Brian Luu
CS03B

Project 8.7:

For faster sorting of letters, the U.S. Postal Service encourages companies that send large volumes of mail to use a bar code denoting the ZIP code (see Figure 8).

The encoding scheme for a five-digit ZIP code is shown in Figure 9. There are full-height frame bars on each side. The five encoded digits are followed by a check digit, which is computed as follows: Add up all digits, and choose the check digit to make the sum a multiple of 10. For example, the sum of the digits in the ZIP code 95014 is 19, so the check digit is 1 to make the sum equal to 20.

Each digit of the ZIP code, and the check digit, is encoded according to the table at right, where 0 denotes a half bar and 1 a full bar. Note that they represent all combi- nations of two full and three half bars. The digit can be computed easily from the bar code using the column weights 7, 4, 2, 1, 0. For example, 01100 is

$0 \times 7 + 1 \times 4 + 1 \times 2 + 0 \times 1 + 0 \times 0 = 6$
The only exception is 0, which would yield 11 according to the weight formula.

Write a program that asks the user for a ZIP code and prints the bar code. Use : for half bars, | for full bars. For example, 95014 becomes

```
||:|:::|:|:||::::::||:|::|:::|||
```

Your program should also be able to carry out the opposite conversion: Translate bars into their ZIP code, reporting any errors in the input format or a mismatch of the digits.

Using How To 3.1

a.) Find out which method you are asked to supply.

b.) Specify the public interface.

c.) Document the public interface.

d.) Determine instance variables.

e.) Implement constructors and methods.

f.) Test your class.

Project 8.7 Design

- Find out which methods you are asked to supply.
    - A method that'll convert a string representation of a barcode to a numeric value.

  - A method that’ll convert a numeric value of a barcode to a string representation of a barcode.
  - A method that’ll give us the appropriate value for the check digit
- Specify the public interface
  - Public class PostalBarCode{
  - Public PostalBarCode(){}
  - Public PostalBarCode(String BarCode){}
  - Public PostalBarCode(int NumericBarCode){}
  - Private void initialize2D(){}
  - Private void markColumns(int row, int column7, int column4, int column2, int column1, int column0){}
  - Private boolean comparison(int[] a, int[]b){}
  - private int getValue(int[] c){}
  - private void getNth(){}
  - public int PostalDecoder(){}
  - public String PostalEncoder(){}
  - public String PostalEncoder(int NumericBarCode){}
  - public int PostalDecoder(String BarCode){}
  - public String toString(){}
  - }
- Document the public interface
  - Public class PostalBarCode{
  - /**
  - * Default constructor that'll initialize the instance * variables to default variables.
  - */
  - Public PostalBarCode(){}
  - /**
  - * Constructor that'll initialize the bar code.
  - * @param BarCode String representation of the bar code.
  - */
  - Public PostalBarCode(String BarCode){}
  - /**
  - * Constructor that'll initialzie the numeric version of the * bar code.
  - * @param NumericBarCode numeric representation of the bar * code.
  - */
  - Public PostalBarCode(int NumericBarCode){}
  - /**
  - * 2D array that matches the string representation of the bar * code to the numerical version and vice versa.

  - */
  - Private void initialize2D(){}
  - /**
  - * Function that'll initialize the 2D array to the given parameters.
  - * @param row the row is the representation of the numeric value of the string.
  - * @param column7 the first bar is a representation of 7.
  - * @param column4 the second bar is a representation of 4.
  - * @param column2 the third bar is a representation of 2.
  - * @param column1 the fourth bar is a representation of 1.
  - * @param column0 the fifth bar is a representation of 0.
  - */
  -
  - Private void markColumns(int row, int column7, int column4, int column2, int column1, int column0){}
  - /**
  - * Integer array comparison.
  - * @param c the first integer array that is going to be compared.
  - * @param b the second integer array that is going to be compared.
  - * @return true if they hold the same elements false if not.
  - */
  - Private boolean comparison(int[] a, int[]b){}
  - /**
  - * Method that'll match the String representation of the bar code to the numeric value of the bar code.
  - * @param c String representation of the bar code.
  - * @return Numeric value that matches the string representation of the bar code.
  - */
  - private int getValue(int[] c){}
  - /**
  - * Method that'll store the in an array according to it's 10^nth place.
  - * Then it'll add all the digits together and append a value to make the total = to a multiple of 10.
  - */
  - private void getNth(){}
  - /**
  - * Method that'll turn the string representation of the bar code to a numeric value.

  - * @return numeric value of the bar code.
  - */
  - public int PostalDecoder(){}
  - /**
  - * Method that'll turn the numeric value into the string representation of the bar code.
  - * @return String representation of the bar code.
  - */
  - public String PostalEncoder(){}
  - /**
  - * Method that'll convert the numeric value to a string representation of the bar code.
  - * @param NumericBarCode Numeric value of bar code.
  - * @return String representation of bar code.
  - */
  - public String PostalEncoder(int NumericBarCode){}
  - /**
  - * Method that'll convert the string representation of the bar code to a numeric value.
  - * @param BarCode String representation of bar code.
  - * @return Numeric value of bar code.
  - */
  - public int PostalDecoder(String BarCode){}
  - /**
  - * Print function.
  - */
  - public String toString(){}
  - }
- Determine the instance variables
  - Private int match[][]
  - Private int NumericBarCode
  - Private int String BarCode
  - Private int postalCode[]
- Implement constructors and methods
  - public class PostalBarCode{
  - private int match[][];
  - private int NumericBarCode;
  - private String BarCode;
  - private int postalCode[];

```

    /**
     * Default constructor that'll initialize the instance variables to default variables.
     */
    public PostalBarCode(){
        this.NumericBarCode = 0;
        this.BarCode = new String();
        this.match = new int [10][5];
        this.postalcode = new int[6];
        initialize2D();
    }

    /**
     * Constructor that'll initialize the bar code.
     * @param BarCode String representation of the bar code.
     */
    public PostalBarCode(String BarCode){
        this.BarCode = BarCode;
        this.match = new int [10][5];
        initialize2D();
    }

    /**
     * Constructor that'll initialzie the numeric version of the bar code.
     * @param NumericBarCode numeric representation of the bar code.
     */
    public PostalBarCode(int NumericBarCode){
        this.match = new int[10][5];
        this.BarCode = new String();
        this.postalcode = new int[6];
        this.NumericBarCode = NumericBarCode;
        initialize2D();
    }

    /**
     * 2D array that matches the string representation of the bar code to the
  numerical version and vice versa.
```

```
○     */
○    private void initialize2D(){
○        markColumns(0, 1,1,0,0,0);
○        markColumns(1, 0,0,0,1,1);
○        markColumns(2, 0,0,1,0,1);
○        markColumns(3, 0,0,1,1,0);
○        markColumns(4, 0,1,0,0,1);
○        markColumns(5, 0,1,0,1,0);
○        markColumns(6, 0,1,1,0,0);
○        markColumns(7, 0,1,1,0,0);
○        markColumns(8, 1,0,0,0,1);
○        markColumns(9, 1,0,1,0,0);
○    }
○
○    /**
○     * Function that'll initialize the 2D array to the given parameters.
○     * @param row the row is the representation of the numeric value of the string.
○     * @param column7 the first bar is a representation of 7.
○     * @param column4 the second bar is a representation of 4.
○     * @param column2 the third bar is a representation of 2.
○     * @param column1 the fourth bar is a representation of 1.
○     * @param column0 the fifth bar is a representation of 0.
○     */
○    private void markColumns(int row, int column7, int column4, int column2, int column1, int column0){
○        match[row][0] = column7;
○        match[row][1] = column4;
○        match[row][2] = column2;
○        match[row][3] = column1;
○        match[row][4] = column0;
○    }
○
○    /**
○     * Integer array comparison.
○     * @param c the first integer array that is going to be compared.
○     * @param b the second integer array that is going to be compared.
○     * @return true if they hold the same elements false if not.
○      */
○     private boolean comparison(int[] c, int[] b){
○         for(int i = 0; i < 4; i++){
○             if(c[i] != b[i])
○                 return false;
○         }
○         return true;
○     }
○
○     /**
○      * Method that'll match the String representation of the bar code to the numeric value of the bar code.
○      * @param c String representation of the bar code.
○      * @return Numeric value that matches the string representation of the bar code.
○      */
○     private int getValue(int[] c){
○         for(int i = 0; i < 10; i++){
○             int temp[] = match[i];
○             if(comparison(c, temp)){
○                 return i;
○             }
○         }
○         return 0;
○     }
○
○     /**
○      * Method that'll store the in an array according to it's 10^nth place.
○      * Then it'll add all the digits together and append a value to make the total = to a multiple of 10.
○      */
○     public void getNth(){
○         //Storing the 1st place of the the integer.
○         postalCode[4] = NumericBarCode%10;
○         //Storing the 10th place of the the integer.
○         postalCode[3] = (NumericBarCode%100)/10;
○         //Storing the 100th place of the the integer.
○           postalCode[2] = (NumericBarCode%1000)/100;
○           //Storing the 1,000th place of the the integer.
○           postalCode[1] = (NumericBarCode%10000)/1000;
○           //Storing the 10,000th place of the the integer.
○           postalCode[0] = NumericBarCode/10000;
○
○           int temp = 0;
○           for(int i = 0; i < 5; i++){
○               temp += postalCode[i];
○           }
○
○           temp %= 10;
○           temp = Math.abs(10-temp);
○           postalCode[5] = temp;
○
○       }
○
○       /**
○        * Method that'll turn the string representation of the bar code to a numeric value.
○        * @return numeric value of the bar code.
○        */
○       public int PostalDecoder(){
○           int j = 1;
○           int temp[] = new int[5];
○           for(int k = 0 ; k < 5; k++){
○               for(int i = 0; i < 5; i++){
○                   if(BarCode.charAt(j) == '|'){
○                       temp[i] = 1;
○                   }
○                   else{
○                       temp[i] = 0;
○                   }
○                   j++;
○               }
○
○               int temp2 = getValue(temp);
```

```
            NumericBarCode += temp2;
            NumericBarCode *= 10;

            temp = new int[5];
        }

        NumericBarCode /= 10;
        return NumericBarCode;
    }

    /**
     * Method that'll turn the numeric value into the string representation of the bar code.
     * @return String representation of the bar code.
     */
    public String PostalEncoder(){
        BarCode += "|";

        getNth();

        //First loop to loop through the indicis of the integer.
        //Ex.) 95014 --> 9 --> 5 --> 0 --> 1 --> 4
        for(int i = 0; i < 5; i++){
            //Second loop to match the array to the correct bar output.
            for(int j = 0; j < 5;j++){
                if(match[postalCode[i]][j] == 1){
                    BarCode += "|";
                }
                else{
                    BarCode += ":";
                }
            }
        }

        BarCode +=  "|";
        return BarCode;
    }

    /**
     * Method that'll convert the numeric value to a string representation of the bar code.
     * @param NumericBarCode Numeric value of bar code.
     * @return String representation of bar code.
     */
    public String PostalEncoder(int NumericBarCode){
        this.NumericBarCode = NumericBarCode;
        this.match = new int[10][5];
        initialize2D();
        return PostalEncoder();
    }

    /**
     * Method that'll convert the string representation of the bar code to a numeric value.
     * @param BarCode String representation of bar code.
     * @return Numeric value of bar code.
     */
    public int PostalDecoder(String BarCode){
        this.BarCode = BarCode;
        return PostalDecoder();
    }

    /**
     * Print function.
     */
    public String toString(){
        return (String) ("Bar Code Numeric: " + NumericBarCode +"\t\tBar Code Representation: ( " +  BarCode + " )");
    }
}
```

- Test your class.