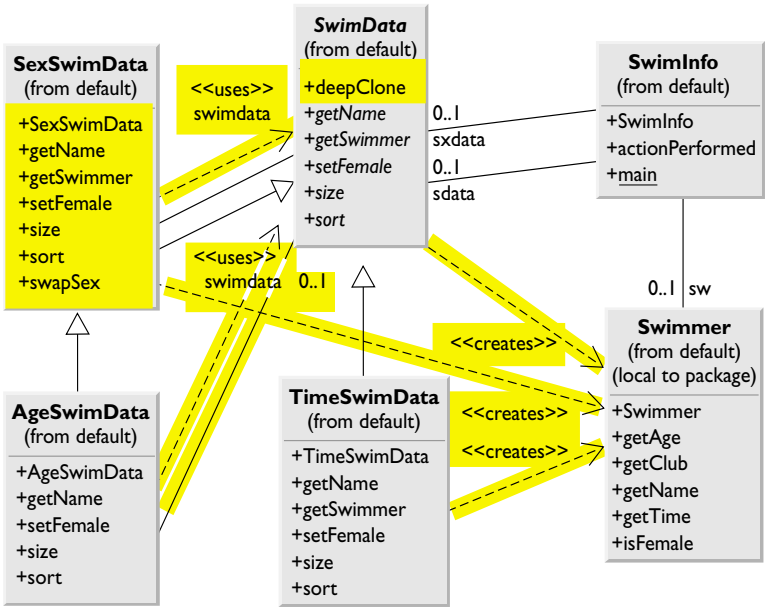

Figure 8.5 The UML diagram for the various SwimData classes.

You should also note that you are not limited to the few subclasses demonstrated here. You could easily create additional concrete classes and register them with whatever code selects the appropriate concrete class. In our previous example program, the user is the deciding point or factory, because he simply clicks on one of several buttons. In a more elaborate case, each concrete class could have an array of characteristics, and the decision point could be a class registry or *prototype manager,* which examines these characteristics and selects the most suitable class. In Java, you can load additional classes dynamically at runtime and make them available through this registry.

You could also combine the Factory Method pattern with the Prototype pattern to have each of several concrete classes use a concrete class different from those available.

Prototype Managers

A prototype manager class can be used to decide which of several concrete classes to return to the client. It can also manage several sets of prototypes at once. For example, in addition to returning one of several classes of swimmers,