

Tests

Personnage

Cas de test : Personnage::testInitWorking

CI : $\text{nom} = \text{"Ryan"} \wedge \text{l} = 30 \wedge \text{h} = 30 \wedge \text{p} = 30 \wedge \text{a} = 10 \wedge \text{v} = 100 \wedge \text{f} = 100$

Operation : $P0 \stackrel{def}{=} \text{init}(\text{nom}, \text{l}, \text{h}, \text{p}, \text{f}, \text{v}, \text{a})$

Oracle :

$\text{nom}(P0) = \text{"Ryan"}$
 $\text{largeur}(P0) = 30$
 $\text{profondeur}(P0) = 30$
 $\text{hauteur}(P0) = 30$
 $\text{pointsDeVie}(P0) = 100$
 $\text{force}(P0) = 100$
 $\text{argent}(P0) = 10$

Cas de test : Personnage::testInitFailing

CI : $\text{nom} = \text{"Joe"} \wedge \text{l} = 30 \wedge \text{h} = 30 \wedge \text{p} = 30 \wedge \text{a} = 10 \wedge \text{v} = 100 \wedge \text{f} = 100$

Operation : $P0 \stackrel{def}{=} \text{init}(\text{nom}, \text{l}, \text{h}, \text{p}, \text{f}, \text{v}, \text{a})$

Oracle :

$\text{nom} \neq \text{"Alex"} \wedge \text{nom} \neq \text{"Ryan"}$
Une exception est levee

Cas de test : Personnage::testInitFailing

CI : $\text{nom} = \text{"Alex"} \wedge \text{l} = -5 \wedge \text{h} = 30 \wedge \text{p} = 30 \wedge \text{a} = 10 \wedge \text{v} = 100 \wedge \text{f} = 100$

Operation : $P0 \stackrel{def}{=} \text{init}(\text{nom}, \text{l}, \text{h}, \text{p}, \text{f}, \text{v}, \text{a})$

Oracle :

$\text{l} \leq 0$
Une exception est levee

Cas de test : Personnage::testInitFailing

CI : $\text{nom} = \text{"Alex"} \wedge \text{l} = 30 \wedge \text{h} = -5 \wedge \text{p} = 30 \wedge \text{a} = 10 \wedge \text{v} = 100 \wedge \text{f} = 100$

Operation : $P0 \stackrel{def}{=} \text{init}(\text{nom}, \text{l}, \text{h}, \text{p}, \text{f}, \text{v}, \text{a})$

Oracle :

$\text{h} \leq 0$
Une exception est levee

Cas de test : Personnage::testInitFailing

CI : $\text{nom} = \text{"Alex"} \wedge \text{l} = 30 \wedge \text{h} = 30 \wedge \text{p} = -5 \wedge \text{a} = 10 \wedge \text{v} = 100 \wedge \text{f} = 100$

Operation : $P0 \stackrel{def}{=} \text{init}(\text{nom}, \text{l}, \text{h}, \text{p}, \text{f}, \text{v}, \text{a})$

Oracle :

$\text{p} \leq 0$
Une exception est levee

Cas de test : Personnage::testInitFailing

CI : $\text{nom} = \text{"Alex"} \wedge \text{l} = 30 \wedge \text{h} = 30 \wedge \text{p} = 30 \wedge \text{a} = -10 \wedge \text{v} = 100 \wedge \text{f} = 100$

Operation : $P0 \stackrel{def}{=} \text{init}(\text{nom}, \text{l}, \text{h}, \text{p}, \text{f}, \text{v}, \text{a})$

Oracle :

$\text{a} < 0$

Une exception est levee

Cas de test : Personnage::testInitFailing

CI : nom = "Alex" \wedge l = 30 \wedge h = 30 \wedge p = 30 \wedge a = 10 \wedge v = 0 \wedge f = 100

Operation : P0 $\stackrel{def}{=}$ init(nom,l,h,p,f,v,a)

Oracle :

v \leq 0

Une exception est levee

Cas de test : Personnage::testInitFailing

CI : nom = "Alex" \wedge l = 30 \wedge h = 30 \wedge p = 30 \wedge a = 10 \wedge v = 100 \wedge f = -8

Operation : P0 $\stackrel{def}{=}$ init(nom,l,h,p,f,v,a)

Oracle :

f \leq 0

Une exception est levee

Cas de test : Personnage::RetraitVieWorking

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

Operation : P0 $\stackrel{def}{=}$ retraitPdV(personnage, 3)

Oracle :

argent(P0) = 7 = (10-3)

Cas de test : Personnage::RetraitVieFailing

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

Operation : P0 $\stackrel{def}{=}$ retraitPdV(personnage, -5);

Oracle :

-5 < 0

Une exception est levee

Cas de test : Personnage::RetraitArgentWorking

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

Operation : P0 $\stackrel{def}{=}$ retraitArgent(personnage, 3)

Oracle :

pointsDeVie(P0) = 7 = (10-3)

Cas de test : Personnage::RetraitArgentFailing

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

Operation : P0 $\stackrel{def}{=}$ retraitArgent(personnage, -5);

Oracle :

-5 < 0

Une exception est levee

Cas de test : Personnage::RetraitArgentFailing

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

Operation : P0 $\stackrel{def}{=}$ retraitArgent(personnage, 108);

Oracle :

108 > 100

Une exception est levee

Cas de test : Personnage::DepotAgentWorking

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

Operation : P0 $\stackrel{def}{=}$ depotArgent(personnage, 3)

Oracle :

pointsDeVie(P0) = 7 = (10-3)

Cas de test : Personnage::DepotAgentFailing

CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

```

Operation : P0  $\stackrel{def}{=}$  depotArgent(personnage, -5);
Oracle :
-5 < 0
Une exception est levee

Cas de test : Personnage::ramasserObjetWorking
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100),
    obj = Objet::init("arme",10,0)
Operation : P0  $\stackrel{def}{=}$  ramasserObjet(personnage, obj)
Oracle :
objetEquipe(personnage) = obj
force(personnage) = 110

Cas de test : Personnage::ramasserObjetFailing1
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)
    obj = Objet::init("sous",0,40)
Operation : P0  $\stackrel{def}{=}$  ramasserObjet(personnage, obj);
Oracle :
Objet:est_equipable(obj) = false
Une exception est levee

Cas de test : Personnage::ramasserObjetFailing2
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)
    p = Personnage::init("Ryan",10,10,10,100,100)
    personnage = ramasser_perso(p)
    obj = Objet::init("arme",10,0)
Operation : P0  $\stackrel{def}{=}$  ramasserObjet(personnage, obj);
Oracle :
est_equipe_perso(personnage) = true
Une exception est levee

Cas de test : Personnage::ramasserArgentWorking
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100),
    obj = Objet::init("piece",0,40)
Operation : P0  $\stackrel{def}{=}$  ramasserObjet(personnage, obj)
Oracle :
objetEquipe(personnage) = obj
force(personnage) = 110

Cas de test : Personnage::ramasserArgentFailing1
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)
    personnage = retraitVie(personnage,200)
    obj = Objet::init("soussou",0,40)
Operation : P0  $\stackrel{def}{=}$  ramasserArgent(personnage, obj)
Oracle :
estVaincu(P0) = true
Une exception est levee

Cas de test : Personnage::ramasserArgentFailing2
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)
    obj = Objet::init("soussou",30,0)
Operation : P0  $\stackrel{def}{=}$  ramasserArgent(personnage, obj);
Oracle :
Objet:est_DeValeur(obj) = false
Une exception est levee

Cas de test : Personnage::ramasserPersoWorking
CI : personnage = init("Alex", 10, 10, 10, 10, 100, 100)

```

```

        perso2 = init("Ryan", 10, 10, 10, 10, 100, 100)
Operation : P0  $\stackrel{def}{=}$  ramasserPerso(personnage, perso2)
Oracle :
    persoEquipe(P0) = perso2

Cas de test : Personnage::ramasserPersoFailing1
CI :   personnage = init("Alex", 10, 10, 10, 10, 100, 100)
        perso2 = init("Ryan", 10, 10, 10, 10, 100, 100)
        personnage = personnage.retraitPdV(10000);
Operation : P0  $\stackrel{def}{=}$  ramasserPerso(personnage, perso2 )
Oracle :
    estVaincu(P0) = true
    Une exception est levee

Cas de test : Personnage::jeterWorking
CI :   personnage = init("Alex", 10, 10, 10, 10, 100, 100)
        obj = Objet::init("arme",10,0)
        personnage = ramasserObjet(personnage,obj)
Operation : P0  $\stackrel{def}{=}$  jeter(personnage)
Oracle :
    persoEquipe(P0) = null
    force(P0) = 100
    objetEquipe(P0) = false

Cas de test : Personnage::jeterFailing1
CI :   personnage = init("Alex", 10, 10, 10, 10, 100, 100)
Operation : P0  $\stackrel{def}{=}$  jeter(personnage)
Oracle :
    estEquipeObjet(personnage) = false
    estEquipePerso(personnage) = false
    Une exception est levee

```

Gangster

```
Cas de test : Gangster::testInitWorking
CI : nom = "Slick"  $\wedge$  l = 10  $\wedge$  h = 10  $\wedge$  p = 10  $\wedge$  f = 10  $\wedge$  v = 10
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  nom(P0) = "Slick"
  largeur(P0) = 10
  hauteur(P0) = 10
  profondeur(P0) = 10
  pointsDeVie(P0) = 10
  force(P0) = 10
  argent(P0) = 0

Cas de test : Gangster::testInitFailing1
CI : nom = ""  $\wedge$  l = 10  $\wedge$  h = 10  $\wedge$  p = 10  $\wedge$  f = 10  $\wedge$  v = 10
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  nom(P0) == ""
  Une exception est levee

Cas de test : Gangster::testInitFailing2
CI : nom = "Slick"  $\wedge$  l = -1  $\wedge$  h = 10  $\wedge$  p = 10  $\wedge$  f = 10  $\wedge$  v = 10
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  l  $\leq$  0
  Une exception est levee

Cas de test : Gangster::testInitFailing3
CI : nom = "Slick"  $\wedge$  l = 10  $\wedge$  h = -1  $\wedge$  p = 10  $\wedge$  f = 10  $\wedge$  v = 10
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  h  $\leq$  0
  Une exception est levee

Cas de test : Gangster::testInitFailing4
CI : nom = "Slick"  $\wedge$  l = 10  $\wedge$  h = 10  $\wedge$  p = -1  $\wedge$  f = 10  $\wedge$  v = 10
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  p  $\leq$  0
  Une exception est levee

Cas de test : Gangster::testInitFailing5
CI : nom = "Slick"  $\wedge$  l = 10  $\wedge$  h = 10  $\wedge$  p = 10  $\wedge$  f = -1  $\wedge$  v = 10
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  f  $\leq$  0
  Une exception est levee

Cas de test : Gangster::testInitFailing6
CI : nom = "Slick"  $\wedge$  l = 10  $\wedge$  h = 10  $\wedge$  p = 10  $\wedge$  f = 10  $\wedge$  v = -1
Operation : G0  $\stackrel{def}{=}$  init(nom,l,h,p,f,v)
Oracle :
  v  $\leq$  0
  Une exception est levee
```

Bloc

```

Cas de test : Bloc::testInitWorking
CI : type = TYPE.VIDE  $\wedge$  o = null
Operation : B0  $\stackrel{def}{=}$  init(type,o)
Oracle :
    type(B0) = TYPE.VIDE
    objet(B0) = 0

Cas de test : Bloc::testInitFailing1
CI : type = TYPE.VIDE  $\wedge$  o = Objet::init("machin",100,0)
Operation : B0  $\stackrel{def}{=}$  init(type,o)
Oracle :
    type== TYPE.VIDE  $\wedge$  o  $\neq$  null
    Une exception est levee

Cas de test : Bloc::testInitFailing2
CI : type = TYPE.OBJET  $\wedge$  o = null
Operation : B0  $\stackrel{def}{=}$  init(type,o)
Oracle :
    type== TYPE.OBJET  $\wedge$  o == null
    Une exception est levee

Cas de test : Bloc::RetirerObjetWorking
CI : b = init(TYPE.OBJET,Objet::init("machin",100,0))
Operation : B0  $\stackrel{def}{=}$  retirerObjet(b)
Oracle :
    type(B0) = TYPE.VIDE
    objet(B0) = null

Cas de test : Bloc::RetirerObjetFailing
CI : b = init(TYPE.VIDE,null)
Operation : B0  $\stackrel{def}{=}$  retirerObjet(b)
Oracle :
    type(b)  $\neq$  TYPE.OBJET
    Une exception est levee

Cas de test : Bloc::PoserObjetWorking
CI : b = init(TYPE.VIDE,null)  $\wedge$  o = Objet::init("machin",100,0)
Operation : B0  $\stackrel{def}{=}$  poserObjet(b,o)
Oracle :
    type(B0) = TYPE.OBJET
    objet(B0) = o

Cas de test : Bloc::PoserObjetFailing
CI : b = init(TYPE.OBJET,Objet::init("machin",100,0))  $\wedge$  o = Objet::init("truc",10,0)
Operation : B0  $\stackrel{def}{=}$  poserObjet(b,o)
Oracle :
    type(b)  $\neq$  TYPE.VIDE
    Une exception est levee

```

Objet

```

Cas de test : Objet::testInitWorking1
CI : n = "machin"  $\wedge$  bonus = 10  $\wedge$  valeur = 0
Operation : O0  $\stackrel{def}{=}$  init(n,bonus,valeur)
Oracle :
    nom(O0) = "machin"

```

```

bonus\_force(O0) = 10
valeur\_marchande(O0) = 0

Cas de test : Objet::testInitWorking2
CI : n = "machin" ∧ bonus = 0 ∧ valeur = 10
Operation : O0  $\stackrel{def}{=}$  init(n,bonus,valeur)
Oracle :
  nom(O0) = "machin"
  bonus\_force(O0) = 0
  valeur\_marchande(O0) = 10

Cas de test : Objet::testInitFailing1
CI : n = "" ∧ bonus = 0 ∧ valeur = 10
Operation : O0  $\stackrel{def}{=}$  init(n,bonus,valeur)
Oracle :
  n == ""
  Une exception est levee

Cas de test : Objet::testInitFailing2
CI : n = "truc" ∧ bonus = 10 ∧ valeur = 10
Operation : O0  $\stackrel{def}{=}$  init(n,bonus,valeur)
Oracle :
  bonus > 0 ∧ valeur > 0
  Une exception est levee

Cas de test : Objet::testInitFailing3
CI : n = "truc" ∧ bonus = -1 ∧ valeur = 0
Operation : O0  $\stackrel{def}{=}$  init(n,bonus,valeur)
Oracle :
  bonus < 0
  Une exception est levee

Cas de test : Objet::testInitFailing4
CI : n = "truc" ∧ bonus = 0 ∧ valeur = -1
Operation : O0  $\stackrel{def}{=}$  init(n,bonus,valeur)
Oracle :
  valeur < 0
  Une exception est levee

```

Terrain

```

Cas de test : Terrain::testInitWorking
CI : p = 50, h = 100, l = 50
Operation : T0  $\stackrel{def}{=}$  init(l,h,p)
Oracle :
  largeur(T0) = l
  hauteur(T0) = h
  profondeur(T0) = p

Cas de test : Terrain::testInitFailing1
CI : p = 0, h = 100, l = 50
Operation : T0  $\stackrel{def}{=}$  init(l,h,p)
Oracle :
  p < 50
  Une exception est levee

Cas de test : Terrain::testInitFailing2
CI : p = 50, h = 100, l = 0

```

Operation : $T0 \stackrel{def}{=} \text{init}(l, h, p)$
Oracle :
 $l < 50$
 Une exception est levee

Cas de test : Terrain::testInitFailing3
CI : $p = 50, h = 10, l = 50$
Operation : $T0 \stackrel{def}{=} \text{init}(l, h, p)$
Oracle :
 $h < 100$
 Une exception est levee

Cas de test : Terrain::testInitFailing4
CI : $p = 60, h = 100, l = 50$
Operation : $T0 \stackrel{def}{=} \text{init}(l, h, p)$
Oracle :
 $(p \% 50) \neq 0$
 Une exception est levee

Cas de test : Terrain::testInitFailing5
CI : $p = 50, h = 100, l = 60$
Operation : $T0 \stackrel{def}{=} \text{init}(l, h, p)$
Oracle :
 $(l \% 50) \neq 0$
 Une exception est levee

Cas de test : Terrain::modifierBlocWorking
CI : $t = \text{init}(500, 100, 500) \wedge b = \text{Bloc}::\text{init}(\text{TYPE.VIDE}, \text{null})$
Operation : $T0 \stackrel{def}{=} \text{modifierBloc}(t, 10, 10, b)$
Oracle :
 $\text{bloc}(T0, 10, 10) = b$

Cas de test : Terrain::modifierBlocFailing1
CI : $t = \text{init}(500, 100, 500) \wedge b = \text{Bloc}::\text{init}(\text{TYPE.VIDE}, \text{null})$
Operation : $T0 \stackrel{def}{=} \text{modifierBloc}(t, -1, 10, b)$
Oracle :
 $x < 0$
 Une exception est levee

Cas de test : Terrain::modifierBlocFailing2
CI : $t = \text{init}(500, 100, 500) \wedge b = \text{Bloc}::\text{init}(\text{TYPE.VIDE}, \text{null})$
Operation : $T0 \stackrel{def}{=} \text{modifierBloc}(t, 1000, 10, b)$
Oracle :
 $x > \text{largeur}(t)$
 Une exception est levee

Cas de test : Terrain::modifierBlocFailing3
CI : $t = \text{init}(500, 100, 500) \wedge b = \text{Bloc}::\text{init}(\text{TYPE.VIDE}, \text{null})$
Operation : $T0 \stackrel{def}{=} \text{modifierBloc}(t, 10, -1, b)$
Oracle :
 $y < 0$
 Une exception est levee

Cas de test : Terrain::modifierBlocFailing4
CI : $t = \text{init}(500, 100, 500) \wedge b = \text{Bloc}::\text{init}(\text{TYPE.VIDE}, \text{null})$
Operation : $T0 \stackrel{def}{=} \text{modifierBloc}(t, 10, 1000, b)$
Oracle :
 $y > \text{profondeur}(t)$
 Une exception est levee


```

Cas de test : Terrain::modifierBlocFailing5
CI : t = init(500,100,500) ∧ b = null
Operation T0  $\stackrel{def}{=}$  modifierBloc(t, 10,10, b)
Oracle :
    b == null
    Une exception est levee

```