```
-- select meta-model types of input and output
modeltype AD uses 'http://www.eclipse.org/uml2/5.0.0/UML';
modeltype mitigation uses 'http://sse.uni-due.de/mitigationtemplate';
modeltype hazardanalysis uses 'http://sse.uni-due.de/FHA';
modeltype HRD uses 'http://sse.uni-due.de/HRD';
--- define transformation function signature
--- AD^hmr is the set of activity diagrams containing
--- hazard-mitigating requirements out of which a
--- Hazard Relation Diagram hazRD is to be created.
--- CM^h is a set of partial mitigations needed to append mitigation partitions
--- fha(actD^fr) is the results of the hazard analyses that was conducted on the
--- on the hazard-inducing requirements actD^fr which yield contextual information
--- about the hazard h that was mitigated in CM^h
transformation generateHazardRelationDiagram
    (in actD^hmr:AD, in CM^h:template, in fha(actD^fr): hazardanalysis, out hazRD:HRD);

main() {
    --- select mitigated hazard by checking which hazard is referenced in first
    --- partial mitigation in the array
    let haz = h ∈_t pm_1^h | pm_1^h ∈ CM^h
    --- check if all templates reference same hazard
    foreach pm_i^h ∈ CM^h {
        if h ∈_t pm_i^h ≠ haz {
            throw error("Multiple Conceptual Mitigations detected. Aborting");
            return hazRD = ∅;
        }
    }
    --- check if the mitigated hazard is part of the hazard analysis results
    --- assume the mitigated hazard is not part of the hazard analysis results
    boolean hazardFound = false
    foreach res ∈ fha(actD^fr) {
        if res = haz { hazardFound = true }
    }
    --- if hazard wasn't found, Hazard Relation Diagram cannot be created
    if hazardFound = false {
        throw error
        ("Mitigated Hazard is not part of Hazard Analysis Result Set. Aborting.");
        return hazRD = ∅;
    }
    --- check if hazard-mitigating requirements are referenced
    --- in some mitigation template.
    foreach actD_i^hmr ∈ AD^hmr {
        --- assume activity diagram is not referenced in mitigation template
        boolean adReferenced = false
        foreach pm_i^h ∈ CM^h {
            if actD_i^hmr ∈_t pm_i^h {
                adReferenced = true
            }
        }
        if adReferenced = false {
            throw error("No Mitigation Template available for Hazard-Mitigating
                Requirements. Cannot create mitigation partition. Aborting.");
        return hazRD = ∅;
```

```
54              }
55          }
```

Listing 6 Pseudo-Code Signature sig of the QVTo Script q^{hrd}.