

第25回全国高専プログラミングコンテスト自由部門 2019

Code on the Tablet

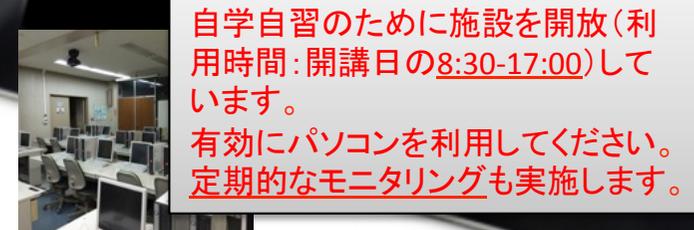


はじめに

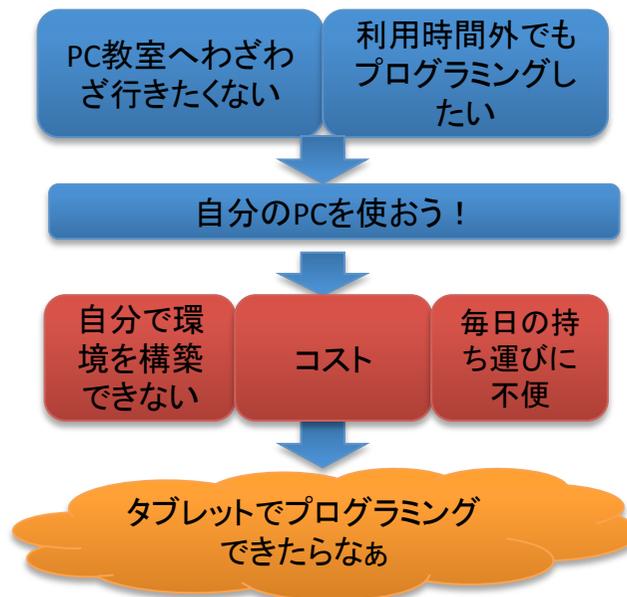
場所・時間・環境の制約

学校で行われているプログラミング授業のやり方・問題について指摘し、現状の問題を解決する手段を提案します。

- **プログラミングの重要性**
みなさんはスマホ、家電製品などいつも当たり前のように利用しています。それらはどれもプログラムが大きく関わっており、プログラミング技術者はさらに必要になっています。
- **学校でのプログラミング授業の現状**
現在の学校ではPCを用いたプログラミング授業が行われており、本校ではPC教室に用意されたPCを利用しています。PC教室による授業には以下の問題があります。
 - **場所の制約**
プログラムを作成するため、PC教室に移動しないと行けない。固有教室では受けられない。
 - **時間の制約**
PC教室の解放時間は限られており、自由に利用することはできません。
 - **環境の制約**
生徒用PCの購入を義務づける大学もありますが、学生個人がPC教室と同等の環境を構築するのは容易ではありません。
 - **コストの制約**
また、PCを個人に買わせることにはまだまだ抵抗があります。これらの問題から、プログラム学習を十分に行えない学生が出てきています。
- **タブレットの普及**
 - タブレット端末はPCに比べると、安価で持ち運びに適しており、学校単位での導入も進んでいます。
 - しかし、現在はタブレット端末での本格的なプログラミング開発環境は提供されていません。



場所・時間・環境の制約(本校の例)



私たちは、タブレット上で本格的なプログラミング学習ができるシステムを提供します

システムの目的

いつでも・どこでも・瞬時にコーディング

本システムは、タブレット上でプログラミング学習を行うJavaプログラミング環境です。プログラムのコーディングからコンパイルまでをタブレット上で行い、LMSと連携して成績の管理を行うことを主な機能として、いつでも・どこでも・瞬時に利用できることを目的としています。

学生の利点

- タブレットアプリとすることで**いつでも・どこでも・瞬時に**利用が可能
- 学習記録の把握が容易
- アプリを用いて予習・復習が可能

教員の利点

- 学生の学習評価が容易
- 穴埋め問題・テストケースの作成が容易
- 問題文・穴埋め問題・テストケース(学習コンテンツ)の管理が容易

アプリが提供する機能

- LMSとの連携による授業記録の管理・学習コンテンツの利用
- タブレット上でのプログラム統合開発環境(エディタ・コンパイラ)

サーバおよびPCが提供する機能

- LMSによる各学生の学習記録・学習コンテンツの蓄積・公開
- オンラインジャッジによるコンパイル
- プログラム穴埋め問題とテストケースの自動生成

対象とするユーザ

- タブレットを所有しているプログラミング初心者
- プログラミング授業を行う教員



いつでも



どこでも

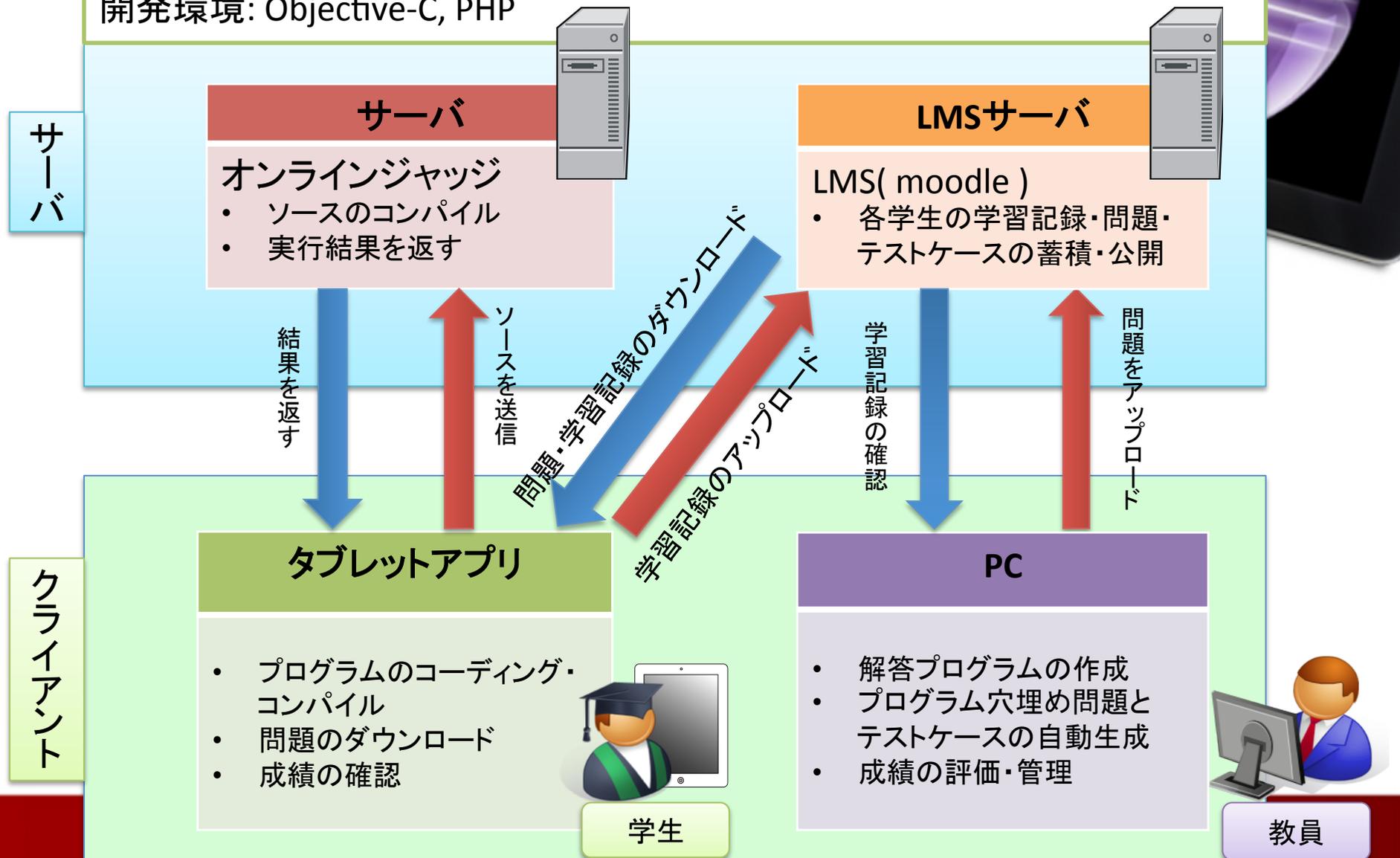


瞬時に

本格的なコーディング体験を提供します

システム構成図

動作環境: iPad(iOS7以降)
サーバ(オンラインジャッジ・LMS) : Linux
開発環境: Objective-C, PHP



LMS

Moodleによる学習コンテンツ・記録の利用・管理

LMS(Moodle)によって、学習記録・学習コンテンツを蓄積・公開することで学生・教員のプログラミング授業を支援します。



Moodleの採用

本システムではLMSにMoodleを採用します。これは、多くの高専がMoodleを利用しているからです。Moodleを採用することで、本システムと高専との親和性は高くなり、多くの高専での採用が期待されます。

学生に提供するサービス

- **学習コンテンツの利用**
教員がアップロードした問題文・穴埋め問題・テストケースを学生がダウンロードして利用できます。
- **学習記録の蓄積**
学生が問題をやり終えたら、問題のソースと成績が、日時と共にLMSサーバに保存されます。

教員に提供するサービス

- **学習コンテンツの公開**
教員が、問題文・穴埋め問題・テストケースをLMSサーバにアップロードして、学生に公開することができます。
- **学生の学習状況の把握**
学生がサーバに記録してきた学習記録を教員が確認することで学生の学習状況が把握できます。



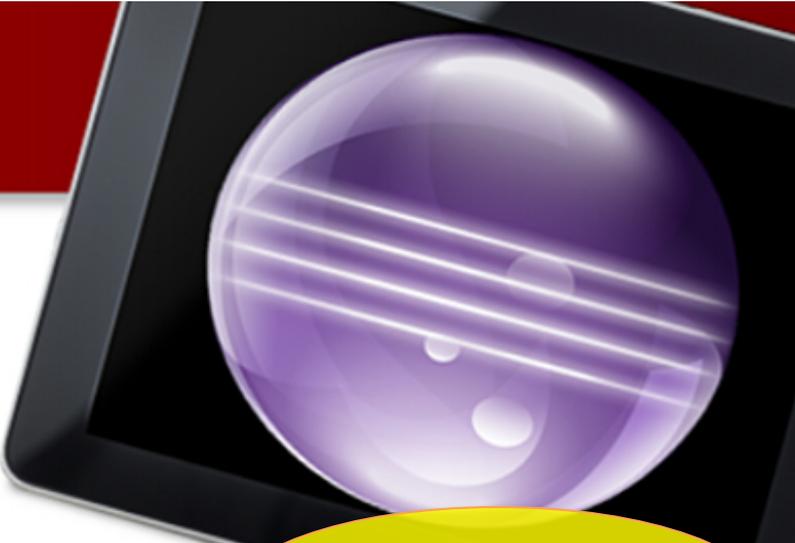
エディタ

タブレットでも効率的なコーディングを実現

タブレット上のプログラム統合開発環境が提供する機能の内の1つがエディタです。
タブレットの特徴を活かしたインターフェースを提供します。

コーディング支援

- シンタックスハイライト
 - プログラムの構造に応じて色分けを行います。
 - こうすることで、プログラムの可読性が高まります。
- オートインデント
 - 自動でインデントを行います。
 - タブを入力する必要がなくなり、プログラムの見た目も良くなります。
- シンタックスチェック
 - 構文的なエラーを検出します。
 - 編集集中にエラーに気付くため、いちいちコンパイルしてエラーを確認する手間が省けます。



```
File Undo Redo Build Run
1 import java.io.*;
2 class Main {
3     public static void main (String[] args)
4     throws Exception {
5         System.out.println("Hello, World!");
6         BufferedReader reader = new
7         BufferedReader(new
8         InputStreamReader(System.in));
9         int n = Integer.parseInt(
10        reader.readLine());
11        for (int i=0; i<=n; i++) {
12            double r;
13            r = Math.exp(Math.log(2)*i);
14            System.out.format("%.0f ", r);
15        }
16    }
17 }
```

色分け・段落で
コードを見やすく

エラーを検出

* イメージ図

エディタ

タブレットの特徴を活かした操作

タブレット入力の操作性はどうしても、PCのキーボードと比較して劣る部分があります。

タブレット操作の特徴を活かして、PC操作と遜色のないスムーズな操作を提供します。

入力支援

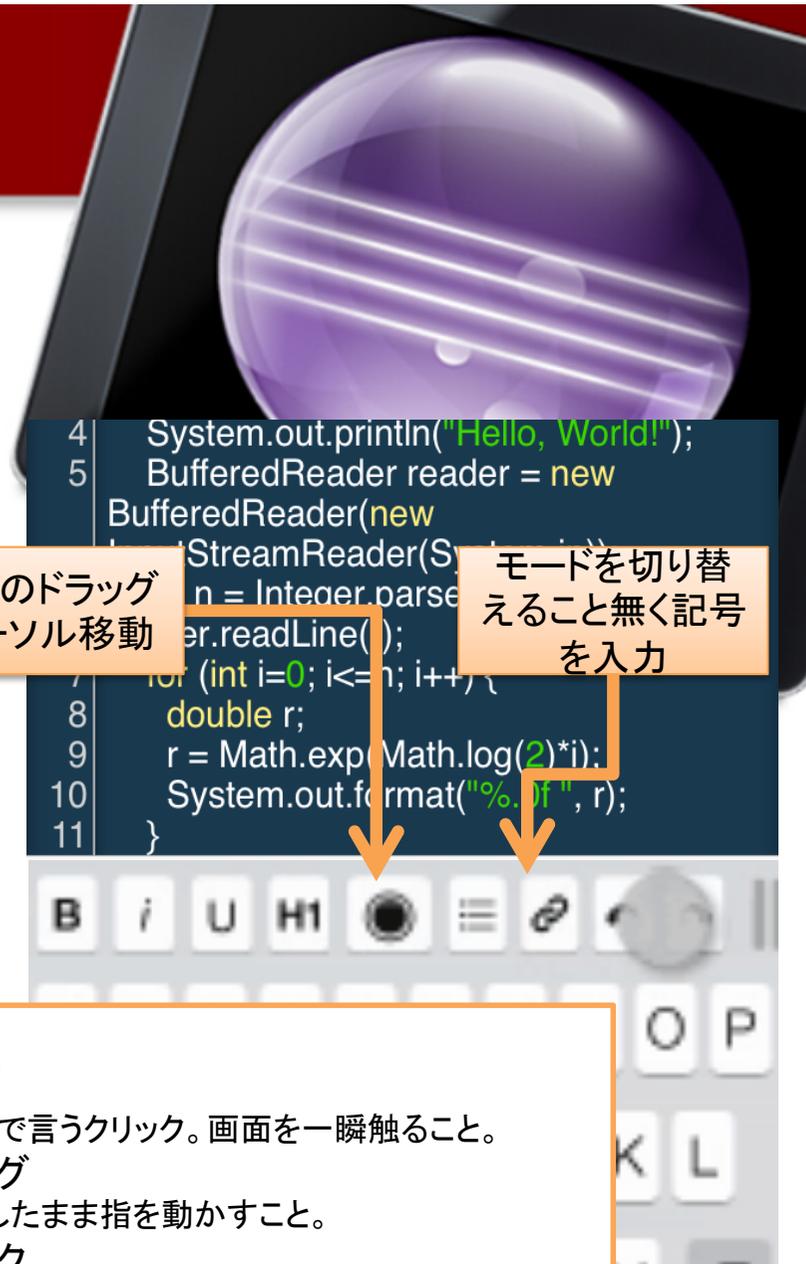
- **ボタンのドラッグでカーソル移動**
 - キーボード中に用意したボタンをドラッグすることで、キーボードをしまうことなくスムーズなコーディングが可能です。
- **記号の入力欄を設置**
 - キーボード上部に記号入力欄を用意することで、いちいち切り替えをせずにコーディングを行うことができます。
- **予約語のフリック入力**
 - タブレットのフリック入力を利用して、プログラムの予約語を入力する機能を提供します。
- **スニペット(単語予測)入力**
 - 任意の単語を登録しておいて予測変換できるようにします。
 - 単語の登録で予測変換を行うので、自分好みのカスタマイズが可能です。

ボタンのドラッグ
でカーソル移動

モードを切り替
えること無く記号
を入力

用語説明

- **タップ**
マウスで言うクリック。画面を一瞬触ること。
- **ドラッグ**
タップしたまま指を動かすこと。
- **フリック**
タッチパネルを指先でスライドしたりパツと払ったりすること。



コンパイラ

タブレット上でのコンパイルを実現

作成されたプログラムは即座に動作確認することが必要です。タブレット上では、動作制限があるため、2種類のコンパイル方法を提供します。

標準的なプログラム

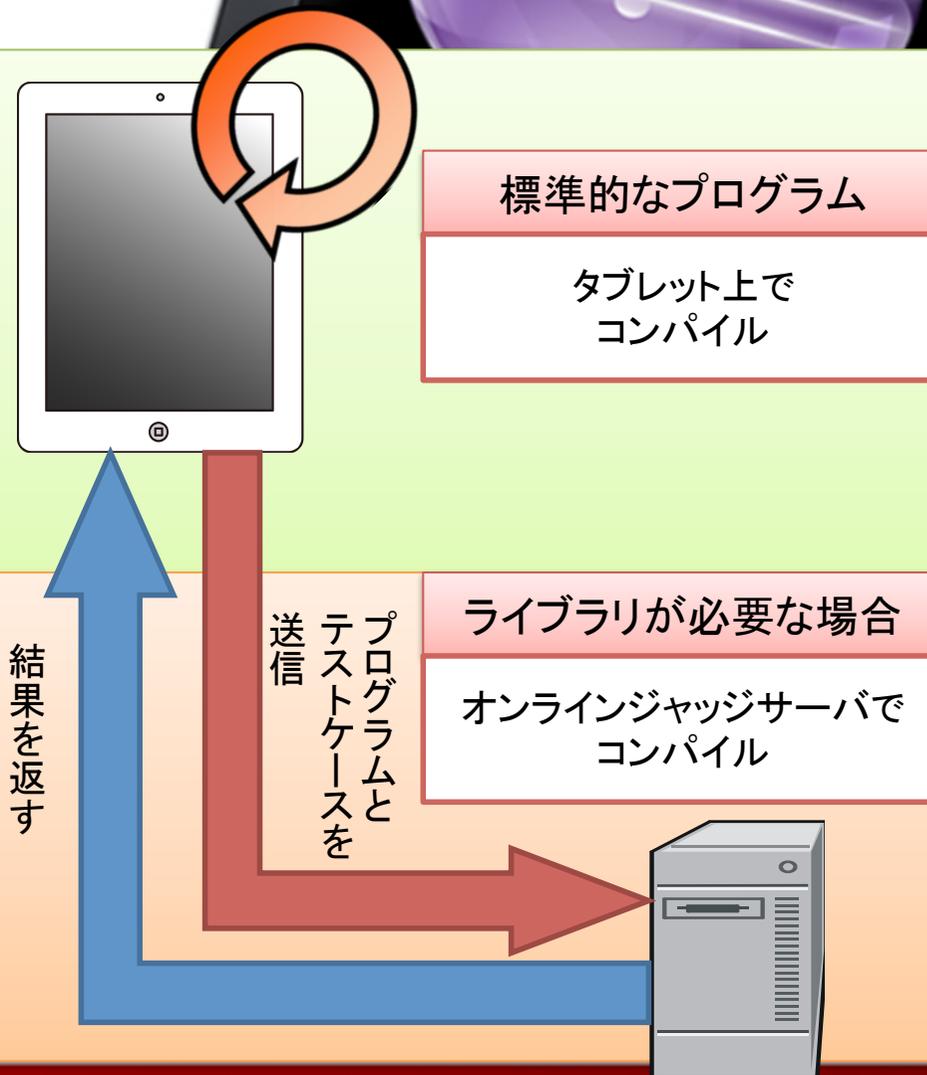
- タブレットアプリの中にコンパイルを用意しておくことで、タブレット上でのJavaプログラムのコンパイルを実現します。
- この機能を有する類似アプリは見つかりません。つまり、これは**初めての試みである**ことを意味します。
- 実行環境の制約があるので、この機能はライブラリを必要としない標準的なプログラムのみ制限します。

外部ライブラリを必要とする場合

- タブレットでコーディングを行ったプログラムとテストケースをサーバに送信し、サーバ側でコンパイルを行います。
- テストケースで解答が適切かを判断し、結果をタブレットに返します。

機能制限:

コンソール出力のプログラムのみ対象



学習コンテンツの作成

教員の負担を軽く

教員が学習コンテンツを作成する時、できるだけ負担を軽くする機能を提供します。

- 教員の手間を省きたい
 - 教員は学習コンテンツを全て手入力で作成してきました。しかし教員のする仕事はコンテンツを作成することだけではありません。
 - 中でもテストケースは、条件分岐を網羅する数だけ作成が必要なので、大変時間を必要とします。
 - そこで私たちは**穴埋め問題**と**テストケース**の自動生成機能を提供します。
- 解答プログラムの構文解析
 - 自動生成を行うために、教員が作成した問題の模範解答プログラムを構文解析します。
- テストケース・穴埋め問題の自動生成
 - 解析結果からテストケースを生成し、教員の問題作成意図を適用して穴埋め問題を生成します。
- LMSサーバへの自動アップロード
 - 作成した問題はサーバにワンクリックで登録ができます。



	問題文の作成	解答プログラムの作成	穴埋め問題の作成	テストケースの作成
従来	× 手動	× 手動	× 手動	× 手動
本システム	× 手動	× 手動	○ 自動	○ 自動

構文解析によるテストケースと穴埋め問題の生成

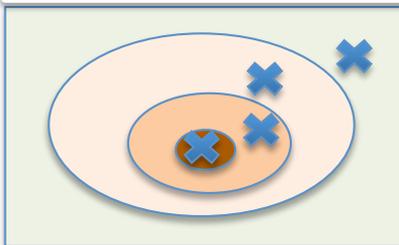
構文解析を行うことで、適切な問題作成を実現します。

Source

```
boolean isLeapYear(int y){
    if (y%4==0 && y%100!=0 || y%400==0)
        return true;
    else
        return false;
}
```



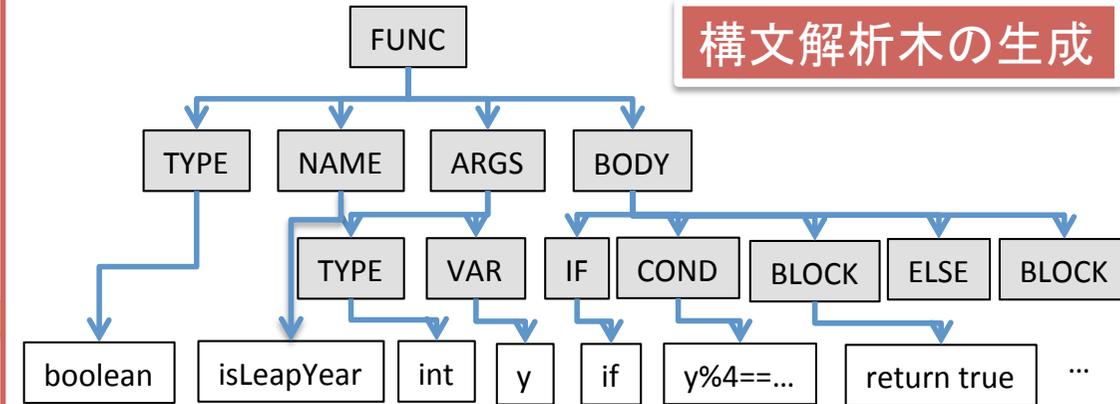
全領域のテストを生成



テストケース

```
public void testIsLeapYear() {
    assertEquals(true,b.isLeapYear(2012));
    assertEquals(false,b.isLeapYear(2014));
    assertEquals(true,b.isLeapYear(2000));
    assertEquals(false,b.isLeapYear(1900));
}
```

構文解析木の生成



問題作成意図の選択

- 仮引数
- メソッドの戻り値
- 演算子

穴埋め

```
boolean isLeapYear(int y){
    if (y%4==0 && y%100!=0 || y%400==0)
        return true ;
    else
        return false ;
}
```

問題の趣旨にあった葉(leaf)を空欄に



	テストの整合性	テストの偏り	穴埋め問題の整合性	問題の意図	作業負担
手動	○偏りなし	×テスト固定	○	○	× 手動
ランダム	×偏りあり	○テスト任意	×	×	○ 自動
構文解析	○偏りなし	○テスト任意	○	○(項目選択)	△ 半自動

学習環境・類似システムとの比較

学習環境の比較、類似システムとの比較を行い、本システムの有用性を示します。

■本システムと様々な学習環境との比較

	どこでも 利用可能	いつでも 利用可能	環境構築が容易	課題提出が容易	起動時間が短い
本システム	○	○	○	○	○
PC教室	×	×	○	×	×
自分のノートPC	△	○	×	×	△
自分のデスクトップPC	×	△	×	×	×

■本システムと類似システムとの比較

	コーディング	端末上での コンパイル	サーバを用いた コンパイル	LMSとの連携
本システム	○	○	○	○
JVM(アプリ)	○	×	○	×
Eclipse	○	○	×	×

まとめ

本システムはJavaプログラミング学習をする初心者のコーディング・実行・学習記録の蓄積、プログラミング授業を行う教員の学習コンテンツの作成・学生の学習記録の把握を支援します。

学生への支援

- 学習コンテンツの利用
 - 学生はいつでもどこでも瞬時に利用が可能なので、予習復習への応用が可能です。
- 学習記録の蓄積・閲覧
 - やってきた成果をサーバに蓄積し、時に振り返ることで自分の学習状況を把握できます。
- タブレット上での統合開発環境を提供
 - タブレット上の1つのアプリがコーディングからコンパイル・実行までをサポートします。
- オンラインジャッジによるコンパイル
 - タブレット上コンパイルの制約をオンラインジャッジによるコンパイルがカバーします

教員への支援

- 問題の作成・公開
 - フォーマットにより問題の作成を効率化し、問題の公開が可能です。
- 問題テストケースの自動生成
 - 問題の模範解答から穴埋め問題・テストケースを自動生成し、サーバにアップします。
- 学習記録の閲覧
 - 学生の学習記録を閲覧することで、全体の学習状況の把握が容易に可能です。



Code on the Tablet

場所・時間・環境にとらわれない新しいプログラミング授業の体験を提供します