

David MONTET
Yannick AZEAU

IUP1
Mdp : nteiup1

TP

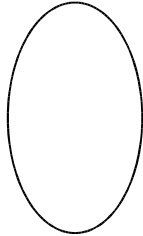
SPECIFICATION FORMELLE AVEC B

Système de contrôle de clients

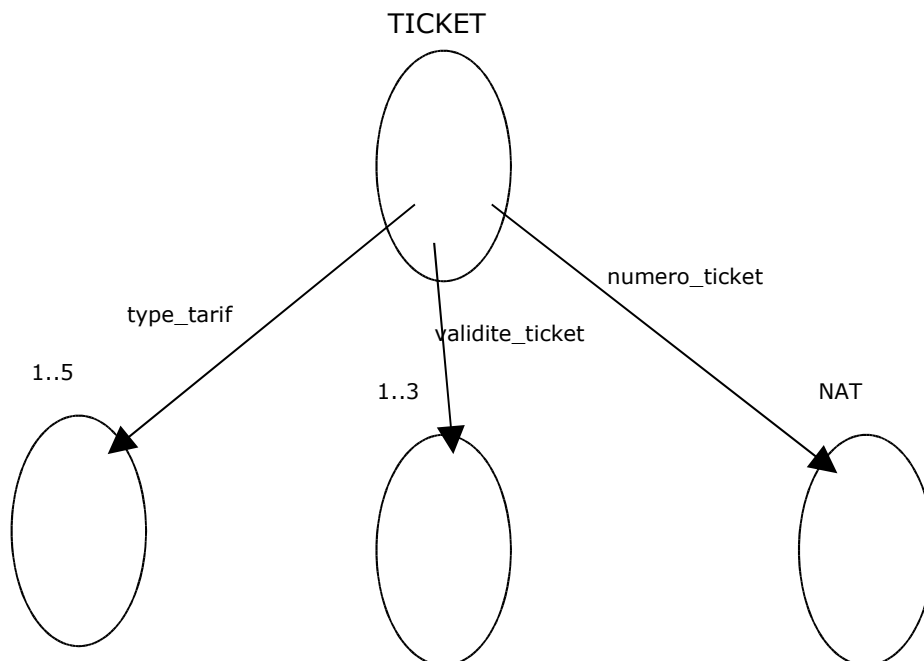
Modélisation Mathématiques

- VOYANT_M

COULEUR



- CODAGE_M



type_tarif : fonction partielle non injective
type_tarif : TICKET \rightarrow 1..5

validite_ticket : fonction partielle non injective
validite_ticket : TICKET \rightarrow 1..3

numero_ticket : fonction partielle injective
numero_ticket : TICKET \rightarrow NAT

VOYANT_M

```
MACHINE
  VOYANT_M
SETS
  COULEUR = {Rouge, Vert}
VARIABLES
  couleur_courante
INVARIANT
  couleur_courante : COULEUR
INITIALISATION
  couleur_courante := Vert
OPERATIONS

  set_vert = /* Met le voyant au vert */
  BEGIN
    couleur_courante := Vert
  END;

  set_rouge = /*Met le voyant au rouge */
  BEGIN
    couleur_courante := Rouge
  END;

  cc <-- couleur = /* Retourne la couleur du voyant */
  BEGIN
    cc := couleur_courante
  END
END
```

CODAGE_M

```
/* iup1
   Azeau Yannick
   Montet David */
MACHINE
  CODAGE_M(TICKET)

CONSTRAINTS
  card(TICKET) : NAT

VARIABLES
  type_tarif,
  valideite_ticket,
  numero_ticket

INVARIANT
  type_tarif      : TICKET +-> 1..5 &
  valideite_ticket : TICKET +-> 1..3 &
  numero_ticket   : TICKET >+> NAT

INITIALISATION
  type_tarif      :={} ||
  valideite_ticket :={} ||
  numero_ticket   :={}

OPERATIONS
  generer_code (tt, type) = /* Attribue au ticket t un code incluant le type de
tarif type, la validité 1 un entier quelconque non déjà utilisé */
  PRE
    tt : TICKET &
    tt  /: dom(type_tarif) &
    type : 1..5
  THEN
    type_tarif := type_tarif \/ {tt |-> type} ||
    valideite_ticket := valideite_ticket \/ { tt |-> 1} ||

    ANY numero WHERE numero : NAT & numero /: ran(numero_ticket)
  THEN
    numero_ticket := numero_ticket \/ {tt |-> numero}
  END
END;

  changer_type_tarif (tt, nvtype) = /* Attribue au ticket t un nouveau type
nvtype de tarif. Le ticket doit avoir une validité donnée */
  PRE
    tt : TICKET &
    tt  : dom(type_tarif) &
    nvtype : 1..5
  THEN
    type_tarif(tt) := nvtype
  END;

  invalider_code(tt) = /* Supprime du systme le ticket t*/
  PRE
    tt : TICKET &
    tt  : dom(type_tarif)
  THEN
    /* Suppression des 3 relations */
    numero_ticket := numero_ticket - {tt |-> numero_ticket(tt)} ||
    type_tarif := type_tarif - {tt |-> type_tarif(tt)} ||
    valideite_ticket := valideite_ticket -{ tt |-> valideite_ticket(tt)}
  END
```

```
END
```

COMPTEUR_M

```
/* iup1
   Azeau Yannick
   Montet David */
MACHINE
  COMPTEUR_M
CONSTANTS
  MAX_PLACES
PROPERTIES
  MAX_PLACES : NAT & MAX_PLACES < MAXINT
VARIABLES
  nb_client_interieur /* Nombre de clients à l'intérieur du centre */
INVARIANT
  nb_client_interieur : 0..MAX_PLACES
INITIALISATION
  nb_client_interieur := 0
OPERATIONS

  incrementer = /* Incrémente le compteur de 1 */
    PRE
      nb_client_interieur + 1 <= MAX_PLACES
    THEN
      nb_client_interieur := nb_client_interieur + 1
    END;

  decrements = /* Incrémente le compteur de 1 */
    PRE
      nb_client_interieur - 1 >= 0
    THEN
      nb_client_interieur := nb_client_interieur - 1
    END
END
```

CONTROLEUR_M

```
/* iup1
   Azeau Yannick
   Montet David */
MACHINE
  CONTROLEUR_M(TICKET)  /* Controle des tickets en entrée et en sortie */

CONSTRAINTS
  card(TICKET) : NAT

INCLUDES
  COMPTEUR_M
  CODAGE_M(TICKET)

OPERATIONS

  lire_entree(tt) =
  PRE
    tt : dom(validite_ticket) &          /* Ticket valide */
    nb_client_interieur < MAX_PLACES & /* Il y a assez de place */
    valide_ticket(tt) : 1..2
  THEN
    IF valide_ticket(tt)=1 THEN
      valide_ticket(tt) :=2
    END
    ||
    incrementer()
  END;

  lire_sortie(tt) =
  PRE
    tt : dom(validite_ticket)          /* Ticket valide */
  THEN
    IF valide_ticket(tt)=2 THEN
      valide_ticket(tt) :=3
    END IF
    ||
    decrementer()
  END;

  r <-- rejet_ticket(tt) =
  PRE
    tt : dom(validite_ticket)
  THEN
    IF valide_ticket(tt)=3 THEN
      r := 'INVALIDE'
    END IF
  END

END

END
```