

Ohjelmoinnin perusteet & J2SE 5.0



Kurssin ohjelmointikielenä on Java SE (versio 5.0).

Kurssin ensimmäinen osa (luvut 1-9) käsittelevät ohjelmoinnin perusteita Java-kielellä ja jälkimmäinen osa (luvut 10-18) tutustuttaa opiskelijan olio-ohjelmoinnin käsitteisiin sekä muutama Java SE 5.0 -versiossa julkaistuihin ominaisuuksiin.

Teoria sisältää monipuolisia esimerkkejä oppittavasta aiheesta. Luvuissa 4-8 ja 11 on lopussa esimerkkiohjelma, jossa harjoitellaan siihen saakka Javasta opittuja asioita. Luku 9 keskittyy taas pelkästään Ristinolla-pelin toteutukseen suunnittelusta testaamiseen.

Kurssi sisältää 47 ohjelmoinnin harjoitustehtävää sekä 124 monivalintatehtävää. Kurssin suorittaminen kestää arviolta 100-160 tuntia.

Kurssin sisällysluettelo:

1 Johdanto 1.1 Kurssin esittely 1.2 Vinkkejä opiskeluun 1.3 Tietokone ja tietokoneohjelmat 1.4 Lähdekoodista tietokoneen ymmärtämään muotoon	2 Algoritmin idea 2.1 Ongelmasta algoritmiin 2.2 Algoritmita lähdekoodiin	3 Johdanto Javaan 3.1 Java-kielestä lyhyesti 3.2 Ohjeet itsenäiseen harjoitteluun 3.3 Lähdekoodin osat ja rakenne
4 Tyyppejä, muuttujia, lauseita ja lausekkeita 4.1 Tyypeistä ja muuttujista 4.2 Lauseita ja lausekkeita 4.3 Pieni Java-ohjelma tähän asti opitusta	5 Tulostamista, lukemista ja operaattoreita 5.1 Lyhyesti: luokka ja metodi 5.2 Tiedon tulostamista 5.3 Näppäimistösyötteiden lukemista Scanner-luokan välineillä 5.4 Operaattoreita 5.5 Pieni Java-ohjelma tähän asti opitusta	6 Rakenteiset lauseet 6.1 Lohko 6.2 Valintalauseet 6.3 Toistolauseet 6.4 Pieni Java-ohjelma tähän asti opitusta
7 Metodit 7.1 Metodin määrittely 7.2 Metodin kutsuminen 7.3 Parametrien välittäminen 7.4 Metodien kuormittaminen 7.5 Pieni Java-ohjelma tähän asti opitusta	8 Taulukot 8.1 Taulukot käyttöön 8.2 Taulukko käytössä 8.3 Taulukoiden kopioinnista 8.4 Moniulotteiset taulukot 8.5 Pieni Java-ohjelman tähän asti opitusta	9 Ohjelmien tekemisestä 9.1 Ristinolla: ongelman analysointi 9.2 Ristinolla: algoritmin suunnittelu 9.3 Ristinolla: toteutus 9.4 Ristinolla: testaaminen
10 Oliot ongelman ratkaisussa 10.1 Proseduraalinen ohjelmointi vs. olio-ohjelmointi 10.2 Olio 10.3 Luokka 10.4 Tietojen suojaamisesta ja kapseloinnista 10.5 Abstrahointi 10.6 Periytyminen	11 Luokka, olion malli 11.1 Luokka ja kuinka se tehdään 11.2 Luokasta olio 11.3 Luokan testaaminen 11.4 Taulukko olion säilytyspaikkana 11.5 String-luokan metodeita 11.6 Pieni Java-ohjelma tähän asti opitusta	12. Luokan jäsenistä ja niiden näkyvyydestä 12.1 Luokan jäseniä 12.2 Näkyvyyttä määreistä
13. Periytyminen 13.1 Johdanto periytymiseen 13.2 Aliluokan määrittely 13.3 Monimuotoisuus eli polymorfismi 13.4 Abstraktit luokat 13.5 Object	14. Rajapintaluokat 14.1 Johdanto 14.2 Rajapintaluokan käyttäminen	15 Pakkaukset 15.1 Mikä on pakkaus? 15.2 Valmiiden pakkausten käyttäminen 15.3 Omien pakkausten määrittely 15.4 Miten kääntäjä löytää pakkaukset? 15.5 Pakkausten viittausalueesta
16. Poikkeuksista ja niiden käsittelystä 16.1 Mikä on poikkeus? 16.2 Poikkeusluokista 16.3 Poikkeuksen heittäminen 16.4 Poikkeusten käsitteleminen	17. tietovirroista ja Tiedostoista 17.1 Käsitteitä 17.2 Tekstimuotoisen tiedon lukemista ja kirjoittamista 17.3 Tiedostojen käsittelyä java.io.File-luokan välineillä	18 Pinosta generinen 18.1 Johdanto pino-esimerkkiin 18.2 Dynaaminen generisyys Object-luokan avulla 18.3 Vilkaisu Javan parametroituihin luokkamalleihin