메인클래스

package JavaClass;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Base64 bencoder = new Base64();  
 Hex hencoder= new Hex();  
 ASCII aencoder = new ASCII();  
 Decimal dencoder = new Decimal();  
 Binary Bencoder = new Binary();  
  
 byte[] inputdata = new byte[6];  
 inputdata[0] = 0x12;  
 inputdata[1] = 0x34;  
 inputdata[2] = 0x56;  
 inputdata[3] = 65;  
 inputdata[4] = 66;  
 inputdata[5] = 67;  
  
 String res1 = bencoder.encode(inputdata);  
 String res2 = hencoder.encode(inputdata);  
 String res3 = aencoder.encode(inputdata);  
 String res4 = dencoder.encode(inputdata);  
 String res5 = Bencoder.encode(inputdata);  
  
 System.*out*.println("Base64 result: " + res1);  
 System.*out*.println("Hex result: " + res2);  
 System.*out*.println("ASCII result: " + res3);  
 System.*out*.println("Decimal result: " + res4);  
 System.*out*.println("Binary result: " + res5);  
 }  
}

아스키클래스

package JavaClass;  
  
public class ASCII extends Encoder {  
  
 @Override  
 public byte[] CodeWord(byte[] in){  
 return in;  
 }  
  
 @Override  
 public String toString(int value){  
 return (char)value + "";  
 }  
}

Base64클래스

package JavaClass;  
  
public class Base64 extends Encoder{  
 Base64(){  
 symbol\_map = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/=";  
 }  
  
 @Override  
 public byte[] CodeWord(byte[] in){  
 byte [] word = new byte[4];  
 word[0] = (byte) ((in[0] & 0xfc) >> 2);  
 word[1] = (byte) (((in[0] & 0x3) << 4) | ((in[1] & 0xf0)>> 4));  
 word[2] = (byte) (((in[1] & 0xf) << 2) | ((in[2] & 0xc0)>>2));  
 word[3] = (byte) (in[2] &0x3f);  
 return word;  
 }  
  
 @Override  
 public String encode(byte[] in){  
 byte[] word;  
 byte[] in\_code = new byte[3];  
 String result = "";  
  
 for (int i = 0 ; i < in.length ; i = i+3) {  
 in\_code[0] = in[i];  
 in\_code[1] = in[i+1];  
 in\_code[2] = in[i+2];  
  
 word = CodeWord(in\_code);  
  
 for (int j = 0 ; j < 4; j++) {  
 result += toString(word[j]);  
 }  
 }  
 return result;  
 }  
}

Binary클래스

package JavaClass;  
  
public class Binary extends Encoder{  
 Binary() {  
 symbol\_map = "01";  
 }  
  
 @Override  
 public byte[] CodeWord(byte[] in){  
 byte[] out = new byte[in.length \* 8];  
  
 for(int i = 0; i < in.length; i++)  
 for(int j = 7; j >= 0; j--)  
 out[i\*8 + j] = (byte)((in[i] & (1 << (7 - j))) >> (7 - j));  
 return out;  
 }  
}

Decimal클래스

package JavaClass;  
  
public class Decimal extends Encoder {  
 Decimal(){  
 symbol\_map = "0123456789";  
 }  
  
 @Override  
 public byte[] CodeWord(byte[] in) {  
 byte[] out = new byte[in.length \* 3];  
  
 for(int i = 0; i < in.length; i++) {  
 out[i \* 3] = (byte)(in[i] / 100);  
 out[i \* 3 + 1] = (byte)(in[i] / 10 % 10);  
 out[i \* 3 + 2] = (byte)(in[i] % 10);  
 }  
 return out;  
 }  
}

Hex클래스

package JavaClass;  
  
public class Hex extends Encoder{  
 Hex(){  
 symbol\_map = "0123456789ABCDEF";  
 }  
  
 @Override  
 public byte[] CodeWord(byte[] in){  
 byte[] word = new byte[in.length \* 2];  
  
 for(int i = 0; i <in.length; i++){  
 word[i\*2] = (byte)(in[i] >> 4);  
 word[i\*2 + 1] = (byte)(in[i] & 0x0F);  
 }  
 return word;  
 }  
}

Encoderble (인터페이스)

package JavaClass;  
  
public interface Encoderble {  
  
 public byte[] CodeWord(byte[] in);  
 public String toString(int value);  
 public String encode(byte[] in);  
}

Encoder(부모클래스(abstract와 주석을 지울시) + 추상클래스)

package JavaClass;  
public abstract class Encoder implements Encoderble {  
 protected String symbol\_map;  
  
 @Override  
 abstract public byte[] CodeWord(byte[] in);//{  
// return in;  
// }  
  
 @Override  
 public String toString(int value){  
 return symbol\_map.charAt(value) + "";  
 }  
  
 @Override  
 public String encode(byte[] in) {  
 String res = "";  
 byte[] out = CodeWord(in);  
 for(int i = 0; i < out.length; i++) {  
 res += toString((int)out[i]);  
 }  
 return res;  
 }  
}