한양대학교

OBJECT-ORIENTED SYSTEMS DESIGN (Lab4-2)

Heejin Park

Hanyang University

4-3

Extend the class *Date* as follows and create a class *OverloadingDemo*.

[*Date*]

1. Create two overloaded *dateOK()'s* whose headings are as follows.

   private boolean dateOK(String month, int day, int year)

   private boolean dateOK(int month, int day, int year)

   Return true if the month is between 1 and 12, the day is between 1 and 31, and the year is between 1000 and 9999. Otherwise, return false. Use the method *monthOK()* on the next page if needed.

2. Create a method *monthOK()* whose heading is as follows.

private boolean monthOK(String month)

Return true if *month* is one of the names of months and false, otherwise.

3. Create three overloaded *setDate()'s* whose headings are as follows.

public void setDate(String month, int day, int year)

public void setDate(int month, int day, int year)

Public void setDate(int year)

Use the method *dateOK()* to check if parameters are legal. If so, store them into instance variables. (Use *this* if needed.) Otherwise, print out "Fatal Error" and exit. For **setDate(int year),** assume that the date is January, 1st. (You need to check if parameters are legal even for **setDate(int year)**.)

4. Create a method **public void** setMonth(**int** month): If *month* is between 1 and 12, store it. Otherwise, print out "Fatal Error" and exit.

5. Create a method **public void** setDay(**int** day): If *day* is between 1 and 31, store it. Otherwise, print out "Fatal Error" and exit.

6. Create a method **public void** setYear(**int** year): If *year* is between 1000 and 9999, store it. Otherwise, print out "Fatal Error" and exit.

7. Update the method **public void readInput()** such that it checks the input by the method **dateOK()** before they are stored. If it is false, it asks the user to input again by printing out “Illegal date. Reenter input.”

4–3

[*OverloadingDemo*]

Write a class *OverloadingDemo* as follows and run it with the input on the right.

```
public class OverloadingDemo {
    public static void main(String[] args) {
        Date date1 = new Date(),
             date2 = new Date(),
             date3 = new Date();
        date1.setDate(2022);
        System.out.println(date1);
        date2.setDate("April", 6, 2022);
        System.out.println(date2);
        date3.readInput();
        System.out.println(date3);
    }
}
```

<input and output>

```
January 1, 2022
April 6, 2022
Enter month, day and year
Do not use a comma.
Apri 6 2022
Illegal date. Reenter input.
Enter month, day and year
Do not use a comma.
April 32 2022
Illegal date. Reenter input.
Enter month, day and year
Do not use a comma.
April 6 2022
April 6, 2022
```

4-4

Extend the class *Date* as follows and create a class *ConstructorDemo*.

[*Date*]

1. Create five constructors *Date()'s* whose headings are as follows.

public Date() : Initialize such that the date is January 1st, 1000.

public Date(int month, int day, int year)

public Date(String month, int day, int year)

public Date(int year): Initialize such that the date is January, 1st, **year**.

public Date(Date aDate): If *aDate* is null, print out "Fatal Error." and exit. Otherwise, store the parameter.

[*ConstructorDemo*]

Write a class *ConstructorDemo* as follows and run it.

```
public class ConstructorDemo {
    public static void main(String[] args) {
        Date date1 = new Date("December", 16, 1770),
             date2 = new Date(1,27,1756),
             date3 = new Date(1882),
             date4 = new Date(date1),
             date5 = new Date();
        System.out.println("Whose birthday is " + date1 + "?");
        System.out.println("Whose birthday is " + date2 + "?");
        System.out.println("Whose birthday is " + date3 + "?");
        System.out.println("Whose birthday is " + date4 + "?");
        System.out.println("The default date is " + date5 + ".");
    }
}
```

<output>

```
Whose birthday is December 16, 1770?
Whose birthday is January 27, 1756?
Whose birthday is January 1, 1882?
Whose birthday is December 16, 1770?
The default date is January 1, 1000.
```

4–5

Create classes *Pet* and *PetDemo* defined as follows.

[*Pet*]

1. Create three instance variables: **private String** *name*, **private int** *age*, and **private double** *weight*.

2. Create a method **public String** *toString()*: It returns "Name: *name*₩n Age: *age* years ₩nWeight: *weight* pounds".

3. Create five constructors *Pet()* whose heading is as follows.

public Pet(String name, int age, double weight):

Print out “Error: Negative age or weight.” and exit if the age or the weight is less than 0. Otherwise, store the parameters.

public Pet(String name):

Store parameters assuming that the age and weight are 0.

public Pet(int age):

Print out “Error: Negative age.” and exit if the age is less than 0. Otherwise, store parameters assuming that the name is “No name yet.” and the weight is 0.

public Pet(double weight):

Print out “Error: Negative weight.” and exit if the weight is less than 0. Otherwise, store parameters assuming that the name is “No name yet.” and the age is 0.

public Pet():

Initialize the name as “No name yet.” and age and weight as 0.

4. Create four methods whose headings are as follows.

public void set(String name, int age, double weight):

Print out “Error: Negative age or weight.” and exit if the age or the weight is less than 0. Otherwise, store the parameters into instance variables.

public void setName(String name):

Store **name**.

public void setAge(int age):

Print out “Error: Negative age.” and exit if the age is less than 0. Otherwise, store **age**.

public void setWeight(double weight):

Print out “Error: Negative weight.” and exit if the weight is less than 0. Otherwise, store **weight**.

5. Create a method **public String** *getName()*: It returns *name*.

6. Create a method **public int** *getAge()*: It returns *age*.

7. Create a method **public double** *getWeight()*: It returns *weight*.

4–5

[*PetDemo*]

Write a class *PetDemo* on the right and run it with the input on the next page.

```
import java.util.Scanner;

public class PetDemo {
    public static void main(String[] args) {
        Pet usersPet = new Pet("Jane Doe");
        System.out.println("My records on your pet are incomplete.");
        System.out.println("Here is what they currently say:");
        System.out.println(usersPet);

        Scanner keyboard = new Scanner(System.in);
        System.out.println("Please enter the pet's name:");
        String name = keyboard.nextLine();
        usersPet.setName(name);
        System.out.println(usersPet);
        System.out.println("Please enter the pet's age:");
        int age = keyboard.nextInt();
        usersPet.setAge(age);
        System.out.println(usersPet);
        System.out.println("Please enter the pet's weight:");
        double weight = keyboard.nextDouble();
        usersPet.setWeight(weight);
        System.out.println("My records now say:");
        System.out.println(usersPet);
    }
}
```

<input and output>

```
My records on your pet are
incomplete.
Here is what they currently say:
Name: Jane Doe
Age: 0 years
Weight: 0.0 pounds
Please enter the pet's name:
Fang Junior
Name: Fang Junior
Age: 0 years
Weight: 0.0 pounds
Please enter the pet's age:
5
Name: Fang Junior
Age: 5 years
Weight: 0.0 pounds
Please enter the pet's weight:
87.5
My records now say:
Name: Fang Junior
Age: 5 years
Weight: 87.5 pounds
```