

SAVONIA



OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN ALA

RUST-OHJELMOINTI -KURSSI TYKO-VIIKOLLE

Kurssin teoria, suunnittelu ja toteutus

TEKIJÄ/T Joonas Rissanen

Koulutusala Tekniikan ja liikenteen ala	
Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma	
Työn tekijä Joonas Rissanen	
Työn nimi Rust-ohjelmointi -kurssi TYKO-viikolle	
Päiväys	25.3.2025
	32/0
Yhteistyötaho Savonia	
<p>Tämän opinnäytetyön tavoitteena oli toteuttaa Savonia-ammattikorkeakoululle ohjelmointikurssi osana Kuopion alueen lukioiden TYKO-viikkoa. Kurssi toteutettiin hybridimallilla, jossa yhdistettiin lähi- ja etäopetusta.</p> <p>Työn viitekehysessä keskitytään tietojenkäsittelyn perusteisiin, jotka tukevat olennaisesti kurssin sisältöä ja opetusta. Viitekehys käsittelee, miten tieto esitetään tietokoneen muistissa käyttäen binäärijärjestelmää, mukaan lukien kokonais- ja liukuluvut sekä tekstin esittäminen ASCII- ja Unicode-standardeilla. Lisäksi käsitellään tietokoneohjelmissa käytettäviä satunnaislukuja ja niiden generointitapoja: näennäissatunnaislukugeneraattoreita sekä aitoja satunnaislukugeneraattoreita. Myös tietorakenteet, erityisesti taulukot ja vektorit, muodostavat keskeisen osan teoriaosuudesta.</p> <p>Viitekehys toimii opintojakson luentojen ja käytännön harjoitusten perustana, ja sen tarkoituksena on antaa opiskelijoille selkeä käsitys siitä, miten tieto rakentuu tietoteknisissä järjestelmissä, mikä puolestaan tukee ohjelmoinnin perusteiden omaksumista.</p> <p>Lopputuloksena järjestettiin viisipäiväinen kurssi, joka koostui luentodioista, esimerkkikoodeista, harjoitustehtävistä sekä Moodle-tenteistä. Kurssimateriaalit ja tentit laadittiin tukemaan ohjelmoinnin perusteiden oppimista ja käsitteiden sisäistämistä.</p>	
Avainsanat Moodle, Savonia, Rust-ohjelmointikieli, Ohjelmoinnin perusteet, TYKO-viikko	

SISÄLTÖ

1	JOHDANTO.....	5
2	SAVONIA	6
3	TYÖKALUT	7
3.1	Moodle.....	7
3.1.1	Historia	7
3.1.2	Pedagogiikka	7
3.1.3	Tekninen katsaus.....	8
3.2	Rust Playground.....	8
4	TIEDON ESITYS TIETOKONEISSA.....	9
4.1	Binääriluvut.....	10
4.2	Negatiiviset binääriluvut	11
4.2.1	Yhden komplementti	11
4.2.2	Kahden komplementti	12
4.3	Liukuluvut	12
4.4	Tekstin esittäminen	13
5	SATUNNAISLUVUT OHJELMOINNISSA	15
5.1	Näennäissatunnaislukugeneraattorit.....	15
5.2	Aidot satunnaislukugeneraattorit	15
6	TIETORAKENTEET	17
6.1	Taulukot.....	17
6.2	Vektorit Rustissa	18
7	TOTEUTUS	19
7.1	Lähtökohdat ja tavoitteet	19
7.2	Kurssin toteutunut aikataulu	20
7.3	Toteutusympäristön valinta ja perusteet	21
7.4	Opetusmateriaalin toteutus	21
7.4.1	Luentomateriaali	21
7.4.2	Harjoitustehtävät	22
7.4.3	Tentit ja kurssin arviointi	23
7.5	Moodle-rakenne	23
7.6	Testaus ja viimeistely	28
8	POHDINTA.....	29

LÄHTEET	30
---------------	----

KUVALUETTELO

Kuva 1: 32-bittisen ja 64-bittisen liukulukuesitys	13
Kuva 2: Rust taulukko nimeltään varit, joka sisältää arvot ”punainen”, ”sininen” ja ”vihreä”	17
Kuva 3: Esimerkki taulukon indeksien käytöstä	17
Kuva 4: Esimerkki varit muuttujan indeksi paikan 0 arvon muuttamisesta.....	18
Kuva 5: Alkoiden lisääminen vektoriin	18
Kuva 6: Kurssin etusivu.	23
Kuva 7: Navigointipalkki.....	24
Kuva 8: Aloita tästä-osio.	24
Kuva 9: Materiaalit-osio.	25
Kuva 10: Esimerkki maanantain päiväkohtaisesta osiosta.....	26
Kuva 11: Esimerkki tiistain päiväkohtaisen tentin aloitussivusta.	26
Kuva 12: Esimerkki tiistain päiväkohtaisen tentin ensimmäisestä kysymyksestä.	27
Kuva 13: Esimerkki tiistain päiväkohtaisen tentin vastauksien yhteenvedosta.	27
Kuva 14: Edistymisen seuranta vimpain.....	27

1 JOHDANTO

Opinnäytetyön tavoitteena on toteuttaa Savonia-ammattikorkeakoululle verkkokurssi, jossa opetaan ohjelmoinnin perusteita osana TYKO-viikkoa. TYKO-viikko eli työelämä- ja korkeakouluviikko järjestetään Kuopion alueen lukiolaisille.

Savonian tietotekniikan ala tarjoaa lukio-opiskelijoille kurssin "Savonia Code Academy 2.0 intro – koodia on kaikkialla". Tämä kurssi kohdistetaan lukioiden toisen vuoden opiskelijoille, ja sen tarkoituksena on tutustuttaa heidät korkeakouluopiskeluun sekä kehittää heidän työelämätaitojaan. Kurssi koostuu viisipäiväisestä kokonaisuudesta, joka sisältää kaksi ohjattua live-päivää sekä kolme itsenäiseen opiskeluun tarkoitettua päivää. Kurssin tavoitteena on antaa opiskelijoille vahva perusta ohjelmointiosaamiseen sekä innostaa heitä jatkamaan aiheen opiskelua myös tulevaisuudessa. Samalla kehitetään opiskelijoiden itseohjautuvuutta ja valmiuksia käyttää digitaalisia oppimisympäristöjä, mikä on olennaista niin korkeakouluopinnoissa kuin työelämässäkin.

Koska kyseessä on lyhyt intensiivikurssi, kurssimateriaalissa keskitytään keskeisiin ohjelmointikäsitteisiin, kuten muuttujien hallintaan, ohjauksrakenteisiin, funktioihin sekä tietorakenteisiin. Viitekehyyksessä syvennytään näihin käsitteisiin tietojenkäsittelyn perusteiden kautta, jolloin muodostetaan teoreettinen pohja kurssin materiaalille ja luento-osuuksille. Teoria näkyy käytännössä kurssimateriaaleissa, Moodle-tenteissä, esimerkkikoodeissa sekä opetustilanteissa, joita järjestetään sekä luo- kassa että Zoomissa.

Teoriaosassa tarkastellaan, miten tieto esitetään tietokoneen muistissa perustietotyyppien näkökulmasta. Lisäksi perehdytään desimaalilukujen ja binäärilukujen väliseen suhteeseen sekä niiden merkitykseen ohjelmoinnissa. Koska satunnaislukujen tuottaminen ja erilaisten tietorakenteiden hyödyntäminen ovat keskeisiä ohjelmointitaidon osa-alueita, myös näiden periaatteita avataan viitekehyyksessä tarkemmin.

Kurssilla käytetään ohjelmointikielenä Rust-ohjelmointikieltä. Rustin käytön perusteluina olivat sen kasvava merkitys ohjelmointimaailmassa sekä mahdollisuus tarjota opiskelijoille uudenlaista näkökulmaa ohjelmoinnin perusteisiin verrattuna perinteisempiin opetuskieliin, kuten Python tai C#.

2 SAVONIA

Savonia-ammattikorkeakoulu aloitti toimintansa väliaikaisena ammattikorkeakouluna vuonna 1992 (Jääskeläinen 2012 ym., 11). Sen asema vakinaistettiin vuonna 1998 (Savonia 2022). Vuonna 2004 nimi muutettiin Pohjois-Savon ammattikorkeakoulusta Savonia-ammattikorkeakouluksi (Jääskeläinen 2012 ym., 18). Vuonna 2023 Savoniassa oli yhteensä 596 työntekijää, joista 303 oli opetushenkilöstöä, 173 työskenteli tutkimus-, kehitys- ja innovaatiotoiminnassa (TKI) ja 120 korkeakoulupalveluissa (HTP-henkilöstö) (Savonia-ammattikorkeakoulu Oy 2024, 16). Koulutusta tarjotaan neljällä kampuksella: Kuopiossa, Musiikkikeskuksella, Iisalmessa ja Varkaudessa (Savonia 2025a). Tällä hetkellä Savoniassa opiskelee yli 8000 opiskelijaa (Savonia n.d.).

Savoniassa on mahdollista opiskella kuutta eri alaa. Näihin kuuluvat liiketalouden ala, luonnonvara-ala, matkailu- ja ravitsemisala, muotoilun koulutus, musiikki ja tanssi, Sosiaali- ja terveysala, tekniikan sekä Savonian Master school:n ylemmät ammattikorkeakoulu-tutkinnot. (Savonia n.d.) Vuonna 2024 Savoniassa oli 48 AMK-tutkinto-ohjelmaa sekä 25 YAMK-tutkinto-ohjelmaa (Savonia 2025b). Tutkinnot voidaan suorittaa päivätoteutuksena, monimuotototeutuksena tai verkkototeutuksena ja kaikissa toteutusmuodoissa tutkinnon tavoitteet pysyvät samana (Savonia 2024.). Savoniasta valmistui 1374 AMK-tutkintoa ja 241 YAMK-tutkintoa 2024 vuoden aikana. 72,3 % opiskelijoista valmistuvat tutkinnon suunnitellussa aikataulussa. (Savonia 2025b)

Savonian strategian 2025–2028 mukaisesti korkeakoulu painottaa vastuullisuutta, vaikuttavuutta ja työelämäläheisyyttä koulutuksen ja tutkimuksen kehittämisessä (Savonian viestintä 2025, 4.). Savonia pyrkii laajentamaan sekä tutkintokoulutusta että TKI-toimintaa siten, että niiden vaikutuksia arvioidaan eri tasoilla. Tarkastelussa huomioidaan henkilöstön asiantuntemus, opiskelijoiden oppimispolut, työelämän tarpeet sekä laajempi yhteiskunnallinen merkitys (Savonian viestintä 2025, 8.) Savonia panostaa koulutuksen saavutettavuuteen, kansainvälisiin opiskelijoihin ja uudenlaisiin työelämäyhteyksiin, jotka tukevat alueellista elinvoimaa (Savonian viestintä 2025, 10.). Kokonaisturvallisuus ja vihreä siirtymä ovat keskeisiä teemoja, joiden avulla Savonia haluaa toimia kestäväen yhteiskunnan rakentajana ja innovaatiokeskuksena (Savonian viestintä 2025, 12.).

3 TYÖKALUT

3.1 Moodle

Moodle on avoimen lähdekoodin verkko-oppimisalusta, eli Learning Management System (LMS), jota käytetään laajasti eri puolilla maailmaa opetuksen tukena. Ohjelmisto on alun perin kehitetty tarjoamaan opettajille välineet luoda verkko-oppimiskokemuksia Internetin välityksellä. Moodle on GNU GPL -lisensoitu ja siksi vapaasti saatavilla: sen voi ladata, käyttää ja muokata maksutta. (Moodle Docs 2010)

3.1.1 Historia

Moodle sai alkunsa 2000-luvun taitteessa australialaisen kasvatusteknologian kehittäjän Martin Dougiamasin visiona. Dougiamas varttui kaukana lähimmästä koulusta ja perehtyi etäopetukseen jo lapsena radioyhteyksien kautta, mikä herätti halun hyödyntää Internetiä koulutuksen saavutettavuuden parantamiseksi. 1990-luvun lopulla Dougiamas työskenteli Curtin University of Technology -yliopistossa ja kohtasi tuolloin verkko-opetuksen työkaluissa rajoitteita – erityisesti kokemukset WebCT-alustasta olivat turhauttavia. (Moodle n.d.)

Ensimmäiset Moodlen prototyypit syntyivät Dougiamasin gradu- ja jatko-opintojen yhteydessä. Hän käytti aluksi työnimeä Moodle muodostettuna lyhenteestä ”Martin’s Object-Oriented Dynamic Learning Environment”. Myöhemmin lyhenteen merkitys muutettiin muotoon ”Modular Object-Oriented Dynamic Learning Environment”. (Moodle n.d.) Moodlen nimi tulee englannin kielestä ja se viittaa myös luovaan ja vapaamuotoiseen työskentelytapaan, jossa edetään ilman tarkkaa suunnitelmaa ja tehdään havaintoja matkan varrella. Tätä ajatusta myös Moodle pyrkii välittämään käyttäjilleen. (Moodle Docs 2010)

Vuoteen 2008 mennessä Moodlella oli jo yli puoli miljoonaa käyttäjää, ja Moodle oli vakiinnuttanut asemansa yhtenä maailman johtavista oppimisalustoista. Teknisesti Moodlea kehitettiin jatkuvasti vastaamaan kasvaviin vaatimuksiin. Vuonna 2010 julkaistiin merkittävä päivitys Moodle 2.0. Samoihin aikoihin Moodlen rekisteröityneiden käyttäjien määrä ylitti miljoonan käyttäjän rajapyykin. Kunnes vuonna 2015 Moodlen globaalit käyttäjämäärät nousivat jälleen uuteen virstanpylvääseen: rekisteröityneitä käyttäjiä oli jo 18 miljoonaa ja Moodlesta tuli tilastojen valossa maailman käytetyin LMS. 2020-luvulle tultaessa COVID-19-pandemia loi äkillisen tarpeen etäopetukselle ja samaan aikaan Moodlen käyttäjämäärä oli noussut jo noin 190 miljoonaan. (Moodle n.d.)

3.1.2 Pedagogiikka

Moodlen pedagogiikka perustuu sosiaaliseen konstruktionismiin, jonka mukaan oppiminen tapahtuu yhteisöllisesti, vuorovaikutuksessa muiden oppijoiden kanssa. Tämä pedagoginen lähestymistapa korostaa oppijoiden aktiivista roolia sekä toisiltaan oppimista, mikä ilmenee Moodlen tarjoamissa työkaluissa, kuten keskustelufoorumeissa, wikeissä ja tietokannoissa. Moodle pyrkii tukemaan joustavaa ja mukautuvaa oppimisympäristöä, jossa opettajat voivat muokata kurssirakennetta ja sisältöjä opiskelijoiden tarpeiden mukaisesti. Järjestelmä kannustaa oppimiseen tarjoamalla mahdollisuuksia ilmentää tietoa esimerkiksi tehtävien, projektien ja vertaisarvioinnin kautta. Oppiminen nähdään prosessina, jossa aktiivinen osallistuminen ja yhteisöllinen tiedonrakentaminen ovat keskeisiä. (Moodle Docs 2006)

3.1.3 Tekninen katsaus

Moodle 4.5 edellyttää vähintään PHP-versiota 8.1.0 sekä Sodium-laajennuksen käyttö on pakollista. Tietokantapalvelimista Moodle 4.5 tukee PostgreSQL:ää versiosta 13 alkaen, MySQL:ää versiosta 8.0 alkaen, MariaDB:tä versiosta 10.6.7 alkaen, Microsoft SQL Serveriä versiosta 2017 alkaen sekä Oracle Databasea versiosta 19c alkaen. (Moodledev 2024)

Moodle toimii kaikkien standardeja noudattavien verkkoselainten kanssa. Säännöllisesti testattuja selaimia ovat työpöytäympäristössä Chrome, Firefox, Safari ja Edge sekä mobiililaitteilla MobileSafari ja Google Chrome. (Moodledev 2024)

3.2 Rust Playground

Rust Playground on avoimen lähdekoodin projekti, joka tarjoaa selainpohjaisen käyttöliittymän Rust-kääntäjälle. Käyttöliittymä koostuu koodieditorista, joka tukee syntaksikorostusta. Koodin voi suorittaa joko debug- tai release-tilassa käyttäen Rustin nykyistä stable-, beta- tai nightly-versiota. Playgroundissa on saatavilla 100 ladattua crates.io-kirjastoa sekä Rust Cookbookin kirjastot ja kaikki näiden kirjastojen riippuvuudet. (Github n.d.) Koodia voidaan tallentaa ja jakaa GitHub Gistin kautta muille käyttäjille (Rust-lang n.d.).

Käyttöliittymä on rakennettu hyödyntäen React-kirjastoa, ja palvelinpuolella on käytetty Axum-nimistä web-frameworkia, jolla käsitellään käännös- ja suoritustehtävät. Jokainen ohjelmakäännös- ja suoritustehtävä ajetaan omassa Docker-kontissaan. Kontit on konfiguroitu rajoittamaan muistin käyttöä, suoritusaikaa ja verkkoyhteyksiä, millä pyritään ehkäisemään järjestelmän väärinkäyttöä. (Github n.d.)

4 TIEDON ESITYS TIETOKONEISSA

Tietojenkäsittelylaitteet käyttävät perusyksikkönä binäärijärjestelmän bittiiä tiedon esitykseen. Kahdeksan bitin binääriluvulle on annettu oma nimitys ja sitä kutsutaan tavuksi. (Jyväskylän yliopisto n.d.) Laitteet käsittelevät ja tallentavat tietoa käyttämällä näitä binäärisiä esityksiä. Binäärijärjestelmän käyttö mahdollistaa loogisten operaatioiden, kuten AND, OR ja NOT, suorittamisen. Nämä operaatiot perustuvat George Boolean kehittämään Boolean algebraan. Laite, joka käsittelee syötearvoja ja tuottaa niiden perusteella loogisen operaation lopputuloksen, tunnetaan loogisena porttina. Portteja voidaan toteuttaa monin eri tavoin, esimerkiksi mekaanisilla hammasratkaisuilla, releillä tai valoon perustuvilla järjestelmillä. Nykyiset tietokoneet käyttävät portteja, jotka ovat pieniä sähköisiä piirejä, joissa binääriset arvot 0 ja 1 esitetään erilaisina jännitetasoina. (Brooshear & Brylow 2015, 33.)

Loogisten operaatioiden toimintaa kuvataan totuustaulujen avulla. Totuustaulu esittää kaikki mahdolliset syötteiden yhdistelmät sekä niiden mukaiset ulostulot, mikä tekee operaatioiden käyttäytymisestä yksikäsitteisen. (Aalto Yliopisto 2004)

Taulukko 1 & 2. AND- ja OR-Operaattorien totuustaulu

AND-operaattori		
A	B	TULOS
0	0	0
1	0	0
0	1	0
1	1	1

OR-operaattori		
A	B	TULOS
0	0	0
1	0	1
0	1	1
1	1	1

Taulukko 3 & 4. XOR- ja NOT-operaattorien totuustaulu

XOR-operaattori		
A	B	TULOS
0	0	0
1	0	1
0	1	1
1	1	0

NOT-operaattori	
A	TULOS
1	0
0	1

4.1 Binääriluvut

Binäärijärjestelmä, jota kutsutaan myös 2-järjestelmäksi, sisältää kaksi arvoa: 0 ja 1. Näitä merkkejä kutsutaan biteiksi. Kun useita bitejä yhdistetään, niistä muodostuu binäärilukuja. (Jyväskylän yliopisto n.d.) Binääriluvun jokainen bitti edustaa tiettyä kahden potenssia sijaintinsa perusteella.

Taulukko 5: luvun 2 potenssit ja niiden arvot vähenevässä järjestyksessä

Bitti	5.bitti	4. bitti	3. bitti	2.bitti	1.bitti
Potenssi	2^4	2^3	2^2	2^1	2^0
Arvo	16	8	4	2	1

Binääriluku 11000110_2 voidaan tulkita seuraavasti:

Taulukko 6. Taulukossa esitetään 8-bittinen binääriluku ja sen desimaalivastineiden arvot

Bitti	1	1	0	0	0	1	1	0
Arvo	128_{10}	64_{10}	0	0	0	4_{10}	2_{10}	0

Yhteenlaskettuna binääriluku 11000110_2 vastaa desimaalilukua:

$$128_{10} + 64_{10} + 0 + 0 + 0 + 4_{10} + 2_{10} + 0 = 198_{10}$$

binääriluvun jokainen bitti edustaa tiettyä kahden potenssia, ja näiden arvojen summa muodostaa luvun kokonaisarvon. (Peda n.d.; Brooshear & Brylow 2015, 53-54.)

Desimaalijärjestelmän mukaan luku 198_{10} tulkitaan seuraavasti:

$$198_{10} = (1 * 10^2) + (9 * 10^1) + (8 * 10^0)$$

Desimaaliluvuissa jokainen paikka kertoo, kuinka monta kertaa kyseisen paikan arvo esiintyy luvussa. Oikeanpuoleisin numero (tässä tapauksessa 8) sijaitsee ykkösten paikalla (10^0) ja kertoo, montako yksikköä siinä on. Seuraava numero (9) sijaitsee kymmenien paikalla (10^1), joten se tarkoittaa yhdeksää kymmentä. Vastaavasti vasemmanpuoleisin numero (1) sijaitsee satojen paikalla (10^2), mikä tarkoittaa sataa. (Brooshear & Brylow 2015, 53-54.)

Desimaaliluvun muuntaminen binääriluvuksi voidaan tehdä jakamalla luku toistuvasti kahdella ja kirjaamalla jakojäännökset. Tämä prosessi jatkuu, kunnes osamäärä on nolla. Binääriluku muodostetaan lukemalla jäämät alhaalta ylöspäin.

$$\frac{198_{10}}{2} = 99, \text{jakojäännös } 0$$

$$\frac{99_{10}}{2} = 49, \text{jakojännös } 1$$

$$\frac{49_{10}}{2} = 24, \text{jakojännös } 0$$

$$\frac{24_{10}}{2} = 12, \text{jakojännös } 0$$

$$\frac{12_{10}}{2} = 6, \text{jakojännös } 0$$

$$\frac{6_{10}}{2} = 3, \text{jakojännös } 0$$

$$\frac{3_{10}}{2} = 1, \text{jakojännös } 1$$

$$\frac{1_{10}}{2} = 1, \text{jakojännös } 1$$

Lukemalla jäämät alhaalta ylöspäin saadaan binääriluku 11000110. Näin ollen desimaaliluku 198 vastaa binäärilukua 11000110. (Lahtonen 1998; Brooshear & Brylow 2015, 54.)

4.2 Negatiiviset binääriluvut

Tietotekniikassa on sovittu tapa esittää sekä positiivisia että negatiivisia lukuja binäärimuodossa varaamalla yksi bitti ilmaisemaan etumerkkiä. Tyypillisesti tämä on kaikkein vasemmanpuoleisin bitti, joka kertoo luvun olevan positiivinen, jos sen arvo on 0, ja negatiivinen, jos se on 1. (Oulun yliopisto, n.d.) Kahden komplementti (two's complement notation) on nykyajan tietokoneissa yleisimmin käytetty menetelmä. (Brooshear & Brylow 2015, 59.)

4.2.1 Yhden komplementti

Komplementti tarkoittaa binääriluvun vasta-arvoa eli sen muuttamista vastakkaiseksi arvoksi. Komplementointi voidaan tehdä sekä positiivisille että negatiivisille binääriluvuille, jolloin luvun etumerkki vaihtuu vastakkaiseksi. Yhden komplementin järjestelmässä luvun etumerkki määräytyy ensimmäisen bitin perusteella: 0 tarkoittaa positiivista ja 1 negatiivista lukua. Pelkkä etumerkkibitin vaihtaminen ei kuitenkaan riitä muuttamaan lukua vastakkaiseksi, vaan koko bittiesitys muunnetaan kääntämällä jokainen bitti päinvastaiseksi: 0 muuttuu 1:ksi ja 1 muuttuu 0:ksi. (Fenwick 2014, 30.)

Yhden komplementin etuna on, että positiivisten ja negatiivisten lukujen määrä jakautuu tasaisesti. Esimerkiksi yhden tavun lukujoukko on välillä -127 ja +127. Haittapuoli on kuitenkin se, että nolla voidaan esittää kahdella eri tavalla, mikä vaikeuttaa laskutoimituksia, että vertailuoperaatioita. (MOOC-keskus 2022)

Taulukko 7: Yhden komplementti 10110_2 Binääriluvusta

	4. bitti	3. bitti	2. bitti	1. bitti	0. bitti
Alkuperäinen arvo	1	0	1	1	0
komplementti arvo	0	1	0	0	1

4.2.2 Kahden komplementti

Kahden komplementissa positiiviset luvut esitetään tavallisessa binäärimuodossa. Negatiiviset luvut saadaan kääntämällä vastaavan positiivisen luvun bitit ja lisäämällä 1. Sama menetelmä toimii myös toiseen suuntaan: negatiivisen luvun itseisarvo saadaan kääntämällä sen bitit ja lisäämällä 1. (MOOC-keskus 2022) Vasemmanpuoleinen bitti toimii merkkibittinä, jossa arvo 0 tarkoittaa positiivista lukua ja arvo 1 negatiivista. (Oulun yliopisto, n.d.) Kahden komplementin etuna yhden komplementtiin verrattuna on se, että nolalla on vain yksi esitysmuoto, mikä selkeyttää muun muassa yhteenlaskutoimituksia. Tämä esitystapa ei kuitenkaan ole täysin ongelmaton, sillä negatiivisia lukuja on yksi enemmän kuin positiivisia. Esimerkiksi yhden tavun lukujoukko kattaa arvot välillä -128 ja $+127$. (MOOC-keskus 2022)

Esimerkki kahden komplementin muodostamisesta binääriluvulle:

Alkuperäinen arvo:

10110_2

Komplementoidaan alkuperäinen arvo:

01001_2

Lisätään $+1_2$:

$01001_2 + 1_2 = 01011_2$

Kahden komplementti:

01011_2

4.3 Liukuluvut

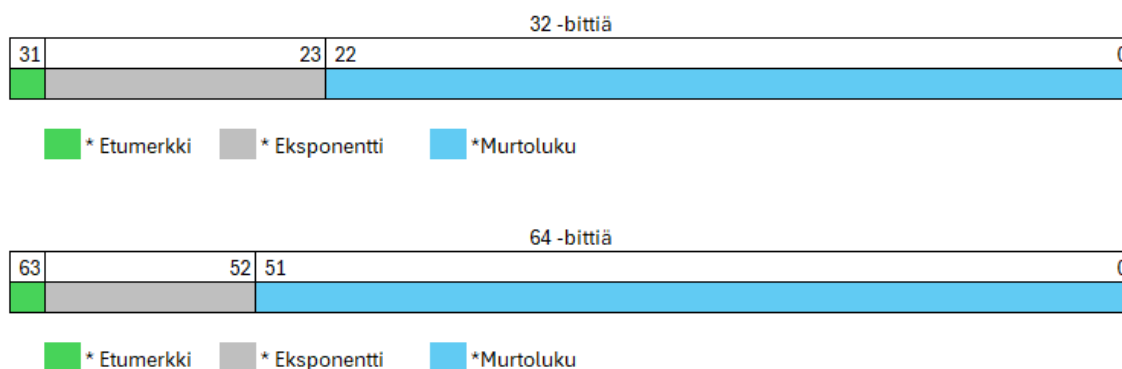
Liukuluvut ovat vain likimääräinen esitys reaalityyppisistä luvuista, ja niillä on sekä kokoon että tarkkuuteen liittyviä rajoituksia. Toisin kuin matemaattiset reaalityyppiset luvut, joiden välillä on aina äärettömän paljon muita lukuja, liukuluvuilla on vain rajallinen määrä mahdollisia arvoja. Esimerkiksi 32-bittisessä liukulukuformaattissa lukualueella $1,0-2,0$ on vain noin 16 miljoonaa eri arvoa, mikä tarkoittaa, että hyvin lähellä toisiaan olevat luvut voivat pyöristyä samaksi. Tämä voi aiheuttaa ongelmia tarkkuutta vaativissa laskuissa. (Fenwick 2014, 103.)

Liukuluvut esitetään binäärijärjestelmässä samalla tavalla kuin desimaaliluvut, mutta käyttäen binääristä pilkkua, jota kutsutaan radix-pisteeksi. Radix-pisteen oikealla puolella olevat numerot kuvaavat

murto-osia, joiden arvot pienenevät edetessä oikealle. Radix-pisteen oikealla puolella olevat numerot kuvaavat murto-osia, joiden arvot pienenevät edetessä oikealle. Ensimmäinen numero radix-pisteen jälkeen vastaa puolta ($\frac{1}{2}$), seuraava neljäsosaa ($\frac{1}{4}$), sitä seuraava kahdeksasosaa ($\frac{1}{8}$) ja niin edelleen, aina puoliintuen edelliseen verrattuna. Tämä tarkoittaa, että jokaisen sijainnin arvo on kaksi kertaa pienempi kuin edellinen. (Brooshear & Brylow 2015, 56.)

IEEE 754 -standardi määrittelee liukulukuaritmetiikan periaatteet ja on keskeinen tekninen standardi liukulukulaskennan kannalta. Sen kehitti Institute of Electrical and Electronics Engineers (IEEE), ja se vahvistettiin vuonna 1985. Tänä päivänä IEEE 754 -standardi on yleisimmin käytössä oleva tapa esittää reaali-lukuja tietokoneilla. Se on käytössä monissa eri järjestelmissä, kuten Intel-pohjaisissa PC-tietokoneissa, Applen Mac-laitteissa sekä monilla Unix-käyttöjärjestelmää hyödyntävillä alustoilla. (Geeksforgeeks.org 2020)

IEEE 754 -standardin mukaiset yksinkertaisen (32-bittinen) ja kaksinkertaisen (64-bittinen) tarkkuuden liukulukuesitykset koostuvat yhden bitin etumerkistä, eksponenttiosasta (8 bittiä yksinkertaisessa ja 11 bittiä kaksinkertaisessa muodossa) sekä murtolukuosasta (23 bittiä yksinkertaisessa ja 52 bittiä kaksinkertaisessa muodossa). (Oulun yliopisto, n.d.)



Kuva 1: 32-bittisen ja 64-bittisen liukulukuesitys

IEEE 754 -standardin mukaisessa 32-bittisessä liukulukuesityksessä pienin mahdollinen positiivinen normaalimuotoinen luku on noin 1.17549×10^{-38} . Vastaavasti suurin mahdollinen positiivinen luku 32-bittisessä muodossa on noin 3.40282×10^{38} . Jos laskutoimitus tuottaa suuremman luvun, tapahtuu ylivuoto (overflow), mikä voi johtaa äärettömyyteen (∞) tai muihin virhetilanteisiin. 64-bittisessä liukulukuesityksessä käytettävissä oleva lukualue on huomattavasti laajempi. Pienin esitettävä positiivinen normaalimuotoinen luku on noin 2.22507×10^{-308} , ja suurin mahdollinen luku on noin 1.79769×10^{308} . (Haataja ym. 2002, 395-396.)

4.4 Tekstin esittäminen

Teksti esitetään tietokoneissa käyttämällä koodeja, joissa jokaiselle merkkijärjestelmän symbolille annetaan yksilöllinen bittiesitys. Historiallisesti käytössä on ollut useita eri koodauksia, mikä aiheutti yhteensopivuusongelmia. Näiden ratkaisemiseksi kehitettiin American Standard Code for Information Interchange (ASCII), joka käyttää seitsemän bitin pituisia kuvioita esittämään englannin aakko-

set, numerot ja välimerkit. (Brooshear & Brylow 2015, 46.) Seitsemän bitin kuvoihin mahtuu yhteensä 128 erilaista merkkiä, joista suurin osa koostuu kirjoitusmerkeistä, kuten kirjaimista, numeroista ja välimerkeistä. Näiden lisäksi noin 32 merkkiä on varattu erityistarkoituksiin, kuten tietoliikenteen ohjaukseen ja perusmuotoiluun, kuten rivinvaihtoihin ja sarkaimiin. (Fenwick 2014, 136.) Myöhemmin ASCII laajennettiin kahdeksan bitin muotoon, mikä mahdollisti ylimääräisten merkkien lisäämisen, kuten eurooppalaisten kielten erikoismerkkien. (Brooshear & Brylow 2015, 46.) ASCII laajenuksen myötä merkkimäärä kasvoi 224:ään, kun 8-bittisen kooditilan yläpuolisko varattiin uusille merkeille (Fenwick 2014, 138.).

Taulukko 8: Esimerkkejä 8-bittisistä ASCII merkeistä

8-bittisiä ASCII Merkkejä			
Binääriluku	Kirjain	Binääriluku	Kirjain
01000001	A	01100001	a
01000010	B	01100010	b
01000011	C	01100011	c
01000100	D	01100100	d
01000101	E	01100101	e
00110000	0	00110001	1
00110010	2	00110011	3

ASCII ei kuitenkaan riitä kaikille kielille, erityisesti aasialaisille ja muille laajoille kirjoitusjärjestelmille. Tästä syystä on kehitetty Unicode standardi, joka voi käyttää jopa 21 bitin mittaisia koodeja ja mahdollistaa siten kaikkien maailman kirjoitusjärjestelmien kattavan tuen. Unicode Transformation Format 8-bit (UTF-8) säilyttää yhteensopivuuden alkuperäisen ASCII:n kanssa, sillä se esittää ASCII-merkit edelleen 8 bitillä, mutta laajentuu tarpeen mukaan 16, 24 tai 32 bitin koodeiksi monimutkaisempia merkistöjä varten. (Brooshear & Brylow 2015, 47) Vuonna 2024, Unicode 16.0.0 version myötä, standardi sisälsi 154 998 merkkiä (Unicode 2024).

5 SATUNNAISLUVUT OHJELMOINNISSA

Satunnaisuus tarkoittaa tapahtumia, jotka tapahtuvat ilman ennustettavaa kaavaa tai selkeää syy-seuraussuhdetta. Se liittyy läheisesti todennäköisyyteen ja sattumaan, ja sitä voidaan tarkastella eri näkökulmista, kuten subjektiivisena uskomuksena, tilastollisena ilmiönä tai objektiivisena mahdollisuutena. Arkipuheessa satunnaisuus ja sattuma saatetaan nähdä samana asiana, mutta filosofisesti ne eivät vastaa toisiaan. Jopa deterministisessä järjestelmässä, jossa kaikki tapahtumat ovat periaatteessa ennalta määrättyjä, voi ilmetä satunnaisuutta, jos lopputulokset ovat monimutkaisia tai mahdottomia ennustaa. (Stanford Encyclopedia of Philosophy 2010)

Satunnaislukuja käytetään monilla eri aloilla monipuolisesti. Niitä hyödynnetään esimerkiksi simulaatioissa, joissa tarvitaan realistisia satunnaisilmiöitä, sekä otantatutkimuksissa, joissa ne auttavat valitsemaan edustavan joukon suuresta aineistosta. Lisäksi satunnaislukuja käytetään numeerisessa analyysissä, tietokoneohjelmoinnissa, päätöksenteossa, Kryptografiassa sekä taiteellisissa sovelluksissa, kuten tietokonegrafiikassa ja musiikissa. Myös viihteessä, kuten arpajaisissa ja peleissä, satunnaisuus on keskeinen osa kokemusta. (Knuth 1997, 1-2.)

Koska tietokoneet toimivat tarkasti annettujen ohjeiden mukaan, niiden on vaikea tuottaa sattumanvaraisuutta. Sattuman luomiseen on kehitetty kaksi tapaa – näennäissatunnaislukugeneraattorit ja aidot satunnaislukugeneraattorit. (Random n.d.)

5.1 Näennäissatunnaislukugeneraattorit

Näennäissatunnaislukugeneraattorit (Pseudo-random number generator, PRNG) ovat algoritmeja, joilla voidaan luoda satunnaislukuja deterministisesti. Ne tuottavat lukujonon, joka näyttää tilastollisesti satunnaiselta ja jakautuu tasaisesti välillä 0 ja 1. Vaikka luvut vaikuttavat satunnaisilta, ne syntyvät matemaattisen kaavan mukaisesti. (Luengo, 2022, 144.)

Hyvän näennäissatunnaislukugeneraattorin (PRNG) tulee täyttää tietyt keskeiset vaatimukset. Ensinnäkin sen on tuotettava lukuja, jotka tilastollisesti käyttäytyvät kuin aidot satunnaisluvut. Generaattorin tulee pystyä tuottamaan mahdollisimman pitkä sekä toistumaton lukujono. Tehokkuuden näkökulmasta PRNG:n tulee olla laskennallisesti nopea ja vaatia vähäistä muistinkäyttöä. (Luengo, 2022, 145.)

Mersenne Twister on tunnetuimpia Näennäissatunnaislukugeneraattoreita, jonka kehittivät Makoto Matsumoto ja Takuji Nishimura vuonna 1997 (asecuritysite 2025). Se perustuu 19 937 bitin siemenarvoon, joka säilytetään 624 alkion taulukossa, missä jokainen arvo on 32-bittinen. Algoritmin nimi juontaa juurensa sen erittäin pitkään jaksoon, joka on $2^{19937} - 1$, mikä on Mersennen alkuluku. (how n.d.)

5.2 Aidot satunnaislukugeneraattorit

Aidot satunnaislukugeneraattorit (True Random Number Generators, TRNG) ovat järjestelmiä, jotka tuottavat satunnaislukuja reaali maailman ilmiöistä kerätyn datan avulla. Tietokone voi tallentaa esi-

merkiksi näppäimistön painalluksia, hiiren liikkeitä, verkkoliikenteen viiveitä tai keskeytysten ajoituksia, ja näiden perusteella se tuottaa satunnaisia bittijonoja. Nämä bittikuviot kootaan entropia-altaaseen, josta niitä voidaan käyttää satunnaisuuden lähteenä. (Aalto yliopisto n.d.)

Entropia-allas on tietorakenne, johon tietokone kerää epädeterministisiä tietoja, kuten hiiren liikkeitä ja näppäinpainalluksia, satunnaislukugeneraattorin siemenenä käytettäväksi (Hackaday 2017). Entropia-altaan täyttymiseen vaikuttaa se, kuinka paljon tietokone pystyy keräämään reaali maailman vaihtelua. Jos tietokoneessa ei tapahdu tarpeeksi vaihtelua, entropia-altaan täyttäminen voi olla hidasta, ja jos bittejä kulutetaan nopeammin kuin niitä kertyy, allas voi ehtyä. Tämä voi muodostua ongelmaksi esimerkiksi virtuaaliympäristöissä, kuten pilvipalveluissa, joissa ei ole suoraa pääsyä fyysisiin ilmiöihin. (Aalto yliopisto n.d.)

6 TIETORAKENTEET

Tietorakenteet ovat keskeinen osa tietojenkäsittelytiedettä ja ohjelmistokehitystä, sillä ne määrittelevät tavat, joilla data organisoidaan, tallennetaan ja käsitellään tehokkaasti. Ne koostuvat erilaisista rakenteista, kuten taulukoista, linkitetyistä listoista, pinoista ja hajautustauluista. Oikean tietorakenteen valinta vaikuttaa algoritmien suorituskykyyn ja ohjelmistojen tehokkuuteen. (IBM 2024)

6.1 Taulukot

Taulukko (array) on yksi yleisimmistä tietorakenteista ohjelmoinnissa. Se on kokoelma saman tyyppisiä alkioita. Yksiulotteinen taulukko on yksinkertainen rivi alkioita, joissa jokainen alkio on yksilötävissä indeksin perusteella. Taulukon voidaan ajatella myös laajenevan moniulotteiseksi rakenteeksi, kuten kaksiulotteiseksi matriisiksi, jossa indekseillä viitataan riveihin ja sarakkeisiin (Brooshear & Brylow 2015, 284.) Taulukoiden perusominaisuuksia ovat kiinteä koko ja tehokas satunnaishaku (random access) indeksin avulla. Taulukon rakenne heijastaa tietokoneen muistin järjestystä, minkä ansiosta sitä voidaan käyttää pohjana muiden tietorakenteiden toteutuksessa (Laaksonen 2022, 4.).

Taulukko 9: Yksiulotteinen taulu, jossa alkiot 0:sta 5:een muodostavat sanan "TERVE"

Index:	0	2	3	4	5
Alkio:	"T"	"E"	"R"	"V"	"E"

Rustin taulukot ovat kiinteäkokoisia tietorakenteita, joissa elementtien tyyppi ja määrä on määriteltävä jo kääntämisen yhteydessä. Ne esitetään muodossa [T; N], jossa T edustaa alkioden tyyppiä ja N taulukon kokoa. Koska koko on ennalta määrätty, Rustin taulukot eivät tue dynaamista laajentamista. (GeeksforGeeks 2024)

Uuden taulukon voi luoda seuraavasti (Rust-lang 2025):

```
let mut varit = ["punainen", "sininen", "vihreä"];
```

Kuva 2: Rust taulukko nimeltään varit, joka sisältää arvot "punainen", "sininen" ja "vihreä"

Taulukon arvoja voi käyttää, kutsumalla muuttujan nimeä ja indeksi numeroa hakasulkujen sisällä (Rust-lang 2025).

```
println!("Ensimmäinen väri: {}", varit[0]);
println!("Toinen väri: {}", varit[1]);
println!("Kolmas väri: {}", varit[2]);
```

Kuva 3: Esimerkki taulukon indeksien käytöstä

Taulukon indeksin arvoa voi muuttaa, mikäli alkuperäinen muuttuja on asetettu muutettavaksi "mut" sanalla ja arvo vastaa taulun tyyppiä (Rust-lang 2025).

```
varit[0] = "musta";
```

Kuva 4: Esimerkki varit muuttujan indeksi paikan 0 arvon muuttamisesta

6.2 Vektorit Rustissa

Rustin vektorit (`Vec<T>`) ovat dynaamisesti muunneltavissa olevia tietorakenteita, jotka säilyttävät saman tyyppiset arvot jatkuvana tietosarjana muistissa. Vektorien koko voi muuttua ajonaikaisesti, mikä erottaa ne kiinteäkokoisista taulukoista. (The Rust Programming Language n.d.)

Uuden vektorin voi luoda joko tyhjänä `Vec::new()`-funktioilla tai käyttämällä `vec!`-makroa, joka mahdollistaa alkioiden määrittämisen suoraan. Esimerkiksi `let v = vec![1, 2, 3];` luo vektorin, joka sisältää kokonaisluvut 1, 2 ja 3. Rust pystyy päättämään vektorin alkioiden tyyppin automaattisesti annettujen arvojen perusteella. (The Rust Programming Language n.d.)

Vektoriin voi lisätä alkioita `push`-metodilla, mutta jotta vektorin arvoa voi muuttaa, sen on oltava muuttuva. (The Rust Programming Language n.d.)

```
let mut v = Vec::new();
v.push(5);
v.push(10);
```

Kuva 5: Alkoiden lisääminen vektoriin

Vektorien arvoja voidaan lukea joko indeksoinnin (`v[i]`) tai `get`-metodin avulla. Indeksointi palauttaa suoran viitteen arvoon, mutta jos indeksi ylittää sallitun rajan, ohjelma kaatuu (`panic`). `get`-metodi puolestaan palauttaa `Option<T>`-tyypin, joka mahdollistaa turvallisen virheen käsittelyn. (The Rust Programming Language n.d.)

Rustin muistinhallinnan ja lainauksen säännöt vaikuttavat vektorien käsittelyyn. Jos esimerkiksi pidetään viitettä vektorin alkioon ja yritetään lisätä uusi alkio `push`-metodilla, Rust estää tämän, koska vektori saattaa varata uuden muistialueen ja siirtää olemassa olevat alkiot uuteen paikkaan. Tämä voisi johtaa virheellisiin viitteisiin. (The Rust Programming Language n.d.)

7 TOTEUTUS

7.1 Lähtökohdat ja tavoitteet

"Savonia Code Academy 2.0 Intro"-kurssi toteutettiin osana Kuopion alueen lukioiden TYKO-viikkoa, joka järjestettiin 7.–11.4.2025. TYKO-viikon (työelämä- ja korkeakouluviikko) tarkoituksena on tutustuttaa lukiolaiset työelämään ja korkeakouluopintoihin. Tämän kurssin kohderyhmänä olivat toisen vuoden lukiolaiset, jotka osallistuivat TYKO-viikon korkeakouluopintoihin. Kurssin laajuus oli yksi opintopiste (1 op). Se tarjosi opiskelijoille mahdollisuuden saada ensikosketuksen ohjelmointiin sekä tutustua ammattikorkeakouluopiskeluun Savoniassa.

Kurssi toteutettiin hybridimallilla, joka sisälsi sekä etä- että lähiopetuspäiviä. Tällä järjestelyllä pyrittiin joustavasti vastaamaan opiskelijoiden erilaisiin oppimistarpeisiin ja tarjoamaan mahdollisuus tutustua Savonia-ammattikorkeakoulun tiloihin lähiopetuspäivien aikana.

Kurssin suunnitteluvaiheessa Savonia-ammattikorkeakoulun kone-, sähkö- ja ICT-alan koulutuspäällikkö Pasi Liimatainen ehdotti kurssin toteuttamista Rust-ohjelmointikielellä. Rust valikoitui kurssin ohjelmointikieleksi erityisesti sen vahvan tyyppityksen vuoksi, sen katsottiin olevan olennaista ohjelmoinnin perusteiden opiskelussa. Se auttaa opiskelijoita jo alusta lähtien omaksumaan hyviä ohjelmointikäytäntöjä ja välttämään yleisiä virheitä. Rust on moderni ja kysytty kieli, joten sen oppimisesta on hyötyä opiskelijoiden tulevaisuuden kannalta. Kurssin toteuttaminen Rustilla antoi myös minulle mahdollisuuden kehittää opetustaitojani ja syventää omaa osaamistani kyseisessä ohjelmointikielessä.

7.2 Teorian hyödyntäminen kurssikokonaisuudessa

Kurssin suunnittelun ja toteutuksen tukena hyödynnettiin opinnäytetyön aikana tutkittua teoriaa, joka toimi sisällöllisenä ja pedagogisena perustana useille opetuskokonaisuuksille. Teoreettisen viitekehityksen tarkoituksena oli syventää opetettavia aiheita sekä tarjota opiskelijoille taustatietoa ohjelmoinnin ilmiöistä ja teknisistä yksityiskohdista, jotka eivät aina tule esiin pelkän käytännön koodauksen kautta. Teoriaa esitettiin opiskelijoille sekä kirjallisessa materiaalissa että luentojen aikana suullisesti, keskustelun ja esimerkkien lomassa. Kaikkea teoriaa ei kuitenkaan käsitelty jokaisessa muodossa, vaan esitystapa valittiin kulloinkin aiheen ja opetustilanteen mukaan.

Esimerkiksi tietotyyppejä käsiteltäessä teoriaa hyödynnettiin havainnollistamaan, miksi kahden kokonaisuuden jakaminen ei aina tuota odotettua tulosta ilman sopivaa tietotyyppiä, kuten liukulukua. Tämä avasi tilaisuuden keskustella tietotyyppien roolista ohjelmoinnissa ja siitä, miten ne vaikuttavat laskennan tarkkuuteen ja ohjelman käyttäytymiseen.

Rustin tietotyyppien, kuten $u32$ ja $i32$, esittely tarjosi mahdollisuuden käsitellä positiivisten ja negatiivisten lukujen käsittelyä ohjelmointikielessä. Vaikka kahden komplementin binaarijärjestelmää ei varsinaisesti käsitelty, siihen liittyvää taustaa oli mahdollista ammentaa opetuksen valmistelussa. Näin teoria tuki opetusta myös silloin, kun oppitunneilla pysyteltiin abstraktiotasolla.

Satunnaislukujen käsittelyn yhteydessä teoria tarjosi luontevan mahdollisuuden esitellä satunnaisuuden merkitystä ohjelmoinnissa. Luennoilla tuotiin esiin ero pseudorandom- ja aidosti satunnaisten lukujen välillä, sekä pohdittiin, millaisissa sovelluksissa satunnaisuus on keskeisessä roolissa – esimerkiksi pelien logiikassa tai tietoturvasovelluksissa.

Tietorakenteiden osalta päädyttiin pitämään tarkastelu selkeästi Rust-ohjelmointikielen tarjoamissa rakenteissa. Tämä ratkaisu tehtiin ennen kaikkea käytännön syistä: yleinen tietorakenneteoria olisi vienyt opetuksen kurssin tavoitteista liian kauas ja Rustin tarjoamien rakenteiden avulla teoria pystyttiin sitomaan suoraan kurssin käytännön harjoituksiin.

7.3 Kurssin toteutunut aikataulu

Kurssi toteutettiin viikon mittaisena kokonaisuutena maanantaista perjantaihin. Jokainen opetuspäivä kesti klo 9.00–15.00 ja noudatti yhtenäistä rakennetta, jossa aamupäivisin käytiin läpi teoriaa ja päivän päätteeksi opiskelijat tekivät itsenäisesti harjoitustehtäviä sekä suorittivat päivän Moodle-tentin.

Savonia määritteli viikon yleisen aikataulurungon, jota kurssien vetäjät täydensivät oman opintojaksonsa tarpeiden mukaisesti. Seuraavassa taulukossa esitetään kurssin toteutunut aikataulu päivätasolla:

Taulukko 10: Viikon opetussuunnitelma

Maanantai 7.4.2025	Tiistai 8.4.2025	Keskiviikko 9.4.2025	Torstai 10.4.2025	Perjantai 11.4.2025
Savonia-AMK	ETÄNÄ	ETÄNÄ	Savonia-AMK	ETÄNÄ
Klo 9.00 TYKO-viikon aloitus tervetuloa! Savonia-AMK Kampusydän (microkati 1) TAI Musiikkikeskus Tervetuloa - Savonian esittely - Jalkautuminen ryhmiin ja siirtyminen luokkatiloihin opintojakson vastuuopettajan johdolla - Läsnaolijoiden tarkistus - Työskentelyn aloitus Savonian ohjeiden ohjeiden mukaan	klo 9.00-10.00 Luento Zoomissa 10.00-11.00 Itsenäistä opiskelua	klo 9.00-10.00 Luento Zoomissa 10.00-11.00 Itsenäistä opiskelua	klo 9.00-11.00 Opiskelua opintojakson ohjelman mukaisesti	klo 9.00-10.00 Luento Zoomissa 10.00-11.00 Itsenäistä opiskelua
PAIKKA: SAKKY klo 11.00- 12.00 Ruokailu	klo 11.00- 12.00 Ruokailu	klo 11.00-12.00 Ruokailu	PAIKKA: SAKKY klo 11.00- 12.00 Ruokailu	klo 11.00-12.00 Ruokailu

Työskentely jatkuu kampuksella Savonian ohjeiden mukaan	klo 12.00-15.00 Itsenäistä opiskelua	klo 12.00-15.00 Itsenäistä opiskelua	klo 12.00-14.00 Opiskelua opintojakson ohjelman mukaisesti Klo 14.00-15.00 yhteinen ohjelma Kampusydämessä	klo 12.00-15.00 Itsenäistä opiskelua TYKO-viikko päättyy
---	--------------------------------------	--------------------------------------	---	---

7.4 Toteutusympäristön valinta ja perusteet

Kurssin varsinaisena opintoympäristönä käytetään Savonian oletuksena tarjoamaa Moodlea, joka toimii keskeisenä alustana kurssimateriaalien jakelussa, tehtävien hallinnassa sekä opiskelijoiden etenemisen seurannassa.

Harjoitustehtävien ja esimerkkikoodien toteutusympäristöksi valittiin Rust Playground. Tämä valinta perustui ennen kaikkea käytännöllisiin syihin: Rust Playground on helppokäyttöinen, eikä sen käyttö vaadi opiskelijoilta erillisten ohjelmistojen asentamista. Rust Playground mahdollistaa valmiiden koodiesimerkkien ja tehtävien jakamisen kätevästi linkkien kautta, mikä tukee joustavaa opetusta ja materiaalien saavutettavuutta. Käyttämällä Rust Playgroundia vältetään mahdolliset yhteensopivuusongelmat ja muut tekniset haasteet, jotka voisivat haitata opiskelijoiden oppimiskokemusta tai häiritä kurssin aikataulussa pysymistä.

7.5 Opetusmateriaalin toteutus

7.5.1 Luentomateriaali

Kurssin luentomateriaali laadittiin luentokalvoiksi Savonian Powerpoint-pohjaa käyttämällä ja materiaalia syntyi yhteensä 139 kalvoa. Kalvojen sisältö koostui teoriaosuuksista, aihetta havainnollistavista kuvista sekä yhteensä 23 käytännön esimerkistä. Esimerkit toteutettiin Rust Playground -ympäristössä, mikä mahdollisti niiden jakamisen opiskelijoille helposti linkkien avulla. Jokainen käytännön esimerkki sisälsi koodin lisäksi kattavat kommentit, joiden avulla opiskelijat voivat seurata esimerkkien toimintalogiikkaa ja ymmärtää teoriaa paremmin käytännön sovellusten kautta.

Luentomateriaalien aihealueet jaettiin päivittäin selkeisiin kokonaisuuksiin, jotta opiskelijoiden oppiminen olisi mahdollisimman johdonmukaista ja aihepiirit erottuisivat toisistaan. Alla olevassa taulukossa on kuvattu päivittäiset teemat ja niiden yhteydessä käsitellyt keskeiset aiheet.

Taulukko 11: Päivän aiheet

PÄIVÄ	AIHE
Maanantai	<ul style="list-style-type: none"> • Mitä on ohjelmointi? • Muuttujat • Perustietotyypit • String ja &str • Tekstin tulostaminen
Tiistai	<ul style="list-style-type: none"> • Perusoperaattorit • Vertailuoperaattorit • Loogiset operaattorit • Ehdolliset lauseet
Keskiviikko	<ul style="list-style-type: none"> • Toistolauseet (for ja while-silmukka) • Taulukot • Vektorit
Torstai	<ul style="list-style-type: none"> • Oman syötteen käsittely • Funktiot
Perjantai	<ul style="list-style-type: none"> • Match-rakenne • Satunnaisnumerot

Päivittäisten aiheiden valinnassa hyödynnettiin Savonian C#-ohjelmoinnin perusteet -kurssin luentorakennetta sekä Rustin virallista dokumentaatiota, "Rust bookia" (<https://doc.rust-lang.org/book/title-page.html>). Suunnittelun keskeisenä tavoitteena oli varmistaa, että valitut aiheet kattavat ohjelmoinnin perusteiden keskeisimmät osa-alueet ja muodostavat johdonmukaisesti etenevän oppimiskokonaisuuden.

7.5.2 Harjoitustehtävät

Kurssin aikana laadittiin yhteensä 23 harjoitustehtävää, joiden tarkoituksena oli auttaa opiskelijoita syventämään luennoilla käsiteltyä teoriaa käytännön kautta. Harjoitustehtävät julkaistiin Moodle-ympäristössä selkeästi päiväkohtaisen materiaalin yhteydessä, minkä lisäksi tehtävien pohjat olivat saatavilla päivän viimeisillä luentokalvoilla.

Harjoitustehtävät toteutettiin asteittain vaikeutuvina. Päivän ensimmäiset tehtävät suunniteltiin helpommiksi, jotta opiskelijat pääsisivät nopeasti soveltamaan juuri opittua teoriaa käytännössä. Päivän viimeiset tehtävät olivat haastavampia ja vaativat opiskelijoilta aikaisemmin käsitellyn tiedon soveltamista sekä laajempaa ongelmanratkaisukykyä.

7.5.3 Tentit ja kurssin arviointi

Kurssin varsinaisena arviointimenetelmänä käytettiin päiväkohtaisia Moodle-tenttejä, jotka mittasivat päivän teoriaosuuden ymmärtämistä. Tenteissä hyödynnettiin monipuolisia kysymystyypppejä, kuten monivalintakysymyksiä, tosi/epätosi-väittämiä sekä yhdistämistehtäviä.

Kurssin hyväksytyt suorittaminen edellytti, että opiskelija sai jokaisesta päiväkohtaisesta tentistä täydet pisteet. Tenttien suoritusyrityksiä ei kuitenkaan rajoitettu, vaan opiskelijat saivat yrittää tenttejä uudelleen rajattomasti. Tämä ratkaisu tehtiin oppimista ja motivaatiota tukevaksi valinnaksi.

Moodle-tentteihin päädyttiin, koska vaihtoehtona ollut harjoitustehtävien automaattinen testaus osoittautui teknisesti hankalaksi toteuttaa Rust Playgroundissa. Lopullinen päätös valita Moodle-tentit arviointimenetelmäksi nähtiin opiskelijaystävällisenä ratkaisuna, joka mahdollisti opiskelijoille enemmän vapautta kokeilla ja tutkia koodia ilman automaattisen testauksen rajoitteita. Tämä ratkaisu nähtiin pedagogisesti onnistuneena.

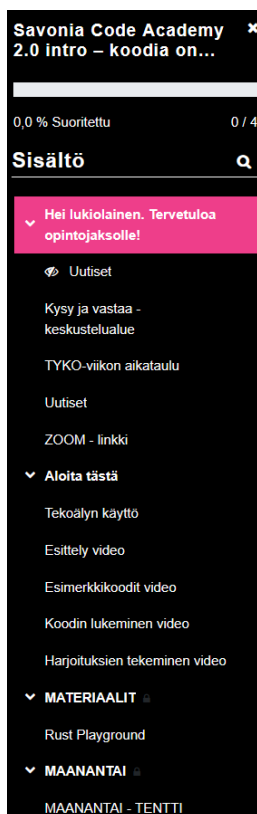
7.6 Moodle-rakenne

Käyttäjän saapuessa kurssin etusivulle, hänelle aukeaa valkea pohjainen esittelysivu, jolla kerrotaan, mistä kurssista on kyse ja kuinka kurssin rakenne etenee. Samasta näkymästä löytyy myös kätevästi opiskelijalle päiväkohtainen kalenteri, QA-keskustelualue sekä Zoom-linkki etäluentoja varten.

The screenshot shows the Moodle course homepage. The top navigation bar includes the Savonia logo, 'ETUSIVU TYÖPÖYTÄ', a search bar, and user profile 'JR'. The left sidebar contains a course overview with '0,0 % Suoritettu' and a table of contents. The main content area features a large header image of students, followed by the course title and a welcome message. Below the message, there is a brief description of the course and a 'Uutiset' section with a 'Pilotoitu opiskelijoilla' tag.

Kuva 6: Kurssin etusivu.

Sivun vasemmasta laidasta löytyy navigointipalkki, josta löytyvät kurssisivun kaikki osiot. Pääosiot ovat: "Hei lukiolainen, Tervetuloa opintojaksolle!", "Aloita tästä", "Materiaalit", "Maanantai", "Tiistai", "Keskiviikko", "Torstai" sekä "Perjantai".



Kuva 7: Navigointipalkki.

Kurssin etusivulla opiskelijaa ohjeistetaan tutustumaan "Aloita tästä" -osioon. Tässä osiossa esitellään yhteenveto kurssin sisällöstä ja viikkoteemoista päiväkohtaisesti, sekä annetaan ohjeet Moodlen käyttöön mahdollisten ongelmatilanteiden varalta. Lisäksi kerrotaan, kuinka kurssista saa suorituserkinnän. Osion lopussa on myös Esittelyvideo, esimerkkikoodista video ja video koodin lukemisen sekä harjoituksen tekemisen tueksi, mikäli opiskelija kaipaa lyhyttä virkistystä etäpäivinä.

SAVONIA ETUSIVU TYÖPÖYTÄ Etsi kurseja JR Muokkauksia

Savonia Code Academy 2.0 intro – koodia on... 0,0 % Suoritettu 0 / 4

Sisältö

- Hei lukiolainen. Tervetuloa opintojaksolle!
 - Uutiset
 - Kysy ja vastaa - keskustelualue
 - TYKO-viikon aikataulu
 - Uutiset
 - ZOOM - linkki
 - Aloita tästä**
 - Tekoälyn käyttö**
 - Esittely video
 - Esimerkkikoodit video
 - Koodin lukeminen video
 - Harjoituksien tekeminen video
 - MATERIAALIT
 - Rust Playground
 - MAANANTAI
 - MAANANTAI - TENTTI

Aloita tästä

Tervetuloa mukaan

Tämän kurssin jälkeen osaat

- ymmärtää ohjelmoinnin peruskäsitteet
- käyttää Rust-ohjelmointikieltä perustoinnoissa
- soveltaa opittuja taitoja käytännön ohjelmointitehtävissä

Kurssin osiot

- Maanantai:**
 - Kurssin aloitus: Kampussydän klo 9.00
 - Käydään läpi ohjelmoinnin käsitteet, muuttujat, tietotyypit (mm. String ja &str) sekä tekstin tulostus.
- Tiistai:** Opetellaan perus-, vertailu- ja loogiset operaattorit sekä ehdolliset lauseet.
- Keskiviikko:** Harjoitellaan toistolauseita, for- ja while-silmukoita, taulukoita ja vektoreita.
- Torstai:** Opitaan lukemaan käyttäjän syöte, käsittelemään tekstiä ja numeroita sekä kirjoittamaan funktioita.
- Perjantai:** Käydään läpi match-rakenne ja satunnaislukujen käyttö.

Mistä apua?

Jos tarvitset apua opiskeluun liittyvissä asioissa, vilkaise ensin Kysy ja vastaa -palsta. Siellä saattaa joku olla kysynyt samaa asiaa jo aiemmin! Voit myös kysyä apua muilta oppijoilta.

Jos et löydä vastausta kysymyksiisi näistä ohjeista, tehtävänannoista tai Kysy ja vastaa -keskustelualueelta, niin voit olla yhteydessä kurssin vastuupet-tajaan tai teknisissä ongelmatilanteissa Savonian Service Deskiin Sanni-palvelun kautta.

Kurssin sisältöön ja oppimiseen liittyvät asiat: etunimi.sukunimi@savonia.fi
Tekniset ongelmat oppimisympäristössä: <http://sanni@savonia.fi>

Opintopisteet

Kun olet tehnyt hyväksytysti kaikki opintojakson suorittamiseksi vaaditut tehtävät, saat opintopisten. Näet kaikki vaaditut arvioitavat tehtävät opintojakson kurssialueen oikean reunan lohkovalikon kautta.

Opettaja kirjaa opintosuorituksesi rekisteriin kuukauden kuluessa opintojakson päättymisestä. Näet kirjatut opintosuorituksesi OmaOpintopalku - palvelussa (vaatii kirjautumisen) osoitteessa <https://opintopalku.fi/oma-opintopalku>. OmaOpintopalkusta voit jakaa tiedon suorituksesta esimerkiksi omalle nro:lle. Omatietojien kirjautumisessa Omatietojen voi olla viivettä. Sivutka nro:lle näet huti nro:lle nähtävää

Kuva 8: Aloita tästä-osio.

Materiaalit-osiossa kurssin luentokalvot ovat saatavilla PDF-tiedostoina, esimerkkikoodit sekä harjoitustehtävien vastaukset ZIP-tiedostona, ja mukana on myös linkki Rust Playgroundiin.

The screenshot shows the 'MATERIAALIT' section of a course page. At the top, there's a search bar and navigation icons. Below the header, a text block explains that materials like lecture slides, code examples, and exercise solutions are available. A date indicator shows 'Saatavilla 7. huhtikuu 2025 alkaen'. The main content area lists three folders: 'Luentokalvot' (Lecture slides), 'Koodit' (Code), and 'Rust Playground'. Each folder has a 'Lataa kansio' button. The 'Luentokalvot' folder contains 'Ohjelmointikurssin luentokalvot.pdf'. The 'Koodit' folder contains 'Esimerkkikoodit', 'Esimerkkikoodit.zip', and 'Harjoitustehtävien vastaukset', with the latter containing 'Harjoitustehtävä vastaukset.zip'. The 'Rust Playground' folder is also listed.

Kuva 9: Materiaalit-osio.

Kurssi sisältävää jokaiselle viikonpäivälle oman osionsa maanantaista perjantaihin. Näissä osioissa on lyhyt kuvaus päivän aiheista sekä sivunumerot, joiden avulla opiskelijat löytävät kyseisen päivän sisällön ja siihen liittyvät harjoitustehtävät luentokalvoista. Osiot avautuvat vasta kurssin aloituspäivänä. Jotta tentti olisi opiskelijoille helposti löydettävissä, päiväkohtainen tentti on sijoitettu ensimmäiseksi linkiksi ja se avautuu vasta kyseisenä päivänä klo 12.00. Tenttiä seuraavat linkit esimerkkikoodeihin ja lopuksi linkit harjoitustehtäviin.

The screenshot shows the 'MAANANTAI' section of the course page. At the top, there's a navigation menu with 'Kurssi', 'Asetukset', 'Osallistujat', 'Arvioinnit', 'Raportit', and 'Lisää'. Below the header, a text block describes the Monday session's focus on deepening programming fundamentals, covering String and &str, and using exercise solutions. It specifies that the lecture slides are on pages 15-48 and exercise solutions are on slide 48. A date indicator shows 'Saatavilla 7. huhtikuu 2025 alkaen'. Below this, a list of links is provided: 'MAANANTAI - TENTTI' (with 'Saada arvosana' and 'Saada hyväksytty arvosana' options) and five example links for 'MAANANTAI - esimerkki1' through 'MAANANTAI - esimerkki5'.

Kuva 10: Esimerkki maanantain päiväkohtaisesta osiosta.

Kun tentti avautuu, opiskelija voi aloittaa sen painamalla "Tee tentti" -nappia. Tenttien suorittamisessa ei ole rajoituksia, vaan niitä voi tehdä niin monta kertaa kuin tarpeen.

The screenshot shows the course page for 'TIISTAI - TENTTI' on the Savonia Code Academy platform. The page is in Finnish and displays the following information:

- Course Info:** Työpöytä / Kurssit / Peppi / 2024-12 / JA00BS63-3001 / TIISTAI / TIISTAI - TENTTI
- Progress:** 0,0 % Suoritettu (0 / 1)
- Content:** A list of course materials including 'keskustelualaue', 'TYKO-viikon aikataulu', 'Uutiset', 'ZOOM - linkki', and 'Aloita tästä' (Tekoälyn käyttö, Esittely video, Esimerkkikoodit video, Koodin lukeminen video, Harjoituksien tekeminen video).
- Navigation:** 'Takaisin kurssin etusivulle' (Back to course home), 'Tee tentti' (Take test), and 'Harjoituksien tekeminen video' (Video for making exercises).
- Test Details:** Arviointitapa: Korkein arvosana; Hyväksymisraja: 10,00 / 10,00.
- Footer:** 'Olet kirjautunut nimellä Joonas Rissanen.: Opiskelija (Palaa omaan rooliisi)' and social media links for Facebook, X, Instagram, YouTube, and LinkedIn.

Kuva 11: Esimerkki tiistain päiväkohtaisen tentin aloitussivusta.

Tentinäkymä noudattaa Moodlen tentti-oletusasua. Kurssin tentin asetuksien mukaan yksi kysymys näkyy aina yhdellä sivulla.

The screenshot shows the test page for 'TIISTAI - TENTTI' on the Savonia Code Academy platform. The page is in Finnish and displays the following information:

- Course Info:** Työpöytä / Kurssit / Peppi / 2024-12 / JA00BS63-3001 / TIISTAI / TIISTAI - TENTTI
- Navigation:** 'Takaisin kurssin etusivulle' (Back to course home), 'Takaisin' (Back), and 'Seuraava sivu' (Next page).
- Test Question:** 'Kysymys 1' (Question 1) with a score of 2,00. The question asks: 'Mikä seuraavista koodiriveistä laskee oikein muuttujien a ja b summan ja tallentaa sen muuttujaan summa?' (Which of the following code lines calculates the sum of variables a and b correctly and stores it in the variable summa?).
- Options:**
 - a. let summa = a + b;
 - b. let summa = a - b;
 - c. let summa = a * b;
 - d. let summa = a / b;
- Footer:** 'Olet kirjautunut nimellä Joonas Rissanen.: Opiskelija (Palaa omaan rooliisi)' and social media links for Facebook, X, Instagram, YouTube, and LinkedIn.

Kuva 12: Esimerkki tiistain päivakohtaisen tentin ensimmäisestä kysymyksestä.

Suoritettuaan tentin opiskelijalle esitetään yhteenveto omista vastauksistaan sekä saatu arvosana.

TENTTI

TIISTAI - TENTTI

[← Takaisin kurssin etusivulle](#)

Aloitettiin	keskiviikko 12. maaliskuu 2025, 10.04
Tila	Palautettu
Valmis	keskiviikko 12. maaliskuu 2025, 10.05
Suoritus aika	47 sekuntia
Arvosana	10,00 pistettä maksimista 10,00 (100%)

Kysymys 1
Oikein
Pisteet 2,00
kokonaispisteistä 2,00
[Merkitse kysymys](#)
[Muokkaa kysymystä](#)

Mikä seuraavista koodiriveistä laskee oikein muuttujien **a** ja **b** summan ja tallentaa sen muuttujaan **summa**?

- a. let summa = a + b; ✓
- b. let summa = a - b;
- c. let summa = a * b;
- d. let summa = a / b;

Vastauksesi on oikein.
Oikea vastaus on: let summa = a + b;

Kuva 13: Esimerkki tiistain päivakohtaisen tentin vastauksien yhteenvedosta.

Opiskelijat näkevät kurssin sivun oikeassa reunassa vimpaimen, josta he voivat kätevästi seurata kurssin edistymistä. Kurssin neljän tentin vuoksi seurantapalkkista löytyy neljä lohkoa, jotka on värikoodattu seuraavasti: punainen osoittaa, että tentti ei ole hyväksytysti suoritettu, vihreä tarkoittaa, että tentti on suoritettu hyväksytysti, keltainen viestii tentin olevan kesken ja sininen kertoo, että tentti ei ole vielä aloitettu tai se ei ole auki.

Edistymisen seuranta

NYT

Vie hiiren kursori lohkon päälle tai näpäytä saadaksesi lisätietoa.

[Yleiskatsaus opiskelijoista](#)

Kuva 14: Edistymisen seuranta vimpain.

7.7 Testaus ja viimeistely

Työ alkoi itsenäisesti luentokalvojen tuottamisesta. Olin ensin edennyt noin 20 prosenttia, jolloin pidimme palaverin ohjaajan kanssa, ja esittelin tähänastisen tuotokseni. Palaverissa sain arvokasta palautetta, jonka pohjalta tein tarvittavat korjaukset ja täsmennyksiä materiaaliin. Tämän jälkeen pidimme toisen palaverin, kun mielestäni tuotos oli saattanut lähemmäs lopullista muotoa. Toinen tapaaminen tuotti lisää muutosehdotuksia, joiden avulla sain aineistosta entistä johdonmukaisemman.

Kun luentomateriaalit oli viimeistelty, siirryin työstämään kurssin toteutusta Moodlessa. Kurssin päivärakenteen avulla Moodlen sisältö ja ulkoasu hahmottui ripeästi. Näin syntyi Moodle-kurssi, joka sisälsi luentomateriaalit, esimerkkikoodit ja harjoitustehtävät. Seuraavaksi ohjaaja ohjeisti esittelemään tuotoksen myös Savonian yhteyshenkilölle. Sovimme yhteisen palaverin, jossa kävimme koko kurssin sisällön läpi. Palaverin aikana sain pääosin myönteistä palautetta ja vain muutamia pieniä muutosehdotuksia esitettiin.

Lopulliseen testaukseen osallistui ohjaaja, varaohjaaja sekä Savonian yhteyshenkilöt, jotka kaikki testasivat kurssin tentit ja hyväksyivät luentomateriaalit. Näin varmistimme, että kaikki osa-alueet toimivat suunnitellusti. Jäljelle jäi enää aikataulun viimeistely, jonka jälkeen kurssi voidaan ottaa käyttöön opiskelijoiden suoritettavaksi.

8 POHDINTA

Tämä opinnäytetyö on ollut henkilökohtaisesti merkittävä ja opettavainen projekti, joka on tarjonnut minulle ainutlaatuisen mahdollisuuden syventyä tiedon käsittelyyn ja Rust-ohjelmointikieleen sekä sen soveltamiseen opetusmateriaalin tuotannossa. Olen erittäin kiitollinen siitä, että sain toteuttaa tällaisen kokonaisuuden. Työ on tuonut esiin omat korkeat tavoitteeni ja toivon, että ne ovat näkyneet myös opiskelijoille. Kurssin suunnittelun, materiaalien laatimisen ja itse opetuksen vetämisen kautta olen kokenut, että tämä projekti on ollut eräänlainen haaveiden täytyminen.

Työn laajuus oli kuitenkin merkittävä haaste. Aikataulun hallinta osoittautui usein haastavaksi, sillä tuntien paukuessa tuntui välillä, että edistyminen pysähtyi kokonaan. Vaikka aihe oli mielenkiintoinen, laaja kokonaisuus vaati enemmän aikaa kuin alun perin kuvittelin. Jos kyse olisi ollut merkittävästi suppeammasta aiheesta, olisi eteneminen ollut sujuvampaa ja nopeampaa. Tämä kokemus on kuitenkin opettanut minulle tärkeitä ajanhallinnan ja projektinhallinnan taitoja, joita voin hyödyntää tulevaisuudessa.

Yksi työn kohokohdista oli luentokalvojen suunnittelu ja toteutus. Panostin siihen, että kalvot olisivat houkuttelevia, toimivia ja ajankuvaan sopivia. Jokaisessa kalvossa pyrin esittämään sisällön ytimekkäästi, käyttäen visuaalisia elementtejä, kuten kuvia, muotoja ja jopa emojiä korostamaan pääkohtia. Tämän ansiosta kalvojen sisältö pysyi selkeänä ja helposti seurattavana, mikä oli erityisen tärkeää opiskelijoiden näkökulmasta. Koen, että onnistuin erityisesti tässä osa-alueessa ja kalvot muodostivat vahvan perustan koko kurssin opetukselle.

Kurssin kokonaisuuden kannalta Rust osoittautui erittäin sopivaksi ohjelmointikieleksi, erityisesti tämän suppean (1 op) kurssin puitteissa. Kielessä on kaikki tarpeellinen perustietojen opettamiseen, mikä mahdollisti opetusmateriaalin tiiviin ja informatiivisen suunnittelun. Toisaalta, mikäli kurssi olisi ollut laajempi (esimerkiksi 5 op), olisi ollut tarpeen paneutua myös Rustin syvempiin ominaisuuksiin, kuten omistajuuden hallintaan. Tällöin kurssin aloitustasolla olisi saattanut olla tarpeen valita toinen ohjelmointikieli, joka olisi sopinut paremmin perusasioiden opettamiseen.

Yhteenvedona tämä opinnäytetyö on ollut vaativa, mutta samalla erittäin palkitseva prosessi. Se on tarjonnut mahdollisuuden kehittyä ohjelmoinnin osaajana sekä tutustua opetusmaailmaan. Vaikka projektin laajuus ja siihen liittyvät aikataulut haastoivat, olen oppinut paljon sekä itsestäni että työkentelytavoistani. Uskon, että nämä opit ovat arvokkaita tulevaisuuden haasteita ajatellen ja olen ylpeä lopputuloksesta.

LÄHTEET

Työssä on käytetty tekoälyä seuraavasti: ChatGPT 2025. OpenAI. GPT-4o. Käytetty kielentarkistukseen, Maaliskuu 2025. <https://chat.openai.com>

Jääskeläinen, Vuotilainen, Alanko, Rissanen, Vornanen, Likitalo, Diov, Asikainen, Tolppi. 2012. Savonia 20-vuotisjuhla-julkaisu. Savonia-ammattikorkeakoulu, Viestintäpalvelut. Saarijärven Offset Oy. Viitattu 12.3.2025.

Savonia 2022. Juhla-artikkeli: Muistoja vuosilta 2012–2013. Savonia.fi verkkosivu. <https://www.savonia.fi/artikkelit/juhla-artikkeli-osa-1/>. Viitattu 12.3.2025.

Savonia-ammattikorkeakoulu Oy. 2024. tilinpäätös ja toimintakertomus. <https://www.savonia.fi/app/uploads/2024/06/Tilinpaatos-2023-allekirjoitettu.pdf>. Viitattu 12.3.2025.

Savonia 2025. Savonian vuosi 2024 lukuina. Savonia.fi verkkosivu. <https://www.savonia.fi/vuosikertomus-2024/>. Viitattu 13.2.2025.

Savonia n.d. Löydä koulutuslasi. Savonia.fi verkkosivu. <https://www.savonia.fi/opiskele-tutkinto/loyda-koulutuslasi/>. Viitattu 13.2.2025.

Savonia 2025. Savonia murskasi tutkintoennätykset – Suomen toiseksi paras tulos. Savonia.fi verkkosivu. <https://www.savonia.fi/uutiset/savonia-murskasi-tutkintoennatykset-suomen-toiseksi-paras-tulos/>. Viitattu 13.2.2025.

Savonia 2024. Savonia koulutusesite 2025. Esite. https://issuu.com/savonia/docs/savonia_koulutusesite_2025-pages. Viitattu 12.3.2025.

Savonian viestintä. 2024. Savonia strategia 2025-2028. <https://www.savonia.fi/app/uploads/2024/08/Savonia-Strategia-2025-2028-FINAL.pdf>. Viitattu 12.3.2025.

Moodle Docs 2010. About Moodle FAQ. docks.moodle.org verkkosivu. Päivitetty 14.3.2025. https://docs.moodle.org/405/en/About_Moodle_FAQ. Viitattu 15.3.2025.

Moodle n.d. The Moodle Story. moodle.com verkkosivu. <https://moodle.com/about/the-moodle-story>. Viitattu 15.3.2025.

Moodle Docs 2006. Pedagogy. docs.moodle.org verkkosivu. Päivitetty 25.1.2025. <https://docs.moodle.org/405/en/Pedagogy>. Viitattu 15.3.2025.

Moodledev 2024. Moodle 4.5. moodledev.io verkkosivu. <https://moodledev.io/general/releases/4.5#server-requirements>. Viitattu 15.3.2025.

Github n.d. The Rust Playground. github.com verkkosivu. <https://github.com/rust-lang/rust-playground>. Viitattu 23.3.2025.

Rust-lang n.d. The Rust Playground. play.rust-lang.org verkkosivu. <https://play.rust-lang.org/help>. Viitattu 24.3.2025.

Jyväskylän yliopisto n.d. TIEP114 Tietokoneen rakenne ja arkkitehtuuri. tim.jyu.fi verkkosivu. <https://tim.jyu.fi/view/kurssit/tie/tiep114/materiaalia/Luentomateriaali>. Viitattu 13.2.2025.

- Brooshear, J.G. & Brylow, d. 2015. Computer Science - An Overview 12th ed. Pearson Education.
- Aalto Yliopisto 2004. Digitaalitekniikan perusteet - luento 3. legacy.spa.aalto.fi verkkosivu. <http://legacy.spa.aalto.fi/sig-legacy/digis/printtaa/luento3/luento3.html>. Viitattu 13.3.2025.
- Peda n.d. Luvut biteiksi. peda.net verkkosivu. <https://peda.net/iitti/lukio/oppiaineet/tietotekniikka/atp/bl/luvut-biteiksi>. Viitattu 13.3.2025.
- Lahtonen 1998. Lukujärjestelmistä. appro.mit.jyu.fi verkkosivu. <https://appro.mit.jyu.fi/1999/johdatusohjelmointiin/materiaalia/lukujarjestelmat.html>. Viitattu 13.3.2025.
- Fenwick, P. 2014. Introduction to Computer Data Representation. Bentham Science Publishers. Viitattu 14.3.2025.
- MOOC-keskus 2022. Tietokoneen toiminnan perusteet 2022. tito-perusteet-2022.mooc.fi verkkosivu. <https://tito-perusteet-2022.mooc.fi/luku-3/2-kokonaisluvut-ja-liukuluvut>. Viitattu 15.3.2025.
- Oulun yliopisto n.d. Bitit ja lukujärjestelmät. lovelace oulu.fi verkkosivu. <https://lovelace oulu.fi/tietokonej%C3%A4rjestelm%C3%A4t/tietokonej%C3%A4rjestelm%C3%A4t/bitit-ja-lukuj%C3%A4rjestelm%C3%A4t/>. Viitattu 15.3.2025.
- GeeksforGeeks 2020. IEEE Standard 754 Floating Point Numbers. geeksforgeeks.org verkkosivu. Päivitetty 16.3.2020. <https://www.geeksforgeeks.org/ieee-standard-754-floating-point-numbers/>. Viitattu 14.3.2025.
- Haataja, J., Heikonen, J., Leino, Y., Rahola, J., Ruokolainen, J. & Savolainen, V. 2002. NUMEERISSET MENETELMÄT KÄYTÄNNÖSSÄ. Picaset Oy. Viitattu 14.3.2025.
- Unicode 2024. Unicode 16.0.0. Unocide.org verkkosivu. <https://www.unicode.org/versions/Unicode16.0.0/>. Viitattu 17.3.2025.
- Stanford Encyclopedia of Philosophy 2010. Chance versus Randomness. plato.stanford.edu verkkosivu. Päivitetty 8.2.2018. <https://plato.stanford.edu/entries/chance-randomness/>. Viitattu 16.3.2025.
- Knuth, D. 1997. The Art of Computer Programming vol. 2 3rd ed. Addison-Wesley. Viitattu 16.3.2025.
- Random n.d. Introduction to Randomness and Random Numbers. random.org verkkosivu. <https://www.random.org/randomness/>. Viitattu 16.3.2025.
- Luengo, E. 2022. A brief and understandable guide to pseudo-random number generators and specific models for security. <https://doi.org/10.1214/22-SS136>. Viitattu 17.3.2025.
- Asecuritysite 2025. Mersenne Twister. asecuritysite.com verkkosivu. https://asecuritysite.com/principles_pub/twister. Viitattu 17.3.2025.
- How n.d. What is Mersenne Twister?. how.dev verkkosivu. <https://how.dev/answers/what-is-mersenne-twister>. Viitattu 17.3.2025.
- Aalto Yliopisto n.d. Tietokone ja satunnaisuus. plus.cs.aalto.fi verkkosivu. <https://plus.cs.aalto.fi/cs-eyJ404/2022/04-FrekvenssianalyysiSatunnaisuus/05-Satunnaisuus/>. Viitattu 16.3.2025.

Hackaday 2017. What Is Entropy And How Do I Get More Of It?. Hackaday.com verkkosivu. <https://hackaday.com/2017/11/02/what-is-entropy-and-how-do-i-get-more-of-it/>. Viitattu 23.3.2025.

IBM 2024. What is a data structure?. Ibm.com verkkosivu. <https://www.ibm.com/think/topics/data-structure>. Viitattu 17.3.2025.

Laaksonen, A. 2022. Tietorakenteet ja algoritmit. Opetusmateriaali. Helsingin yliopisto. Viitattu 17.3.2025.

Rust-lang 2025. Primitive Type array. Doc.rust-lang.org verkkosivu. <https://doc.rust-lang.org/std/primitive.array.html>. Viitattu 24.3.2025.

GeeksforGeeks 2024. Rust - Array. geeksforgeeks.org verkkosivu. Päivitetty 27.2.2024. <https://www.geeksforgeeks.org/rust-array/>. Viitattu 17.3.2025.

The Rust Programming Language n.d. Storing Lists of Values with Vectors. doc.rust-lang.org verkkosivu. <https://doc.rust-lang.org/book/ch08-01-vectors.html>. Viitattu 17.3.2025.