Split `is_native_to_class()` to two functions depending on if extras is NULL or not

I profiled server for a couple of turns of `europe_1900WWI` scenario. `is_native_to_class()` clocks over 5% of the CPU time. It's typically called with NULL extras parameter, so it really doesn't do much per invocation, but it runs extremely often. One of the few things it does do, is checking that extras parameter is NULL. We can get away from that extras is NULL -check by introducing sibling function that does not take extras parameter at all. In practice I'm about to drop extras parameter from `is_native_to_class()` and to introduce new `is_native_to_class_with_extras()` that does take extras parameter. Most existing `is_native_to_class()` callers will keep on calling it, but without NULL extras parameter. Some callers are changed to call `is_native_to_class_with_extras()`.