiden GitHub-tilit kuitenkin

Liite 1

yhdistyvät varsinaisesti luokkahuoneeseen vasta ensimmäisen tehtävän linkin hyväksytyä.



Kuva 12. Luokkahuoneeseen osallistuvien mahdollisten muiden opettajien kutsuminen.



Kuva 13. Luokkahuoneeseen oppilaiden lisääminen olemassa olevasta oppimisen hallintajärjestelmästä joko suoraan tai tekstinä.

Tehtävät GitHub Classroomiin luodaan GitHub-lähdekoodiarkistoista, jotka voivat olla joko organisaatiotililtä tai muualta GitHubista kopioituna mallipohjana. Tehtävät tarkistetaan automaattisesti ohjelmointikielikohtaisesti testiskripteinä, jotka suoritetaan jokaisen tallennuksen (engl. push) jälkeen GitHubin toimet (engl. actions) toiminnon avulla. Testit on luotava itse, mutta niitä on helppo uusiokäyttää (kuva 13, 14 ja 15).

Liite 1

Let's set up the basics for your assignment.

Assignment creation steps

- Assignment basics
- Starter code and environment
- Grading and feedback

Assignment title *

Conditional basics

Student assignment repositories will have the prefix: `conditional-basics`

Deadline

dd / mm / yyyy, --, --

(Optional) If left blank, there will be no deadline. Date format: 12/31/2022, 11:59 PM

Individual or group assignment

Individual assignment

Cancel Continue

Kuva 14. Luokkahuoneen tehtävien luominen kolmessa vaiheessa: perustiedot.

Add your starter code and choose an optional online IDE.

Assignment creation steps

- Assignment basics
- Starter code and environment
- Grading and feedback

Add a repository to give students starter code

Your assignment will be created with empty student repositories if you don't add starter code.

If the repository exists outside of your organization it must be a template, a copy will be created in your organization.

Accepted assignments will be forks of the assignment repository. Check our guide to learn how students can [sync changes](#) to the starter code after accepting the assignment.

Find a GitHub repository

jooa-thesis/Conditionals-Loops

Repository visibility

When a public repository is selected as starter code, student assignment repository forks must also be public. Create a private copy of the starter code if you wish for student repositories to be private (only visible to the student and the classroom owners)

Public

Public repositories will be visible to everyone, including other students.

Grant students admin access to their repository

Editing this after assignments are created will not retroactively change permissions.

Copy the default branch only

Only the default branch of your repository will be included in your students assignments. If unchecked, all branches will be copied.

Add a supported editor

Automatically include a link to an editor in students' repositories to give them a one-click experience for getting started coding, running, and collaborating on their code.

Select an editor

← Back Cancel Continue

Kuva 15. Luokkahuoneen tehtävien luominen kolmessa vaiheessa: aloituskoodi ja käyttöoikeudet.

Liite 1

Set up autograding and feedback.

Assignment creation steps

- ✓ Assignment basics
- ✓ Starter code and environment
- Grading and feedback**

Add autograding tests [New Give feedback](#)

Autograding tests help provide feedback for students immediately upon submission using [GitHub Actions](#). Add a test to enable autograding.

Conditionals and loops - pytest ✎ 🗑

+ Add test ▾

Choose an autograding method:

- input_output g tests are run [New Give feedback](#)
- run_command an assignment
- java ype/Kyiv)
- node
- python ing test runs for all students from the assignment dashboard
- c
- cpp

Protected file paths [New Give feedback](#)

Assign protected file paths to monitor whether students change tests or other important files. A "Protected file(s) modified" label will be applied to student submissions that change protected files on the assignment dashboard.

Example: `github/**/*`

Type file path

🗑

Enable feedback pull requests

A pull request will automatically be created on all student repository submissions. Pull requests allow you to answer questions and provide feedback.

Kuva 16. Luokkahuoneen tehtävien luominen kolmessa vaiheessa: automaattitestit ja niiden suojaus muutoksilta.

Kun testit on luotu oikein ja oppilaiden GitHub-tunnukset yhdistetty luokkahuoneeseen, näkyvät oppilaiden tulokset sekä tehtävän suorittaneet kootusti tehtäväsivulla (kuva 16).

Liite 1

The screenshot displays the GitHub Classroom interface for an assignment titled "Conditional basics". At the top, there is a navigation bar with the "Classroom" logo and "GitHub Education" text. A yellow banner below the header contains a warning message: "Assignment acceptance and starter code repositories will be changing on June 17, 2024. Review the changes and prepare your assignments." Below this, the breadcrumb "Classrooms / jooa-thesis-classroom / Conditional basics" is visible. The main heading "Conditional basics" is followed by a search bar and a status indicator "Active". A URL bar shows "https://classroom.github.com/a/ygatdVfc" with icons for "Run tests", "Edit", and "Download".

The "Assignment Details" section features four summary cards:

- Students total:** 4 Rostered, 0 Added students
- Accepted assignments:** 1 Students
- Assignment submissions:** 1 Submitted, 0 Not submitted
- Passed students:** 1/1 Passed (indicated by a green progress bar)

Below these cards is a filter bar with a search input "Search for an assignment" and several filter buttons: "Filter by unlinked accounts", "Filter by accepted", "Filter by passing", and "Sort".

The "Classroom roster" section shows a single student entry:

- Profile: jooajaa@gmail.com (Submitted)
- Avatar: @jooajaa-tester
- Activity: Latest commit 2 months ago ✓
- Commits: 4 commits
- Progress: 1/1
- Repository icon: Repository

Kuva 17. Luokkahuoneen tehtäväkohtainen näkymä ja oppilaiden tulokset.

Esimerkki GitHub Classroom luokkahuoneesta

Jotta opettaja saa kaikkien kurssille osallistuvien oppilaiden käyttäjätunnukset, on kätevintä aloittaa esimerkiksi GitHubin tarjoamalla esimerkkitoteutuksella. Tehtävässä opetetaan käyttämään GitHubia ja tulkkamaan sekä testaamaan Python-koodia (kuva 17).

The screenshot shows a GitHub repository page for a course. The repository name is 'git-github-fundamentals-Git-GitHub-Fundamentals' and it is public. The repository was generated from 'jooaj/Git-GitHub-Fundamentals'. The repository has 1 branch (master) and 0 tags. The commit history shows several commits, including updates to README.md and initial commits for .gitignore, hello.py, and hello_test.py. The README file is open, showing the course overview and learning outcomes. The README content includes:

The Basics of GitHub

Course overview and learning outcomes

The goal of this course is to give you a brief introduction to GitHub. We'll also provide you with materials for further learning and a few ideas to get you started on our platform.

How to Git

1. Make sure you have Git installed by just writing "git" in any command-line interface
2. Copy this repository from the top of this page - either with the http- or ssh-link
3. Then, if you don't already have, make sure you either: 3.1 generate an ssh-key pair and copy the public-key to GitHub (access settings) 3.2 or generate an oauth access token (from developer setting)
4. When you have git ready and security keys set up, just write `git clone <your-repository-link>` into your CLI
5. When making changes, you mainly add, commit and push (`git add .`, `git commit -m "<commit-message>"` and `git push`)

Autograding Example: Python

This example project is written in Python, and tested with pytest.

The assignment

The tests are failing right now because the method isn't outputting the correct string. Fixing this up will make the tests green.

Setup command

```
sudo -H pip3 install pytest
```

Run command

```
pytest
```

- then make fixes to according source code

Notes

- pip's install path is not included in the PATH var by default, so without installing via `sudo -H`, pytest would be unaccessible.

Git and GitHub

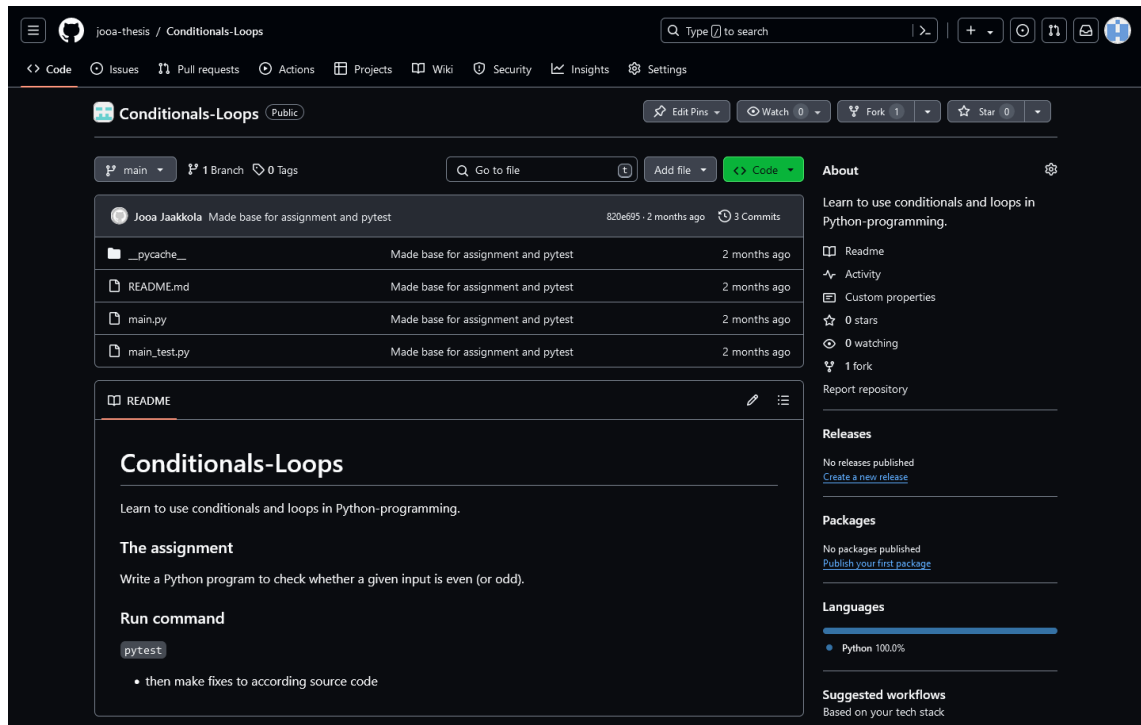
Git is a distributed Version Control System (VCS), which means it is a useful tool for easily tracking changes to your

The right sidebar shows repository statistics: 0 stars, 0 watching, 2 forks, and 0 releases. It also lists suggested workflows for Django and Python packages using Anaconda.

Kuva 18. Lähdekoodiarkisto ensimmäisestä tehtävästä: <https://github.com/jooa-thesis/git-github-fundamentals-Git-GitHub-Fundamentals/tree/master>.

Liite 2

Tehtävänantoja on pääasiassa kuitenkin suotavaa luoda itse omalle organisaatiolililleen, jotta ne säilyisivät tallessa uusiokäyttöä sekä päivityksiä varten. Esimerkkinä itse luodusta tehtävästä on yksinkertainen ohjausrakennetehtävä Python-ohjelmointikielellä ja Pytest-testauskirjastolla (kuva 18).



Kuva 19. Lähdekoodiarkisto toisesta tehtävästä: <https://github.com/jooa-thesis/Conditionals-Loops>.