

Invoking methods

AudioFile

fileName: String

AudioFile(String)

play(): void

changeVolume(int): void

“A method call is also known as message passing. It is conceptualized as a message (the name of the method and its input parameters) being passed to the object for dispatch.”

https://en.wikipedia.org/wiki/Object-oriented_programming#Dynamic_dispatch.2Fmessage_passing

```
AudioFile af =  
    new AudioFile("mysong.mp3");
```

```
af.play();
```

```
af.changeVolume(11);
```

“it is the responsibility of the object, not the external code, to 'on-demand' select the procedural code to run/execute in response to a method call, typically by looking up the method at run time in a table associated with the object. This feature is known as dynamic dispatch, and distinguishes an object from an abstract data type (or module), which has a fixed (static) implementation of the operations for all instances.”

AudioFile

fileName: String

AudioFile(String)

play(): void

changeVolume(int): void

- `changeVolume` is a method
- `af` is a reference to an `AudioFile` object

`af.changeVolume(11)`

.. invokes the `changeVolume` method on/associated with `af`.

All other `Audio File` objects (if any) remain the same

af1

af2

AudioFile

fileName: "mysong.mp3"

volume: 7

AudioFile

fileName: "othersong.mp3"

volume: 7

```
af1.changeVolume(11);
```

af1

af2

AudioFile

fileName: "mysong.mp3"

volume: 11

AudioFile

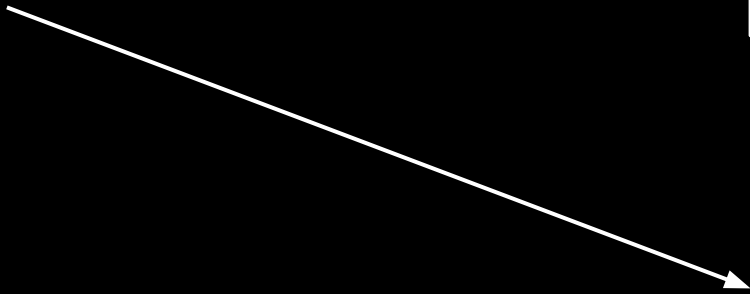
fileName: "othersong.mp3"

volume: 7

```
af2.changeVolume(1);
```

af1

af2



AudioFile

fileName: "mysong.mp3"

volume: 11

AudioFile

fileName: "othersong.mp3"

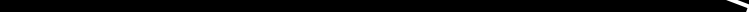
volume: 1

```
AudioFile af3 = af1;
```

af1

af2

af3



AudioFile

fileName: "mysong.mp3"

volume: 11

AudioFile

fileName: "othersong.mp3"

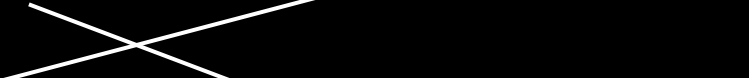
volume: 1


```
af3.changeVolume(9);
```

af1

af2

af3



AudioFile
fileName: "mysong.mp3"
volume: 9

AudioFile
fileName: "othersong.mp3"
volume: 1