|  |
| --- |
| **TAD Game** |
| Game {(p1, p2, …, pn), atyp, mean} |
| {inv: n = size((p1, p2, …, pn)) 2 ≤ n ≤ 100 ( mean = matchValue(patyp) matchValue(pi)} |
| Operaciones básicas   * Game Entero → Game * setPlayers Player[] x Game → Game * tryToInsert Game x Player → Game x Player |

Operaciones

|  |
| --- |
| **Game(n)**  “Crea un game con capacidad para la cantidad de jugadores dada”  Pre: n 2 ≤ n ≤ 100  Post: game = {(p1, p2, …, pn), -1, 0} p1…pn Player p1…pn |
| **setPlayers(players,game)**  “Modifica la lista de jugadores de la partida por la que llega como parámetro”  Pre: game = {(p1, p2, …, pn), …} players = (pl1, pl2, …, plm) pl1,pl2,…,plm Player p1,p2,…,pm  Post: game = {(pl1, pl2, …, plm), …} |
| **tryToInsert(p,game)**  “Intenta insertar un jugador al juego. Se inserta en el caso de que su desviación sea menor al jugador con mayor desviación, de manera que elimina a este último del juego. Retorna el jugador que queda fuera del juego”  Pre: p Player p game = {(p1, p2, …, pn), …} p1…pn  Post: Si matchValue(patyp)  matchValue(p), game = {(p1, p2, …, patyp-1, p, patyp+1, …, pn), …} player = patyp. De lo contrario, game = {(p1, p2, …, pn), …} player = p |

|  |
| --- |
| **TAD Pickaxe** |
| Pickaxe {Name = <name>, Type = <type>, Range = <range>, Supplies = <supplies>} |
| {inv: } |
| Operaciones básicas   * Pickaxe → Pickaxe * getSupplies Pickaxe → Entero * getName Pickaxe → Texto * getRange Pickaxe → Texto * getType Pickaxe → Texto * useWeapon Pickaxe → Pickaxe * isEmpty Pickaxe → booleano |

Operaciones

|  |
| --- |
| **Pickaxe()**  “Crea una Pickaxe”  Pre:  Post: Pickaxe = {Name = “Pickaxe”, Type = ”Default”, Range = “Basic”, Supplies = 1} |
| **getSupplies(Pickaxe)**  “Retorna la munición actual del hacha”  Pre: Pickaxe = {…, Supplies = <supplies>}  Post: <supplies> |
| **getName(Pickaxe)**  “Retorna el nombre del hacha”  Pre: Pickaxe = {…, Name = <name>, …}  Post: <name> |
| **getRange(Pickaxe)**  “Retorna el rango del hacha”  Pre: Pickaxe = {…, Range = <range>, …}  Post: <range> |
| **getType(Pickaxe)**  “Retorna el tipo del hacha”  Pre: Pickaxe = {…, Type = <type>, …}  Post: <type> |
| **useWeapon(Pickaxe)**  “Usa el hacha”  Pre: Pickaxe = {Name = <name>, Type = <type>, Range = <range>, Supplies = <supplies>}  Post: Pickaxe = {Name = <name>, Type = <type>, Range = <range>, Supplies = <supplies>} |
| **isEmpty(Pickaxe)**  “Verifica si el arma está vacía”  Pre: Pickaxe = {Name = <name>, Type = <type>, Range = <range>, Supplies = <supplies>}  Post: Pickaxe = True si <supplies> , False de lo contrario |

|  |
| --- |
| **TAD StackWeapon** |
| StackWeapon = {(a1, a2, …, an), top} |
| {inv: |
| Operaciones básicas   * StackWeapon - → StackWeapon * push StackWeapon x Weapon → StackWeapon * top StackWeapon → Weapon Pickaxe * pop StackWeapon → StackWeapon x Weapon * size StackWeapon→ Entero * useWeapon StackWeapon → StackWeapon |

Operaciones

|  |
| --- |
| **StackWeapon()**  “Crea una StackWeapon”  Pre:  Post: StackWeapon = {a1} |
| **push(s,w)**  “Añade un arma al StackWeapon”  Pre: s = {a1, a2, …, an}, w Weapon  Post: s = {a1, a2, …, an, w} |
| **top(s)**  “Retorna el elemento en el top del StackWeapon”  Pre: s = {a1, a2, …, an} s = {a1}  Post: Weapon an si s = {a1, a2, …, an}, Pickaxe a1 si s = {a1} |
| **pop(s)**  “Elimina la última arma añadida al StackWeapon y la retorna”  Pre: s {a1}, es decir, s = {a1, a2, …, an}  Post: s = {a1, a2, …, an-1}, Weapon an |
| **size(s)**  “Retorna la cantidad de armas en el StackWeapon”  Pre: s = {a1, a2, …, an}  Post: n |
| **useWeapon(s)**  “Usa la última arma añadida al StackWeapon, y la elimina si se queda sin munición y no es un hacha”  Pre: s = {a1, a2, …, an} s = {a1}  Post: s = {a1} si s = {a1}. De lo contrario, si getSupplies(an) = 1, s = {a1, a2, …, an-1}; de lo contrario s = {a1, a2, …, an={…, Supplies = <supplies>-1}} |

**DISEÑO DE CASOS DE PRUEBAS UNITARIAS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba 1: Añade elementos correctamente a la Stack | | | | |
| Clase | Método | Escenario | Entrada | Resultado |
| StackWeapon | Push(Weapon w) | Se ha creado un StackWeapon | W tiene  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30 | El elemento del top es un Rifle común llamado M16 con 30 municiones.  La Stack es de tamaño 2. |
| StackWeapon | Push(Weapon w) | Se ha creado un StackWeapon que tiene 2 armas, el hacha que viene por defecto y la siguiente arma:  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30 | W tiene  Nombre= “SMG”  Range=”Rare”  Type=”SMG”  Supplies=30 | El elemento del top es un Rifle raro llamado SMG con 30 municiones.  La Stack es de tamaño 3. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba 2: Elimina elementos satisfactoriamente de la Stack | | | | |
| Clase | Método | Escenario | Entrada | Resultado |
| StackWeapon | +Weapon Pop() | Se ha creado un StackWeapon | Ninguna | Lanza una excepción porque no se puede eliminar el hacha de la Stack. |
| StackWeapon | +Weapon Pop() | Se ha creado un StackWeapon y se añadió la siguiente arma:  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30 | Ninguna | El elemento del top de la Stack es el hacha y la Stack es de tamaño 1. |
| StackWeapon | +Weapon Pop() | Se ha creado un StackWeapon y se añadió la siguiente arma:  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30  Y luego la siguiente arma:  Nombre= “SMG”  Range=”Rare”  Type=”SMG”  Supplies=30 | Ninguna | El elemento del top de la Stack es el M16 y la Stack es de tamaño 2. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba 2: El método use weapon decrementa las municiones del arma y la elimina del Stack cuando se quede sin municiones. | | | | |
| Clase | Método | Escenario | Entrada | Resultado |
| StackWeapon | + void useWeapon() | Se ha creado una StackWeapon | Ninguna | El elemento del top es el hacha por lo tanto sus municiones siguen siendo uno y la Stack es de tamaño 1 |
| StackWeapon | + void useWeapon() | Se ha creado una StackWeapon.  Se le ha añadido la siguiente arma:  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30  Y se usa 29 veces | Ninguna | El elemento del top es el M16, al que le queda una munición. |
| StackWeapon | + void useWeapon() | Se ha creado una StackWeapon.  Se le ha añadido la siguiente arma:  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30  Y se usa 30 veces | Ninguna | El elemento del top es el hacha, la Stack es de tamaño 1. |
| StackWeapon | + void useWeapon() | Se ha creado una StackWeapon.  Se le ha añadido la siguiente arma:  Nombre= “M16”  Range=”Common”  Type=”Rifle”  Supplies=30  Y luego la siguiente arma:  Nombre= “SMG”  Range=”Rare”  Type=”SMG”  Supplies=30  Y se usa 29 veces el arma del top. | Ninguna | El elemento del top es el SMG y su atributo supplies=1 |
| StackWeapon | + void useWeapon() | El mismo que el anterior, pero el arma del top se usa 30 veces | Ninguna | El elemento del top es el M16 y su elemento supplies es 30 |
| StackWeapon | + void useWeapon() | El mismo que el anterior, pero el arma del top se usa 29 veces  Y Luego se le añade un arma  Nombre= “Rocket Launcher”  Range=”Rare”  Type=”Granade Launcher”  Supplies=1  Y se usa el arma 2 veces. | Ninguna | El elemento del top es el M16 y su elemento supplies es 30 |