12.

a.)
```
boolean subClasseDe( Classe sup) {
    if (this == sup) //reflexiva
        return true;
    /* if ( sup == superClasse)
        return true; */
    // transitiva
    if (superClasse != null)
        return superClasse.subClasseDe(sup);
    else
        return false;
}
```

b)
```
Classe tipoExp( TabSimb<Classe> tenv) {
    Classe tobj = objeto.tipoExp(tenv);
    Metodo m = tobj.get(nomeMetodo);
    if(m == null)
        // erro de método não existe
    if( m.nomeParams.length != args.length)
        // erro de aridade
    for (int i=0; i<args.length; i++) {
        Classe targ = args[i].tipoExp(tenv);
```

```
        Classe tparam = am.tipoParams[i];
        if( ! targ.subClasseDe(tparam))
            //erro de tipo no arg i

    }
    return am.tipoRet;

}

c)  void checaTipo(Classe tipoThis) {
        TabSimb<Classe> tenv =
                new TabSimb<Classe>();
        tenv.inserir("this", tipoThis);
        for (int i=0; i < tipoParams.length; i++)
            tenv.inserir(nomeParams[i], tipoParams[i]);
        Classe tcampo;
        for ( Exp exp : campo)
            tcampo = exp.tipoExp(tenv);

        if(! tcampo.subClasseDe(tipoRet))
            // erro tipo da ultima exp

    }

    Para incluir campos (estilo de Java):
        for ( String campo: tipoThis.campos.keys())
            tenv.inserir( campo, tipoThis.campos.get(campo));
```