Programación II - Trabajo Práctico Integrador

2do. Cuatrimestre 2021

SEGUNDA PARTE

Tad Persona clase abstracta extiende de Object

IREP: nombe, dni, esPresidente, asistioAVotar no pueden estar vacíos o ser nulos.

Atributos:

String nombre;

int dni;

boolean esPresidente;

boolean asistioAVotar;

Operaciones:

Presidente(int dni, String nombre) // Constructor.

Getters and Setters

Boolean equals():

Tad Votante extiende de persona

IREP: Edad, tieneEnfermedad, esTrabajador no pueden estar vacíos o nulos, turnoAsignado recibe mesa y horario y estos deben cumplir que son validos. Mesa de 1 a 4 y hora de 8 a 17 hs.

Atributos:

Int edad;

boolean tieneEnfermedad;

boolean esTrabajador;

Tupla<Integer, Integer> turnoAsignado;

Operaciones:

Votante(int dni, string nombre, int edad, boolean tieneEnfermedad, boolean esTrabajador); // Constructor

Getter and Setters

String toString()

Tad Presidente extiende de Persona

IREP: El presidente no puede ser nulo tiene que esta en una mesa.

Atributos:

Operaciones:

Presidente(int dni, string nombre) // Constructor

String toString() // Del presidente

Tad Tupla <X, Y>

IREP: x e y no pueden estar vacíos.

Atributos:

final X x;

final Y y;

Operaciones:

Tupla(X x, Y y) // constructor.

Getters and setters.

String toString() // De x = Mesa y de y = hora.

Tad Mesa clase abstracta

IREP: el numero de mesa es único, presidente no puede estar nulo, franjasHorarias tienen que estar entre 8 a 17 hs, maximoVotantesPorFranjaHoraria tiene que ser mayor a cero y en sus clases hijas según su cantidad de cupos por hora, tipoDeMesa tiene que ser correcto, y en el Map la franjaHoraria tiene que ser valida.

Atributos:

int numero;

Pesidente presidente. // Tad Presidente

Int[] franjasHorarias.

Int maximoVotantesPorFranjaHoraria.

String tipoDeMesa.

HashMap<Integer, Integer> DNIsAsignadosPorFranjaHoraria; // (fraja-dni)

Operaciones:

Mesa(Presidente presidente, int numero, string tipoMesa, int maximoVotantesPorFranjaHoraria) // constructor.

Abstract boolean aceptaVotante(Votante votante) // retorna true si el votante cumple la condición de la mesa.

Abstract esDeTipo(string tipoMesa) // retorna true si el tipo de mesa es valido.

======= Metodos de Mesa ========

Void registrarVotante(int horario, int dni) // Registra horario ,dni y cuenta los votantes asignados.

Boolean tieneAsignado() // comprueba si esta asignado el dni.

In obtenerHorarioDisponible() // obtiene el horario disponible.

Getter and Setters.

boolean equals(Object obj).

String toString()

Tad MesaEnfPreex extiende de Mesa

IREP: el tipo de mesa tiene que ser par Enf\_preex y el cupo mayor a cero y menor igual a 20 por franja horaria.

Operaciones:

MesaEnf\_Preex(Votante presidenteMesa) // constructor.

boolean aceptaVotante(Votante votante) // sobre-escritura.

Boolean esDeTipo(string tipoMesa) // sobre-escritura.

Tad MesaGeneral extiende de Mesa

IREP: el tipo de mesa tiene que ser para General y el cupo mayor a cero y menor igual a 30 por franja horaria.

Operaciones:

MesaGeneral(Presidente presidente, int numero) // constructor.

boolean aceptaVotante(Votante votante) // sobre-escritura.

Boolean esDeTipo(string tipoMesa) // sobre-escritura.

Tad MesaMayores extiende de Mesa

IREP: el tipo de mesa tiene que ser para Mayor65 y el cupo mayor a cero y menor igual a 10 por franja horaria.

Operaciones:

MesaMayores(Presidente presidente, int numero) // constructor.

boolean aceptaVotante(Votante votante) // sobre-escritura.

Boolean esDeTipo(string tipoMesa) // sobre-escritura.

Tad MesaTrabajadores extiende de Mesa

IREP: el tipo de mesa tiene que ser para Trabajadores y el cupo mayor a cero por franja horaria.

Operaciones:

MesaTrabajadores(Presidente presidente, int numero) // constructor.

boolean aceptaVotante(Votante votante) // sobre-escritura.

Boolean esDeTipo(string tipoMesa) // sobre-escritura.

Int obtenerHorarioDisponible() // sobre-escritura

Tad SistemaVotacion

IREP: Al agregar un votante su turno tiene que ser valido. Si las clases que componen el sistema cumplen su irep tiene que funcionar.

Cuando agrego una mesa en mesasPorNro Si la clase mesa cumple su irep debe funcionar, en conclusión cuando agrego un votante o una mesa si las clases secundarias cumplen su invariante de representación en SistemaDeTurnos los valores que llegan son correctos y al agregar votantes y mesa serian validos.

Atributos:

String nombreSistema;

Diccionario<Integer, Votante> votantesPorDNI => clave dni, value un votante.

Diccionario<Integer, Mesa> mesasPorNro => clave numero de mesa, value Mesa

Operaciones:

SistemaVotacion(String nombreSistema) // constructor.

void registrarVotante(int dni, String nombre, int edad, boolean tieneEnfermedad, boolean esTrabajador) // guarda un votante en votantesPorDNI.

boolean verificarPresidentDeMesa(int dni) // Verifica si el presidente tiene turno o esta registrado.

int agregarMesa(String tipoMesa, int dni) // Agrega las mesas según el tipo de mesa valida y devuelve un numero de mesa. También registra un votante y asigna turno al presidente.

Tupla<Integer, Integer> asignarTurno(int dni) // devuelve el turno , mesa y franja horaria.

int asignarTurnos() // Asigna un turno sin ninguna condición y retorna la cantidad de turnos.

boolean votar(int dni) // vota y devuelve true.

int votantesConTurno(String tipoMesa) // devuelve la cantidad de turnos asignados a un tipo de mesa determinado.

Tupla<Integer, Integer> consultaTurno(int dni) // Devuelve Mesa y franja horaria o null.

Map<Integer, List<Integer>> asignadosAMesa(int numMesa) // devuelve una Map cuya clave es la franja horaria y el valor es una lista con los DNI de los votantes asignados a esa franja.

List<Tupla<String, Integer>> sinTurnoSegunTipoMesa() // Devuelve una Lista de Tuplas donde se vincula el tipo de mesa con la cantidad de votantes sin turno que esperan ser asignados a ese tipo de mesa.

String toString() //

======== Metodos privados del sistema ========

Tupla<Integer, Integer> obtenerTurno(Votante votante) // busca un horario disponible y retorna el turno.

Votante obtenerVotante(int dni) // si esta registrado el votante lo devuelve.

***Comisión: 04.***

***Abel Fernando Acevedo: 2018 / 28881785.***