

4000번의 도전
드디어 찾아낸 반응 단백질



2012 인텔 국제과학경진대회 고든무어상 수상
(Intel ISEF 2012, Gordon E. Moore Award)



Java 프로그래밍 수업내용 정리 및 실습 과제

책을 보거나 인터넷을 찾아보면서 배우고 거기에 저만의 방식을 추가했어요.



JACK ANDRAKA
IS CREATING A REVOLUTION



AN ANTONIO SPURS
처음에는 막막한 느낌이지만 시간이 지나면 감이 오거든요.



더하기 빼기... 그게 거의 전부죠.



저는 여러분에게 컴퓨터를 통해 어떻게
세상을 바꿀 수 있는 지 이야기하고 싶습니다

그러나 사실, 구글이나 위키피디아 등을 이용해
알고 싶은 모든 것을 찾을 수 있고

GOOGLE, WIKIPEDIA
AND STACKOVERFLOW

2020.04.27.월

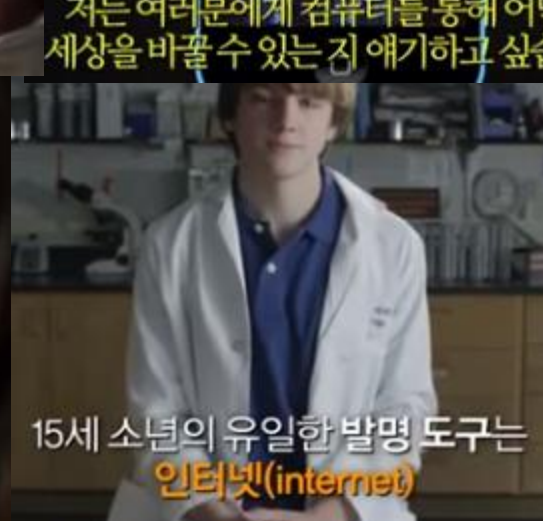
B반 송명훈



처음 배울때 감이 안나는게 뭐가 있을까요?



프로그래밍은 천재들만 할 수 있는것이 아니에요. 의지력이 필요할 뿐입니다.



15세 소년의 유일한 발명 도구는 “더 좋은 진단 키트를 만들고 싶었다.”
인터넷(internet)
- 잭 안드라카

입문 제 14장 Java 표준 클래스

1. 수업 내용 정리
2. 실습 화면 캡처
3. 연습문제 14-1
4. 연습문제 14-2



1. 수업 내용 정리

1. 수업 내용 정리 [제14장. Java 표준 클래스]

Date와 Calendar, SimpleDateFormat class

[1] <https://hyeonstorage.tistory.com/204>
[2] http://tcpschool.com/java/java_api_calendar

1. Date class

- 1) Java SE 1.0 버전부터 제공된 날짜와 시간을 계산하는 class^[1]
- 2) 많은 문제점이 있다고 지적되어 현재 대부분 methods의 사용이 권장되지 않음 (deprecated)^{[1] [2]}
- 3) Date class의 constructor^[1]
 - A. Date() : 현재의 날짜와 시간을 저장한 instance를 생성
 - B. Date(long msec)
: 1970년 1월 1일 0시 0분 0초부터 millisecond 단위로 하여 경과한 날짜와 시간을 저장한 instance를 생성
- 4) Date class의 주요 method^[1]

method	설 명
boolean after(Date when)	when의 날짜가 현재 날짜 이후면 true, 아니면 false 반환
boolean before(Date when)	when의 날짜가 현재 날짜 이전이면 true, 아니면 false 반환
int compareTo(Date anotherDate)	다른 날짜 객체와 비교하여 음수, 양수, 0의 결과를 반환
int compareTo(Object o)	다른 객체와 비교하여 음수, 양수, 0의 결과를 반환
boolean equals(Object obj)	날짜의 값을 비교하여 그 결과를 반환
long getTime()	1970년 1월 1일 0시 0분 0초로부터의 시간을 millisecond 단위로 반환
void setTime(long time)	time의 시간을 1970년 1월 1일 0시 0분 0초로부터 경과한 시간(millisec)으로 설정한다

1. 수업 내용 정리 [제14장. Java 표준 클래스]

Date와 Calendar, SimpleDateFormat class

2. Calendar class

- 1) Java SE 1.1 버전부터 제공된 날짜와 시간을 계산하는 abstract class [\[3\]](#)이며 abstract class로 선언된 이유는 나라마다 사용하는 달력 체계가 조금씩 다를 수 있기 때문 [\[2\]](#)
- 2) Date class처럼 날짜와 시간에 관한 정보를 표현할 때 사용되며 Date class에서 deprecated된 method나 constructor들 중 같은 기능의 method가 Calendar class에서 제공된다 [\[3\]](#)
- 3) Abstract class이므로 instance를 직접 생성할 수는 없으나, getInstance() method를 통해 현재 날짜와 시간의 instance 생성이 가능 [\[3\]](#)
- 4) Date class와 마찬가지로 Calendar class도 아래와 같은 여러 문제점을 갖고 있음 [\[2\]](#)
 - A. Calendar class의 instance는 immutable object가 아니므로 값이 수정될 수 있다.
 - B. 윤초(leap second)와 같은 특별한 상황을 고려하지 않았다.
 - C. Calendar class에서는 월(month)을 나타낼 때부터 1월부터 12월을 0부터 11까지로 표현해야 한다.
- 5) Java SE 8 버전에서는 새로운 날짜와 시간 API 인 java.time package를 제공해 문제점을 모두 해결했고, 다양한 기능을 지원하는 다수의 하위 package를 포함하고 있다. [\[2\]](#)
- 6) Calendar class의 모든 field는 class field(= static field) ⇨ instance 생성 없이도 바로 사용가능 [\[2\]](#)

[2] http://tcpschool.com/java/java_api_calendar

[3] <https://hyeonstorage.tistory.com/205>

1. 수업 내용 정리 [제14장. Java 표준 클래스]

Date와 Calendar, SimpleDateFormat class

2. SimpleDateFormat class

- 1) Java SE 1.1 버전부터 제공된 날짜와 시간을 계산하는 abstract class [\[3\]](#)이며 abstract class로 선언된 이유는 나라마다 사용하는 달력 체계가 조금씩 다를 수 있기 때문 [\[2\]](#)
- 2) Date class처럼 날짜와 시간에 관한 정보를 표현할 때 사용되며 Date class에서 deprecated된 method나 constructor들 중 같은 기능의 method가 Calendar class에서 제공된다 [\[3\]](#)
- 3) Abstract class이므로 instance를 직접 생성할 수는 없으나, getInstance() method를 통해 현재 날짜와 시간의 instance 생성이 가능 [\[3\]](#)
- 4) Date class와 마찬가지로 Calendar class도 아래와 같은 여러 문제점을 갖고 있음 [\[2\]](#)
 - A. Calendar class의 instance는 immutable object가 아니므로 값이 수정될 수 있다.
 - B. 윤초(leap second)와 같은 특별한 상황을 고려하지 않았다.
 - C. Calendar class에서는 월(month)을 나타낼 때부터 1월부터 12월을 0부터 11까지로 표현해야 한다.
- 5) Java SE 8 버전에서는 새로운 날짜와 시간 API 인 java.time package를 제공해 문제점을 모두 해결했고, 다양한 기능을 지원하는 다수의 하위 package를 포함하고 있다. [\[2\]](#)
- 6) Calendar class의 모든 field는 class field (= static field) ⇨ instance 생성 없이도 바로 사용가능 [\[2\]](#)

[2] http://tcpschool.com/java/java_api_calendar

[3] <https://hyeonstorage.tistory.com/205>

1. 수업 내용 정리 [제14장. Java 표준 클래스]

Object class와 Wrapper class

3. Object class

- 1) Java에서, 모든 class는 Object class로부터 상속받는다
- 2) 모든 Instance는 Object type variable에 대입 가능
- 3) 모든 class는 Object로부터 toString()이나 equals() method를 상속 받는다.
- 4) 자신이 작성한 class를 이용해 문자열이나 값의 비교를 하기 위해서는 toString()이나 equals() method를 Override하여 처리해야 한다.

4. Wrapper class

- 1) Primitive data type에 대응하는 wrapper class가 java.lang package에 존재
- 2) Primitive type과 wrapper class의 data는 valueOf()나 ~ vlaue() method로 명시적으로 casting 가능
- 3) Primitive type과 wrapper type은 서로 AutoBoxing / AutoUnboxing 기능에 의해 암묵적으로 casting된다.



2. 실습 화면 캡처

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-1. 시간 (long형)

```
Ex1.java x
1 package chapter13;
2
3 public class Ex1 {
4     public static void main(String[] args) throws InterruptedException {
5         long start = System.currentTimeMillis();
6         // 시간이 걸리는 처리
7         Thread.sleep(3000);
8
9         long end = System.currentTimeMillis();
10        System.out.println("처리 시간 : " + (end - start) + " ms");
11    }
12 }
13
```

Problems Javadoc Declaration Console x

<terminated> Ex1 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (2020. 4. 27. 5:00:00 PM)
처리 시간 : 3000 ms

14-2. 날짜 (Date class)

```
Ex2.java x ※ java.sql.Date 와 혼동하지 말 것
1 package chapter13;
2
3 import java.util.Date;
4
5 public class Ex2 {
6     public static void main(String[] args) {
7         Date now = new Date();
8         System.out.println(now);
9         System.out.println(now.getTime());
10        Date past = new Date(131662225935L);
11        System.out.println(past);
12    }
13 }
14
```

Problems Javadoc Declaration Console x

<terminated> Ex2 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (2020. 4. 27. 5:00:00 PM)
Mon Apr 27 13:58:15 KST 2020
1587963495221
Thu Sep 22 01:23:45 KST 2011

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-3. 날짜 (Calendar class)

```

Main.java  Ex3.java
1 package chapter13;
2
3 import java.util.Calendar;
4 import java.util.Date;
5
6 public class Ex3 {
7
8     public static void main(String[] args) {
9         // 현재의 년도 표시
10        Date now = new Date();
11        Calendar calendar = Calendar.getInstance();
12        calendar.setTime(now);
13
14        int year = calendar.get(Calendar.YEAR);
15        System.out.println("올 해는 " + year + "년 입니다");
16        // 지정한 날의 Date형의 값을 얻기
17        calendar.set(2010, 8, 22, 1, 23, 45);
18        calendar.set(Calendar.YEAR, 2011);
19        Date past = calendar.getTime();
20        System.out.println(past.toString());
21    }
22 }
23
Problems  Javadoc  Declaration  Search  Console
<terminated> Ex3 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\W
올 해는 2020년 입니다
Thu Sep 22 01:23:45 KST 2011
```

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-4. SimpleDateFormat

```
Ex4.java x
2
3 import java.text.ParseException;
4 import java.text.SimpleDateFormat;
5 import java.util.Date;
6
7 public class Ex4 {
8     public static void main(String[] args) throws ParseException {
9         // 오늘의 날짜
10        Date now = new Date();
11        SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
12
13        String s = format.format(now);
14        System.out.println(s);
15
16        // 지정일시의 문자열을 파싱하여 Date형으로 변환
17        Date d = format.parse("2011/09/22 01:23:45");
18    }
19 }
20
```

Problems Javadoc Declaration Console x

<terminated> Ex4 [Java Application] C:\WProgram Files\Java\jre1.8.0_251\bin\javaw.exe (2020. 4. 27. 오후 2:04)
2020/04/27 14:04:42

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-5. 모든 class의 조상 Object

```
Ex5.java x
1 package chapter13;
2
3 public class Ex5 {
4     static class Empty {    // extends object가 생략 됨
5
6     }
7
8     public static void main(String[] args) {
9         Empty e = new Empty();
10        String s = e.toString();
11        System.out.println(s); // s.toString()이 호출 됨
12    }
13 }
14
```

Problems @ Javadoc Declaration Console x

<terminated> Ex5 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\chapter13.Ex5\$Empty@15db9742

14-6. Object로 다형성 이용

```
Ex6.java x
1 package chapter13;
2
3 public class Ex6 {
4
5     public static void main(String[] args) {
6         // 다형성
7         Object o1 = new Hero();
8         Object o2 = new Wizard();
9         Object o3 = "안녕하세요";
10    }
11 }
```

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-7. 다형성

```
Ex7.java x
1 package chapter13;
2
3 public class Ex7 {
4
5     public static void main(String[] args) {
6         // 다형성
7         Object o1 = new Hero();
8         Object o2 = new Wizard();
9         Object o3 = "안녕하세요";
10
11         printAnything(o2);
12     }
13
14     private static void printAnything(Object object) {
15         // 어떤 타입도 좋으니, 인수로 1개를 받아서 화면에 표시
16         System.out.println(object.toString());
17     }
18
19 }
```

Problems @ Javadoc Declaration Console x

<terminated> Ex7 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\chapter13.Wizard@15db9742

System.out.println()의 원리

```
public void println( @Nullable String x) {
    if (getClass() == PrintStream.class) {
        writeln(String.valueOf(x));
    } else {
        synchronized (this) {
            print(x);
            newLine();
        }
    }
}
```

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-8. 기본 문자열로 표시

```
J Ex8.java ✕
1 package chapter13;
2
3 public class Ex8 {
4     public static void main(String[] args) {
5         Hero hero = new Hero();
6         hero.setName("켈타스");
7         hero.setHp(100);
8         // hero.toString() 대신 hero 만 넣어도 됨
9         System.out.println(hero.toString());
10        System.out.println(hero);
11    }
12 }
13
```

Problems | @ Javadoc | Declaration | Console ✕

<terminated> Ex8 [Java Application] C:\Program Files\Java\jre1.8

chapter13.Hero@15db9742
chapter13.Hero@15db9742

14-9. 문자열 표현을 정의

```
J Hero.java ✕
1 package chapter13;
2
3 public class Hero extends Character {
4
5     public void attack(Monster monster) {
6         System.out.println(this.getName() + "의 공격");
7         System.out.println("적에게 10포인트의 데미지를 주었다!");
8         monster.hp -= 10;
9     }
10
11     @Override
12     public String toString() {
13         // toString()을 오버라이드
14         return "이름 : " + this.getName() + "/HP : " + this.getHp();
15     }
16 }
```


2. 실습 화면 캡처 [제14장. Java 표준 클래스]

```
public class Hero extends Character {
    @Override
    public boolean equals(Object obj) {
14-11. if (this == obj) {
이름이 같으면 return true;
같은 걸로 }
취급하도록 if (obj instanceof Hero) {
equals 재정의 if (this.name.equals(hero.name)) {
return true;
}
```

```
return false;
}
```

```
3 public class Ex10 {
4     JavaProgramming/src/chapter13/Ex10.java
5     public static void main(String[] args) {
6         Hero h1 = new Hero();
7         h1.setName("켈타스");
8         h1.setHp(100);
9
10        Hero h2 = new Hero();
11        h2.setName("켈타스");
12        h2.setHp(100);
13
14        if (h1.equals(h2) == true) {
15            System.out.println("같은 내용입니다");
16        } else {
17            System.out.println("다른 내용입니다");
18        }
19    }
20 }
21 }
```

14-10. 같은 것인지 똑같이 생긴

것인지의 차이 (==과 equals 의 차이)

Problems Javadoc Declaration Search Console
<terminated> Ex10 [Java Application] C:\Program Files\Java\jre1.

다른 내용입니다

```
3 public class Ex10 {
4
5     public static void main(String[] args) {
6         Hero h1 = new Hero();
7         h1.setName("켈타스");
8         h1.setHp(100);
9
10        Hero h2 = new Hero();
11        h2.setName("켈타스");
12        h2.setHp(100);
13
14        if (h1.equals(h2) == true) {
15            System.out.println("같은 내용입니다");
16        } else {
17            System.out.println("다른 내용입니다");
18        }
19    }
20 }
21 }
```

Problems Javadoc Declaration Search Console
<terminated> Ex10 [Java Application] C:\Program Files\Java\jre1.

같은 내용입니다

2. 실습 화면 캡처 [제14장. Java 표준 클래스]

14-12. 기본형 → 오브젝트 변환

```
J Ex12.java ✕
1 package chapter13;
2
3 public class Ex12 {
4
5     public static void main(String[] args) {
6         int i1 = 15;
7         Integer i2 = Integer.valueOf(i1);
8         int i3 = i2.intValue();
9         System.out.println("i1 = " + i1);
10        System.out.println("i2 = " + i2);
11        System.out.println("i3 = " + i3);
12    }
13 }
```

Problems @ Javadoc Declaration Search Console ✕

<terminated> Ex12 [Java Application] C:\Program Files\Java\jre1

```
i1 = 15
i2 = 15
i3 = 15
```

14-13. AutoBoxing과 AutoUnboxing

```
J Ex13.java ✕
1 package chapter13;
2
3 public class Ex13 {
4     public static void main(String[] args) {
5         int i1 = 15;
6         Integer i2 = i1;
7         int i3 = i2;
8         System.out.println("i1 = " + i1);
9         System.out.println("i2 = " + i2);
10        System.out.println("i3 = " + i3);
11    }
12 }
13
```

Problems @ Javadoc Declaration Search Console ✕

<terminated> Ex13 [Java Application] C:\Program Files\Java\jre1

```
i1 = 15
i2 = 15
i3 = 15
```



3. 연습문제 14-1

3. 연습문제 14-1

14-1. 문제

: main() method만 가지는 class Main을 정의하고, 이하의 순서로 현재의 100일 후의 날짜를 "서기 2011년 09월 24일"과 같은 형식으로 표시하시오.

1. 현재의 날짜를 Date 형으로 얻는다
2. 얻은 날짜 정보를 Calendar에 설정한다
3. Calendar에서 **일(day)**값을 얻는다
4. 얻은 값에 100을 더한 값을 Calendar의 **일**에 설정한다.
5. Calendar의 날짜정보를 Date 형으로 변환한다
6. SimpleDateFormat을 이용하여 Date instance의 내용을 표시한다

3. 연습문제 14-1

```
1 package chapter13;
2
3 import java.text.SimpleDateFormat;
4 import java.util.Calendar;
5 import java.util.Date;
6
7 public class Main {
8
9     public static void main(String[] args) {
10         // 1. 현재의 날짜를 Date형으로 얻는다
11         Date now = Calendar.getInstance().getTime();
12         // 2. 얻은 날짜 정보를 Calendar에 설정한다
13         Calendar calendar = Calendar.getInstance();
14         calendar.setTime(now);
15         // 3. Calendar에서 일(day)값을 얻는다
16         int day = Calendar.DAY_OF_YEAR;
17         // 4. 얻은 값에 100을 더한 값을 Calendar의 일에 설정한다
18         calendar.add(day, 100);
19         // 5. Calendar의 날짜정보를 Date형으로 변환한다.
20         Date addDayToDate = calendar.getTime();
21         // 6. SimpleDateFormat을 이용하여 Date 인스턴스의 내용을 표시한다.
22         SimpleDateFormat dateFormat = new SimpleDateFormat("서기 yyyy년 MM월 dd일");
23         System.out.println(dateFormat.format(addDayToDate));
24     }
25 }
26
27
```

Problems @ Javadoc Declaration Search Console

<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (2020. 4. 28. 오전 1:5

서기 2020년 08월 06일



4. 연습문제 14-2

4. 연습문제 14-2

14-2. 문제

: 계좌번호를 표시하는 String형 필드 accountNumber와 잔액을 표시하는 int형 필드 balance를 가지는 은행계좌 class를 작성하시오. 그리고, 이 class에 method 선언을 추가해, 다음의 1.과 2.의 조건을 만족하도록 수정하시오.

1. 계좌번호 4649, 잔액 1592원 Account instance를 변수 a에 생성하고
System.out.println(a);를 실행하면, 화면에 1592원 (계좌번호 = 4649) 라고 표시
2. 계좌번호가 같으면 값이 같은 것으로 판단할 것. 단, " 4649"처럼 계좌번호의 앞에 공백이 있으면,
공백을 무시하고 비교할 것 ("4549"와 " 4649"는 같은 것으로 본다)
(힌트: java.lang.String class의 trim() method를 이용)

4. 연습문제 14-2 [Account class]

Account class 소스 코드

```
1 package chapter13;
2
3 public class Account {
4     private String accountNumber = "";
5     private int balance = 0;
6
7     public Account(String accountNumber, int balance) {
8         this.accountNumber = accountNumber;
9         this.balance = balance;
10    }
11    public String getAccountNumber() {
12        return accountNumber;
13    }
14    public void setAccountNumber(String accountNumber) {
15        this.accountNumber = accountNumber;
16    }
17    public int getBalance() {
18        return balance;
19    }
20    public void setBalance(int balance) {
21        this.balance = balance;
22    }
```

```
20    public void setBalance(int balance) {
21        this.balance = balance;
22    }
23    @Override
24    public String toString() {
25        StringBuilder sb = new StringBuilder();
26        sb.append(Integer.toString(balance)).append("원 (계좌번호=)");
27        sb.append(accountNumber).append(")");
28        return sb.toString();
29    }
30    @Override
31    public boolean equals(Object obj) {
```

```
30    @Override
31    public boolean equals(Object obj) {
32        if (this == obj) {
33            return true;
34        }
35        if (obj instanceof Account) {
36            Account account = (Account) obj;
37            if (this.accountNumber.equals(account.getAccountNumber().trim())) {
38                return true;
39            }
40        }
41        return false;
42    }
43 }
```

4. 연습문제 14-2 [main method]

결과 확인을 위한 main method와 출력 결과

```
1 package chapter13;
2
3 public class Practice2 {
4
5     public static void main(String[] args) {
6         Account a = new Account("4649", 1592);
7         System.out.println(a);
8
9         Account b = new Account(" 4649", 1592);
10
11         if (a.equals(b)) {
12             System.out.println("a와 b는 같은 계좌번호 입니다");
13         } else {
14             System.out.println("a와 b는 서로 다른 계좌번호 입니다");
15         }
16     }
17 }
18
```

Problems @ Javadoc Declaration Search Console

<terminated> Practice2 [Java Application] C:\Program Files\Java\jre1.8.0_251\bin

1592원 (계좌번호=4649)

a와 b는 같은 계좌번호 입니다