

実験開始

河口研 助教
梶 克彦

kaji@nuee.nagoya-u.ac.jp

IB電子情報館北棟4F 403

内線：3942

まずは各自でやってみましょう

- サンプルシステムの改良
- 例：
 - マップを複数保存できるようにする
 - 掲示板とマップを組み合わせしてみる
- Webシステムの作り方を理解しましょう

新規プロジェクト

- 一からWebシステムを作るとき
 - Eclipseの左側の空白で右クリック-[New]-[Project..]
 - Tomcatプロジェクトを選ぶ
 - プロジェクト名を適当に入れる
 - Finish
 - WEB-INF\web.xmlはどこか他からコピーしてくる
- 既存のプロジェクトをコピーしてもよい
 - コピー後は既存プロジェクトの設定を一部修正
 - プロジェクトを右クリック-[Properties]-[Tomcat]
 - [全般]タブの[コンテキスト名]を
コピー後のプロジェクト名に変更
- http://localhost:8080/project_name/index

新たなControllerを追加したとき

- WEB-INF/web.xmlを編集
 - URLとクラスの対応付け
- Tomcatの設定を更新
 - プロジェクト右クリック-[Tomcatプロジェクト]
-[コンテキスト定義を更新]
- Tomcatの再起動
 - 猫ボタンの一番右

自分のプログラムを バージョン管理しましょう

- 例：sample_helloを自分のアカウントの
スペースで管理
- 今のバージョン管理から切断
 - sample_helloを右クリック-[Team]-[Disconnect]
 - Also delete ...を選んでOK
- 自分のアカウントにコミット
 - sample_helloを右クリック-[Team]-[Share Project]
 - [SVN]-Create a new RepositoryにチェックしてNext
 - 自分のアカウントのURLとユーザ名・パスワードを
入力

Subversion基本知識

- Checkout : プロジェクトをダウンロード
- Commit : ローカルデータをアップロード
Commitする際には必ずコメントを書く
- Update : ローカルデータを最新に更新
- Conflict : 衝突
- Revision : バージョン. コメントを頼りにバージョンを戻したりできる
- Branch : 枝分れして開発

プログラム共有の心得

- 人が**理解できる**プログラムを書く
 - クラス, 関数, 変数名は体を表すように
 - ×String tmp=“filename.txt”;
 - アルゴリズムはできるだけ**簡潔**に
 - ×入れ子がたくさん

```
for(;;){  
    if(){  
        while(){  
            for(;;){...
```

 - プログラム内に**コメント**を書く
- 分担できるように設計する
 - **モジュール性**を高める

本実験でやること

- Javaを使ってWebシステムを作る
- ただし一つ以上のWebサービスと連携する
- 評価項目：
 - Webシステムのクオリティ
 - オリジナリティ, 有用性, シンプルさ
 - 連携させるWebサービスの組み合わせ
 - 正常に動作するか
 - チームによるプロジェクト推進能力
 - 設計, 分担, 協力
 - プログラムの可読性, 構造

このテーマは何の役に立つの？

- 4年生以降の研究推進に役立ちます
- 世の中に存在しない有用なシステムを構想
 - 研究に求められるものは第一にオリジナリティ
- プログラミングの基礎を身につける
 - ほぼすべての研究はプログラミングが必須
- マッシュアップの概念を身につける
 - ほぼすべての研究は様々な知見や技術の組合せ
- 研究成果を社会に提供する窓口を自分で構築できる

大体のスケジュール

- ~~1:基礎知識に関する講習~~
- ~~2-3:サンプルシステムの改良~~
- 4:Webシステム仕様決定
- 5:チームごとにミニプレゼン
- 6-9:システム実装
- 10:最低限のWebシステム完成
- 11-14:デバッグ, 安定性向上, 洗練
- 15:報告書作成
- 16:研究成果報告会

これからチームで作業します

- Webシステムのアイデア出し
- 仕様決定（UML, イメージ図）
- ミニプレゼン準備, 発表
- 分担決定
- 実装を進める
- 随時テスト, 仕様変更
- 注意事項
 - 現実的な仕様にする
 - 各自の興味と能力に合わせて分担する
 - シンプルでも面白いものは作れる

チーム

- 仲良しの馴れ合いではうまくいかない
- チームをまとめ上げるリーダーの存在
- One for All, All for Oneの精神
- 興味や能力に合わせた適材適所
- コミュニケーションが何より重要

Webシステム例

- アカウント管理 : Google, Yahoo!
- マップ : Google Maps, YOLP, OpenStreetMap
- ドキュメント : Google Docs, Evernote
- TODO : Remember The Milk
- メール : Gmail
- ブックマーク : はてなブックマーク
- スライド : SlideShare
- 画像 : Flickr,
- 検索 : Google, Yahoo, Bing
- ストレージ : Dropbox, SkyDrive
- 動画 : YouTube, ニコニコ動画

※これらはほんの一部です

使えそうなWebサービスを探してみましょう

- マイナビ：知っておきたいWebサービスジャンル別リスト

<http://news.mynavi.jp/special/2009/wslists/index.html>

- 自分で探して、使い方を調べて、使ってみる
 - Javaで使えるか？
 - 簡単か？
 - ドキュメントはちゃんとしているか？

他の技術の紹介

- Ajax
 - ページ遷移なしでサーバ・ブラウザ間通信
 - 余裕があればぜひ使ってもらいたい技術
- UML
 - 仕様設計のモデリング言語
- Play! Framework
 - JavaのWebシステムを簡単に作るフレームワーク
- Google App Engine, Slim3
 - クラウドプラットフォーム

随時進捗をヒアリングします

- システムの実現性・新規性があるか
- 分担は妥当か
- 進捗は順調か