비밀번호 암호화

❖ SHA-256

- 일종의 hash 함수
- 단방향으로 만 동작(복원 불가)
- 결과값은 32bit 8개를 병렬로 늘어놓은 값 (32byte)
  - 32bit x 8 = 256bit → 이름이 sha256
  - SHA-512

- 결과값을 16진수 표현 문자열로 변환
  - 64바이트 길이를 가지는 암호생성

❖ Salt

- 비밀번호만 암호화하는 경우 같은 비밀번호에 대해서는 같은 암호화 값이 산출
- 비밀번호 암호화시 랜덤한 값 추가 → Salt
- 랜덤한 8byte 배열 생성 → 16진수 문자열로 변환(16 byte)

❖ SHA256Util.java

```
package edu.iot.common.sec;

public class SHA256Util {
   public static String generateSalt() {
      byte[] salt = new byte[8];

      // 랜덤 8바이트 데이터 생성
      Random random = new Random();
      random.nextBytes(salt);

      StringBuilder sb = new StringBuilder();
      for (int i = 0; i < salt.length; i++) {
         // byte 값을 Hex 값으로 바꾸기
         sb.append(String.format("%02x", salt[i]));
      }
      return sb.toString();
   }
}
```

❖ SHA256을 이용한 비밀번호 암호화

- 사용자 입력 비밀번호 + Salt값

❖ MessageDigest 객체

- 임의의 길이를 가지는 문자열을 동일한 길이의 해시값으로 변환
- 변환 알고리즘으로 SHA-256 사용

```
MessageDigest md = MessageDigest.getInstance("SHA-256");
md.update(bytes);

byte[] byteData = md.digest();

// byte[]를 16진수로 표현되는 문자열로 변환
```

❖ SHA256Util.java

```
package edu.iot.common.sec;

public class SHA256Util {
      public static String getEncrypt(String source, String salt) {
      return getEncrypt(source, salt.getBytes());
   }

   public static String getEncrypt(String source, byte[] salt) {

      String result = "";

      byte[] a = source.getBytes();
      byte[] bytes = new byte[a.length + salt.length];

      System.arraycopy(a, 0, bytes, 0, a.length);
      System.arraycopy(salt, 0, bytes, a.length, salt.length);
```

❖ SHA256Util.java

```
    try {
       MessageDigest md = MessageDigest.getInstance("SHA-256");
       md.update(bytes);

       byte[] byteData = md.digest();
       // 바이트를 문자열로 변환
       StringBuilder sb = new StringBuilder();
       for (int i = 0; i < byteData.length; i++) {
          sb.append(String.format("%02x", byteData[i]));
       }

       result = sb.toString();
    } catch (NoSuchAlgorithmException e) {
       e.printStackTrace();
    }

    return result;
 }
```

❖ 확인

```
public class EncEx1 {

   public static void main(String[] args) {

      String password = "1234";
      String salt = SHA256Util.generateSalt();
      String encPassword = SHA256Util.getEncrypt(password, salt);

      System.out.println("salt : " + salt);
      System.out.println("암호화된 비밀번호: " + encPassword);

   }

}
```