

WETO - WEb Teaching Organiser - kuvaus

Osoite: <https://weto.sis.uta.fi/> (vaatii tällähetkellä Tampereen yliopiston tunnuksen)

Kehitysversion osoite: <https://wetodev.sis.uta.fi/> (vaatii tällähetkellä Tampereen yliopiston tunnuksen)

Lisätietoja: Heikki Hyyrö (heikki.hyyro@uta.fi)



WETO:n kirjautumisnäyttö.

Taustaa

WETO on suunniteltu palvelemaan varsinkin seuraavia tarpeita:

- Alusta kurssitöiden (esim. viikkoharjoitukset, harjoitustyöt) palautukselle.
- Kurssitöiden vertaisarviointi.
- Ohjelmakoodin automaattinen tarkastus.

WETO:n yksi tärkeä päämäärä on tehostaa opettajan työajan käyttöä tarjoamalla apuvälineitä niin tiedon välittämiseen, kurssitöiden vastaanottoon, pisteyttämiseen kuin kurssiarvosteluunkin. Järjestelmä on ainakin ollut tarkoitus pitää kohtalaisen yksinkertaisena, jonka käyttö on sutjakkaa ja täyttää kurssien tärkeimmät perustarpeet.

WETOn kehitys aloitettiin 2000-luvun alussa projektimuotoisesti, ja mukana on ollut toisinaan esim. harjoittelijoita tai IT-hallinnon henkilöitä aputyövoimana. Aktiivinen kehitys ilmeisesti oli hiipunut viimeistään 2007 mennessä. WETO oli kuitenkin jatkuvassa kurssikäytössä, vaikka siinä oli teknisiä ongelmia eikä esim. ohjelmakoodin automaattinen tarkistus ollut toimintakunnossa.

Nykyinen WETO alkoi kehittyä vuodesta 2012 alkaen. Tietotekniikkakilpavalmennuksen yhteydessä monenlaisiin automaattista ohjelmakoodin tarkistusta käyttäviin järjestelmiin tutustunut yliopistonlehtori Heikki Hyrö aloitti projektin, jonka päämääränä oli saada käyttöön automaattista tarkistusta tukeva järjestelmä ohjelmointikursseille. Aluksi tämä yritettiin toteuttaa Moodlen päälle, mutta projektiryhmä piti Moodlea teknisesti ja käyttökokemukseltaankin sekavana, eikä projektin lopputuotosta koskaan otettu kurssikäyttöön. Seuraava askel oli ottaa WETO lähtökohdaksi. Koska WETOn alkuperäinen tekninen toteutus oli paikoitellen heikko ja järjestelmä lisäksi pohjautui vanhentuneeseen viitekehukseen, viitekehys vaihdettiin uudempaan ja valtaosa palvelinkoodista kirjoitettiin uudelleen. Samalla mukaan lisättiin toimiva automaattinen ohjelmakoodin tarkistus. Koodipohjan uudistus vaikuttaisi saavuttaneen tavoitteensa: nykyisin WETO on järjestelmänä vakaa (esim. voi olla kuukausikaupalla ajossa passiivisesti ilman että tarvitsee tehdä muita ylläpitotoimenpiteitä kuin palvelimen Linux-järjestelmän peruspakettien ajoittaiset päivitykset). Viimeisten n. 5 vuoden aikana WETOa on kehitetty pieni askel kerrallaan. Suurin yksittäinen askel tänä aikana on ollut WETOn käyttöliittymän kokonaisvaltainen uudistaminen, joka toteutettiin 2015.

WETO / Ohjelmoinnin tekniikka C

Ohjelmoinnin tekniikka C

Ohjeita / tiedotteita:

Viimeisimmät uutiset:

- Seuraava tentti järjestetään ma 10.12.2018 klo 14-18 mikroluokassa ML1 (Tietopinnin alakerta).
- 20.11.2018 pidetyn tentin palautukset ja arvosanat löytyvät sivun "Tentti" alaisuudesta.
- Syksyn 2018 kurssia koskeva tiedotus tapahtuu alisivun "Syksyn 2018 kurssi" alaisuudessa (sivu näkyy vain sisäänkirjautuneille käyttäjille).
- Kurssi luennoidaan (ohjelmointidemoja jne.) syksyllä 2018 tiistaisin 4.9-13.11 klo 16-19 luentosalissa **Pinni B 3107**. Luennoille suositellaan ottamaan omat kannettavat mukaan, jotta voi itsekin kokeilla ajaa läpikäytäviä esimerkkikoodeja.
 - o Kannettavan esivalmistelu ennen luentoja:
 - Asenna gcc-kääntäjä sivulla [Ohjeita C-ohjelmien kääntämiseen](#) annettujen ohjeiden mukaisesti.
 - Jos halut käyttää samaa editoria kuin luennoitsija, lataa ja asenna Geany-editori sivulta www.geany.org/Download/Releases.

Tenttioikeuden saaminen edellyttää, että olet palauttanut hyväksytyt palautukset vähintään 50% näillä sivuilla julkaistusta harjoitustehtävistä. On kuitenkin hyvä muistaa, että koska tenttitehtävät pohjautuvat pitkälti harjoitustehtäviin, kannattaa tenttimenestyksen kannalta tehtäviä tehdä mieluummin enemmän, kuin minimivaatimuksen verran! Tehtävät tarkistetaan WETOn automaattisella tarkastuksella. Harjoitustehtävät on jaettu neljään osaan, ja edellä mainittua 50% rajaa sovelletaan kuhunkin erikseen.

Harjoitustehtävien neljä osaa ovat:

- Ensimmäinen osa **Osa 1: C-kielen perusteita**.
 - o 20 tehtävää, jotka koskevat pääasiassa alisivulla [Kurssimateriaalia](#) tarjolla olevien luentokalvojen sivuilla 1-40 käsiteltyä asiaa.
 - o Tehtävä vähintään 50% eli 10 tehtävää.
- Toinen osa **Osa 2: Bitit, osoittimet ja tietueet**.
 - o 16 tehtävää, jotka koskevat pääasiassa luentokalvojen sivuja 41-82.
 - o Tehtävä vähintään 50% eli 8 tehtävää.
- Kolmas osa **Osa 3: Dynaaminen muistinhallinta, void-osoittimet ja tiedostot**.
 - o 20 tehtävää, jotka koskevat pääasiassa luentokalvojen sivuja 83-125.
 - o Tehtävä vähintään 50% eli 10 tehtävää.

Ohjelmoinnin tekniikka C -kurssin pääsivu kirjautumattomalle käyttäjälle.

Toiminnallisuudet

WETOn toiminnallisuus on suunniteltu “perinteisen kurssin” tarpeiden mukaiseksi, ja tarkoitus on, että kaikki toiminnot voi tehdä web-käyttöliittymän kautta. Pääpiirteissään WETO tarjoaa seuraavat ominaisuudet:

- Kurssi-ilmoittautumisen / kurssilta pois vetäytymisen.
 - Voi säätää aikarajat, mihin asti uudet opiskelijat saavat kirjautua kurssille, ja mihin mennessä ilmoittautuminen on mahdollista perua itse.
- Tuen ryhmille.
 - Kurssille voi määrittää ryhmiä (esim. jos kurssilla on erillisiä harjoitusryhmiä).
 - Opiskelija voi itse valita ryhmänsä kurssille ilmoittautumisen yhteydessä, tai vaihtoehtoisesti opettaja voi asettaa opiskelijat ryhmiin.
 - On myös mahdollista määrittää “palautusryhmiä”, joiden puitteissa ryhmän jäsenet jakavat saman palautusnäkömän ja pisteet (jos joku jäsen palauttaa tiedoston, muutkin näkevät sen omassa palautusnäkömässään; kun palautus pisteytetään, sama pistemäärä tulee jokaiselle ryhmäläiselle).
- Tavanomaisen, joskin ehkä hieman rudimentaalisen(?), sisällönhallinnan.
 - Kurssin sivujen sisältöä voi muokata “WYSIWYG”-html-editorilla, ja kurssin alaisuuteen voi lisätä vapaasti alisivuja, ladata kuvia/tiedostoja, jne. Sisältö tiedostoihin päivineen näkyy vain sivulle oikeudet omaaville käyttäjille.
 - WETO itsessään tallentaa sivut raaka-HTML-muodossa sellaisenaan, joten järjestelmä ei ainakaan periaatteessa aseta rajoituksia sivujen sisällölle.
 - Kullekin kurssille voi halutessaan määrittää omat CSS-tyylimäärittelyt (tätä harvemmin taidetaan käyttää).
 - Kurssin sivuja on mahdollista määrittää julkisiksi, jolloin kaikki (myös sisäänkirjautumattomat) saavat lukuoikeuden.
 - Kurssin (tai jonkin sen “alipuun”) voi ladata talteen tiedostoon, ja toisaalta ladata tiedostosta takaisin palvelimelle (esim. siirtää palvelimelta toiselle, jos haluaa).
 - Lataukseen (kumpaan suuntaan tahansa) voi määrittää, otetaanko mukaan myös opiskelijadata (arvosanat, vastaukset jne.). Tätä on käytetty mm. siten, että tentti sisältöineen on siirretty erillisestä tentti-WETO-instanssista tavanomaiseen kurssi-WETO-instanssiin, jolloin opiskelijat pääsevät kurssin sivuilla katselemaan tenttipisteidensä lisäksi myös alkuperäistä tenttiä sellaisenaan: tenttitehtäviä ja omia vastauksiaan.
 - Sivuja voi helpohkosti uudelleenjärjestellä kurssin sisällä ja kopioida toisesta kurssista toiseen.
- Melko joustavan mahdollisuuden säätää oikeuksia tarvittaessa yksittäisten käyttäjien tasolla.
 - Esim. voi antaa yksittäiselle Matti/Maija Myöhäiselle lisää aikaa jonkin harjoituksen palautukseen.

- Sivun näkyvyyden voi määrittää ehdolliseksi: sivu näkyy opiskelijalle vain, jos hän on saanut hyväksytyt pisteet jostain määrätystä tehtävästä tai tehtäväkokonaisuudesta.
- Nykyisellään seuraavat tehtävätyypit:
 - Tiedostomuotoinen palautus.
 - Voi määrittää käyttämään automaattista tarkistusta. Tarkistuksen muodon voi säätää aika vapaasti omien skriptien avulla.
 - Tiedoston voi luoda / sitä voi editoida suoraan WETOssa. Tätä ominaisuutta on käytetty esim. sellaisissa tenteissä, joissa opiskelijoilla on ollut käytettävissään ainoastaan verkkoselain, josta pääsee ainoastaan WETOon.
 - Lomakepohjainen palautus (upotetaan kurssisivulle tavanomaisen tekstin joukkoon).
 - Esseetehtävä.
 - Automaattisesti tarkistettava koodi (jolloin lomake = koodieditori).
 - Monivalintatehtävä.
- Vertaisarvioinnin.
 - Tehtävään palautetut vastaukset voi määrittää vertaisarvioitaviksi (sanallinen arvio + numeerinen arvosana). Tällöin, asetuksista riippuen, kukin palauttanut arvioi x muun opiskelijan palaukset nimettöminä, ja mahdollisesti myös omansa
 - Vertaisarvioinnin voi määrittää pakolliseksi: opiskelija saa tehtävästä itselleen pisteitä vain jos on tehnyt ainakin yhden vertaisarvioinnin.
- Melko joustavan mahdollisuuden säätää pistelaskusääntöjä kurssin eri osille.
 - Kurssin sivut muodostavat puumaisen hierarkian.
 - Hierarkian kullekin solmulle voi esim. määrittää arvosanataulukon. Esim. jos solmu on "1. Harjoitukset", ja sen alla on vaikkapa 8 kpl 1 pisteen arvoisia tehtäviä, voi solmuun "1. harjoitukset" määrittää vähimmäispisterajan, mitä sen lapsista (ne 8 tehtävää) pitää yhteensä saada. Toisaalta solmu "1. Harjoitukset" voi puolestaan olla solmun "Harjoitukset" alaisuudessa, ja solmu "Harjoitukset" kurssin pääsivun alaisuudessa. Pääsivun alaisuudessa voisi sitten olla myös vaikkapa solmut "Tentti" ja "Harjoitustyö". Kaikille näille voisi määrittää pisteytyssäännöt (mukaanlukien esim. pistemäärä -> arvosana -tyyppisiä sääntöjä, kuten 5p -> 1, 8p -> 2, 11p -> 3, 14p -> 4 ja 8p -> 5), joiden kautta järjestelmän voi pyytää laskemaan koko kurssiarvosanan harjoitusten, harjoitustyön ja tentin pisteiden pohjalta.
 - Pisteytykseen voi myös määrittää "myöhästymispisteitä" luettelemalla päivämäärä=pistemäärä -pareja. Tällöin pisteidenlasku vähentää vastauksen pisteistä kaikkien palautusajankohtana jo ohi menneiden päivämäärien mukaiset pisteet kumulatiivisesti.
- Erittäin rudimentaalisen keskustelufoorumin.
- Lokin.
 - Pitää kirjaa siitä, milloin kukin käyttäjä katsoo jotain sivua, palauttaa jonkin tiedoston, tallentaa vertaisarvioinnin, jne.

- Opettaja voi tarkastella lokia kurssinsa alaisuudessa ja esim. rajata lokinäkömään koskemaan vain tiettyä ajankohtaa/käyttäjää/toimenpidettä/sivua.

Tekninen toteutus

WETO on Apache Struts 2 -web-ohjelmointikehyksellä (<https://struts.apache.org/>) toteutettu Java Servlet sovellus, jota voi ajaa Java Servlet -yhteensopivilla sovelluspalvelimilla. Käytännössä WETOa on aina ajettu Apache Tomcatilla (<https://tomcat.apache.org/>).

WETO:n data tallennetaan PostgreSQL-tietokantaan.

Automaattisesta tarkistuksesta vastaa erillinen palvelu, joka käyttää joissain IOI (International Olympiad in Informatics) ohjelmointikilpailuissakin käytettyjä apuvälineitä ohjelman turvalliseen ajamiseen ja resurssien (aika, muisti) rajaamiseen/mittaamiseen. Palvelu itsessään on lisäksi asetettu chroot-pohjaiseen hiekkalaatikkoon (JailKit). Wetodev-palvelimella ajetaan samanaikaisesti sekä WETOa että tarkistuspalvelua. Tenteissä käytetään nykyään varmuuden vuoksi järjestelyä, jossa WETO ja tarkistuspalvelu ovat erillisillä palvelimilla, koska esim. huonosti (= liian väljät aikarajat) säädettyjen tehtävien tarkastus voisi kuormittaa palvelinta niin paljon, että tarkistuskuorma alkaisi hidastaa/kankeuttaa samalla palvelimella pyörivän WETO:n kurssisivujen käyttöä.

WETO:n arkkitehtuuri on kohtalaisen modulaarinen. Siihen voi kytkeä joustavasti monta toisistaan riippumatonta tietokantaa (esim. kurssin X data tietokannassa A, kurssin Y data tietokannassa B, jne.) tai monta automaattista tarkistinta. Automaattisen tarkistuksen toteutus on sellainen, että tarkistin kysyy WETO:lta, onko tarkistettavaa, eikä toisinpäin.

WETO:n kehityksessä on noudatettu 2012 alkaen käytäntöä, että on kaksi erillistä WETO-instanssia: weto.sis.uta.fi on peruspalvelin, jolla on harvemmin päivittyvä WETO, ja wetodev.sis.uta.fi sisältää aina uusimman version. Kumpaakin instanssia käytetään aktiivisesti kurssiopetuksessa. Uudet ominaisuudet toteutetaan pienin inkrementein ja otetaan lähes välittömästi käyttöön wetodev-palvelimella. Wetodev-palvelimen rooli onkin toimia uusien ominaisuuksien "live-testaus" palvelimena. Kun wetodev-palvelimella oleva versio on todettu ongelmattomaksi, päivitetään sama uusi versio myös perus-weto-palvelimelle. Kehitystyötä tehdään joustavasti WETOa käyttävien opettajien toiveiden pohjalta.

Kirjautuminen

Wetoon kirjaututaan yliopiston peruspalvelutunnuksella. Lisäksi järjestelmään on mahdollista luoda erillistunnuksia.

Muuta

Käytetty kursseilla:

- TIEY4 Tietotekniikkataidot, 3op
- TIEP1 Lausekielinen ohjelmointi I, 5op
- TIEP3, Tietokantojen perusteet, 5op
- TIEP5 Lausekielinen ohjelmointi II, 5op
- TIETA10, Ohjelmoinnin tekniikka C, 5op
- TIETA11, Ohjelmoinnin tekniikka, C++, 5op
- TIEY1, Opintojen ohjattu suunnittelu, 2op
- TIETS14, Introduction to Formal Specification, 5op