Terrain

// const

**public** **int** getNombreColonnes();

// const

**public** **int** getNombreLignes();

// \pre 1 <= x <= getNombreColonnes()

// \pre 1 <= y <= getNombreLignes()

**public** BlocService getBloc(**int** x, **int** y);

// [invariants]

// \pre x >= 1

// \pre y >= 1

// \post getNombreColonne() = x

// \post getNombreLigne() = y

// \post getBloc(x, y) != NULL

// \post Bloc:getType(getBloc(x, y)) = TYPE.VIDE

// \post Bloc:getTresor(getBloc(x, y)) = TRESOR.RIEN

**public** **void** init(**int** x, **int** y);

Classe de Test – TerrainAbstractTest

PRECONDITIONS

1. Objectif 1 : init
   1. cas positif

condition initiale : aucune

opération : L1.1=init(3, 5)

oracle : pas de message d’erreur

* 1. cas négatif

condition initiale : aucune

opération : L1.1=init(-2, 5)

oracle : nombre de colonnes négatif

1. Objectif 2 : getBloc

2.1 cas positif

Condition initiale : I2.1 = init(3, 5)

Opération : L2.1 = getBloc(I2.1, 2, 2)

Oracle : pas de message d’erreur

2.2 cas négatif

Condition initiale : I2.1 = init(3,5)

Opération : getBloc(I2.2, 4, 5)

Oracle : bloc hors terrain

POST-CONDITION

1. Objectif 3 : init

Condition initiale : aucune

opération : L2.1 = init(3, 5)

oracle : invariants vrais et getNombreColonnes(L2.1) = 3 et getNombreLignes(L2.1) = 5 et getBloc(L2.1, 3, 5) != NULL et Bloc :getType(getBloc(L2.1), 3, 5) = TYPE.VIDE et Bloc :getTresor(getBloc(L2.1), 3, 5) = TRESOR.RIEN