CIS 351-Data Structure-ArrayObjects
Feb 6, 2020

Farzana Rahman

Syracuse University

Arrays of Objects

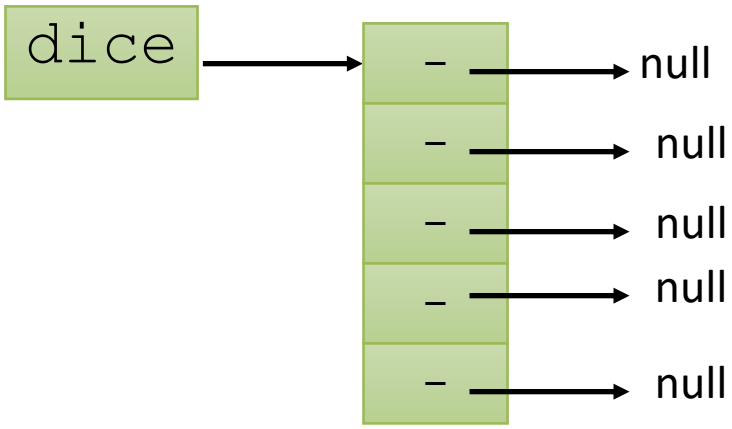
- The following declaration reserves space to store 5 references to `String` objects

```
String[] words = new String[5];
```

- It does NOT create the `String` objects themselves
- Initially an array of objects holds `null` references
- Each object stored in an array must be instantiated separately

Arrays of Objects

An array of `Die` objects when initially declared:

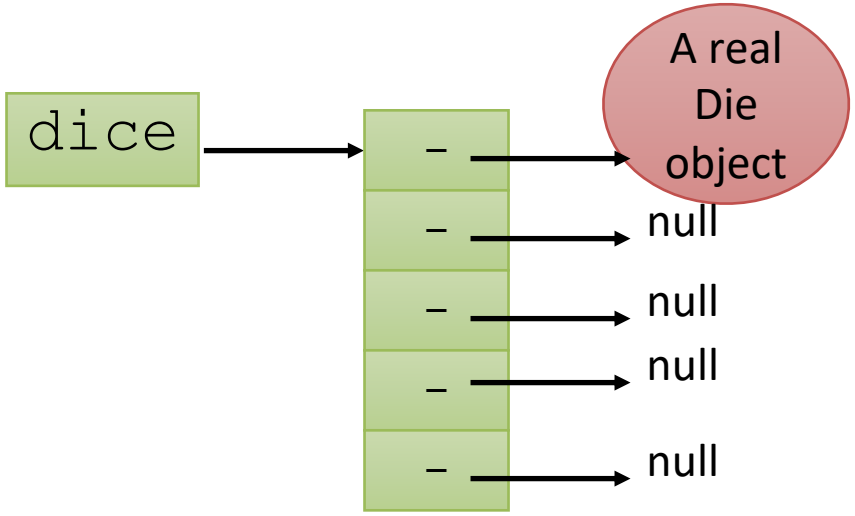



```
At this point, following
  reference would throw
  NullPointerException:
```

```
System.out.println (words[0]);
```

Arrays of Objects

After a `Die` object is created and stored in the array

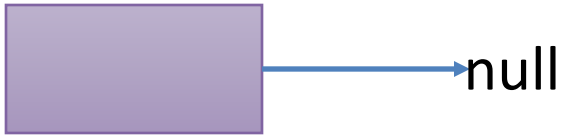


Read the following code carefully

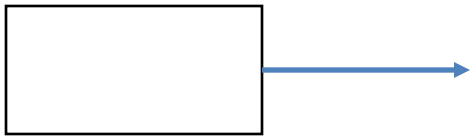
```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",
            diceA[i].getFace(),diceB[i].getFace());
}
```

diceA

diceB

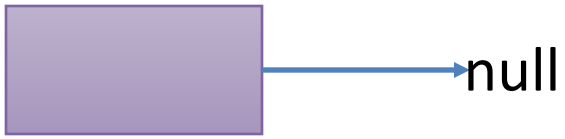



```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```



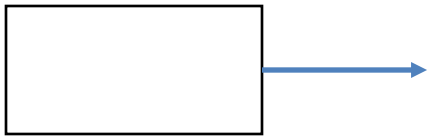
diceA

diceB



single

```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```



diceA

null

null

null

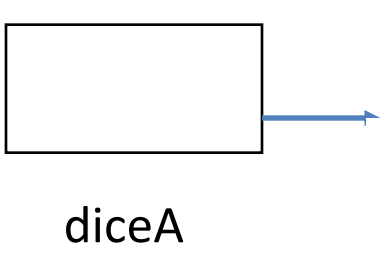
null

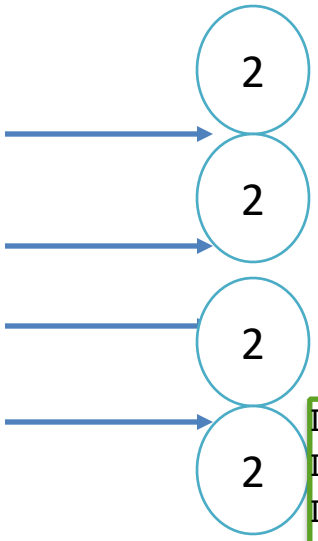
diceB

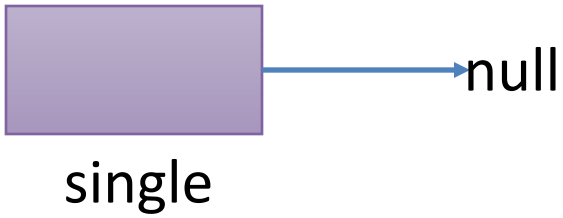
single

null

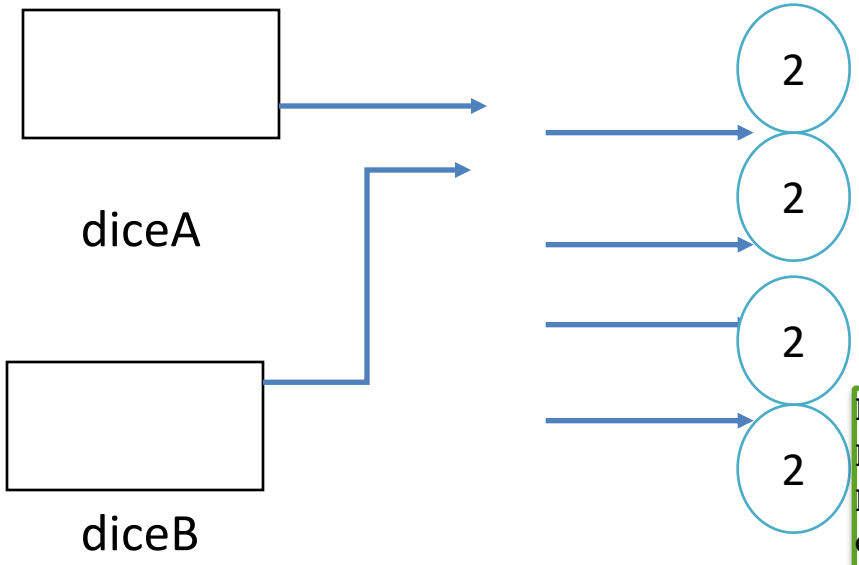
```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```

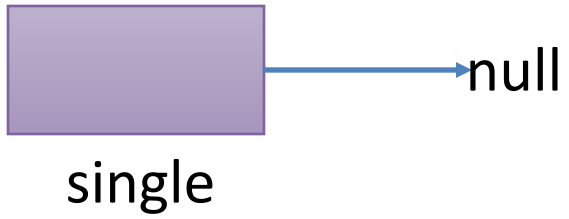




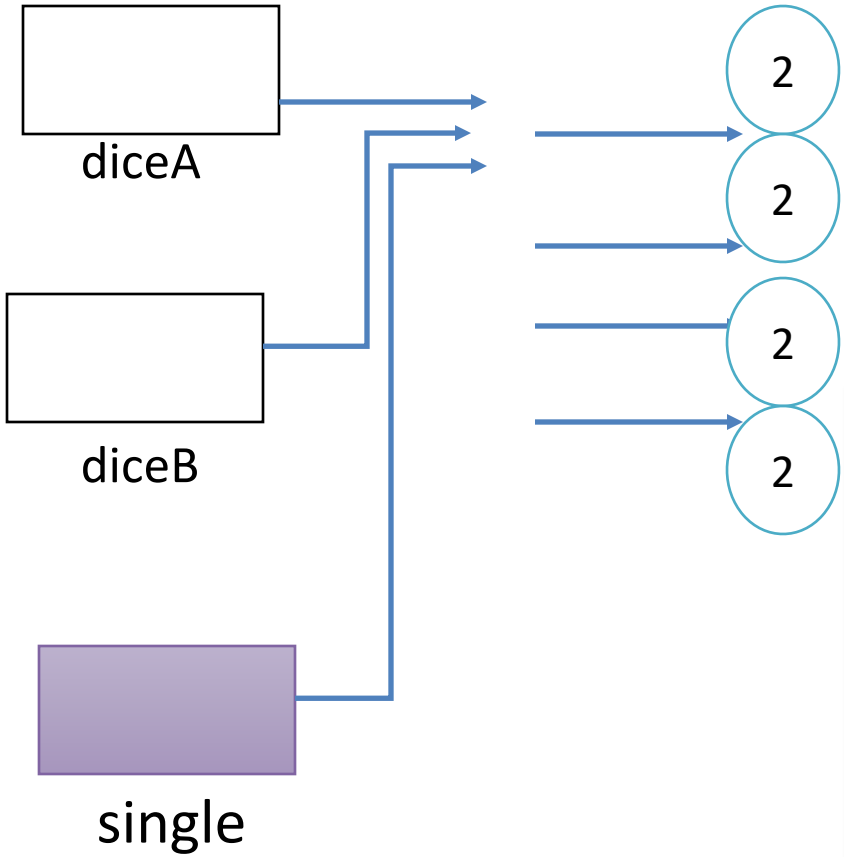



```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```

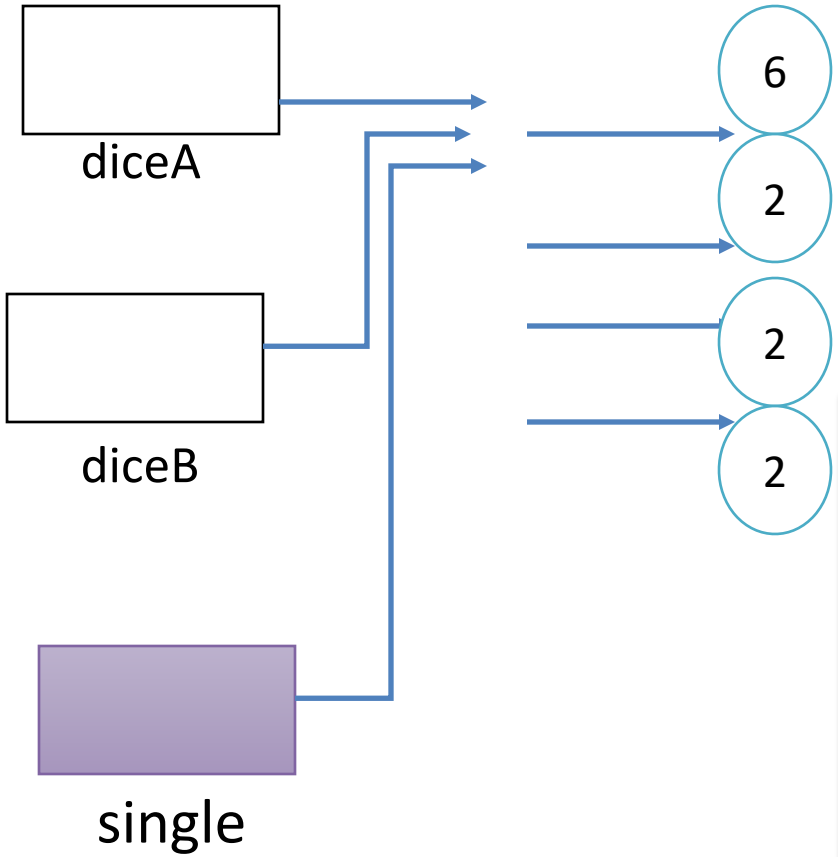





```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```




```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```




```
Die[] diceA;
Die[] diceB;
Die single;
diceA = new Die[4];

for (int i = 0; i < diceA.length; i++)
{
      diceA[i] = new Die(2);
}
diceB = diceA;
single = diceA[0];
single.setFace(6);

for (int i = 0; i < diceA.length; i++)
{
      System.out.printf("A: %d B: %d\n",

diceA[i].getFace(),diceB[i].getFace());
}
```