

2022年度 日本工学院専門学校											
情報処理科 システム開発コース											
プログラミング実習 2											
対象	2年次	開講期	後期	区分	選択	種別	講義	時間数	60	単位	2
担当教員	諸岡 瑞香			実務 経験	有	職種	情報システム担当、システムエンジニア、プログラマ				
担当教員紹介											
<p>プログラマ、システムエンジニアとして、複数のシステム開発経験があります。また、企業内情報システム担当として、社員向けのIT系教育担当にも従事してきました。これらの経験を生かし、IT系職種への進路指導はもちろんのこと、ビジネスマナー、社会人基礎力を身につけていただけるよう学生の皆さんを支援していきたいと思っています。</p>											
授業概要											
<p>ソフトウェアの世界でシステムを開発することは、現実の世界にあるものをコンピュータの中に再現することである。コンピュータの中に再現するにあたり、再現したい現実のものをオブジェクトと捉え、そのオブジェクトをコンピュータの中に作成する代表的なオブジェクト指向プログラミング言語の一つとしてJavaがある。近年はオブジェクト指向でシステムを開発することが一般的となっており、オブジェクト指向の基本概念を理解することが必修である。本講義では、Javaの基本的な文法からクラスを用いたプログラミングを学ぶことで、オブジェクト指向プログラミングの基本的な概念を理解することを目的とする。</p>											
到達目標											
<p>この科目では学生がJavaを使って、プログラムの基本3構造(順次、選択、繰り返し)に則った簡単なプログラムを作成できること、IDE(統合開発環境)による効率的な開発とデバッグができること、クラスを定義しオブジェクトを生成できること、クラスにメソッドを定義して活用したプログラムを作成できること、カプセル化・継承・多態性などを使ったプログラムを作成できること、UMLから仕様を読み取りプログラムを作成し実行できるようになることを目標にしている。</p>											
授業方法											
<p>この授業では、IDE(統合開発環境)を利用してプログラムの開発方法やデバッグ方法を習得するとともに、教科書を中心としてJavaの基本的な文法を説明した後、簡単な例題プログラムの作成を繰り返しながら授業を進める。適時実施する演習課題に対して、自分なりに考え、手を動かしてプログラムを作成することでプログラミングスキルの定着を図り、オブジェクト指向プログラミングで簡単なプログラムを作成できることを目指す。</p>											
成績評価方法											
<p>課題 90% 授業内容の理解度を確保するために実施する 平常点 10% 積極的な授業参加度や課題の提出状況によって評価する</p>											
履修上の注意											
<p>教科書、ノートパソコン、LANケーブルを必ず持参すること。新しい用語の意味を理解し名称を覚えること。簡単なプログラムであったとしても、必ず手を動かして実際にプログラムを作成し、プログラムの実行結果を確認すること。就活で欠席する場合は必ず事前に公欠届の承認印をもらうこと。授業時数の4分の3以上出席しない者は定期試験を受験することができない。</p>											
教科書教材											
中山清喬・国本大悟著：スッキリわかるJava入門、インプレス、2014年出版 ※入手方法別途連絡											
回数	授業計画										
第1回	環境準備、基礎知識：自分のノートパソコンでJavaのプログラムを作成できる										
第2回	式と演算子：演算子を使って、変数に対してさまざまな命令を実行することができる										
第3回	条件分岐と繰り返し：順次・分岐・繰り返しの3つの制御構造を組み合わせたプログラムを作成することができる										
第4回	配列：配列を使用したプログラムを作成することができる										
第5回	メソッド：定義したメソッドを使用することができる。オーバーロードを使用することができる										

2022年度 日本工学院専門学校	
情報処理科 システム開発コース	
プログラミング実習 2	
第6回	複数クラスを用いた開発：クラスを分割し、パッケージに所属させたプログラムを作成することができる
第7回	オブジェクト指向について：ソフトウェアの部品化の考え方、オブジェクトの属性、操作について説明できる
第8回	クラスとインスタンス：定義したクラスのインスタンスを生成することができる
第9回	さまざまなクラス機構：「has-aの関係」で他クラスのオブジェクトを参照するプログラムを作成することができる
第10回	カプセル化：カプセル化することでフィールド(属性)を保護したプログラムを作成することができる
第11回	継承：「is-aの原則」で継承したクラスのプログラムを作成することができる
第12回	高度な継承：抽象クラス、インタフェースを利用したクラスのプログラムを作成することができる
第13回	多態性：多態性のあるクラスを定義し、活用したプログラムを作成することができる
第14回	標準クラス：標準クラスを使用したプログラムを作成することができる
第15回	例外：例外処理を組み込んだプログラムを作成することができる