Дьячков Вадим, 23501/4

RadixConverter.java

```java
package ru.vaddya.converter;

import java.util.ArrayList;
import java.util.Scanner;

public class RadixConverter {

    public static void main(String[] args) {
        RadixConverter converter = new RadixConverter();
        if (args.length == 3) {
            int baseRadix = Integer.parseInt(args[1]);
            int finalRadix = Integer.parseInt(args[2]);

            String res = converter.convert(args[0], baseRadix, finalRadix);
            System.out.format("%s (%d) = %s (%d)\n", args[0], baseRadix, res
                , finalRadix);
        } else {
            Scanner scan = new Scanner(System.in);
            System.out.print("Input the number: ");
            String number = scan.next();

            System.out.print("Input the base radix: ");
            int baseRadix = scan.nextInt();

            System.out.print("Input the final radix: ");
            int finalRadix = scan.nextInt();

            String res = converter.convert(number, baseRadix, finalRadix);
            System.out.format("%s (%d) = %s (%d)\n", number, baseRadix, res,
                finalRadix);
        }
    }

    private static final int ACCURACY = 8;
    private static final char DELIMITER = ',';

    public String convert(String number, int baseRadix, int finalRadix) {
        Parser parser = new Parser(number, DELIMITER);

        return Composer.compose(
                convertIntPart(parser.getIntPart(), baseRadix, finalRadix),
                convertFracPart(parser.getFracPart(), baseRadix, finalRadix)
                    ,
                DELIMITER
        );
    }

    private ArrayList<Integer> convertIntPart(ArrayList<Integer> intPart,
       int baseRadix, int finalRadix) {
        long numberInDecimal = 0;
        long powerOfBaseRadix = 1;
        for (int value : intPart) {
            numberInDecimal += value * powerOfBaseRadix;
            powerOfBaseRadix *= baseRadix;
        }

        ArrayList<Integer> integerPart = new ArrayList<>();
```

```
        while (numberInDecimal != 0) {
            integerPart.add(0, (int) (numberInDecimal % finalRadix));
            numberInDecimal /= finalRadix;
        }

        return integerPart;
    }

    private ArrayList<Integer> convertFracPart(ArrayList<Integer> fracPart,
       int baseRadix, int finalRadix) {
        double numberInDecimal = 0.0;
        long powerOfBaseRadix = baseRadix;
        for (int value : fracPart) {
            numberInDecimal += (double) value / powerOfBaseRadix;
            powerOfBaseRadix *= baseRadix;
        }

        ArrayList<Integer> fractionalPart = new ArrayList<>();
        int accuracy = ACCURACY;
        while (accuracy > 0 && numberInDecimal != 0) {
            numberInDecimal *= finalRadix;
            int diff = (int) numberInDecimal;
            fractionalPart.add(diff);
            numberInDecimal -= diff;
            accuracy--;
        }

        return fractionalPart;
    }
}
```

Parser.java

```
package ru.vaddya.converter;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class Parser {

    private static final Map<Character, Integer> CHARACTER_MAP;

    static {
        CHARACTER_MAP = new HashMap<>(16);
        Character[] chars = {'0', '1', '2', '3', '4', '5', '6', '7',
                '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

        for (int i = 0; i < 16; i++) {
            CHARACTER_MAP.put(chars[i], i);
        }
    }

    private ArrayList<Integer> intPart = new ArrayList<>();
    private ArrayList<Integer> fracPart = new ArrayList<>();

    public Parser(String number, char delimiter) {
        int indexOfDel = number.indexOf(delimiter);
        if (indexOfDel == -1) {
```

```
            indexOfDel = number.length();
        }
        for (int i = 0; i < indexOfDel; i++) {
            intPart.add(0, CHARACTER_MAP.get(number.charAt(i)));
        }
        for (int i = indexOfDel + 1; i < number.length(); i++) {
            fracPart.add(CHARACTER_MAP.get(number.charAt(i)));
        }
    }

    public ArrayList<Integer> getIntPart() {
        return intPart;
    }

    public ArrayList<Integer> getFracPart() {
        return fracPart;
    }
}
```

Composer.java

```
package ru.vaddya.converter;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

public class Composer {

    private static final Map<Integer, Character> INTEGER_MAP;

    static {
        INTEGER_MAP = new HashMap<>(16);
        Character[] chars = {'0', '1', '2', '3', '4', '5', '6', '7',
                '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};

        for (int i = 0; i < 16; i++) {
            INTEGER_MAP.put(i, chars[i]);
        }
    }

    public static String compose(ArrayList<Integer> integerPart, ArrayList<
        Integer> fractionalPart, char delimiter) {
        StringBuilder sb = new StringBuilder();
        if (integerPart.isEmpty()) {
            sb.append('0');
        }
        for (int value : integerPart) {
            sb.append(INTEGER_MAP.get(value));
        }
        if (!fractionalPart.isEmpty()) {
            sb.append(delimiter);
            for (int value : fractionalPart) {
                sb.append(INTEGER_MAP.get(value));
            }
        }
        return sb.toString();
    }
}
```