Département
Informatique

Document de Spécification et de conception

TP-AC
Classe Simple

Binôme 3213

Tristan Pourcelot
Jordan Vincent

3IF - Groupe 2

Année scolaire 2011-2012

Institut National des Sciences Appliquées de Lyon

1 Spécifications et définitions globales de la classe

Notre classe a pour but de permettre de gérer un ensemble d'intervalles.
Un intervalle peut se définir par ses deux bornes, soit $borne_{inf}$ et $borne_{sup}$ qui sont deux entiers signés.
Les intervalles infinis ne sont pas pris en compte.
Autre cas particulier, les intervalles vides (tels que $borne_{inf} = borne_{sup}$) sont pris en compte.

La classe gère le cas particuliers des ensembles d'intervalles disjoints. Deux intervalles disjoints sont deux intervalles qui n'ont aucun point en commun.

L'ensemble des intervalles est trié, afin de faciliter les opérations de recherche et de calcul.

D'un point de vue pratique, cet ensemble est représenté par une liste chaîné de structures du type Interval.

2 Spćifications des méthodes

2.1 Constructeur de recopie : Méthode IntervalSet(IntervalSet)

Cette méthode prend en argument un pointeur vers un objet de type IntervalSet déjà construit. Il alloue l'espace mémoire nécéssaire pour stocker le contenu de l'IntervalSet passé en argument.
CONTRAT : l'objet passé en argument doit être un objet IntervalSet créé et correctement constitué. Toutefois, cet objet peut être vide.

2.2 Ajout d'un intervalle : méthode AddInterval(Interval)

Cette fonction prend en argument une structure de type `Interval` et retourne un boolén témoin du succès de l'opération. Il convient à l'utilisateur de vérifier si l'ajout a été correctement effectué.
La méthode vérifie la validité de l'intervalle (disjonction avec la collection déjà existante, puis, si ce test est correct, l'intervalle est ajouté à l'ensemble et la fonction retourne TRUE. Si ce test n'est pas concluant, l'intervalle est rejeté et la fonction retourne FALSE.

2.3 Calcul de la réunion de deux ensembles d'intervalles : Méthode Union(IntervalSet)

2.4 Intersection de deux ensembles d'intervalles : Méthode Intersection(IntervalSet)

3 Tests Unitaires et fonctionnels

3.1 Test fonctionnel N^o 1 : Fonctionnement normal de la classe

Entrées <-> Sorties attendues

3.2 Test fonctionnel N^o 2 : Fonctionnement aux limites

Entrées <-> Sorties attendues

3.3 Test Unitaire No 1 : Méthode Count

idem

3.4 Test Unitaire No 2 : Méthode Union

pareil