

whether to provide more bits per object for the hashing of objects or more bits per object for implementing fast locking for synchronization. An overview of the Jikes RVM object layout is shown in Figure 10-3.

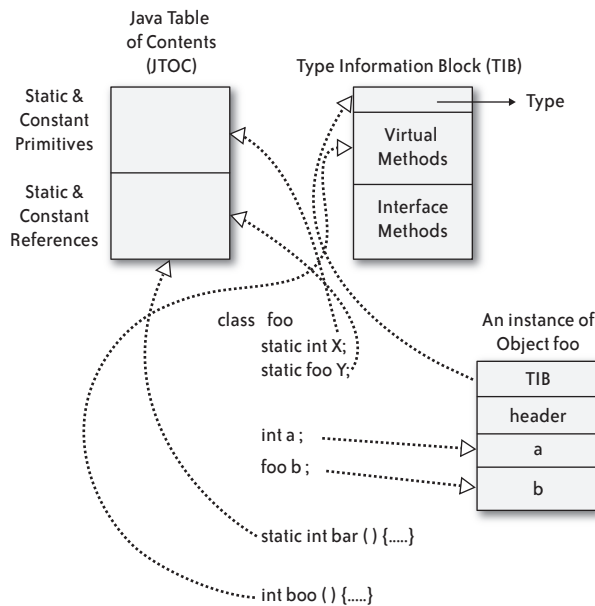


FIGURE 10-3. The layout of objects in Jikes RVM

The default 32-bit object model in Jikes RVM currently uses two words for the object header: the first references the *Type Information Block* (TIB), and the second holds status information for the object on locking, hashing, and garbage collection. After the object header comes the fields of the object. For an array, the first field is the array length and the remainder are the array elements. To avoid a displacement for accesses to an array, the size of which would be the size of the object header and the array length field, all references to objects actually reference a location three words into the object. This allows element zero of an array to be at offset zero within the object, but it also means the object header is always three words behind an object's reference and that the first field of an object is always at a negative offset from the object's reference.

The TIB is responsible for holding the data that is common to every object of a particular type. This data is primarily used for virtual and interface *method dispatch*. Method dispatch is the process of identifying the method that is associated with and should be called for a particular object. Methods within classes are allocated locations within the TIB, in order to enable fast and efficient method dispatch. The TIB also holds values that allow fast runtime type information to be determined, which speeds Java's `instanceof` and `checkcast` operations. It also holds special methods to process an object during garbage collection.