```
    class Person
values
private
        MaennlicheVornamen = {"Hans", "Georg"};
private
        WeiblicheVornamen = {"Heidrun", "Cornelia"}
functions
private
        istMaennlicherVorname : char* → B
        istMaennlicherVorname (v) △
          v ∈ MaennlicheVornamen;
private
        istWeiblicherVorname : char* → B
        istWeiblicherVorname (v) △
          v ∈ WeiblicheVornamen;
private
        keineZeitweiseBigamie : Hochzeit-set → B
        keineZeitweiseBigamie (s) △
          ∀ h1, h2 ∈ s · h1 ≠ h2 ⇒
                        h1.Hochzeitsdatum.groesserGleich (h2.Scheidungsdatum) ∨

                        h2.Hochzeitsdatum.groesserGleich (h1.Scheidungsdatum);
private
        initNachname : Person-set → char*
        initNachname (e) △
          if card e = 1
          then let m ∈ e in
               m.Nachname
          else if card e = 2
               then let Mutter ∈ e be st Mutter.Geschlecht = WEIBLICH in
                    let aktuelleEheDerMutter ∈ Mutter.hochzeit be st
                                aktuelleEheDerMutter.Scheidungsdatum = Datum`invalidDatum in
                    let aktuellerEhemannDerMutter = aktuelleEheDerMutter.ehemann in
                    if aktuellerEhemannDerMutter ∈ e
                    then aktuelleEheDerMutter.Familienname
                    else Mutter.Nachname
               else "NameVomGericht"
instance variables
          public Nachname : char* := initNachname (elter);
          Geburtsname : char*;
          Vorname : char*;
          inv Geschlecht = MAENNLICH ⇒ istMaennlicherVorname (Vorname)
          inv Geschlecht = WEIBLICH ⇒ istWeiblicherVorname (Vorname)
          public Geschlecht : Util`Genus;
          Geburtsort : char*;
          public Geburtsdatum : Datum;
```

$Staatsbuergerschaft : Util`Nation := \text{DE};$
$Bekenntnis : \mathsf{char}^*;$
$Status : Util`Personenstand := \text{LEDIG};$
$\mathsf{inv}\ Status = \text{TOT} \Leftrightarrow Sterbedatum = Datum`invalidDatum$
$Sterbedatum : Datum;$
$Beruf : \mathsf{char}^*;$
$BiometrischeMerkmale : Util`BioDaten;$
$Ausweisnummer : \mathbb{N} := 0;$
$\mathsf{public}\ Wohnsitz : Adresse^* := [];$
$\mathsf{inv\ len}\ Wohnsitz > 0 \Rightarrow$
$\mathsf{self} \in \bigcup \{adr.person \mid adr \in \mathsf{elems}\ Wohnsitz\}$
$\mathsf{public}\ hochzeit : Hochzeit\text{-}\mathsf{set};$
$\mathsf{inv\ card}\ hochzeit > 1 \Rightarrow$
$\mathsf{if}\ Geschlecht = \text{MAENNLICH}$
$\mathsf{then}\ \{\mathsf{self}\} = \{h.ehemann \mid h \in hochzeit\}$
$\mathsf{else}\ \{\mathsf{self}\} = \{h.ehefrau \mid h \in hochzeit\}$
$\mathsf{inv\ card}\ hochzeit > 0 \Rightarrow keineZeitweiseBigamie\ (hochzeit)$
$\mathsf{inv\ card}\ hochzeit > 0 \Rightarrow$
$\mathsf{card}\ \{h \mid h \in hochzeit \cdot$
$h.Scheidungsdatum = Datum`invalidDatum\} \leq$
1
$elter : Person\text{-}\mathsf{set};$
$\mathsf{inv\ card}\ elter \leq 2$
$\mathsf{inv\ card}\ elter > 0 \Rightarrow \mathsf{self} \in \bigcup \{e.kind \mid e \in elter\}$
$\mathsf{inv\ card}\ elter = 0 \Rightarrow Vormund \neq \mathsf{nil}$
$\mathsf{inv\ card}\ elter > 0 \Rightarrow \forall\, e \in elter \cdot$
$Geburtsdatum.minus\ (e.Geburtsdatum) \geq 8$
$\mathsf{inv\ card}\ elter = 1 \Rightarrow \mathsf{let}\ e \in elter\ \mathsf{in}$
$Nachname = e.Nachname$
$\mathsf{inv\ card}\ elter = 2 \Rightarrow$
$\forall\, e \in elter \cdot$
$\mathsf{card}\ e.hochzeit > 0 \Rightarrow$
$\forall\, h \in e.hochzeit \cdot$
$Geburtsdatum.inbetween\ (h.Hochzeitsdatum, h.Scheidungsdatum) \Rightarrow$
$Nachname = h.Familienname$
$kind : Person\text{-}\mathsf{set};$
$\mathsf{inv\ card}\ kind > 0 \Rightarrow \mathsf{self} \in \bigcup \{k.elter \mid k \in kind\}$
$Vormund : [Person];$
$\mathsf{inv}\ Vormund \neq \mathsf{nil} \Rightarrow \mathsf{self} \in Vormund.Muendel$
$Muendel : Person\text{-}\mathsf{set};$
$\mathsf{inv\ card}\ Muendel > 0 \Rightarrow$
$(\forall\, m \in Muendel \cdot m.Vormund \neq \mathsf{nil}\ ) \wedge$
$(\{\mathsf{self}\} = \{m.Vormund \mid m \in Muendel\})$

operations
public

```
        getOpernKarte : ℕ →o 𝔹
        getOpernKarte (Einzahlung) ≜
          return true
        pre Einzahlung ≥ 100 ;
public
        Alter : Datum →o ℕ
        Alter (amDatum) ≜
          return amDatum.minus (Geburtsdatum)
        pre amDatum.groesserGleich (Geburtsdatum) ;
public
        sammleVorfahren : ℕ →o Person*
        sammleVorfahren (level) ≜
          if (level = 0) ∨ (card elter = 0)
          then return []
          elseif (card elter = 1)
          then let e1 ∈ elter in
               return [e1] ⌢ e1.sammleVorfahren (level − 1)
          else let e1 ∈ elter in
               let e2 ∈ elter be st e2 ≠ e1 in
               return [e1, e2] ⌢
                      e1.sammleVorfahren (level − 1) ⌢
                      e2.sammleVorfahren (level − 1)
        pre level ≥ 0 ;
private
        istNeffe : Person →o 𝔹
        istNeffe (n) ≜
          let Geschwister =
                ⋃{e.kind | e ∈ elter} \ {self} in
          let NeffenUndNichten = ⋃{g.kind | g ∈ Geschwister} in
          return (n ∈ NeffenUndNichten) ∧
                 (n.Geschlecht = MAENNLICH);
private
        Cousins : () →o Person-set
        Cousins () ≜
          let Grosseltern = ⋃{e.elter | e ∈ elter} in
          let OnkelUndTanten =
                ⋃{g.kind | g ∈ Grosseltern} \ elter in
          let CousinsUndCousinen = ⋃{o.kind | o ∈ OnkelUndTanten} in
          return {c | c ∈ CousinsUndCousinen · c.Geschlecht = MAENNLICH};
private
```

$istStiefvater : Person \xrightarrow{o} \mathbb{B}$
$istStiefvater\,(s) \triangleq$
  let $Mutter = \iota\, e \in elter \cdot e.Geschlecht = \text{WEIBLICH}$ in
  let $GattenDerMutter = \{h.ehemann \mid h \in Mutter.hochzeit\}$ in
  let $HochzeitenVonSMitMutter =$
    $\{h \mid h \in s.hochzeit \cdot h.ehefrau = Mutter\}$ in
  return $(s.Geschlecht = \text{MAENNLICH}) \wedge$
    $(s \in (GattenDerMutter \setminus elter)) \wedge$
    $(\exists\, h \in HochzeitenVonSMitMutter \cdot$
      $h.Scheidungsdatum.groesserGleich\,(Datum`today\,()))$
end $Person$