Wrapper classes

int is a primitive type. Variables of type int store the actual binary value for the integer you want to represent. int.parseInt("1") doesn't make sense because int is *not* a class and therefore doesn't have any methods.

Integer is a class, no different from any other in the Java language. Variables of type Integer store *references* to Integer objects, just as with any other reference (object) type. Integer.parseInt("1") is a call to the static method parseInt from class Integer (note that this method actually returns an int and not an Integer).

To be more specific, Integer is a class with a single field of type int. This class is used where you need an int to be treated like any other object, such as in generic types or situations where you need nullability.

Note that every primitive type in Java has an equivalent *wrapper* class:

* byte has Byte
* short has Short
* int has Integer
* long has Long
* boolean has Boolean
* char has Character
* float has Float
* double has Double

Wrapper classes inherit from Object class, and primitive don't. So it can be used in collections with Object reference or with Generics.

Since java 5 we have autoboxing, and the conversion between primitive and wrapper class is done automatically. Beware, however, as this can introduce subtle bugs and performance problems; being explicit about conversions never hurts.