

Accessors and Mutators

Accessor and Mutator Methods

- `Accessor` method: does not change the state of its implicit parameter

```
double width = box.getWidth();
```

- `Mutator` method: changes the state of its implicit parameter

```
box.translate(15, 25);
```

Accessor and Mutator Methods

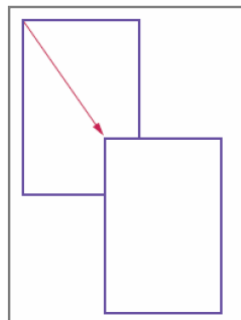


Figure 11:
Using the `translate` Method to Move a Rectangle

Questions?

- Is the `toUpperCase` method of the `String` class an accessor or a mutator?
 - Accessor
- Do you know of any mutator methods in the `String` question?
 - Trick question!
 - There actually aren't any mutators in the `String` class

Object References

Object References

- Describe the location of objects
- The `new` operator returns a reference to a new object

```
Rectangle box = new Rectangle();
```

- Multiple object variables can refer to the same object

```
Rectangle box = new Rectangle(5, 10, 20, 30);  
Rectangle box2 = box;  
box2.translate(15, 25);
```

Continued...

Object References

- Primitive type variables \neq object variables

Object Variables and Number Variables

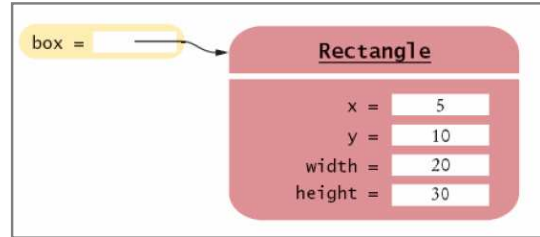


Figure 17:
An Object Variable containing an Object Reference

Object Variables and Number Variables

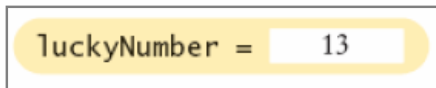


Figure 19:
A Number Variable Stores a Number

Copying Numbers

```
int luckyNumber = 13;  
int luckyNumber2 = luckyNumber;  
luckyNumber2 = 12;
```

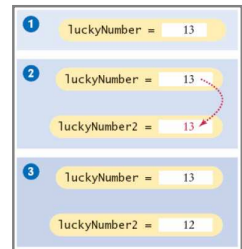


Figure 20:
Copying Numbers

Copying Object References

```
Rectangle box = new Rectangle(5, 10, 20, 30);  
Rectangle box2 = box;  
box2.translate(15, 25);
```

Continued...

Copying Object References

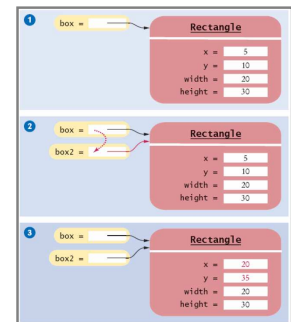


Figure 21:
Copying Object References