科目ナンバリング	U-LAS30 20047 SJ11						
授業科目名<英訳>	プログラミング演習（関数型言語） Programming Practice (Functional Programming)			担当者所属職名・氏名	人間・環境学研究科 准教授 THIES, Holger		
群	情報学科目群	分野(分類)	(各論)			使用言語	日本語
旧群	B群	単位数	2単位	週コマ数	1コマ	授業形態	演習（対面授業科目）
開講年度・開講期	2026・後期	曜時限	水2	配当学年	全回生	対象学生	全学向

（総合人間学部の学生は，全学共通科目として履修登録できません。所属部局で履修登録してください。）

[授業の概要・目的]

関数型プログラミングは、関数を組み合わせてプログラムを構成し、可変状態や副作用を避けることを重視するプログラミングパラダイムである。同じ入力に対して常に同じ出力を返す関数を基礎とするため、数学における関数に近い形で計算を記述できる。そのため、プログラミング作業を論理的な単位に分割して考えやすくなり、より簡潔で信頼性の高いプログラムが書ける。さらに、並列処理や並行処理を含む応用にも適しており、幅広い計算に利用することができる。
本演習では、関数型プログラミングの中心的な考え方と基本的な技法を学ぶ。主な例としてプログラミング言語Haskellを用い、関数の合成による計算、パターンマッチ、再帰、高階関数、遅延評価などの主要な概念を紹介する。また、設計上の選択や概念の違いを比較する際には、他の現代的な関数型プログラミング言語にも適宜言及する。

[到達目標]

関数型言語によるプログラミングを通して、コンピュータを利用することを身につける。また、その裏側にある処理方法などからコンピュータそのものの仕組みを理解する。

[授業計画と内容]

以下のようなトピックについてそれぞれ1、2週の授業を行う予定である。

1. プログラミング・パラダイムと関数型プログラミング
2. 初めてのプログラムの作成と実行
3. 関数の定義と合成
4. 再帰的定義と帰納的な考え方
5. パターンマッチやガードによる定義
6. 基本的なデータ構造
7. 高階関数
8. 評価戦略
9. 型と型推論
10. 代数的データ型と構造化データ
11. 入出力とモナドによる計算
12. 並列・並行プログラミング
13. 現代的な関数型プログラミング言語の比較

授業回数はフィードバックを含め全15回とする。

プログラミング演習（関数型言語）(2)へ続く↓↓↓

プログラミング演習（関数型言語）(2)

[履修要件]

学内のITサービスの利用のマナーを遵守できること。コンピュータの基本的な使い方を習得していることが求められる。ノートPCを持参して授業に参加すること。プログラミングの経験はとくに必要としない。

[成績評価の方法・観点]

評価は授業内課題および宿題課題に基づいて行う。

[教科書]

使用しない

[参考書等]

（参考書）
授業中に紹介する

[授業外学修（予習・復習）等]

基本的には、授業での学習を元に、各トピックに関連するさまざまなプログラムを実際に動かしてみることで理解を深めることが重要である。また、可能であれば事前に、授業で学習する各トピックについて、書籍等に基づいて概念、用語などを調べたり、実際に簡単なプログラムを動かしてみることで、演習にスムーズに取り組めるようにしておくことが望ましい。

[その他（オフィスアワー等）]

関連する科目(全学共通科目): 数理論理学A・B

[主要授業科目（学部・学科名）]