INDIVIDUAL ASSIGNMENT 03

ABEL ANDUALEM   ID 1411130

Detecting Illegal Access to Private Members

Rule: Private members accessible **only within their defining class** (except nested/friend contexts).

Implementation:

1. **Symbol Table:** Store visibility (`private/public`) with each member
2. **Access Check:** When accessing `obj.field` or `obj.method()`:
   - Get member's visibility from symbol table
   - Get current scope/class context
   - **Reject** if: `visibility == private` AND `current_class ≠ defining_class`

Special Cases:

- **Nested Classes:** Inner classes can access outer class's private members
- **Friend Structures:** Language-specific (C++ `friend`, Java package-private)
- **Reflection:** Runtime access may bypass checks (separate mechanism)

Example Error:

```
java
class A { private int x; }
class B { void m() { A a; a.x = 5; } }  // ❌Illegal: x private in A
```