

慶応義塾大学文学部  
応用情報処理V 2018

# ユーザインタフェースの作成

Chapter 11  
2018-12-14

新居雅行 Masayuki Nii Ph.D. in Engineering

 [nii@msyk.net](mailto:nii@msyk.net)  [msyknii](https://www.facebook.com/msyknii)  [msyk\\_nii](https://twitter.com/msyk_nii)

# オブジェクト指向の2つの側面

- ・ **クラスを自分で定義して利用する**
  - ・ これまでに説明した通り
  - ・ 自分で作っているなので、中身を含めて詳細を理解している
- ・ **他の誰かが作ったクラスを利用する**
  - ・ 現実にはこちらの作業が大半を占める
  - ・ 通常は大量のクラスが利用できる状態でプログラミングをスタートする（だから効率的）
  - ・ しかしながら、誰かが作ったものを理解しないといけない
    - ・ ソースコードを読むという選択肢もあるが、限度がある。  
また、ソースコードを公開していないこともある
  - ・ 通常は、クラスを開発した人たちによる「ドキュメント」を参照し、理解して、自分のプログラムで利用する

# ソフトウェアを部品として捉える

- **コンポーネント（ソフトウェアの部品）**
  - 例えば、1つのクラスを利用することで「ポップアップメニュー」と言ったGUI構築部品を利用できる
  - 正確には単一の1クラスで全て賄うわけではないが、利用者にとっては単一のクラス、具体的には単一のインタフェースで利用できるようになっていたものをコンポーネントと呼ぶ
  - 背後ではたくさんのクラスがあるかもしれないが、利用するときには可能な限り隠匿し、考えなくていいようにする
- **Javaは言語だが、多数のコンポーネント群を添付**
  - 例えば、StringやStringBufferも、Javaの世界で利用できるようになっていた「コンポーネント」の1つ
  - 後からコンポーネント群を追加することもできる

# AWTとSwing

---

- **Java標準のGUI構築コンポーネント群**
  - 通常は、以下のコンポーネント群がJavaの世界では使えるようになっていていると考えて良い
- **AWT (Abstract Windowing Toolkit)**
  - 最初のJavaから利用できるもの
  - きちんと作られているが、リリースを急いだのか「必要最低限のGUI構築ツール」的なスペック
  - 後にFlashに負ける原因の1つになってしまう
- **Swing**
  - Java 1.1に追加され、現在も使えるGUI構築ツール
  - かなり高度な機能群だが、リリース当時はかなり重かった

# Javaでの提供クラスの利用方法

- **単にクラス名があるだけでなく、パッケージの考え方**
  - クラス群をパッケージに分類（あらかじめ分類されている）
  - パッケージは階層的に分類される
    - 例：Stringは、java.langというパッケージに所属
    - 階層をドットで区切って表現する
  - クラス指定にパッケージ指定が含まれても良い
    - 例：java.lang.Stringと書いても良い
    - 逆にパッケージ名付きの名前が本来の名前だと思っても良い
- **コードの最初にimportで利用を宣言する**
  - `import javax.swing.JFrame;`
  - `import javax.swing.*; //javax.swingパッケージの全クラス`
  - なお、java.langは省略できるので、StringやSystemなどは、そのままクラス名だけを書ける

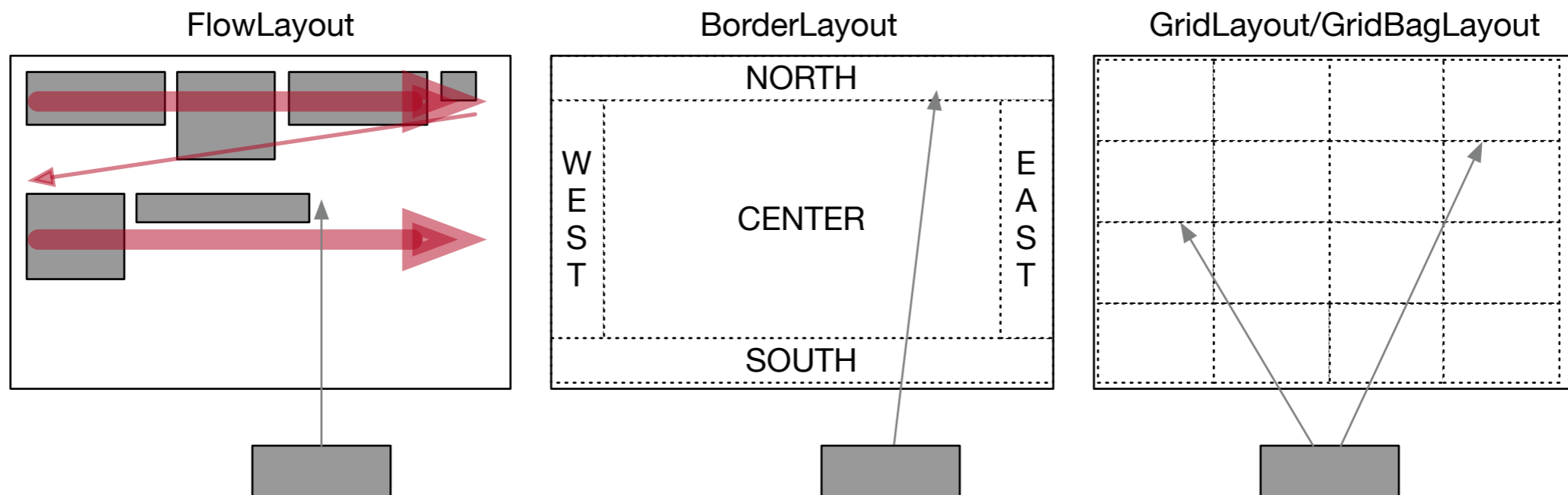
# 各コンポーネントの利用方法

---

- ・ **テキストの解説に、コードの理解に必要なものは記述**
  - ・ ただし、全てを説明することは不可能なくらい機能が多い
- ・ **マニュアルは？**
  - ・ 解説本や、ググるのもあるが...
  - ・ 原本はこちら
    - ・ <https://docs.oracle.com/javase/jp/8/docs/api/>
    - ・ テキストで出てきたクラスについて、理解を深めたい人は是非とも上記ドキュメントを読んでみてください

# レイアウト機能

- ・ 「画面に見える」ということの本質
  - ・ 画面表示されるオブジェクトに、別のオブジェクトを追加することで、見える（レイアウトされる）ようになる
  - ・ レイアウト可能なオブジェクトのクラスには、addメソッドが実装されている
- ・ 画面オブジェクトのレイアウト
  - ・ 座標や大きさを指定することなく、自動的にレイアウトできる機能
  - ・ 簡単なだけでなく、変化するウィンドウの大きさに追従することができる



# 講義のまとめ

---

- ・ **すでに作られたクラス群の使いこなしが極めて重要**
- ・ **JavaでGUIを構築するためのAWTとSwing**
  - ・ 様々な部品が定義されている
  - ・ 開発者が想定した使い方をしなければならない
  - ・ レイアウト機能のような高度な仕組みも用意されている