

参考文献を翻訳したもの(背景や目的に使用できるもの)

How is a project translated?

Translating a C project is as easy as doing a regular build of it. During compilation our system collects all the required data (C code, linked libraries, ...). After a successful build, the collected data can directly be translated to Java.

Such a translation corresponds to a specific configuration of the C project. If the project has multiple configurations (for example, targeting different platforms), it is translated once for each configuration, and then all the resulting translations are merged into one Java project.

→

Cプロジェクトの翻訳は、通常のビルドと同じくらい簡単である。コンパイル中、システムは必要なすべてのデータ（Cコード、リンクライブラリなど）を収集。

構築が成功すると、収集したデータを直接Javaに変換できる。

このような変換は、Cプロジェクトの特定の構成に対応している。プロジェクトに複数の構成（たとえば、異なるプラットフォームを対象とする）がある場合は、構成ごとに1回変換され、結果として得られるすべての変換が1つのJavaプロジェクトにマージされる。

<https://www.mtsystems.com>

卒論 序論、背景の詳しい書き方

<http://d.hatena.ne.jp/kanbayashi/20080204/p1>

序論

論文全体の概要を書く。

背景(問題やそれに対する先行研究も簡単に述べる)、目的、やる事、評価の方法とかを書く。

研究を始める前の段階という位置づけなので「~した」とかじゃなくて「~する」と書く。

実装に立ち入った事はあまり書かない、書くとしても簡単に。

図とか箇条書きは使わない。

書けるなら、結果とContributionも書く。

最後に論文の構成を説明するのもあり。

背景と目的

背景

序論より詳しく。

背景とからめて問題点も説明。

先行研究

目的

先行研究と比べてどんなところに着目してやるかを説明。

実装に立ち上がったことについてはあまり触れない。

図とか箇条書きは使わない。

設計

「～を実現するためには～というような機能が必要だ」というような事を書く。

まだ実装に関する事は書かない。

「設計」という言葉を字面のまま受け取るのではなく、「実装のための問題分析」というようなイメージで考えた方がいいのかも。

実装

設計の話でも実装寄りの話はこっちに書く。

他のことも当然書くけどそれらについては割愛。

評価

何を目的に評価を行ったのかを書いてから、諸条件と結果を書く。

考察も忘れずに。考察では、結果の分析だけでなく、結果は自分たちの目的をどれだけ実現したかとか、どう捉えることができるかというような事も書く。

課題(議論の場合もあり?)

研究について、これから解決しなければならない課題を書く

結論

序論で書いてあった事がそれぞれできたのか書く。できなかつたらできなかつた理由を書く。

序論と対応させて「～した」と書くのが基本。

図とか箇条書きは使わない。

プログラム言語を効率良く学びたいからこそ『2つ同時』に学ぶ!!! 7個のメリット

<http://gabekore.org/double-langue-method>

メリット1: 同じような命令を二回学習でき、頭に残りやすい

メリット2: 二回学習できても、微妙に違いがあるので飽きない

メリット3: 『違いはどこだ?』という比較の観点を持てるので集中しやすい

- メリット4: 同じような命令でも使い方が違う場合があり、それを考えるのが面白くて記憶に残る
- メリット5: 片方にだけある命令や機能があると、特徴として記憶のポイントになる
- メリット6: 一度の学習期間で二つ覚えられるのでチョーお得！！
- メリット7: 一つだけに集中するより二つの方が意外と覚えやすく忘れにくい

廣瀬くんの本稿に則っての背景や目的

タイトル 初学者の学習支援のためのPython Rubyの変換

1 はじめに

1.1 はじめに

今回、私たちは

1.11 背景

現在、様々な言語がある中でpythonとrubyを使用できる人々はプログラマーの中で最も給与が高い。特に、pythonはAIなどの影響により大きな人気があり、両言語はプログラムを初めから学びたい人にとって、取り掛かりやすい言語であると感じた。その中で今回、

1.12 課題と評価

1.2 アプローチ

1.2.1 期待される効果