

```

class Family2PersonTransformation {
    ...

    def private void addRules() {
        ...

        rules.add(new Register2Register(sourceModel, targetModel, corrModel, decision))
        rules.add(new MotherDaughter2Female(sourceModel, targetModel, corrModel, decision))
        rules.add(new FatherSon2Male(sourceModel, targetModel, corrModel, decision))
    }
    ...
}

```

1

```

class MotherDaughter2Female extends FamilyMember2Person {
    ...

    override sourceToTarget(String s) {
        sourceModel.allContents.filter(typeof(Family))
            .forEach [ family |
                val corrregister = family.eContainer.getCorrModelElem
                val females = ECollections.newBasicEList()
                females.addAll(family.daughters)
                if (family.mother != null) females.add(family.mother)
                females.forEach [ member |
                    val corrFemale = member.getOrCreateCorrModelElement(s)
                    val female = getOrCreatePersonElement(family.name + ", " +
                                                                member.name, corrFemale, ...)
                    (corrregister.personElement as PersonRegister).getPersons().add(female)
                ]
            ]
    }

    override targetToSource(String s) {
        targetModel.allContents.filter(typeof(Female)).forEach [ p | ... ]
    }
}

```

2