한양대학교

OBJECT-ORIENTED SYSTEMS DESIGN (Lab5-2)

Heejin Park

Hanyang University

5-7 (display 5.11)

Create classes *ToyClass* and *ClassParameterDemo.*

[*ToyClass*]

1. Create 2 instance variables as follows.

    private String *name.*

    private int *number.*

2. Create two overloaded constructors *ToyClass()'s* whose parameters are as follows.

(String name, int number) : Initialize the instance variable *name* to the value of the parameter *name* and the instance variable *number* to the value of the parameter *number*.

() : Initialize the instance variable *name* to the string "No name yet." and the number to 0.

3. Create a method **set(String name, int number)**: Set the instance variable *name* to the value of parameter *name* and the instance variable *number* to be the value of the parameter *number*.

4. Create a method **toString()**: Return *name* + " " + *number*.

5. Create a method **changer(ToyClass aParameter)**: Set *aParameter.name* to the string "Hot Shot" and *aParameter.number* to 42.

6. Create a method **equals(ToyClass otherObject)**: Return true if the instance variable *name* is the same as *otherObject.name* and the instance variable *number* is the same as *otherObject.number* and false otherwise.

5-7 (display 5.12)

[*ClassParameterDemo*]

Write a class *ClassParameterDemo* that outputs below using the class *ToyClass*.

<output>

Mr. Cellophane 0
Now we call changer with anObject as argument.
Hot Shot 42

5-8 (display 5.17)

Extend *ToyClass* to *ToyClass2* and create *ParametersDemo.*

[*ToyClass2*]

1. Create a method **makeEqual(ToyClass2 anObject)**: Set *anObject.name* to the value of *name* and *anObject.number* to the value of *number*.

2. Create a method **tryToMakeEqual(int number)**: Set the parameter *number* to the value of the instance variable *number*.

5-8 (display 5.16)

[*ParametersDemo*]

Complete the class *ParametersDemo* that prints the output on the next page.

```
public class ParametersDemo {
    public static void main(String[] args) {
        ToyClass2 object1 = new [          ];
        ToyClass2 object2 = new [          ];
        [          ]("Scorpius", 1);
        [          ]("John Crichton", 2);
        System.out.println("Value of object2 before call to method:");
        System.out.println(object2);
        System.out.println("Value of object2 after call to method:");
        object1.[                  ];
        System.out.println(object2);
        int number = 42;
        System.out.println("Value of number before call to method: " + number);
        object1.[                  ];
        System.out.println("Value of number after call to method: " + number);
    }
}
```

5-8 (display 5.16)

<output>

Value of object2 before call to method:
John Crichton 2
Value of object2 after call to method:
Scorpius 1
Value of number before call to method: 42
Value of number after call to method: 42

5-9(Display 5.19, 5.21)

Create a class *Date* by copying the class *Date* from chapter 4.

Create classes *Date*, *Person*, and *PersonDemo* defined as follows.

[*Date*]

Copy the class *Date* from problem 4-4 in Lab 4-2.

[*Person*]

1. Create variables **private String** *name*, **private Date** *born*, and **private Date** *died*.

2. Create two overloaded constructors *Person()'s* whose parameters are as follows.

(String name, Date birth, Date death) :

If *birth* and *death* are consistent, initialize the instance variable **name** to the value of parameter **name,** the instance variable **born** to a new copy of **birth**, and the instance variable **died** to a new copy of **death**. Note that if **death** is null, **died** is null. Otherwise (If *birth* and *death* are not consistent) print out "Inconsistent dates. Aborting." and exit.

(Person original) :

If **original** is null, print out "Fatal error." and exit. Otherwise, initialize instance variables by using the values of the object **original.**

3. Create a method **toString()**: Create a variable String *diedString.* If **died** is null, *diedString* is an empty string. Otherwise, *diedString* is **died.toString().** It returns a string in a "*name, born-diedString*" format. (Consult the output on page 16.)

4. Create a method **equals**(Person **otherPerson)**: It returns false if *otherPerson* is null. Otherwise, it returns whether or not the instance variables **name, born, and died** are the same as those of *otherPerson's.* Use the method **datesMatch()** on the next page if needed.

5. Create a private method **datesMatch(Date date1, Date date2)**: It returns whether or not *date1* and *date2* are equal. Note that it returns true if both of them are null.

6. Create a method **setBirthDate(Date date)**: If *date* and *died* are consistent, **born** gets a new copy of **date**. Otherwise, print out "Inconsistent dates. Aborting." and exit.

7. Create a method **setDeathDate(Date date)**: If *born* and *date* are not consistent, print out "Inconsistent dates. Aborting." and exit. If *date* is null, the instance variable **died** becomes null. Otherwise, **died** gets a new copy of **date**.

8. Create a method **setName(String name)**: Initialize the instance variable **name** to the value of parameter **name.**

9. Create a method **setBirthYear(int year)**: If *born* is null, print out "Fatal Error. Aborting." and exit. Otherwise, change the year of birth to **year** by using the method **setYear()** in the class *Date*. If *born* and *died* are not consistent, print out "Inconsistent dates. Aborting." and exit.

10. Create a method **setDeathYear(int year)**: If *died* is null, print out "Fatal Error. Aborting." and exit. Change the year of death to **year** by using the method **setYear()** in the class *Date*. If *born* and *died* are not consistent, print out "Inconsistent dates. Aborting." and exit.

11. Create a method **getName()**: It returns *name*.

12. Create a method **getBirthDate()**: It returns a new copy of **born**.

13. Create a method **getDeathDate()**: It returns null if *died* is null. Otherwise, it returns a new copy of **died**.

14. Create a private method **consistent(Date birth, Date death)**: It returns false if *birth* is null. It returns true if *death* is null. Otherwise, it returns whether or not the *birth* comes before or equals the *death* by using the method **precedes()** and **equals()** in the class *Date*.

5-9 (display 5.21)

[*PersonDemo*]

Write a class *PersonDemo* that outputs below using the class *Person*.

<output>

A Short List of Composers:
Johann Sebastian Bach, March 21, 1685–July 28, 1750
Igor Stravinsky, June 17, 1882-April 6, 1971
John Adams, February 15, 1947-
Comparing bach and bachTwin:
Distinct copies.
Same data.