TestNotes/pset2/BarCode.java

```java
// BarCode.java

/**
 * PSET2 #4
 * A class representing a postal bar code and its corresponding ZIP code
 *
 * @author Kuljit Takhar
 * @version October 3, 2023
 *
 * */

public class BarCode {

    // Private instance variables representing ZIP code and barcode

    private String myZipCode;
    private String myBarCode;

    /**
     * Constructor for creating BarCode object with a ZIP code or barcode
     * input is a 5-digit ZIP code or a 32-character barcode
     * IllegalArgumentException is thrown if input is invalid
    * */

    public BarCode(String input) {
        if (input.length() == 5 && isValidZipCode(input)) {
            myZipCode = input;
            myBarCode = encode(input);
        } else if (input.length() == 32 && isValidBarCode(input)) {
            myBarCode = input;
            myZipCode = decode(input);
        } else {
            throw new IllegalArgumentException("Invalid input");
        }
    }

    /**
     * Get ZIP code associated with this bar code
     * Return barcode as a string
     * */

    public String getZipCode() {
        return myZipCode;
    }

    public String getBarCode() {
        return myBarCode;
    }

    /**
     * Converts a dight (0-9) to a 5-character bar code segment
     * digit is the digit to convert
     * Return the bar code segment as a string
     */

    private String digitToCode(int digit) {

        // Define the encoding table
        String[] encodingTable = {
            "||:::",
            ":::||",
            "::|:|",
            "::||:",
            ":|::|",
            ":|:|:",
            ":||::",
            "|:::|",
            "|::|:",
            "|:|::"
        };

        return encodingTable[digit];
    }

    private int codeToDigit(String code) {

        // Define the decoding table
        String[] decodingTable = {
            "||:::",
            ":::||",
            "::|:|",
            "::||:",
            ":|::|",
            ":|:|:",
            ":||::",
            "|:::|",
            "|::|:",
            "|:|::"
        };

        for (int i = 0; i < decodingTable.length; i++) {
            if (code.equals(decodingTable[i])) {
                return i;
            }
        }

        // Handle special case for digit 0
        if (code.equals("||:::")) {
            return 0;
        }

        throw new IllegalArgumentException("Invalid barcode segment");
    }

    private boolean isValidBarCode(String barcode) {

        // Check barcode length and format
        if (barcode.length() != 32 || !barcode.matches("[|:]+")) {
            return false;
        }

        // Check frame bars
        if (!barcode.startsWith("|") || !barcode.endsWith("|")) {
            return false;
        }

        // Check barcode segments
        for (int i = 1; i <= 30; i += 5) {
            String segment = barcode.substring(i, i + 5);
            if (codeToDigit(segment) == -1) {
                return false;
            }
        }

        // Check check digit
        String checkSegment = barcode.substring(31);
        int checkDigit = codeToDigit(checkSegment);
        int sum = 0;
        for (int i = 0; i < 30; i += 5) {
            sum += codeToDigit(barcode.substring(i, i + 5));
        }
        if ((sum + checkDigit) % 10 != 0) {
            return false;
        }

        return true;
    }

    private boolean isValidZipCode(String zipCode) {
       // Check zip code format (5 digits)
       return zipCode.matches("\\d{5}");
    }

    private int getCheckDigit(String zipCode) {
        int sum = 0;
        for (int i = 0; i < 5; i++) {
            sum += Character.getNumericValue(zipCode.charAt(i));
        }
        return (10 - (sum % 10)) % 10;
    }

    private String encode(String zipCode) {
        StringBuilder barcode = new StringBuilder("|"); // Start with the frame bar

        // Encode each digit and add to the barcode
        for (int i = 0; i < 5; i++) {
            int digit = Character.getNumericValue(zipCode.charAt(i));
            barcode.append(digitToCode(digit));
        }

        // Add the check digit
        int checkDigit = getCheckDigit(zipCode);
        barcode.append(digitToCode(checkDigit));

        // End with the frame bar
        barcode.append("|");

        return barcode.toString();
    }

    private String decode(String barcode) {
        StringBuilder zipCode = new StringBuilder();

        // Decode each digit segment
        for (int i = 1; i <= 30; i += 5) {
            String segment = barcode.substring(i, i + 5);
            int digit = codeToDigit(segment);
            zipCode.append(digit);
        }

        return zipCode.toString();
    }
}

```