Arrays of Objects

Using the Array Data Structure with Objects

Alwin Tareen

Arrays of Objects

- You have already seen how arrays can be used to store primitive types:

```
int[] nums = new int[10];
double[] scores = new double[8];
```

- Arrays can also be used to store objects of classes that you define.

```
TextMessage[] words = new TextMessage[17];
```

- The following program describes a `DiceGame` class which contains `Die` objects in an array.

The Die Class

```
public class Die
{
    private int faceValue;

    public Die()
    {
        faceValue = 1;
    }

    public int getFaceValue()
    {
        return faceValue;
    }

    public void roll()
    {
        faceValue = (int) (Math.random() * 6) + 1;
    }
}
```

The DiceGame Class

```
public class DiceGame
{
    // instance variable
    private Die[] dice;

    // constructor
    public DiceGame()
    {
        dice = new Die[5];
        for (int i = 0; i < dice.length; i++)
        {
            dice[i] = new Die();
        }
    }
```

The DiceGame Class, Continued

```
    // mutator method
    public void rollDice()
    {
        for (int i = 0; i < dice.length; i++)
        {
            Die cube = dice[i];
            cube.roll();
        }
    }
```

The DiceGame Class, Continued

```
    // the toString() method
    public String toString()
    {
        String result = "";
        for (int i = 0; i < dice.length; i++)
        {
            Die cube = dice[i];
            result += cube.getFaceValue() + " ";
        }
        return result;
    }
}
```

The DiceGameTest Class

```
public class DiceGameTest
{
    public static void main(String[] args)
    {
        DiceGame rack = new DiceGame();
        rack.rollDice();
        System.out.println(rack);
    }
}
```

An Array as an Instance Variable

Instance Variable

- The following statement describes an object instance variable which is declared with `Die` as its type. Recall that arrays can store objects.

```
private Die[] dice;
```

Constructor

- The constructor starts by initalizing the array with a length of 5, and it assigns its reference(memory location) to the variable `dice`.

```
dice = new Die[5];
```

Setting up the Array in the Constructor

Constructor

- The array elements do not contain `Die` objects yet, so they are assigned a value of `null`.

`dice`	`null`	`null`	`null`	`null`	`null`
index	0	1	2	3	4

- The keyword `null` is the default value for an object reference that is empty.

Setting up the Array in the Constructor

Constructor

- Next, the constructor instantiates(creates) five `Die` objects and stores their references in the array.
- Each `Die` object occupies a different array index.

`dice`	`2B94A7`	`87CD9A`	`F9371A`	`31D58E`	`D295A3`
index	0	1	2	3	4

Using a Mutator Method with the Array

The `rollDice()` Method

- This method uses a `for` loop to access each `Die` object within the array.
- Within the `for` loop, an array index is used to retrieve each `Die` object's reference and store it in the local variable `cube`.
- Note that `cube` is declared with data type `Die`.

`dice`	`2B94A7`	`87CD9A`	`F9371A`	`31D58E`	`D295A3`
index	0	1	2	3	4

`cube`	`2B94A7`

Using a Mutator Method with the Array

The `rollDice()` Method

- Then, the `roll()` method from the `Die` class is called on the object reference value stored in the `Die` object `cube`.
- The `roll()` method assigns a random value to the `faceValue` instance variable of that object.

Arrays of Objects: End of Notes