初級プログラミング Ⅰ

第1回 コンピュータとプログラミング

中谷　祐介

本科目の目標

- コンピュータがプログラムによって動くことを理解し，プログラムを作り上げるプログラミングという知的活動ができるようになることを目指す．
  - コンピュータの仕組みとプログラムの働きについて説明できる
  - Javaを通じてプログラミング言語の成り立ちとコンパイラの働きを説明できる
  - データの入出力，プログラムの流れの制御，簡単なデータ構造，文字列の処理，等の基本的な内容を含むプログラミングができる

第1回の構成

◆ 第1講　ハードウェア

- コンピュータとハードウェアに関する知識を習得する.

◆ 第2講　ソフトウェア

- ソフトウェアに関する知識を習得する.

◆ 第3講　プログラミング

- プログラミングに関する知識を習得する.

◆ 第4講　プログラミング環境「@CODE ROOM」

- 本科目のプログラミング環境である「@CODE ROOM」の詳細を理解する.

第1回　コンピュータとプログラミング

第1講　ハードウェア

第1講の学習目標

◆ コンピュータとハードウェアに関する知識を習得する:

- コンピュータとは
- ハードウェアの構成
  - 中央処理装置
  - 記憶装置
  - 入力装置
  - 出力装置
  - 通信制御装置

コンピュータとは（1）

デスクトップPC

ラップトップPC

タブレットPC[1]

スーパーコンピュータ[2]

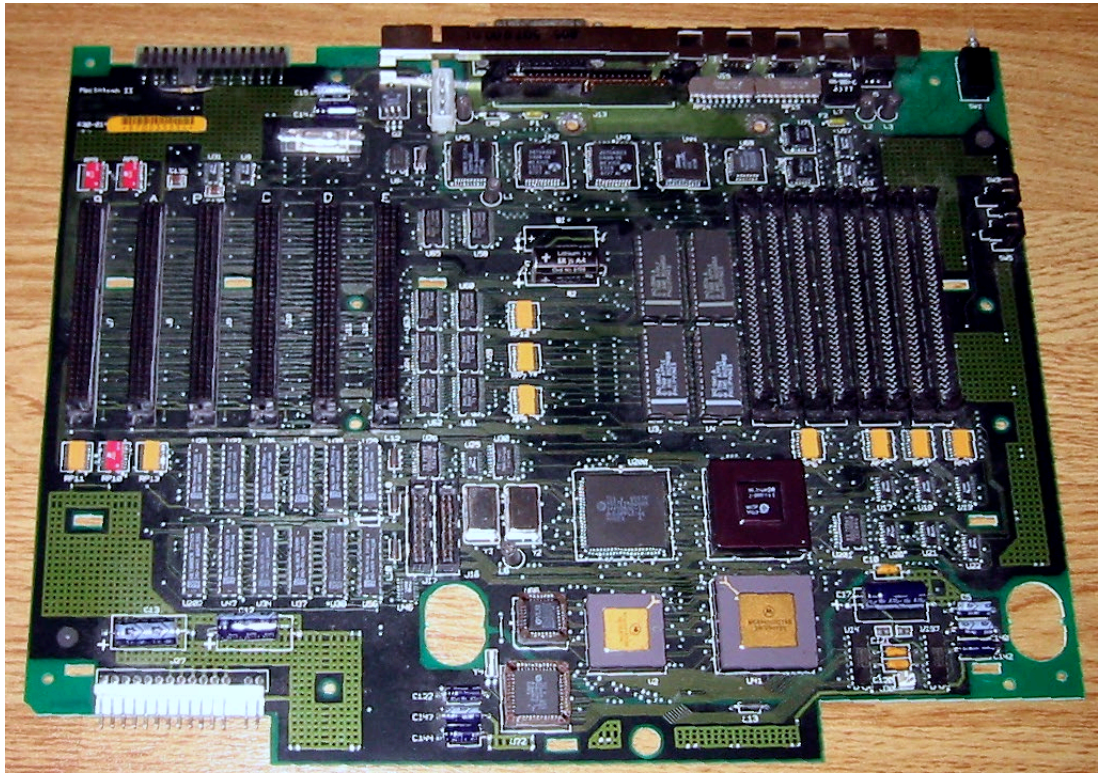
PC：パーソナル・コンピュータ，パソコン

1 Dandaman32による画像（2007）CC BY-SA 3.0
https://commons.wikimedia.org/wiki/File:Lenovo_Thinkpad_X41.png

2 OISTによる画像（2015）CC BY 2.0
https://www.flickr.com/photos/oistedu/33948146551

コンピュータとは（2）

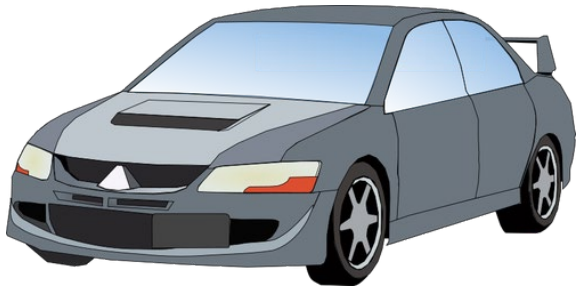
◆PCの蓋を開けると，基盤の上に電子回路素子が並ぶ



基盤の上の電子回路素子[1]

1 Jpkによる写真 (2008) CC BY-SA 3.0
https://commons.wikimedia.org/wiki/File:Macintosh_II_motherboard.jpg

コンピュータとは（3）



自動車

デジタルカメラ

スマートフォン

航空機

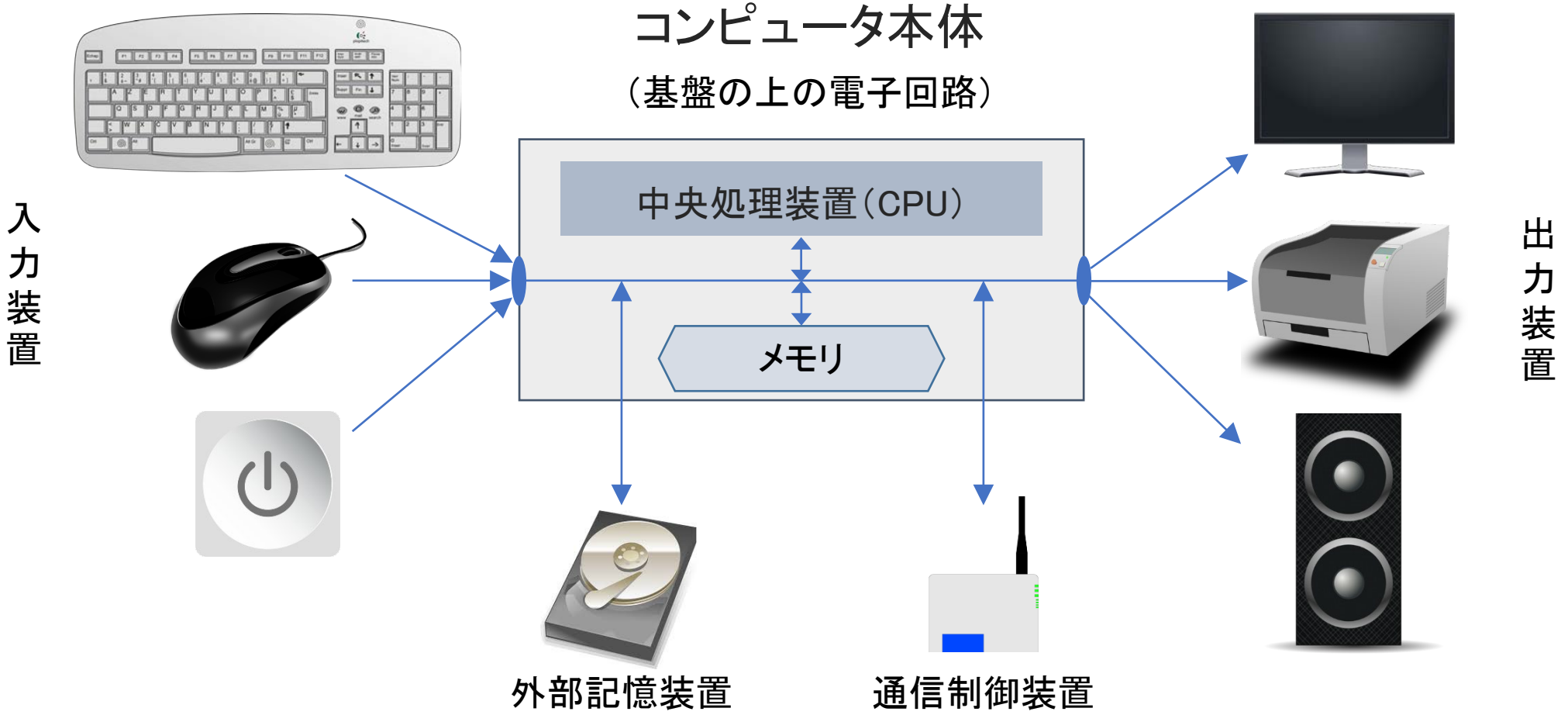
冷蔵庫

ハードウェア（1）

◆ コンピュータのハードウェアは，基本的に，以下の5種類で構成される：

- **中央処理装置（CPU）**：プログラムを実行し，データの処理を行う装置．
  コンピュータの脳．
- **記憶装置**：プログラムやデータを記憶する装置．
  主記憶装置（メモリ）と**外部記憶装置**がある．
- **入力装置**：プログラムやデータをコンピュータに入力する装置．
- **出力装置**：コンピュータでの処理の結果を出力する装置．
- **通信制御装置**：他のコンピュータとの通信を行う装置．

ハードウェア（2）



ハードウェアの構成のイメージ

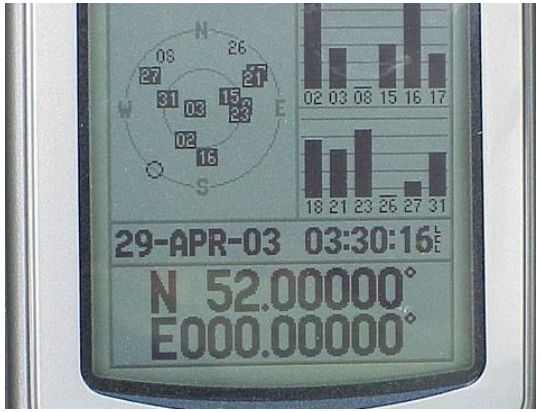
入力装置の例

キーボード

トラックパッド[1]

マウス

マイクロフォン



GPS受信装置

1 Highway of Lifeによる写真 CC BY-SA 3.0
https://commons.wikimedia.org/wiki/File:Macbook_pro_trackpad.jpg

出力装置の例

ディスプレイ

スピーカー

プリンタ

サイン（電源ボタン）

ゴーグル[1]

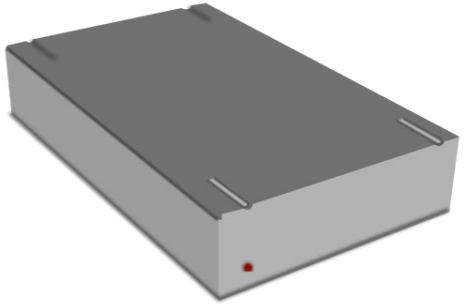
1 Maurizio Pesceによる写真 Samsung Gear VR(2014) CC BY 2.0
https://www.flickr.com/photos/pestoverde/15060706109/

記憶装置・通信制御装置の例

ハードディスク

組込型

可搬型



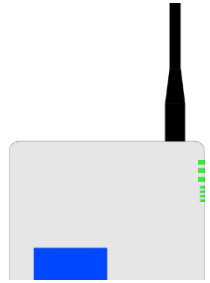
CD / DVD / Blu-ray

組込型ドライブ[1]

外部記憶装置の例

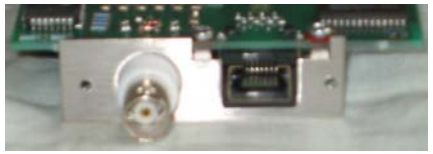
WiFiルータ

外付け型



イーサー送受信

組込型[2]



通信制御装置の例

1 Asim18による写真 Image of an ASUS CD-ROM Drive CD-S520/A4(2008) CC BY-SA 3.0
https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:ASUS_CD-ROM_CD-S520-A4_20080821.jpg

2 Chris Whytehead, Chris 's Acorns による写真 (2012) CC BY-SA 3.0
https://commons.wikimedia.org/wiki/File:I-cubed_EtherLan600_(back).jpg

第1講のまとめ

◆ コンピュータとハードウェアに関する知識を習得した：

- コンピュータ
  - さまざまな種類のコンピュータがある．
  - 身の回りの機器にもコンピュータが組み込まれている．
- ハードウェア
  - コンピュータのハードウェアは，基本的に5つの種類の装置で構成される．
  - コンピュータで使用する入力・出力装置など，具体的なものを確認した．

第1回　コンピュータとプログラミング

第1講　ハードウェア

終わり

第1回　コンピュータとプログラミング

第2講　ソフトウェア

第2講の学習目標

◆ ソフトウェアに関する知識を習得する:

- ソフトウェアとは
- ソフトウェアの種類
  - 基本ソフトウェア
  - 応用ソフトウェア
  - ミドルウェア

ハードウェア（第1講の復習）

◆ ハードウェアの構成

- 中央処理装置（CPU）
- 記憶装置
- 入力・出力装置
- 通信制御装置

} コンピュータ本体とその周辺機器

◆ ハードウェアはただの箱であり，これだけでは動作しない．

◆ ソフトウェアにより，コンピュータを動作させる．

ソフトウェアとは

◆ **ソフトウェア**は，コンピュータを動作させるためのプログラムのこと．

- プログラムが処理するデータ等を含む場合もある．

◆ ソフトウェアは，ハードウェアと対比される．

- ハードウェアは，実体のある機器（コンピュータ，周辺機器）⇒ 物理的な機構
- ソフトウェアは，実体のない目にみえないもの ⇒ 論理的な機構

◆ コンピュータシステム = ハードウェア + ソフトウェア

◆ ソフトウェアの種類

- **基本ソフトウェア**
- **応用ソフトウェア**
- **ミドルウェア**

基本ソフトウェア

- コンピュータを動作させるために必要な基本機能を提供するソフトウェアのこと.
- **オペレーティングシステム（OS）**ともいう.
- ハードウェアの制御，操作を行う.
- OSの例：
  - Windows
  - macOS
  - Unix系OS
  - スマートフォン向けのOS（iOS, Android）

応用ソフトウェア

◆ 一般のユーザが直接利用するソフトウェアのこと.

- ユーザが自身で作成したプログラム
- メーカーその他が提供するパッケージ化されたプログラム

◆ **アプリケーションソフトウェア**ともいう.

◆ 基本ソフトウェアやミドルウェアの機能を利用して，応用上高度な機能を実現する.

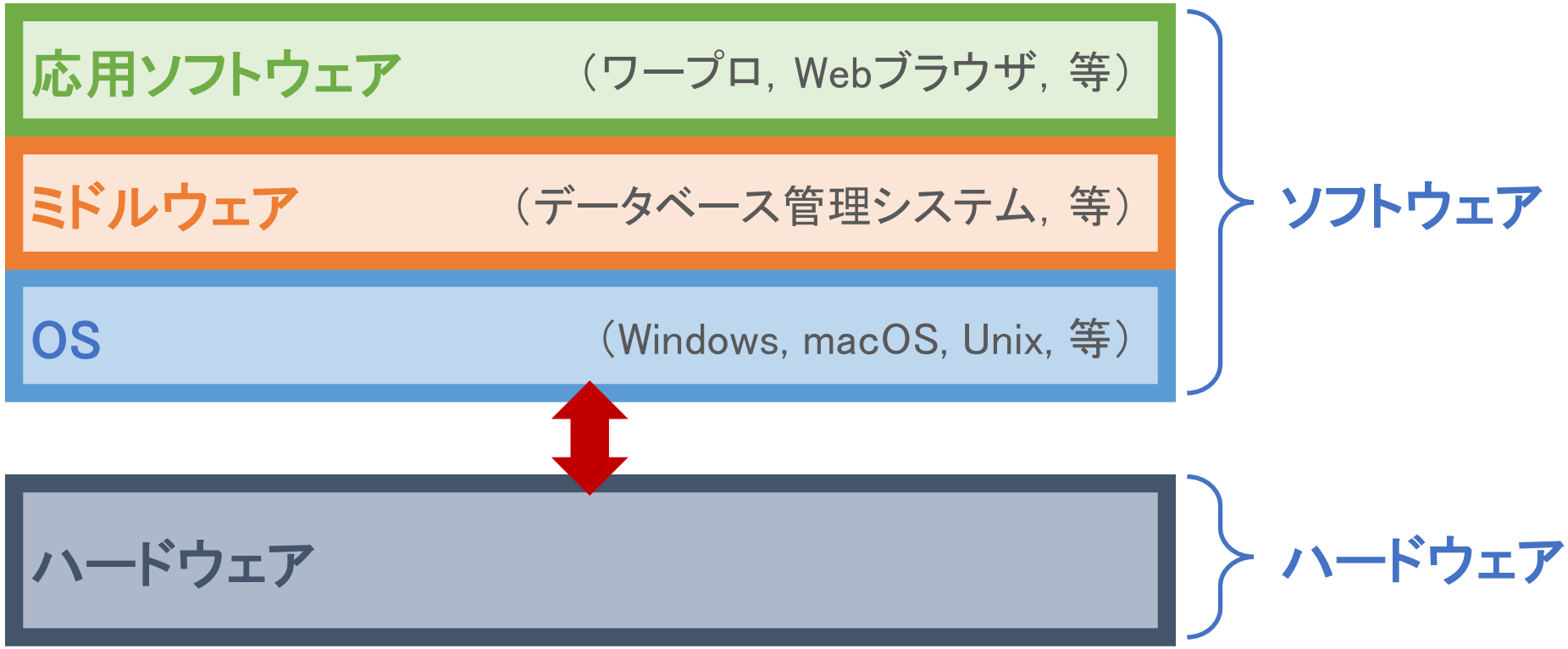
◆ 応用ソフトウェアの例：

- ワードプロセッサ，表計算ソフトウェア，プレゼンテーションソフトウェア
- 電子メールソフトウェア，Webブラウザ
- 各種業務用ソフトウェア

ミドルウェア

- OSと応用ソフトウェアの間で機能するソフトウェアのこと.
  - OSの機能の拡張
  - 応用ソフトウェアの共通機能の集まり
- OSの機能を利用し，特定の分野に共通する高レベルの基本的機能を実現する.
- ミドルウェアの機能の流れ:
  - 応用ソフトウェアからの要求を受け，OSに要求を出す
  - OSの出した結果を応用ソフトウェアに返す
- ミドルウェアの例:
  - データベース管理システム
  - 通信管理システム

コンピュータシステム



コンピュータシステムのイメージ

第2講のまとめ

◆ソフトウェアに関する知識を習得した：

- ソフトウェア
  - コンピュータを動作させるためのプログラムのこと.
  - ソフトウェア ⇔ ハードウェア

- ソフトウェアの種類
  - 基本ソフトウェア（OS）：Windows, macOS, Unix系OS, 等
  - 応用ソフトウェア：ワードプロセッサ, Webブラウザ, 等
  - ミドルウェア：データベース管理システム, 等

第1回　コンピュータとプログラミング

第2講　ソフトウェア

終わり

第1回　コンピュータとプログラミング

第3講　プログラミング

第3講の学習目標

- プログラミングに関する知識を習得する:
  - プログラミング
  - プログラミング言語
    - 種類
    - 歴史
  - 本科目におけるプログラミング
- 本科目で使用する Java の概要を理解する:
  - Java
    - 概要
    - プログラミングの手順
    - 演習環境の概要

プログラミング

- **プログラミング**とは，プログラムを作成する作業のこと．
- プログラミングを行う際は，**プログラミング言語**を使用して記述する．
- プログラミングには，さまざまな**プログラミングパラダイム**が存在する．
- プログラミングパラダイム：
  - プログラムの作り方に関する規範．
  - プログラムの設計手順やプログラムの構造およびプログラムの記述方法を規定するもの．
  - プログラミングの際に，何に着目して問題を整理するのか，何を中心にプログラムを構成するのかの方向付けを与えるもの．
  - 手続き型，オブジェクト指向，関数型，論理型，など．

プログラミング言語

- **プログラミング言語**は，プログラムを記述するための人工言語.
- プログラミング言語は，自然言語と機械語の中間に位置する言語.
- 自然言語:
  - 人間によって日常の意思疎通のために用いられる言語.
  - 人間が理解しやすい.
- 機械語:
  - コンピュータが直接理解できるように2進数で表現された言語.
  - コンピュータが理解しやすい.
- プログラミング言語は，プログラミングパラダイムによって分類することができる.

プログラミング言語の種類

◆手続き型言語

- 処理手続きを命令文で記述する.
- 言語の例 : ALGOL, C, COBOL, FORTRAN.

◆オブジェクト指向言語

- データとそれに対する手続きをカプセル化したオブジェクトとその間のメッセージ通信で記述する.
- 言語の例 : C++, Java, Python, Smalltalk.

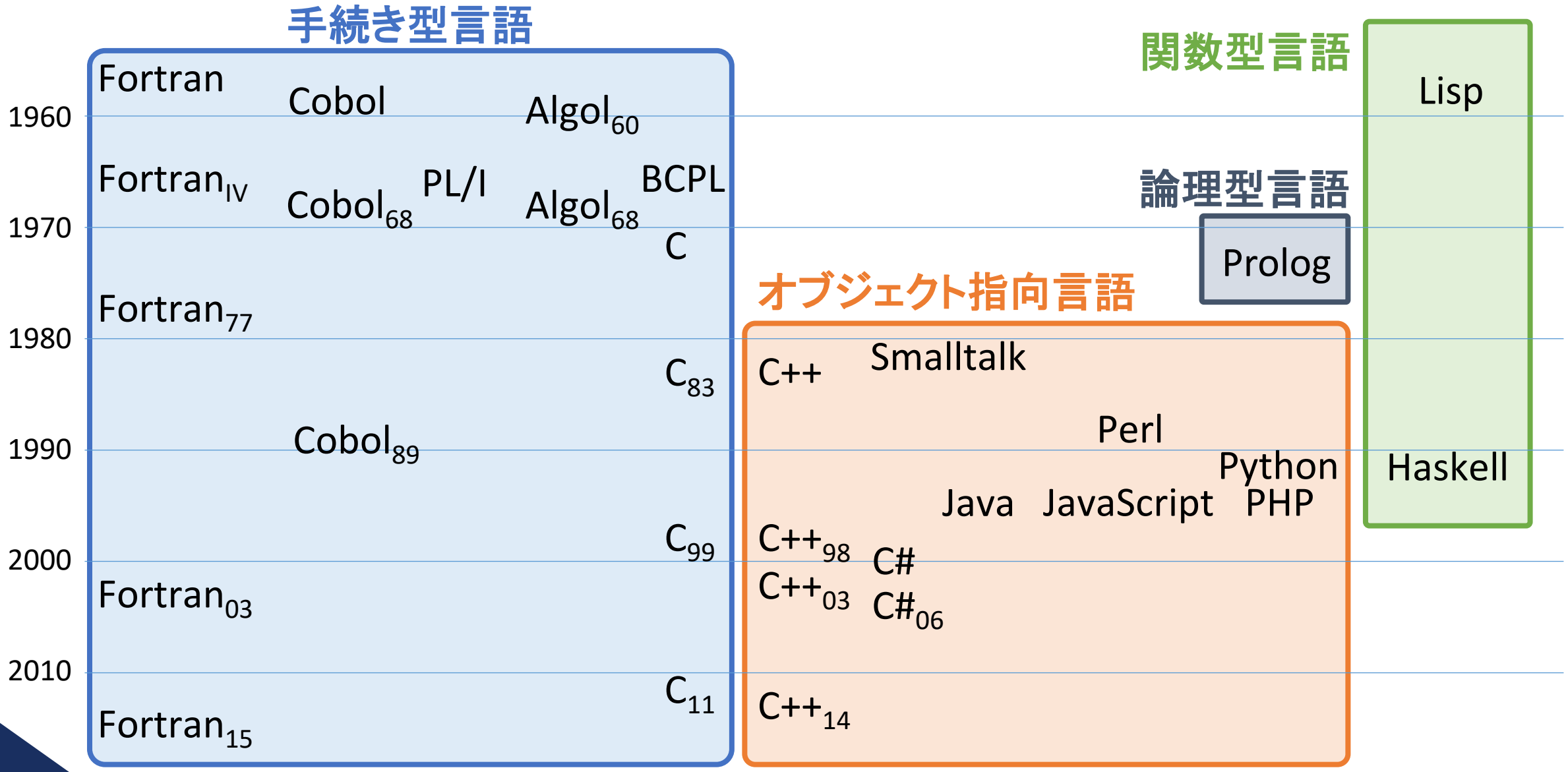
◆関数型言語

- 入出力関係を表現する関数とその呼び出しで記述する.
- 言語の例 : Lisp, Haskell.

◆論理型言語

- 述語論理の推論に基づいて記述する.
- 言語の例 : Prolog.

プログラミング言語の歴史



本科目におけるプログラミング（1）

◆ コンピュータに何かの仕事をさせるためには，プログラミング（プログラムの作成）が必要．

- プログラムを作成し，コンピュータに仕事をさせることを理解する．
- そのために必要な，基本的なプログラミングの知識を習得する．
- プログラミング言語として Java を使用する．

◆ 本科目で習得する内容

- データの入出力，変数，演算
- 制御構造（条件分岐，繰り返し）
- 配列
- 文字，文字列

◆「習うより慣れろ！」

- 手を動かすことで，多くのプログラムを書く．

本科目におけるプログラミング（2）

◆ 各回の進め方

- 講義を受講する.
- 「小テスト」を受験することで，講義の内容を理解できているか確認する.
- 演習環境「@CODE ROOM」でプログラミングを行う:
  - 講義内で示したプログラム例を動かし，内容を理解する.
  - 「練習問題」のプログラムを作成する（成績評価には関係ない）.
  - 「演習課題」のプログラムを作成する（第4回，第8回）.

◆ 演習課題

- 第4回，第8回では，「演習課題」を出題する.
- 課題のプログラムを作成することで，講義の内容を理解できているか確認する.
- 「演習課題」は成績評価に含まれる.

Java

歴史

- 1991年頃，Sun Microsystems[1] 社の内部プロジェクトとして開発を開始.
- 1995年，正式に発表.

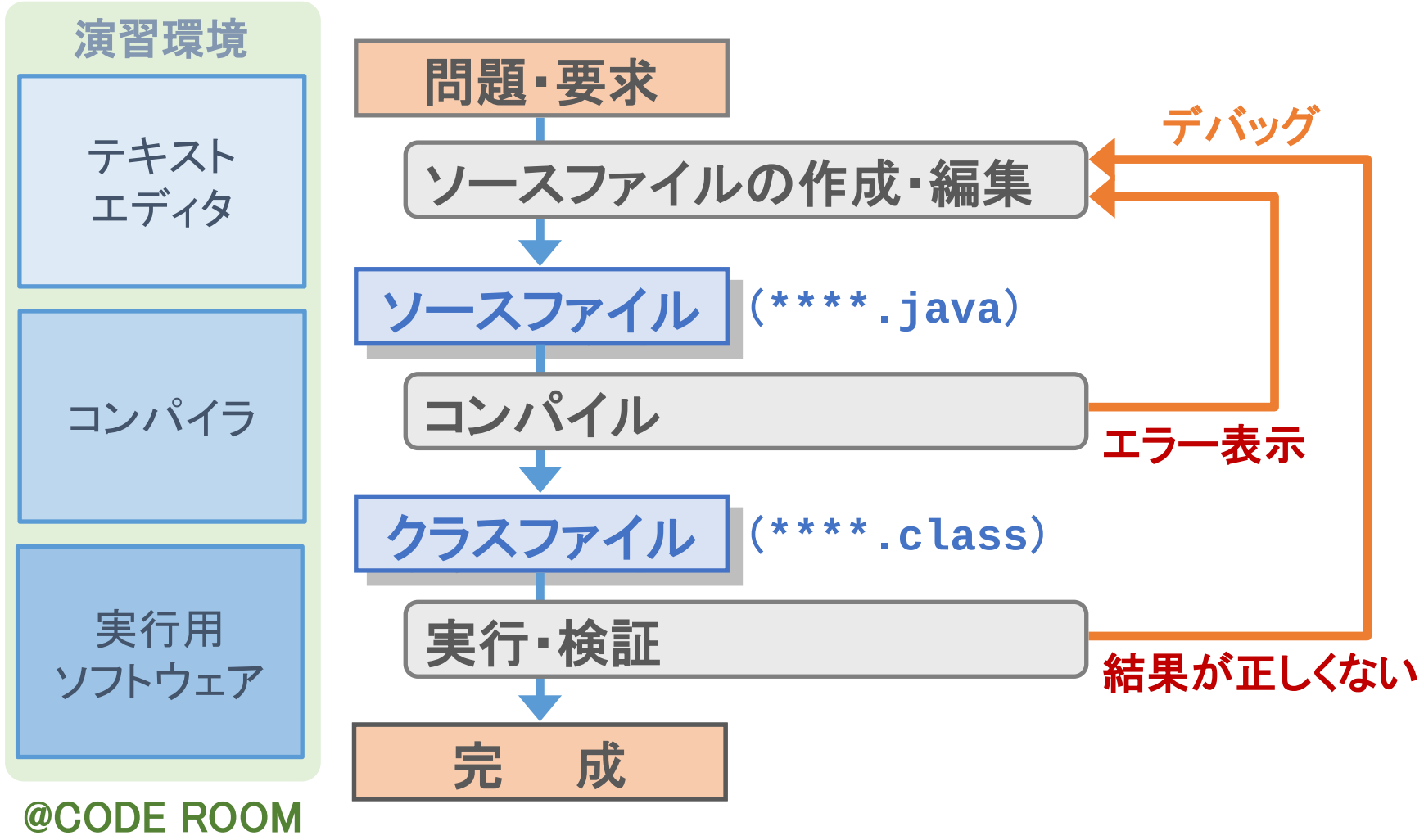
特徴

- オブジェクト指向のプログラミング言語である.
- 実行環境に依存しない.
- 豊富なライブラリが存在する.
- メモリ管理を自動的に行う.

本科目では，Java を用いてプログラミングの基礎を学ぶ.

1 現在は Oracle 社

Javaによるプログラミングの手順



Javaによるプログラミングの手順

演習環境「@CODE ROOM」の概要

◆ 本科目でのプログラミングは，演習環境「@CODE ROOM」で行う．

演習環境「@CODE ROOM」の概要

第3講のまとめ

◆ プログラミング言語に関する知識を習得した:

- プログラミングは，プログラムを作成する作業のことであり，プログラミング言語を使用する.
- プログラミング言語は，プログラムを記述する人工言語であり，プログラミングパラダイムによって分類できる.

◆ 本科目で使用する Java の概要を理解した:

- Java の歴史と特徴
- Java によるプログラミングの手順
- 演習環境「@CODE ROOM」の概要

第1回　コンピュータとプログラミング

第3講　プログラミング

終わり

第1回　コンピュータとプログラミング

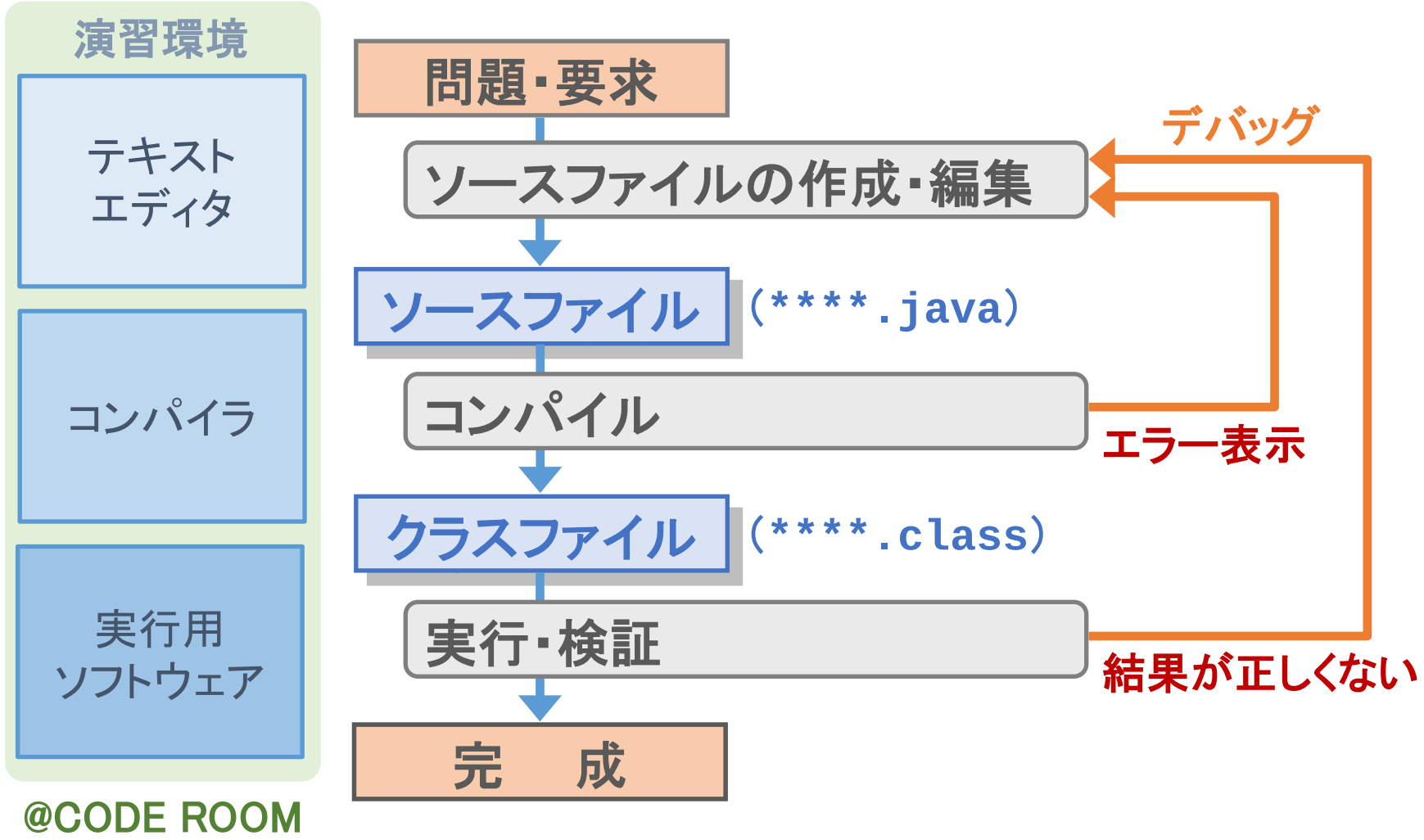
第4講　プログラミング環境「@CODE ROOM」

第4講の学習目標

◆ 本科目のプログラミング環境である「@CODE ROOM」の詳細を理解する:

- プログラミングの手順
  - ソースファイルの作成・編集
  - コンパイル・実行
  - （採点）
- 「@CODE ROOM」の利用方法

Javaによるプログラミングの手順



Javaによるプログラミングの手順

演習環境「@CODE ROOM」の概要（再掲）

◆ 本科目でのプログラミングは，演習環境「@CODE ROOM」で行う．

演習環境「@CODE ROOM」の概要

演習環境「@CODE ROOM」でのプログラミングの手順

例題1-1

標準出力（画面上）に「Hello!」というメッセージを出力（表示）するプログラムを作成してください.

◆ この例題を使用して，演習環境「@CODE ROOM」での Java による具体的なプログラミングの手順を理解する：

- ソースファイルの作成・編集
- コンパイル
- 実行・検証
- （採点）

ソースファイル

◆ プログラムを記述したファイルを**ソースファイル**という.

例題1-1

標準出力(画面上)に「Hello!」というメッセージを出力(表示)するプログラムを作成してください.

◆ 「例題1-1」のプログラムを記述したソースファイルの内容は以下のとおり:

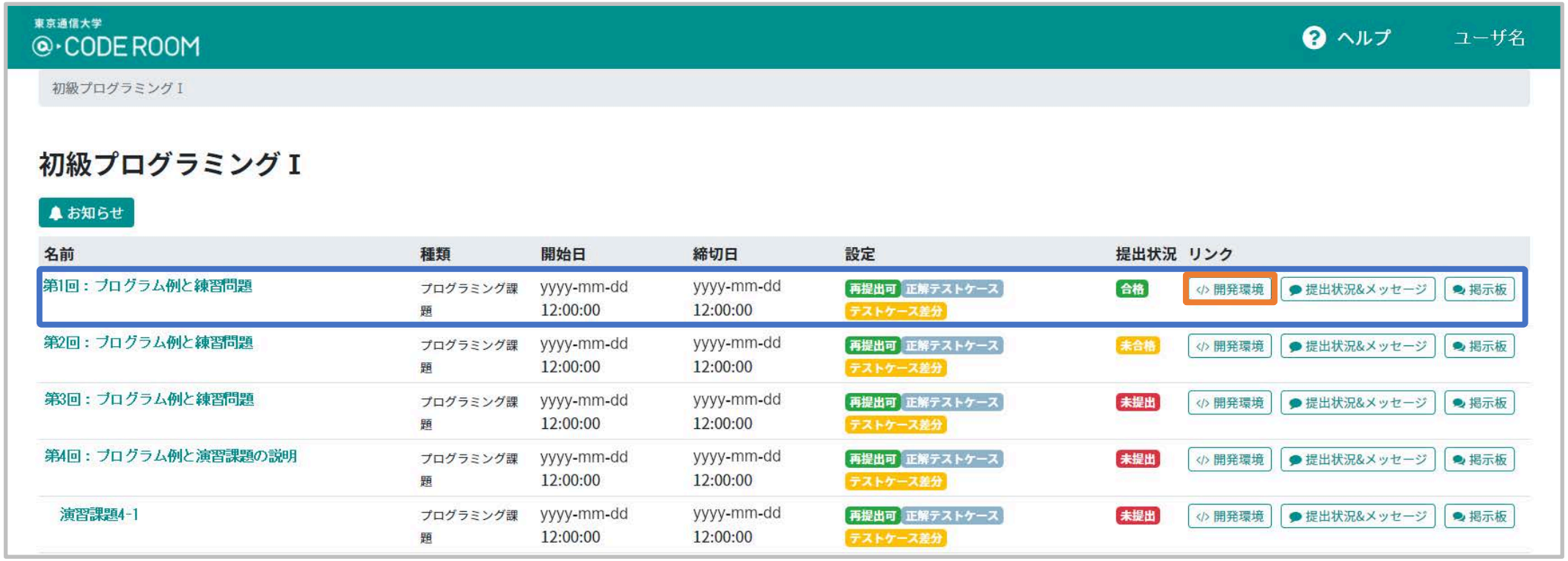
Example_01_01.java

```
1 public class Example_01_01 {
2     public static void main(String[] args) {
3         System.out.println("Hello!");     /* メッセージの表示 */
4     }
5 }
```

省略可能

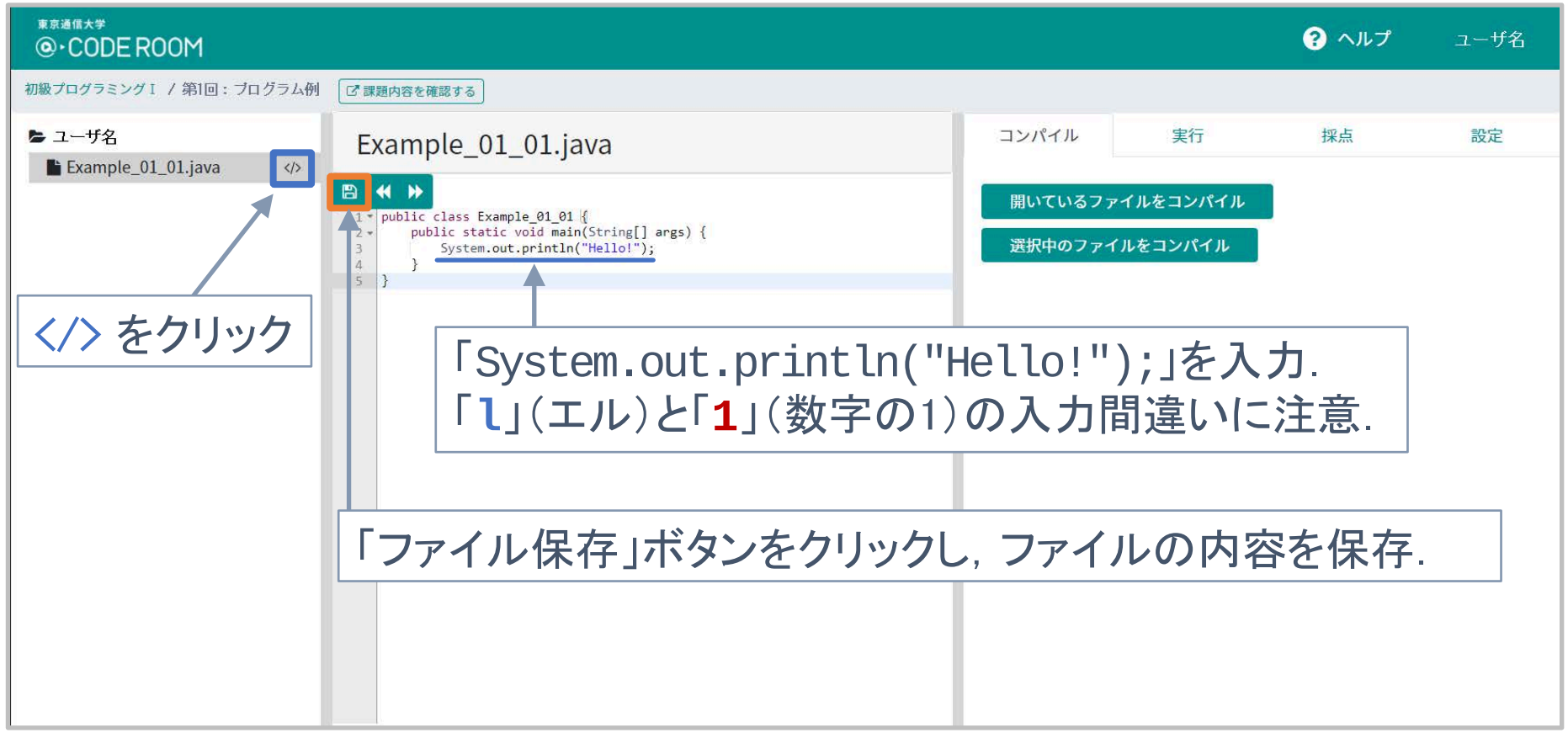
演習環境「@CODE ROOM」の課題一覧

◆ 演習環境「@CODE ROOM」の本科目に進むと，課題の一覧が表示される：
- 各回の講義内で示す「プログラム例」を掲載．
- 第4回，第8回では「演習課題」を出題．それ以外の回では「練習問題」を出題．
- 第1回の「開発環境」に進む．



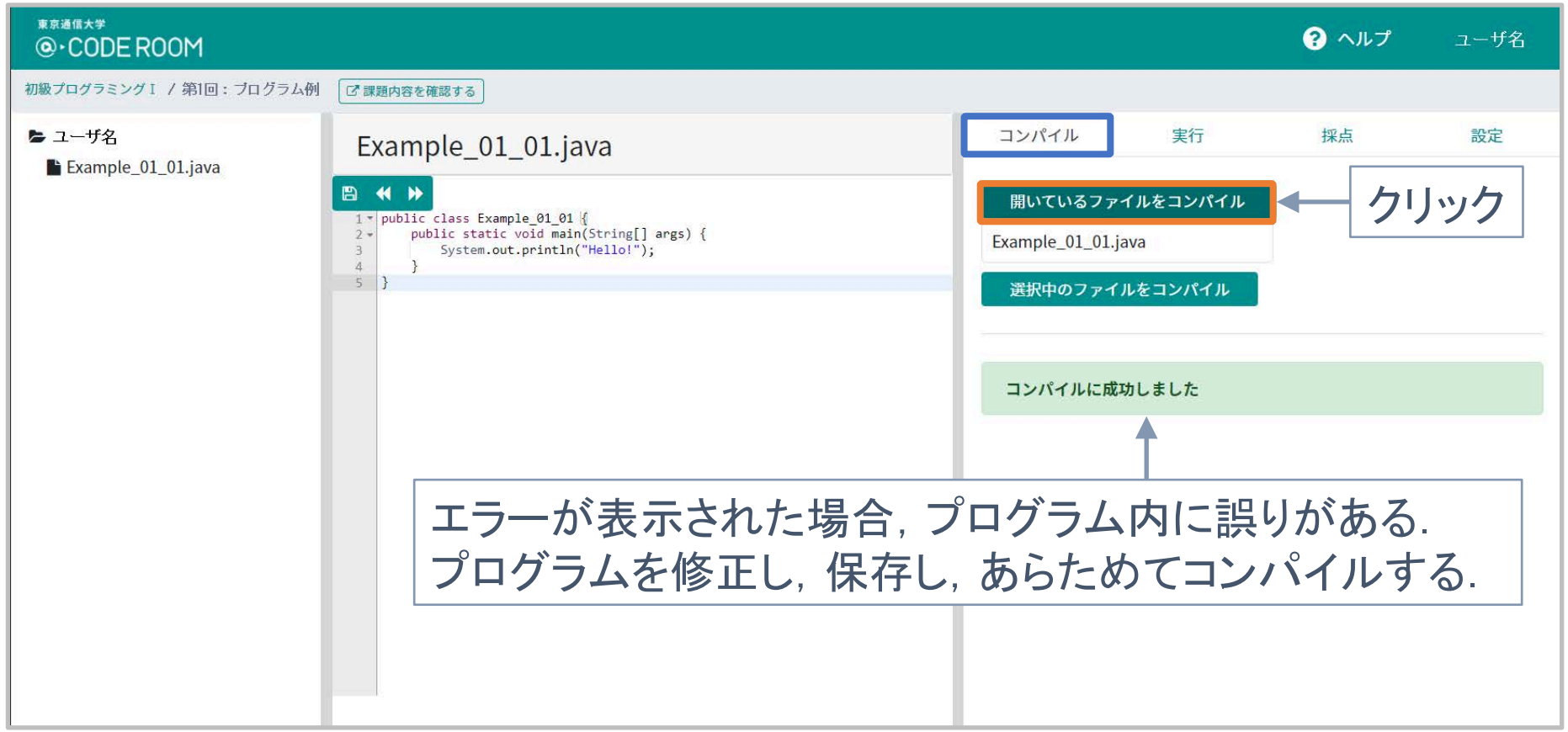
ソースファイルの編集

◆「例題1–1」のソースファイルを開き，プログラムの内容を記述する．
- マウスを「Example_01_01.java」に移動し，</>（ファイルを開く）をクリック．
- プログラムの3行目に「System.out.println("Hello!");」を入力．



コンパイル

- 「コンパイル」タブで，「例題1−1」のソースファイルをコンパイルする．
  - 「Example_01_01.java」を開いた状態で，「開いているファイルをコンパイル」をクリック．
  - 「コンパイルに成功しました」と表示されれば，コンパイル完了．



実行

- 「実行」タブで，「例題1−1」のソースファイルを実行する．
  - 「Example_01_01.java」を開いた状態で，「開いているファイルを実行」をクリック．
  - 「標準出力」に「Hello!」と出力（表示）されれば，成功．

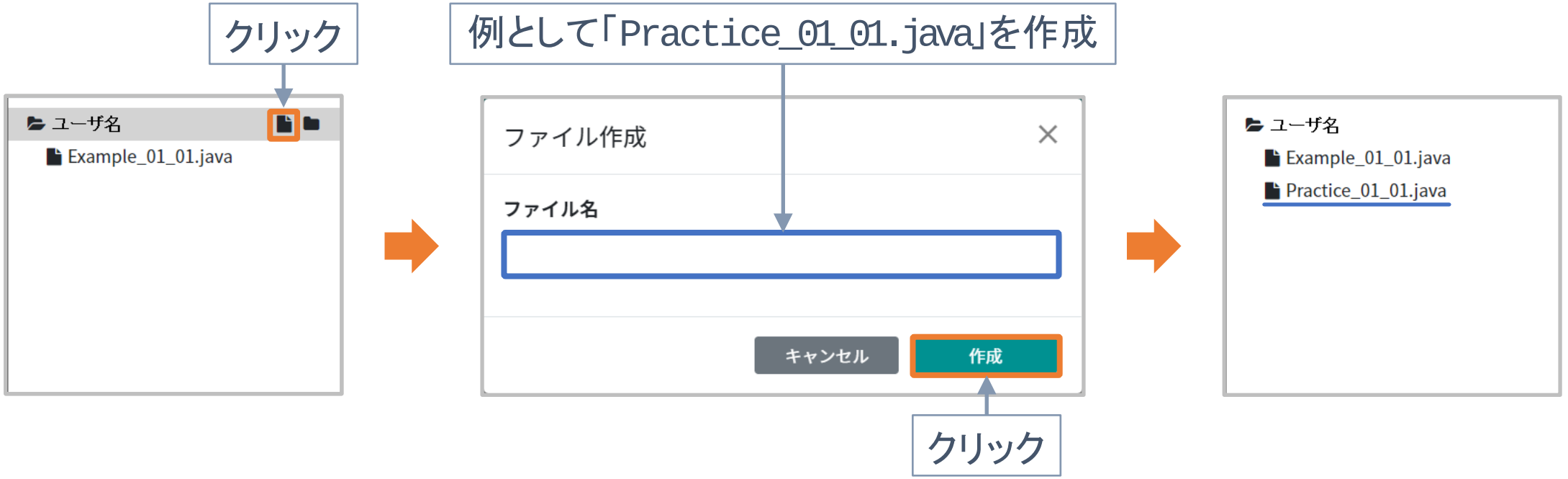
実行

（採点）

- 「採点」タブで，「例題1−1」のソースファイルを採点する．
  - 「Example_01_01.java」を開いた状態で，「開いているファイルを採点」をクリック．
  - 採点を行うことで，ソースファイルの内容を提出したことになる．

ソースファイルの作成

◆「@CODE ROOM」では，各自でソースファイルを作成することができる．
- マウスを「ユーザ名」に移動し，「ファイル作成」ボタンをクリック．
- 「ファイル作成」ウィンドウの「ファイル名」欄に，作成したファイル名を入力し，「作成」ボタンをクリック．
- ファイル一覧に，作成したファイルが表示される．



演習環境「@CODE ROOM」の利用方法（1）

◆ 演習環境「@CODE ROOM」の本科目に進むと，課題の一覧が表示される：

- 各課題の内容は，「名前」欄の課題名をクリックして確認する．
- 課題の「締切日」（講義期間の終了日）を確認する．

演習環境「@CODE ROOM」の利用方法（2）

- 「演習課題」は，「提出状況」欄で提出状況を確認する：
  - 「合格」：採点（提出）した上で「100点」を取っている．
  - 「未合格」：採点（提出）しているが「100点」は取れていない．
  - 「未提出」：採点（提出）していない．

演習環境「@CODE ROOM」の利用方法（3）

◆ 各課題の「演習環境」に進み，プログラミングの作業を行う：
- 各回で示した「プログラム例」の内容を確認する．
- 「練習問題」のプログラムを作成する．
- 「演習課題」のプログラムを作成し，採点（提出）する．

演習環境「@CODE ROOM」の利用方法（4）

◆ 質問する際は，以下に従うこと：

- 各自が作成したプログラムの内容についての質問は，「提出状況&メッセージ」．
- クラス全体に関わる質問は，「掲示板」．

第4講のまとめ

◆ 本科目のプログラミング環境である「@CODE ROOM」の詳細を理解した:

- プログラミングの手順
  - ソースファイルの作成・編集
  - コンパイル・実行
  - （採点）
- 「@CODE ROOM」の利用方法

第1回　コンピュータとプログラミング

第4講　プログラミング環境「@CODE ROOM」

終わり