DESIGN OF UNIT TESTS

Stages

Name	Class	Stage
BSTStage1	BTS_Test	Create a BST
AVLStage1	AVL_Test	Create an AVL
RBTStage1	RBT_Test	Create a RBT
BSTStage2	BTS_Test	Create a BST and insert 3 players Players: {“Jacobo”, “xxx”, 4, “xxx”, 3, 3, 3, 3, 3} {“Juan”, “xxx”, 3, “xx”, 3, 3, 3, 3, 3} {“Sebastian”, “xxx”, 10, “xxx”, 3, 3, 3, 3, 3}
AVLStage2	AVL_Test	Create a AVL and insert 3 players Players: {“Jacobo”, “xxx”, 4, “xxx”, 8, 7, 3, 3, 3} {“Juan”, “xxx”, 3, “xx”, 6, 3, 3, 3, 3} {“Sebastian”, “xxx”, 10, “xxx”, 15, 9, 3, 3, 3}
RBTStage2	RBT_Test	Create a RBT and insert 3 players Players: {"Jacobo", "xxx", 4, "xxx", 8, 7, 12, 3, 3} {"Juan", "xxx", 3, "xx", 6, 3, 10, 9, 3} {"Sebastian", "xx", 10, "xxx", 15, 9, 11, 12, 3} {"Santiago", "ss", 12, "sss", 10, 12, 5, 13, 3}
FIBAStage1	FIBA_Test	Create a FIBA and insert 4 players Players: {"Jacobo", "xxx", 4, "xxx", 8, 7, 12, 3, 3} {"Juan", "xxx", 3, "xx", 6, 3, 10, 9, 3} {"Sebastian", "xx", 10, "xxx", 15, 9, 11, 12, 3} {"Santiago", "ss", 12, "sss", 10, 12, 5, 13, 3} {“Mateo”, “pp”, 23, “ppp”, 12, 4, 6, 5, 3}

Test Cases

Test Objective: Validate the correct creation of a BST

Class	Method	Stage	Input Values	Result
BST	BST(Constructor)	BSTStage1	Comparator<Player> weight = 0 root = null	BST created; constructor method works correctly

Test Objective: Validate the correct insertion of a node in BST

Class	Method	Stage	Input Values	Result
BST	insert (Node<Player>)	BSTStage1	Node<Player> Name = “Jacobo” Last Name = “xxx” Age = 4 Team = “xxx” Points = 3 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in BST; the method insert works correctly
BST	insert (Node<Player>)	BSTStage1	Node<Player> Name = “Juan” Last Name = “xxx” Age = 3 Team = “xxx” Points = 3 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in BST; the method insert works correctly
BST	insert (Node<Player>)	BSTStage1	Node<Player> Name = “Sebastian” Last Name = “xx” Age = 10 Team = “xxx” Points = 3 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in BST; the method insert works correctly

Test Objective: Validate the correct search of a node in BST

Class	Method	Stage	Input Values	Result
BST	Search (Player)	BSTStage2	Node<Player> Name = “xxx” Last Name = “xxxx” Age = 11 Team = “sdfsdf” Points = 3 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	Node not founded, NullPointerException is thrown

BST	Search (Player)	BSTStage2	Node<Player> Name = “Sebastian” Last Name = “xx” Age = 10 Team = “xxx” Points = 3 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	Node founded, return the value of the Node<Player>

Test Objective: Validate the correct remove of a node in BST				
Class	**Method**	**Stage**	**Input Values**	**Result**
BST	Delete (Player)	BSTStage2	Node<Player> Name = “Sebastian” Last Name = “xx” Age = 10 Team = “xxx” Points = 3 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	Node deleted; associations re-established. The new BST root is the node with value = 3

Test Objective: Validate the correct insertion of a node in AVL				
Class	**Method**	**Stage**	**Input Values**	**Result**
AVL	insert (Player)	AVLStage1	Node<Player> Name = “Jacobo” Last Name = “xxx” Age = 4 Team = “xxx” Points = 8 Rebounds = 7 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in AVL; Insert method, balancing and rotations work correctly.
AVL	insert (Player)	AVLStage1	Node<Player> Name = “Juan” Last Name = “xxx” Age = 3 Team = “xxx” Points = 6 Rebounds = 3 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in AVL; Insert method, balancing and rotations work correctly.

AVL	insert (Player)	AVLStage1	Node<Player> Name = “Sebastian” Last Name = “xx” Age = 10 Team = “xxx” Points = 15 Rebounds = 9 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in AVL; Insert method, balancing and rotations work correctly.
AVL	Insert (Player)	AVLStage2	Node<Player> Name = “Santiago” Last Name = “ss” Age = 10 Team = “sss” Points = 12 Rebounds = 10 Assists = 3 Robberies = 3 Blocks = 3	New node inserted in AVL; Insert method, balancing and rotations work correctly.

Test Objective: Validate the correct insertion of a node in RBT				
Class	**Method**	**Stage**	**Input Values**	**Result**
RBT	insertNode (Player)	RBTStage1	Node<Player> Name = “Jacobo” Last Name = “xxx” Age = 4 Team = “xxx” Points = 8 Rebounds = 7 Assists = 12 Robberies = 3 Blocks = 3	New node inserted in RBT; Insert method and rotations work correctly.
RBT	insertNode (Player)	RBTStage1	Node<Player> Name = “Juan” Last Name = “xxx” Age = 3 Team = “xxx” Points = 6 Rebounds = 3 Assists = 10 Robberies = 9 Blocks = 3	New node inserted in RBT; Insert method and rotations work correctly.
RBT	insertNode (Player)	RBTStage1	Node<Player> Name = “Sebastian” Last Name = “xx” Age = 10 Team = “xxx” Points = 15 Rebounds = 9 Assists = 11 Robberies = 12 Blocks = 3	New node inserted in RBT; Insert method and rotations work correctly.

RBT	insertNode (Player)	RBTStage1	Node<Player> Name = “Santiago” Last Name = “ss” Age = 12 Team = “sss” Points = 10 Rebounds = 12 Assists = 5 Robberies = 13 Blocks = 3	New node inserted in RBT; Insert method and rotations work correctly.
RBT	insertNode (Player)	RBTStage2	Node<Player> Name = “Mateo” Last Name = “pp” Age = 23 Team = “ppp” Points = 12 Rebounds = 4 Assists = 6 Robberies = 5 Blocks = 3	New node inserted in RBT; Insert method and rotations work correctly.

Test Objective: Validate the correct search for a player according to the criteria selected in FIBA.

Class	Method	Stage	Input Values	Result
FIBA	Search (Criteria, comparison, value)	FIBAStage1	Criteria = “Points” Comparison = “>” Value = 12	Player founded (Sebastian), return a List with the player in the position 0.
FIBA	Search (Criteria, comparison, value)	FIBAStage1	Criteria = “Robberies” Comparison = “=” Value = 13	Player founded (Santiago), return a List with the player in the position 0.