

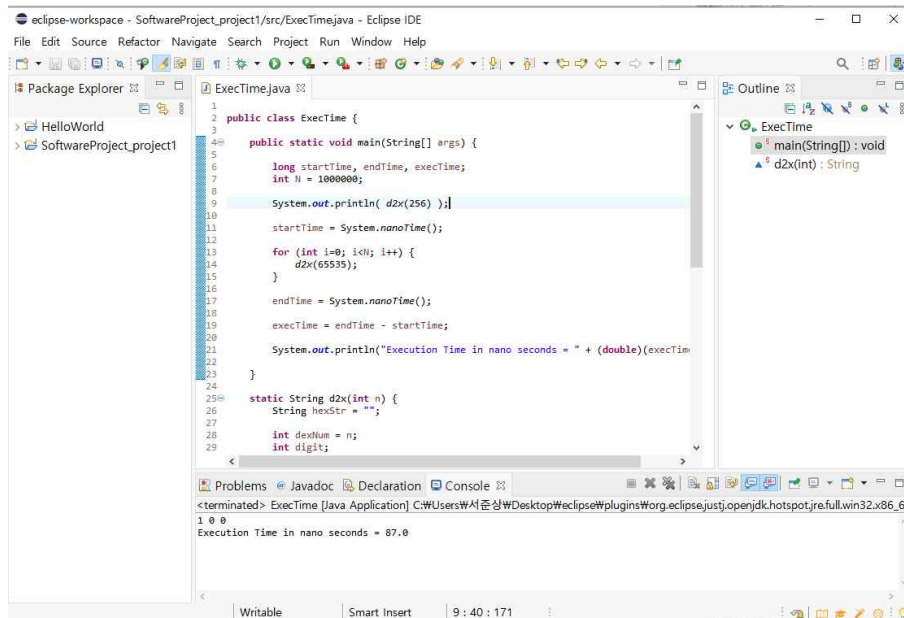
소프트웨어프로젝트

Project 1

소프트웨어학과 학과
서준상 이름
20201672 학번
2021/03/13(토) 날짜

1. 테스트 과정 화면

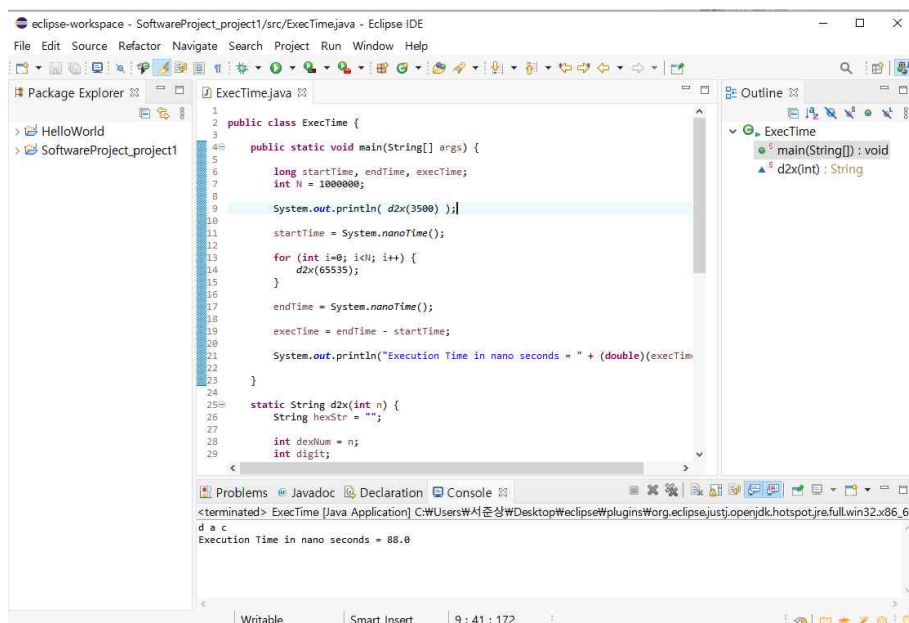
- d2x()가 정확히 동작하는지 여러 숫자로 실험



```
1 public class ExecTime {
2
3     public static void main(String[] args) {
4         long startTime, endTime, execTime;
5         int N = 1000000;
6         System.out.println( d2x(256) );
7         startTime = System.nanoTime();
8         for (int i=0; i<N; i++) {
9             d2x(65535);
10        }
11        endTime = System.nanoTime();
12        execTime = endTime - startTime;
13        System.out.println("Execution Time in nano seconds = " + (double)(execTime
14    }
15
16    static String d2x(int n) {
17        String hexStr = "";
18        int dexNum = n;
19        int digit;
20    }
```

<terminated> ExecTime [Java Application] C:\Users\서준상\Desktop\workspace\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64
Execution Time in nano seconds = 87.0

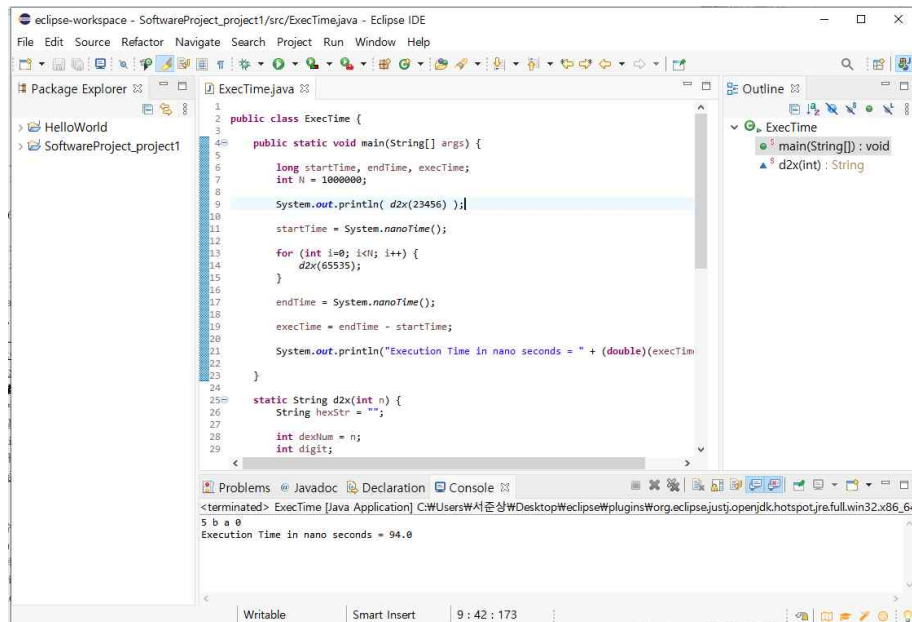
- d2x(256)의 결과
256 = 1×16^2 이므로, 출력된 '1 0 0'이 정확한 값.



```
1 public class ExecTime {
2
3     public static void main(String[] args) {
4         long startTime, endTime, execTime;
5         int N = 1000000;
6         System.out.println( d2x(3500) );
7         startTime = System.nanoTime();
8         for (int i=0; i<N; i++) {
9             d2x(65535);
10        }
11        endTime = System.nanoTime();
12        execTime = endTime - startTime;
13        System.out.println("Execution Time in nano seconds = " + (double)(execTime
14    }
15
16    static String d2x(int n) {
17        String hexStr = "";
18        int dexNum = n;
19        int digit;
20    }
```

<terminated> ExecTime [Java Application] C:\Users\서준상\Desktop\workspace\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64
Execution Time in nano seconds = 88.0

- d2x(3500)의 결과
3500 = $13 \times 16^2 + 10 \times 16^1 + 12 \times 16^0$ 이므로, 출력된 'd a c'가 맞음



- d2x(23456)의 결과

$23456 = 5 \times 16^3 + 11 \times 16^2 + 10 \times 16^1 + 0 \times 16^0$ 이므로,
콘솔에 출력된 '5 b a 0'이 맞는 값입니다.

따라서, 제가 구현한 함수 d2x()가 정상적으로 작동한다는 것을 확인할 수 있습니다.

- 여러 방법으로 d2x() 코드를 짜보고, 실행 시간을 비교

```
31 while (true) {
32     digit = dexNum % 16;
33
34     if (digit == 0) {
35         hexStr = "0" + " " + hexStr;
36     } else if (digit == 1) {
37         hexStr = "1" + " " + hexStr;
38     } else if (digit == 2) {
39         hexStr = "2" + " " + hexStr;
40     } else if (digit == 3) {
41         hexStr = "3" + " " + hexStr;
42     } else if (digit == 4) {
43         hexStr = "4" + " " + hexStr;
44     } else if (digit == 5) {
45         hexStr = "5" + " " + hexStr;
46     } else if (digit == 6) {
47         hexStr = "6" + " " + hexStr;
48     } else if (digit == 7) {
49         hexStr = "7" + " " + hexStr;
50     } else if (digit == 8) {
51         hexStr = "8" + " " + hexStr;
52     } else if (digit == 9) {
53         hexStr = "9" + " " + hexStr;
54     } else if (digit == 10) {
55         hexStr = "a" + " " + hexStr;
56     } else if (digit == 11) {
57         hexStr = "b" + " " + hexStr;
58     } else if (digit == 12) {
59         hexStr = "c" + " " + hexStr;
60     } else if (digit == 13) {
61         hexStr = "d" + " " + hexStr;
62     } else if (digit == 14) {
63         hexStr = "e" + " " + hexStr;
64     } else if (digit == 15) {
65         hexStr = "f" + " " + hexStr;
66     }
67 }
```

```
<terminated> ExecTime [Java Application] C:\Users\서준상\Desktop\ eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
f f f f
Execution Time in nano seconds = 102.0
```

```
Writable Smart Insert 35 : 30 : 607
```

```
<terminated> ExecTime [Java Application] C:\Users\서준상\Desktop\ eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
f f f f
Execution Time in nano seconds = 96.0
```

```
Writable Smart Insert 35 : 30 : 607
```

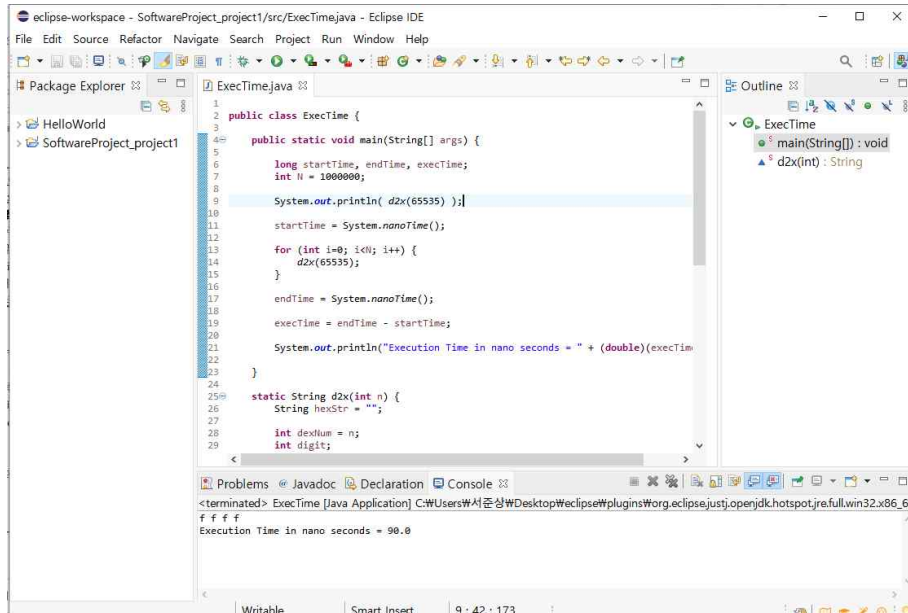
```
<terminated> ExecTime [Java Application] C:\Users\서준상\Desktop\ eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
f f f f
Execution Time in nano seconds = 101.0
```

```
Writable Smart Insert 35 : 30 : 607
```

- digit < 10일 때, hexStr = digit + " " + hexStr을 사용하지 않고, 0부터 9까지의 모든 경우를 일일이 hexStr = "7" + " " + hexStr와 같은 형식으로 작성해 보았습니다.
- 소요 시간은 약 **96 ~ 103 nano seconds**가 나옵니다.
- 80 후반에서 90 초반이 나오던 기존의 코드에 비해 시간이 더 많이 소요되었습니다.

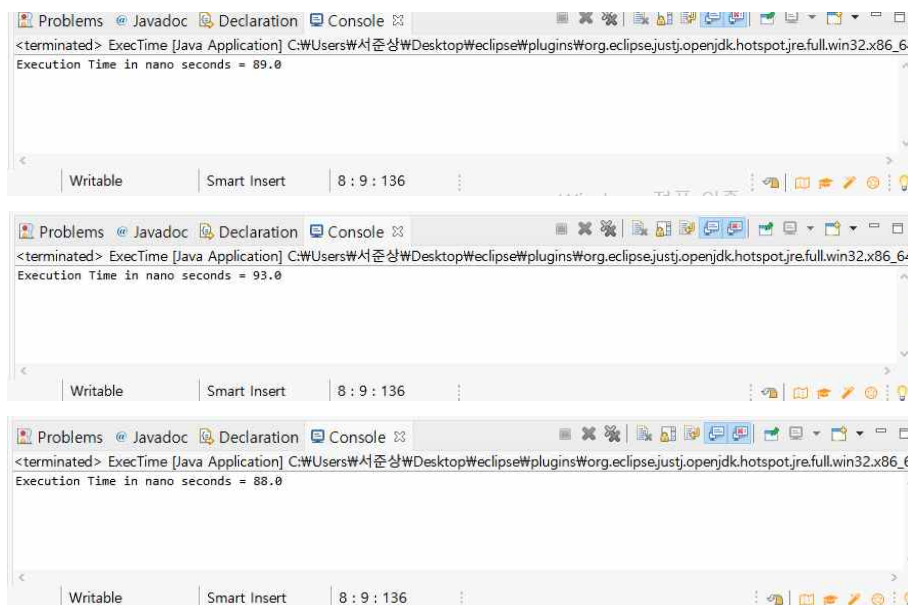
- 프로그램을 두 방법으로 각각 구현/실험

1) Eclipse를 사용하여 개발



- 소스코드는 하단의 [2. 소스 프로그램] 에 나온 것과 동일

- `d2x(65535)`의 결과는 'f f f f'로 출력됩니다.



- eclipse 공지에 올라온 대로 출력문을 넣지 않고 `d2x(65535)`의 실행 시간을 측정했을 때의 결과들로, 약 **87 ~ 93 nano seconds**가 나옵니다.

2) notepad 등의 editor를 이용하여 프로그램 작성 후, cmd 창에서 실행

바탕 화면 > 2학년 1학기 > 소프트웨어프로젝트 > 프로젝트 1

프로젝트 1 검색

이름	수정한 날짜	유형	크기
ExecTime.java	2021-03-13 오후 ...	JAVA 파일	2KB
실습계획	2021-02-15 오전 ...	Microsoft Edge...	263KB
평가표1	2021-02-17 오후 ...	Microsoft Edge...	62KB
프로젝트1	2021-02-17 오후 ...	Microsoft Edge...	202KB
프로젝트1_보고서	2021-03-13 오후 ...	한컴오피스 한...	27KB

```

명령 프롬프트
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\서준상\Desktop\2학년 1학기\소프트웨어프로젝트\프로젝트 1>
C:\Users\서준상\Desktop\2학년 1학기\소프트웨어프로젝트\프로젝트 1>javac ExecTime.java
C:\Users\서준상\Desktop\2학년 1학기\소프트웨어프로젝트\프로젝트 1>java ExecTime
f f f f
Execution Time in nano seconds = 72.0
C:\Users\서준상\Desktop\2학년 1학기\소프트웨어프로젝트\프로젝트 1>

```

바탕 화면 > 2학년 1학기 > 소프트웨어프로젝트 > 프로젝트 1

프로젝트 1 검색

이름	수정한 날짜	유형	크기
ExecTime.class	2021-03-14 오전 ...	CLASS 파일	2KB
ExecTime.java	2021-03-13 오후 ...	JAVA 파일	2KB
실습계획	2021-02-15 오전 ...	Microsoft Edge...	263KB
평가표1	2021-02-17 오후 ...	Microsoft Edge...	62KB
프로젝트1	2021-02-17 오후 ...	Microsoft Edge...	202KB
프로젝트1_보고서	2021-03-13 오후 ...	한컴오피스 한...	27KB

- 명령 프롬프트(command)창을 통해 노트패드로 작성한 .java 파일을 실행해보았습니다. eclipse에서 실행했을 때와 같이, 'f f f f'라는 결과와, 72.0 nano seconds라는 소요 시간 결과를 얻을 수 있습니다.
- command 창을 통해 .java 파일을 구현하는 과정에서, javac ExecTime.java 명령어를 통해 컴파일했을 때, 폴더에 **ExecTime.class**라는 클래스 파일이 중간 결과물로 만들어지는 것을 확인할 수 있습니다.

2. 소스 프로그램

```
public class ExecTime {

    public static void main(String[] args) {

        long startTime, endTime, execTime;
        int N = 1000000;

        System.out.println( d2x(65535) );

        startTime = System.nanoTime();

        for (int i=0; i<N i++) {
            d2x(65535);
        }

        endTime = System.nanoTime();

        execTime = endTime - startTime;

        System.out.println( "Execution Time in nano seconds = " + (double)(execTime/N) );

    }

    static String d2x(int n) {
        String hexStr = "";

        int dexNum = n;
        int digit;

        while (true) {

            digit = dexNum % 16;

            if (digit < 10) {
                hexStr = digit + " " + hexStr;
            } else if (digit == 10) {
                hexStr = "a" + " " + hexStr;
            } else if (digit == 11) {
                hexStr = "b" + " " + hexStr;
            } else if (digit == 12) {
                hexStr = "c" + " " + hexStr;
            } else if (digit == 13) {
                hexStr = "d" + " " + hexStr;
            } else if (digit == 14) {
                hexStr = "e" + " " + hexStr;
            } else if (digit == 15) {
                hexStr = "f" + " " + hexStr;
            }

            dexNum = dexNum / 16;

            if (dexNum == 0) {
                break;
            }
        }

        return hexStr;
    }
}
```

- 코드 내용

d2x() 함수를 구현할 때, 세 가지 변수를 선언했습니다.

첫 번째로, dexNum은 d2x(int n)에서의 n값을 나타내는 변수임과 동시에, 자기 자신을 16으로 나눈 몫을 계속 저장해나가는 변수입니다.

두 번째로, digit은 dexNum을 16으로 나눴을 때의 나머지를 저장하며, 이 나머지가 나타내는 숫자가 바로 16진수의 각 자릿수가 됩니다.

마지막으로, hexStr은, 각 자릿수인 digit을 모두 저장하는 변수로, 초기에 저장된 값은 ""이며, digit을 계산할 때마다 hexStr = "3" + " " + hexStr과 같이, 10진수를 16진수로 변환했을 때의 각 자릿수를 차례대로 저장하게 됩니다.

위에서 설명한 대로, n 값을 16으로 나눌 때마다 나오는 나머지를 while(){}문을 통해 hexStr에 저장하게 되며, 만약 dexNum의 값이 0이 될 경우, 현재까지 저장한 값보다 더 높은 자릿수에 올 숫자가 없다는 뜻이므로, break;를 통해 반복문을 종료합니다.

d2x()의 결과로 스트링 hexStr을 반환합니다.

3. 자체평가표

평 가 표

평가 항목	학생 자체 평가 (리포트 해당 부분 표시 및 간단한 의견)	평가 (빈칸)	점수 (빈칸)
- 실행 시간 동작? - d2x() 정확히 동작? * 여러 테스트 * <u>d2x(65535)의 실행 시간</u> - 충분한 실험? (수행시간이 작은 경우 포함?)	‘1. 테스트 과정 화면’ 파트 d2x(65535)를 실행한 결과 약 87~93 nano seconds 소요. d2x(256), d2x(3500), d2x(23456)으로, 작성한 함수 d2x()가 정확히 동작함을 확인함.		
Eclipse를 이용한 실행	‘ - 프로그램을 두 방법으로 각각 구현/실험’의 ‘ 1) Eclipse를 사용하여 개발’ 파트		
에디터와 명령어를 이용한 실행 - 각 단계별 결과물 과 그 의미는?	같은 부분의 ‘2) notepad 등의 editor를 이용하여 프로그램 작성 후, cmd 창에서 실행’ 파트 javac ExacTime.java에선, ExacTime.class가 생성, java ExacTime를 입력하면, 결과 출력		
리포트 작성 - 평가 항목에 맞게 리포트 작성? - 모든 파일을 하나의 pdf 문서로	평가 항목에 맞게 리포트 작성 하나의 pdf 파일로 만들어 제출할 예정.		
기타 추가 설명 (필요한 경우)			
총평/계	평가자 입장에서 자신의 리포트를 살펴보기가 목적 즉, 평가자가 체크하고자하는 사항을 쉽게 찾아볼 수 있도록 리포트가 기술되어있는지 점검		

* 학생 자체 평가는 점수에 반영되지 않음.

* 학생 스스로 자신의 보고서를 평가하면서, 체계적으로 프로젝트를 마무리하도록 유도하는 것이 목적임.

Eclipse에서 코드를 실행할 때 나온 저장 경로나, 명령 프롬프트 창에서 나온 사용자명 (C:\Users\서준상\Desktop\2학년 1학기\소프트웨어프로젝트\프로젝트 1) 을 통해 제가 직접 과제를 수행했음을 증명할 수 있습니다.