

CS 112/ Section 02

Hilal Ünal

Aslıhan Ekim

Merve Özkılınç

Notes of March 11, 2008 and March 13, 2008:

Pig Latin:

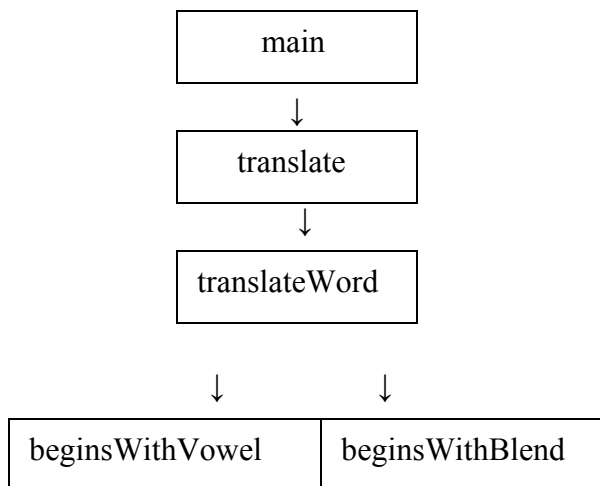
happy → appyhay

want → antway

add → adday

two consonants:

the → ethay



```
import java.util.Scanner;
```

```
public class Piglatin {
```

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

```
        String sentence, result, other;
```

```
        Scanner scan=new Scanner(System.in);
```

```
        do{
```

```
            //to get the input
```

```
            System.out.println();
```

```
            System.out.println("Enter a sentence: ");
```

```
            sentence=scan.nextLine();
```

```

        //to make the translation
        System.out.println();
        result=PigLatinTranslator.translate(sentence);
        System.out.println("Translated sentence " + result);

        //whether to continue or not
        System.out.println("Translate another sentence: (y/n)");
        another=scan.nextLine();
    }while(another.equalsIgnoreCase("y"));
}
}

import java.util.Scanner;

public class PigLatinTranslator {

    public static String Translate(String sentence){
        String result="";
        Scanner scan=new Scanner(System.in);
        sentence=sentence.toLowerCase();
        while(scan.hasNext()){
            result=translateWord(scan.next());
            result+=" ";
        }
        return (result);
    }

    private static String translateWord(String word){
        String result="";
        if(beginswithVowel(word))

```

```

        result=word+ "ay";
    else if(beginswithBlend(word))
        result=word.substring(2) + word.substring(0,2) + "ay";
    else
        result=word.substring(1) + word.charAt(0) + "ay";
    return (result);
}

private static boolean beginswithVowel(String word){
    String vowels= "aeiou";
    char letter=word.charAt(0);
    return(vowels.indexOf(letter)!= -1);//the method returns -1 if the specified character
//cannot be found.
}

private static boolean beginswithBlend(String word){
    return(word.startsWith("bl") || word.startsWith("br") || word.startsWith("ch") ||
word.startsWith("cl") || word.startsWith("cr") || word.startsWith("dr") ||
word.startsWith("dw") || word.startsWith("fl") || word.startsWith("fr") ||
word.startsWith("gl") || .....|| word.startsWith("wr"));
}
}

```

Rational Number:

```

public class RationalNumber {
    private int numerator, denominator ;

    public RationalNumber(int numer, int denom){

        if(denom==0)

```

```

        denom=1;
    if(denom<0){
        numer=numer*-1;
        denom=denom*-1;
    }
    numerator=numer;
    denominator=denom;
    reduce( );
}
private void reduce( ){
    if(numerator!=0){
        int common=gcd(Math.abs(numerator),denominator);
        numerator=numerator/common;
        denominator=denominator/common;
    }
}
private int gcd(int n1,int n2){
//n1>0, n2>0
    while(n1!=n2){
        if(n1>n2)
            n1=n1-n2;
        else
            n2=n2-n1;
    }
    return(n1);
}
public RationalNumber reciprocal( ){
    RationalNumber result=new RationalNumber(denominator,numerator);
    return(result);
// or return(new RationalNumber(denominator,numerator));
}

```

```

public RationalNumber multiply(RationalNumber otherNumber){
    int num=numerator*otherNumber.numerator;
    int denom=denominator*otherNumber.denominator;
    return(new RationalNumber(num,denom));
}
public RationalNumber divide(RationalNumber otherNumber){
    return this.multiply(otherNumber.reciprocal( ));
}
public boolean equals(RationalNumber otherNumber){

return(numerator==otherNumber.numerator&&denominator==otherNumber.denominator);
}
public RationalNumber subtract(RationalNumber otherNumber){
    int commonDenominator=denominator*otherNumber.denominator;
    int num1=numerator*otherNumber.denominator;
    int num2=otherNumber.numerator*denominator;
    int difference=num1-num2;
    return(new RationalNumber(difference,commonDenominator));
}
public RationalNumber addition(RationalNumber otherNumber){
    int commonDenominator=denominator*otherNumber.denominator;
    int num1=numerator*otherNumber.denominator;
    int num2=otherNumber.numerator*denominator;
    int sum=num1+num2;
    return(new RationalNumber(sum,commonDenominator));
}
public String toString( ){
    String result;
    if(numerator==0)
        result="0";

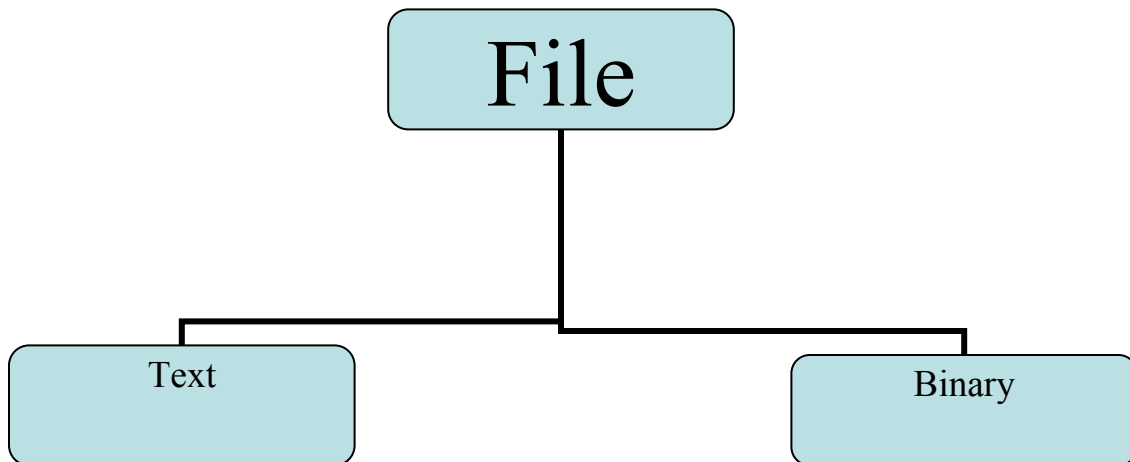
```

```

else
    if(denominator==1)
        result=numerator+" ";
    else
        result=numerator+ "/" + denominator;
return(result);
}
}

```

Reading from a text file:



Mehmet	a	Akif	100
Ayşe	v	Akilli	10

```

import java.io.File;
import java.util.Scanner;
public class ReadData {

    public static void main(String[]args){
        File inputFile=new File("scores.text");
        //java.io.File inputFile=new java.io.File("scores.text"); for this expression import is
        //not used.

        //Create a scanner from the file
        Scanner input=new Scanner(inputFile);

        //Read data from file
        while(input.hasNext()){
            String firstName=input.next();
            String mi=input.next();
            //mi:middle initial
            String lastName=input.next();
            int score=input.nextInt();
            System.out.println(firstName+" "+mi+"."+lastName+": "+score);

        }
        inputFile.close();
    }
}

```

Questions:

- 1) Why did the *translateWord*, *beginsWithVowel*, and *beginsWithBlend* methods declared with private visibility?

Because these methods exist to help the translate method, which is the only true service method in the class, to do its job. They are not intended to provide services directly to clients outside the class.

2) What would happen if the *close()* method is not invoked in the class which a file is read?

If the method *close()* method is not invoked, the program will run without an error.

But it is a good practice to close the file to release the resource on the file.

3) Why don't we need the *get* methods to call the numerator or denominator of the variable *othernumber*?

Since they are in the same class, we don't have to use the *get()* method.

4) How do you create a Scanner to read a data from a file?

To create a Scanner for a file, `new Scanner(new File(filename))` is used.

5) How do you check whether a file already exists?

Use `exists()` in the File Class for this control.