



Figure 10.7 Object creation and deletion

object. For example, a PhoneCall object may self-destruct, when the caller hangs up, or may be cut off (destroyed) by the exchange.

Figure 10.7 shows, as a sequence diagram, the same interaction as the collaboration diagram in Figure 10.1. We have added the destruction of the interface object. When there are no more bikes to add, the control object, MaintainBike, sends a `destroy()` message to the interface object MaintainBikeUI.

The `destroy()` message will only be sent if the condition `[no more bikes to add]` evaluates to true.

*Iteration.* In Chapter 6 we explained message iteration: we can use an asterisk to indicate a message that is sent repeatedly. We can specify how many times the message is sent with an iteration clause. We can also indicate that the message is sent to many objects by using a multiobject icon. If we want to specify that a series of messages is repeated, we can do so by outlining the block of messages and adding an iteration clause. In Figure 10.8 everything inside the rectangle is repeated while there are more bikes to add.

*Conditional behaviour and branching.* We have already seen examples of messages guarded by conditions; in Figure 10.8 the `destroy()` message is guarded by the condition `[no more bikes to add]`. Putting a guard on a message means that the message will only be sent if the Boolean expression in the square brackets