



---

# プログラミング言語

名倉 正剛

日本大学 工学部 情報工学科

70号館7044号室

---



# プログラミング言語 第1回

---

## ■ ガイダンス

□ 授業の目的. 進め方. 日程. 成績評価



# Who am I?

---

## ■ 教員

### □ 名倉 正剛 (61号館404号室)

- nag@cs.ce.nihon-u.ac.jp
- 内線: 8844 ソフトウェアシステム研究室 (の一つ)
- 今年から新任

## ■ オフィスアワー

### □ 金曜4限, 61号館404号室

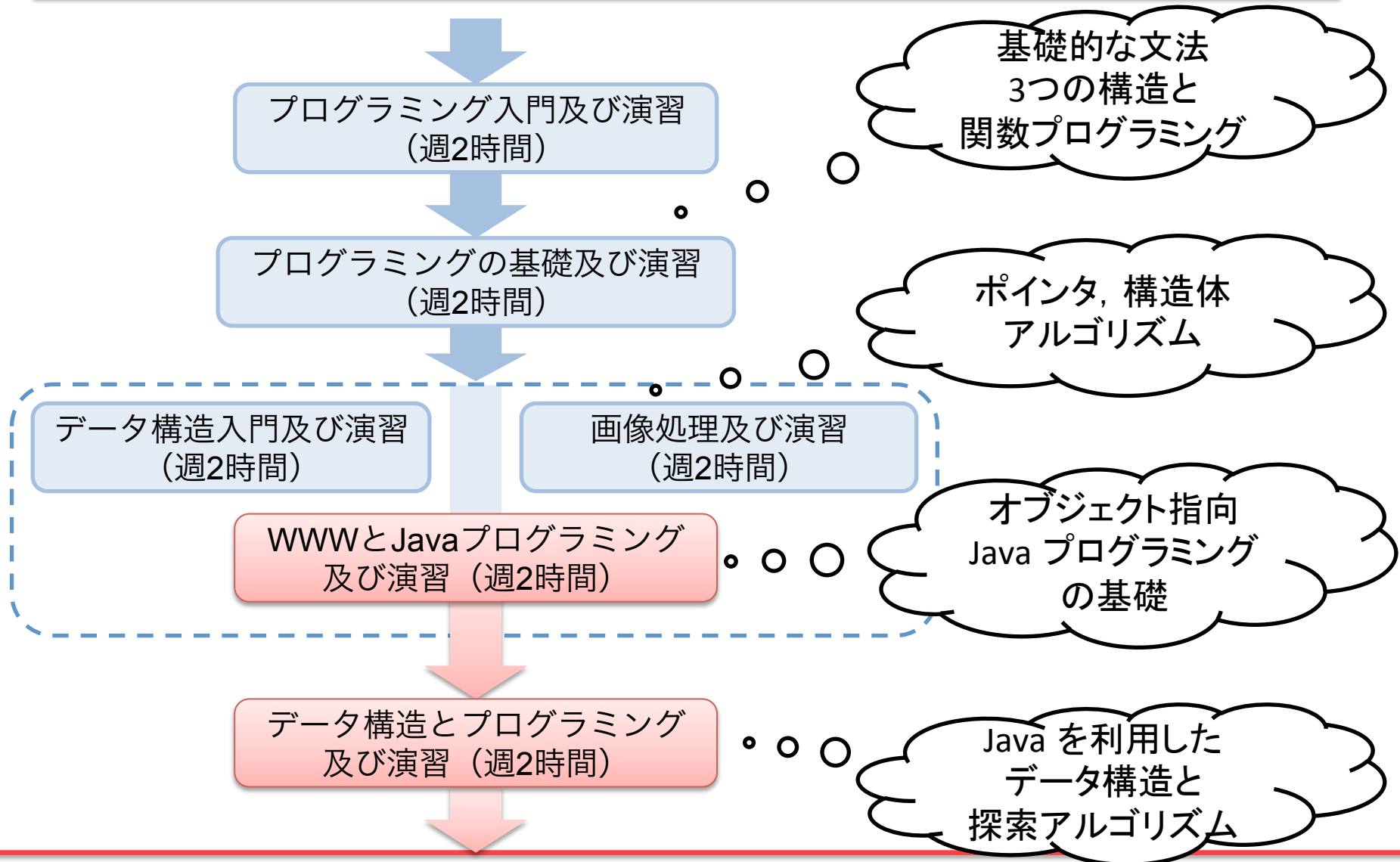
- 他の時間でも, いれば構いません (平日は多分いる).
  - 質問等を直接聞きたい時
  - 授業以外の事で何か質問・相談がある時

...

- コーヒーとかもあるので気軽にどうぞ.

### □ 質問などがある場合は, 研究室に直接来るかメール等で連絡してください.

# プログラミング系科目のカリキュラム





# 授業の目的

---

## ■ プログラミング言語の共通概念の理解

- 今までの科目 : ある特定のプログラミング言語を題材に, コンピュータのプログラミングの考え方を学習

→ でも, 各科目でプログラミングはある程度共通していると思いませんでしたか?

- この科目 : プログラミング言語の体系, 共通概念を習得



## (いきなり) 復習

---

- こんなプログラムを書いてみてください.  
C 言語と, Java 言語の2種類で作成すること.

問題1:

画面上に「Hello, World!!」を表示するプログラム

問題2:

1から入力された数までの総和を求めるプログラム.  
ただし, 引数で整数を受け取り, 総和を返す関数  
を作成して利用するものとする.

# 問題1: Hello, World!! の表示 (C言語)

---



- hello.c ファイルを用意する.
  - main 関数を定義して, 手続きを記述.
  - 標準入出力関数を呼び出して表示する.
- コンパイルして実行する.

```
#include <stdio.h>

int main(void) {
    printf("Hello, World!!");
    return 0;
}
```

# 問題1: Hello, World!! の表示 (Java言語)

---

- Java プログラム名のクラス (Hello クラス) を用意する
  - main メソッドを定義して, 手続きを記述.
  - 標準ライブラリに用意されている println メソッドを呼び出して表示する.
- コンパイルして, JavaVM 上で実行する.

```
public class Hello {  
    public static void main(String[] args){  
        System.out.println("Hello, World!!");  
    }  
}
```



## 問題2: 総和を求めるプログラム(C言語)



- total.c ファイルを用意する.
  - total 関数内で for ループで合計を算出.
  - main 関数を定義して, 手続きを記述.
  - 標準入出力関数を呼び出して入力, 表示する.
- コンパイルして実行する.

```
#include <stdio.h>

int total(int x){
    int i, total = 0;
    for (i = 1; i <= x; i++)
        total += i;
    return total;
}
```

(左からの続き)

```
int main(void) {
    int x, t;
    scanf("%d", &x);
    t = total(x);
    printf("合計=%d\n", t);
    return 0;
}
```

## 問題2: 総和を求めるプログラム (Java 言語)



- Java プログラム名のクラス (Total クラス) を用意する.
  - total メソッド内で for ループで合計を算出.
  - main メソッドを定義して, 手続きを記述.
  - 標準ライブラリに用意されている Scanner を呼び出して, 入力する.
  - 標準ライブラリに用意されている println メソッドを呼び出して表示する.
- コンパイルして, JavaVM 上で実行する.

```
import java.util.Scanner;

public class Total{

    static int total(int x){
        int i, total = 0;
        for (i = 1; i <= x; i++)
            total += i;
        return total;
    }
}
```

(左からの続き)

```
public static void main(String[] args){
    int x, t;
    Scanner sc = new Scanner(System.in);
    x = Integer.parseInt(sc.nextLine());
    sc.close();
    t = total(x);
    System.out.println("合計=" + t);
}
}
```

# 言語が変わると何が違う? (HelloWorld)



## ■ C vs. Java

- 関数 vs. メソッド
- モジュール vs. オブジェクト (クラス): どちらもOS上ではファイルであることに注意
- ネイティブコード実行 vs. バイトコード実行

### 問題1: Hello, World!! の表示 (C言語)



- hello.c ファイルを用意する.
  - main 関数を定義して, 手続きを記述.
  - 標準入出力関数を呼び出して表示する.
- コンパイルして実行する.

```
#include <stdio.h>

int main(void) {
    printf("Hello, World!!");
    return 0;
}
```

プログラミング言語

### 問題1: Hello, World!! の表示 (Java言語)



- Java プログラム名のクラス (Hello クラス) を用意する
  - main メソッドを定義して, 手続きを記述.
  - 標準ライブラリに用意されている println メソッドを呼び出して表示する.
- コンパイルして, JavaVM 上で実行する.

```
public class Hello {
    public static void main(String[] args){
        System.out.println("Hello, World!!");
    }
}
```

プログラミング言語



# 言語が変わると何が違う? (Total)

## ■ C vs. Java (HelloWorld の場合に加えて)

□ 関数定義 vs. メソッド定義

## ■ あとはそれほど違いがない?

□ 制御構造とか, 変数のスコープとか.

### 問題2: 総和を求めるプログラム(C言語)

- total.c ファイルを用意する.
  - total 関数内で for ループで合計を算出.
  - main 関数を定義して, 手続きを記述.
  - 標準入出力関数を呼び出して入力, 表示する.
- コンパイルして実行する.

<pre>#include &lt;stdio.h&gt;  int total(int x){     int i, total = 0;     for (i = 1; i &lt;= x; i++)         total += i;     return total; }</pre>	(左からの続き) <pre>int main(void) {     int x, t;     scanf("%d", &amp;x);     t = total(x);     printf("合計=%d\n", t);     return 0; }</pre>
--	--

プログラミング言語

### 問題2: 総和を求めるプログラム(Java言語)

- Java プログラム名のクラス (Total クラス) を用意する.
  - total メソッド内で for ループで合計を算出.
  - main メソッドを定義して, 手続きを記述.
  - 標準ライブラリに用意されている Scanner を呼び出して, 入力する.
  - 標準ライブラリに用意されている println メソッドを呼び出して表示する.
- コンパイルして, JVM 上で実行する.

<pre>import java.util.Scanner;  public class Total{      static int total(int x){         int i, total = 0;         for (i = 1; i &lt;= x; i++)             total += i;         return total;     } }</pre>	(左からの続き) <pre>public static void main(String[] args){     int x, t;     Scanner sc = new Scanner(System.in);     x = Integer.parseInt(sc.nextLine());     sc.close();     t = total(x);     System.out.println("合計=" + t); }</pre>
---	---

プログラミング言語

# 授業の目的（再掲＋α）

---

## ■ プログラミング言語の共通概念の理解

- 今までの科目：ある特定のプログラミング言語を題材に，コンピュータのプログラミングの考え方を学習
- この科目：プログラミング言語の体系，共通概念を習得

## ■ 達成目標

- コンピュータサイエンスの専門的な研究を行う研究者として知っておくべき，プログラミング言語共通概念の理解
  - ・ 言語共通概念と言語固有概念を意識する



# 授業の日程

※変更する可能性あり

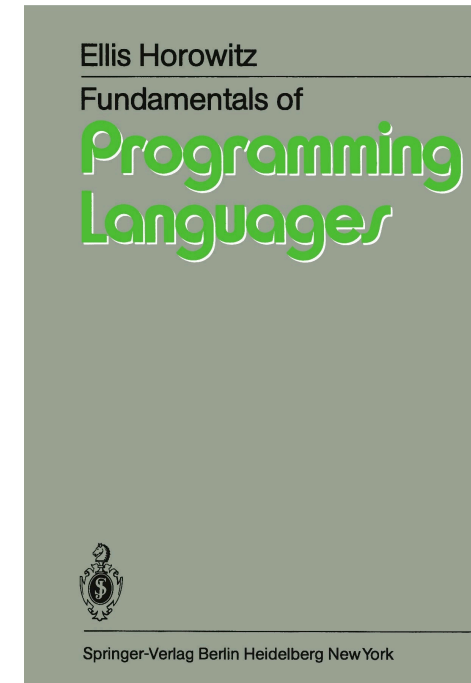
9/17	ガイダンス・復習	11/12	モジュール
9/24	プログラム言語 の歴史	11/19	データ抽象化
10/1	構文, 意味, BNF	11/26	(月曜授業日)
10/8	識別子, 変数	12/3	(土曜授業日)
10/15	ステートメント, 文, 式	12/10	例外処理
10/22	スコープ	12/17	並行, 排他制御
10/29	手続き, 制御構造	12/24	関数型言語, OO 言語, 言語の分類
11/5	中間試験	1/14	まとめ
		1/21	期末試験

## 参考書 ※教科書はなし

---

- "Fundamentals of Programming Languages"

- Ellis Horowitz (著)
- Springer
- \$99.00



- 買っても良いが買う必要はない.

- 読みたければ, 相談してください

- シラバスに日本語の参考書も載せたけれど, こちらもあくまで参考書です.

# 試験と成績

---

## ■ 成績

- 期末試験60%. 中間試験40%

## ■ レポート

- 何回かレポートを課す予定（まだ未定）
- 全て提出する必要がある
- 未提出 ⇒ 不合格（試験の受験資格を失う）
- 再提出は無しにするつもり. 内容にあまりにも不備がある場合は, 未提出扱いになることがある.

## ■ 出席（カードリーダー）

- 原則として4回以上欠席した場合に, 試験の受験資格を失う





# 注意事項

---

- 受講にあたっては、「プログラミングの基礎及び演習」の単位を習得していることが望ましい。
- ポータルサイトに授業資料をアップロードしておく
- ポータルサイトをチェックしておくこと
  - 連絡事項を掲示する場合もある