

インターネットプログラミング
教室2401
水曜日7限19:50-21:20

第1回
2013/9/11

岩井将行

授業資料

- <http://www.cps.im.dendai.ac.jp>
- <http://www.cps.im.dendai.ac.jp/index.php?Classes%2F2013InternetProg>

講師紹介

- <http://cps.im.dendai.ac.jp/index.php?Members%2Fiwai>
- 岩井研究室
- <http://cps.im.dendai.ac.jp>
- 岩井研究室の研究分野
- <http://cps.im.dendai.ac.jp/index.php?Research%2FTopics>
- 連絡先 1号館11F 11107b
- iwaiあっと im.dendai
- 研究室内線2844

講師

- 慶應義塾大学 卒
- 元東京電機大学 戸辺義人先生
OSOITEProject参加
- 東京大学生産技術研究所 助教
- 2013.4月より未来科学部情報メディア学科
准教授

- 詳しくはFacebook 岩井将行
- Twitter @masaiwai

TA・SA・副手

- 加藤 佳祐
- ユビキタスネットワークング研究室
- 東京電機大学 未来科学部 情報メディア学科
- E-mail: Keisuke Kato
<case.unl@gmail.com>

副手紹介

- 鉄谷研究室M2
 - 加藤 佳祐 case先輩

成績

- 出席
- 毎回課題
- 中間試験
- 最終試験 + 最終課題

本日の内容

- Webアプリケーションの全体像
- Java の位置づけ
- Java 最初の一歩

この授業の最終ゴール

プログラミングを通して以下のことを学び取ること
漠然としたアイデアを具体的な「目的」に置き換える能力
「目的」を実現するために必要なプロセスを正確に順序立てて組み立てる能力
組み立てたプロセスを正確かつ簡潔に記述する能力
ネットワークプログラミングを理解する。

「論理的思考・表現能力」をつける。

今日の目標

- オンライン教材を見られるようになります。
- windowsの操作を復習します。
- 資料をみます。
- 初めてのプログラミングを書きます。
- コンパイルします。
- 初めての課題を提出してみます。

ルール

- メールなどではSubject、氏名、学籍番号などを明記
- –社会人として交渉する姿勢が必要。
- 他人のメールやデータを覗いたりして、プライバシーの侵害をしない
- 課題は他人のものを丸写ししない。WEBから丸写ししない。
- 絶対あきらめないでまず自分で考える。
- できた人はできていない人に教えてあげる。
- なにが分かって何が分からないかを明確にする。

スケジュール

第1回Javaプログラムの基礎

第2回条件分岐

第3回繰り返し

第4回メソッド

第5回配列

第6回中間試験

第7回クラスとインスタンス(1)

第8回クラスとインスタンス(2)

第9回スレッド

第10回インタフェース

第11回GUI

第12回例外

第13回スレッドを用いたGUI

第14回予備

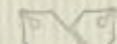
第15回学力考査

Javaプログラミング

- プログラミングとは？
- プログラムを作るための方法
- プログラムとは？
- 決まった手順を決まったように実行
- するようにコンピュータに渡す命令群
- 信号機プログラム
- 信号を赤にしろ→30秒まで→信号を青にしろ→30秒まで→信号を黄色にしろ→3秒まで→(最初にもどれ)

プログラムとは？

- コンピュータが実行する命令の集まり
 - Internet Explorer, MS Word, emacs, iTunes ...
 - ユーザが実行して、操作します
- プログラミング言語
 - プログラムを書くための**人間用の言語**
 - C言語、Java言語、シェルスクリプト ...
- プログラミング
 - プログラミング言語を使ってプログラムを作ること



プログラム

- ・ 命令として与えられている以外のこと是一切しません。
- ・ 命令は細かく与える必要があります。コンピュータは人間のように賢くありません。細かく順序正しく指示しないと
・ いけません。
- ・ コンピュータは命令を忠実に最後までやり遂げます。もし、1万回同じことを繰り返しやるように指示されると、コンピュータはその通り1万回繰り返します。
- ・ それぞれの命令は単純ですが、それを非常に速く行うことができます。

ファイルの種類について

ソースファイル・ソースコード

- Java言語で書かれたプログラムファイル
- *.java

クラスファイル 中間コード

- コンパイラによって生成されたファイル
- *.class

```
import org.w3c.dom.*;  
import org.xml.sax.*;  
import javax.xml.parsers.*;  
  
public class ChessboardDOMPrinter {  
    private DocumentBuilder builder;
```

ソースコード

コンパイル作業

```
XOXOXO  
XOXOXO  
XOXOXO  
XOXOXO
```

中間コード

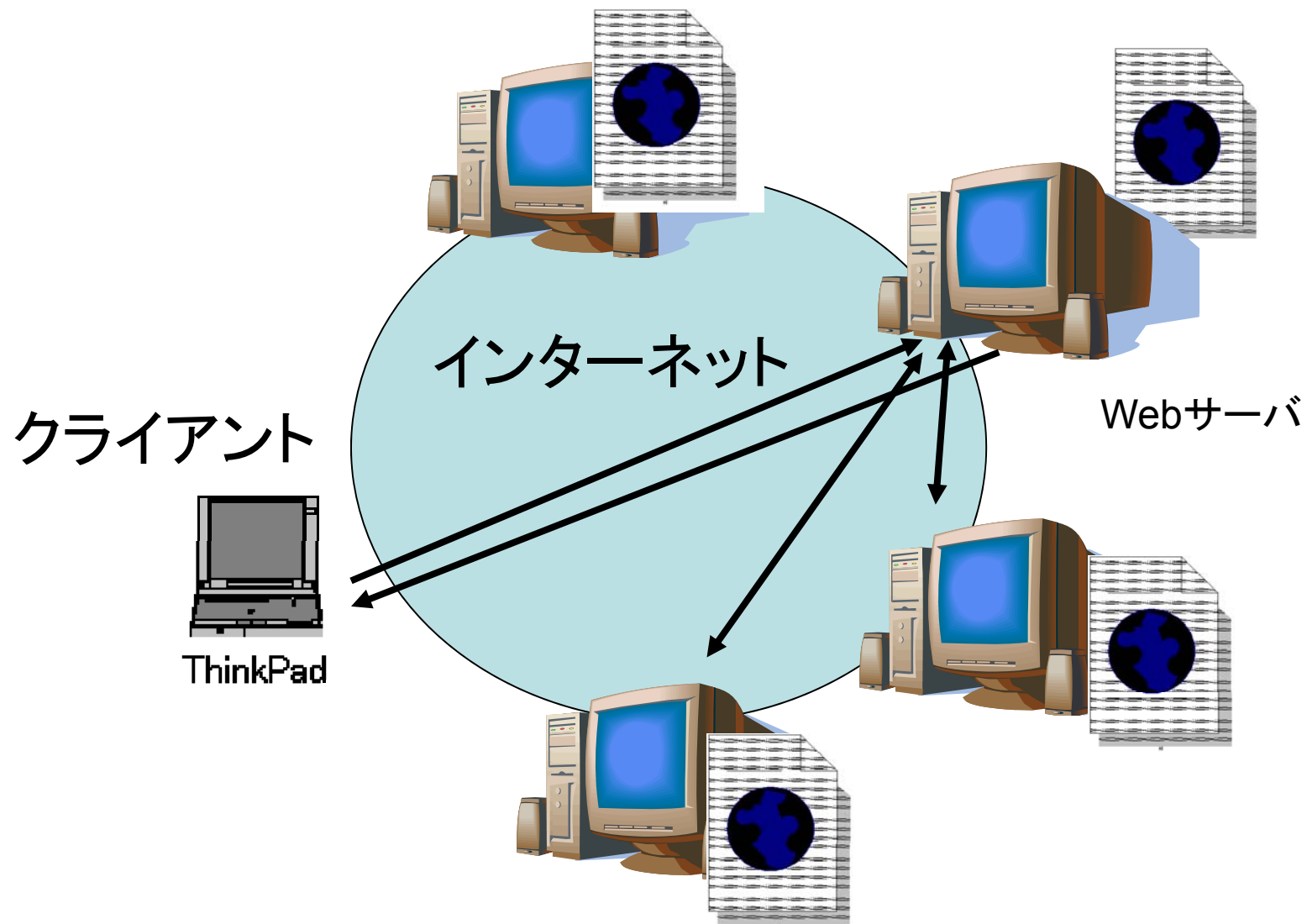
JavaVM

JavaVM

JavaVM

機械に応じた命令後

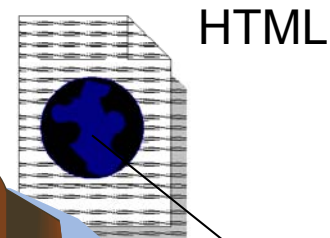
Webアプリケーション全体像



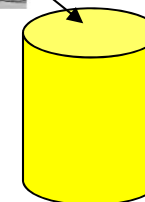
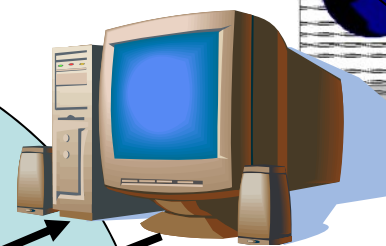
Java アプレット

Applet =
“Application” + “-let”

Webサーバ



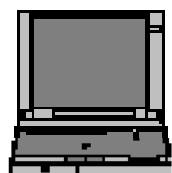
HTML



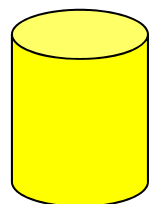
Weather.class

インターネット

クライアント



ThinkPad



Java VM

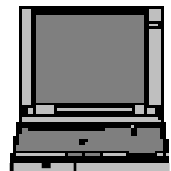
クライアントで実行

Java サーブレット

Servlet =
“Server” + “-let”

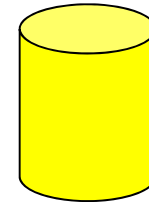
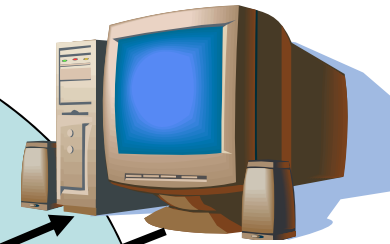
Webサーバ

クライアント



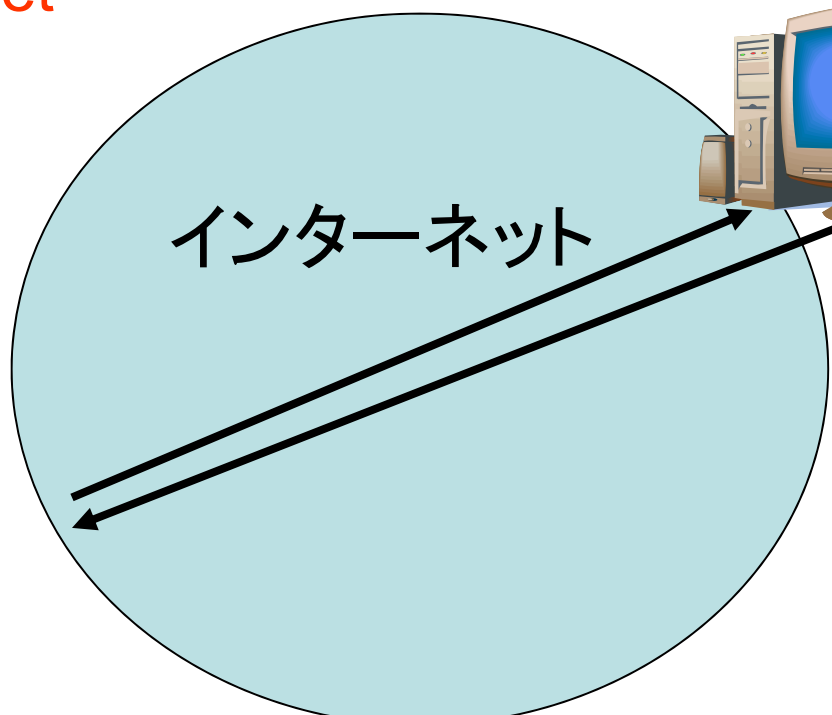
ThinkPad

インターネット



Account.class

サーバで実行



JDK

- JDKはJava Developers Kitの略
- Javaの開発環境として提供
- Javaでの開発のために必要な、コンパイラなどのツール群
- Oracleのホームページから無償でダウンロードが可能
- JDKの構成
 - Javaインタプリタ(java)
 - Javaコンパイラ(javac)
 - Javaデバッガ(jdb)
 - 逆アセンブラ(javah)
 - C言語コードへの変換ツール(javah)
 - ドキュメント生成ツール(javadoc)
 - JDK バージョンからの移行ツール(upgrade)
 - 標準クラス・ライブラリのパッケージ群
- JDKによる開発手順
 - (1) ソース・ファイルのエディット
 - (2) javacによるコンパイル
 - (3) 生成されたバイト・コードの実行

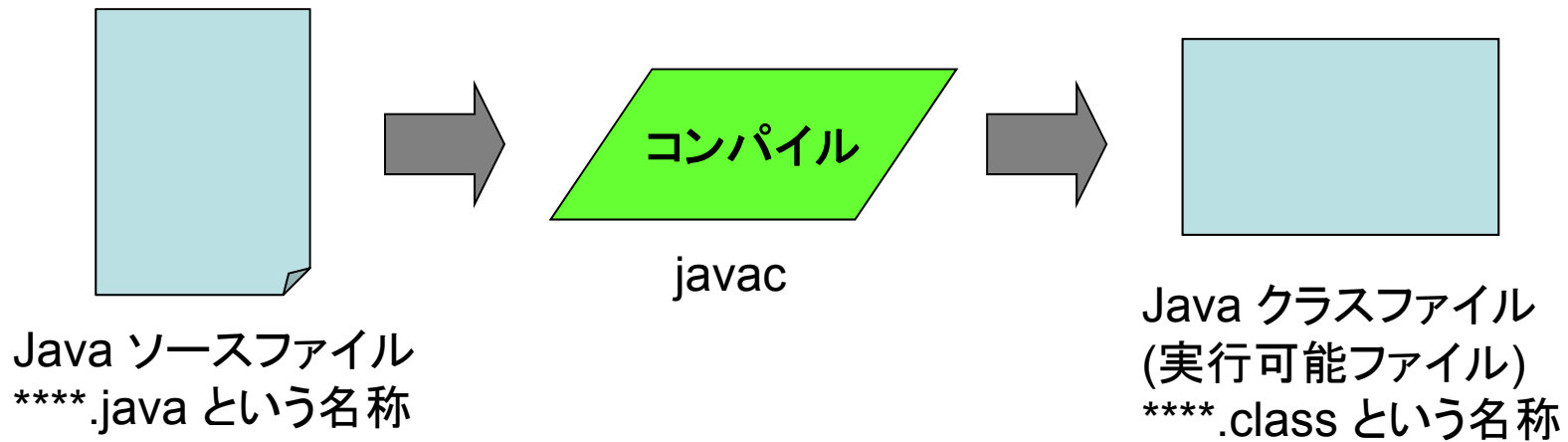
コンピュータ教室での環境

- エディタ TeraPad
- コマンドプロンプト上で、
JDKコマンドを動作させる

数回か後からeclipseを使います。

コンパイル

- 人間が書いたプログラムをコンピュータが理解できるような状態に変換
- 正常にコンパイルできると****.classというファイルができる
- コンピュータが理解できる状態になったプログラムがJavaクラスファイル
- コンパイルするときに, javac というプログラム(コンパイラ)を使用



コマンドプロンプト

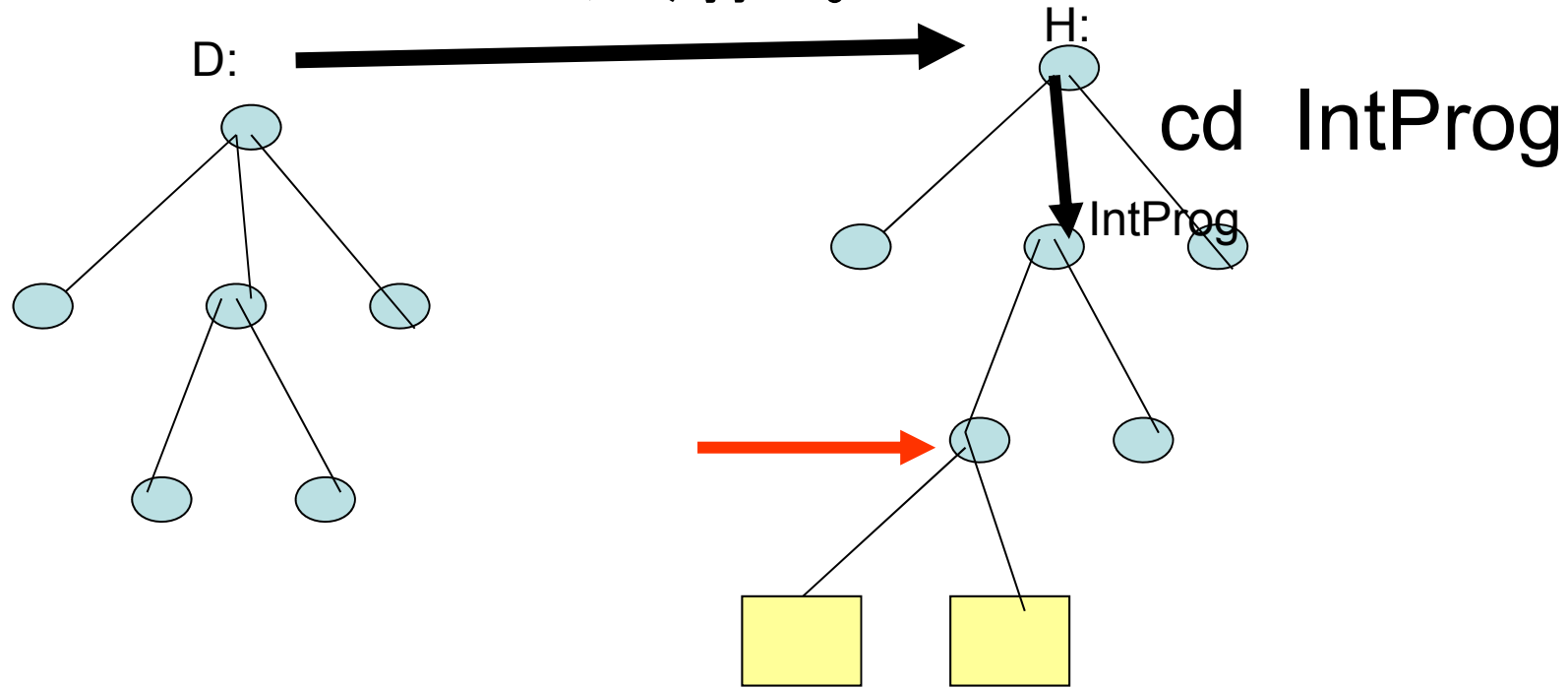
- コマンドプロンプトで以下のコマンドを打ち込んでみよう
- `javac -version`
- `java -version`

コンパイルの実際

- (1) コマンドプロンプトを開く。
- (2) h:
- (3) cd Java
- (4) dir で、Hello.java
があることを確認する。
- (5) javac Hello.java
- (3) エラーがなければ、Hello.class が生成される。
- (4) エラーメッセージがあれば、修正する。

フォルダ=ディレクトリ

H: とだけ打つ。



dir は、そのフォルダ
にあるファイルのリスト
を表示させるコマンド

Javaファイル

HTMLファイル

プログラムの実行

```
java Hello
```

- Hello.java を編集して表示される文字を変えてみよう

Hello.java

```
public class Hello {  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello! ¥”岩井<ん¥" ¥n ");  
        System.out.print("A");  
        System.out.print("B");  
        System.out.print("C");  
    }  
}
```

出力

```
>java Hello
```

```
Hello! “岩井くん”
```

```
ABC
```

Java プログラムの基本

クラス

```
public class Hello {
```

```
    public static void main(String[] args){
```

```
        // この中に、処理内容を書きます
```

```
    }
```

```
}
```

メソッド

注意1: Hello の部分がプログラム名

注意2: String は、文字列。String[] 文字列の配列

注意3: args は、コマンド引数の配列

args[0], args[1]

基本型

- boolean 論理型 (true または false)
- char 整数型(文字型) (0以上65535以下)
- byte 整数型 (-128から127まで)
- int 整数型 (符号付き32ビット)
- long 整数型 (符号付き64ビット)
- float 実数 (単精度浮動小数点型)
- double 実数 (倍精度浮動小数点型)

演算子

- +
- -
- *
- /
- % (余り)

ソースコードをダウンロード

- http://www.cps.im.dendai.ac.jp/index.php?Classes%2F2013NC_InternetProg#ld3383c1
- src2013_1.zip

引数に入れた2つの数字を加算する プログラムを作ろう

```
public class Calc1{  
  
    public static void main(String[] args) {  
        ???????  
  
    }  
  
}
```

String型を int 型に変換する方法

```
int abc = Integer.parseInt(String str)
```

```
String str_a = "12";
```

```
String str_b = "5"
```

```
int ans;
```

```
ans = Integer.parseInt(str_a) + Integer.parseInt(str_b)
```

コマンド入力読み込みCalc2

- 今はわからなくても良いです。
 - BufferedReader クラスのインスタンスreaderを作る
 - reader をオブジェクトを作って、
reader.readLine(); メソッドで読み込む。

ミニ演習

次の計算を答えを表示する
プログラムを作ろう

- $3 + 5$
- $18 - 7$
- 32×5
- $10 \div 2$
- $300 \div 12$ の余り

本日の課題

- AさんとBさんとCさんのテストの点数を入力し平均点を出力するプログラムをつくる。
- ヒント1（文字列連結）
- Calc1.java
- ヒント2（二つの数をたす）
- Calc2.java

- クラス名: Calc3