«enumeration» cPista eEstado «typedef» ushort (unsigned short) desconocido -largo: const short cTorreControl despegando -ancho: const short -LuO: bool enVuelo +cTorreControl(short, short, float, float) aterrizando -c:TorreControl() +cPista(float, float) +autorizarDespeque(cAvion): void ~cPista() +autorizarAterrizaje(cAvion): void +operator==(cAvion): bool -pista +autorizarEstacionamiento(cAvion): void +operator=(cAvion): void +imprimirDetalles(): void -conversor(cAvion): const float +verificarHorario(cAvion): avion -getTimepoldeal(cAvion): const float +operator!=(cAvion): bool +getPosicionFinal(cAvion): float +agregarAvionesListados(cAvion): void +getLuo(): const bool +switchLuo(): void -hangar cl ista #lista: T #cantTotal: short #cantActual: ushort cHangar #checkEliminar: bool cListaAvion +cLista(short, bool) ~cLista() +cHangar(short) 1 -cListaAvion(short, bool) ~cHangar() ~cListaAvion() -ordenar(): void +getCantActual(): short +getCantActual(): const short -noRepetido(T): bool +almacenar(cAvion): bool +operator==(cAvion): bool -resize(short): void +despachar(cAvion): cAvion +agregar(T): void +operatorfl(short): cAvion +eliminar(T): void +operator+(T): void +operator-(T): T +operator[](short): T +operator++(): T +quitar(T): T c∆vion cModelo #largoAvion: short #anchoAvion: short -nombre: const string cCombustible -limiteCarga: float #ID: const string (id) -aceleracion: float +cCombustible(short, short) #helice: short -limitePasajeros: short #pasajerosActual: short ~cCombustible() +cModelo(string, float, float, short, short, short) -getDuracion(): const cFecha #velocidad: short -combustible ~cModelo() +fallaCombustible(short, short): void #estado: eEstado +getLimitePasajeros(): const short #destino: string +getLimiteCarga(): const float #horaSalida: time_t +getCombustible(): const cCombustible modelo: cModelo +getAceleracion(): const void +cAvion(string, float, float, short, string, short, eEstado, short, time_t, short) ~cAvion() -duracion +getPasajerosActual(): const short +getLargoAvion(): const float +getAnchoAvion(); const float eEC +getEstado(): const eEstado +getID(): const string cBiplano +getModelo(): const cModelo cFecha +getCombustible(): const cCombustible #velMaxDesBip: ushort +horarioEsperado: struct tm +getVelocidad(); const short #velMaxAteBip: ushort +getAceleracion(): const float +cFecha(short, short) #inclinacion: float +setDestino(string): void + ~cFecha() +cBiplano(string, float, float, string, short, eEstado, short, time_t, float) +setPasajerosActuales(short): void +getHorarioEsperado(): const time_t + ~cBiplano() +setVelocidad(short): void +getHorarioActual(): time_t +despegar(): void +aterrizar(): void +setEstado(eEstado): void +despegar(): void +estacionar(): void +aterrizar(): void +to_string(): const string +estacionar(): void +operator>(cModelo): void +to_string(): const string +enumToString(eEstado): const string +operator>(cModelo): bool +operator<(cCombustible): bool +getHoraSalida(): const time_t +getDestino(): const string +inputCleaning(): void cCESSNA #velMaxDesCes: ushort #velMaxAteCes: ushort #alerones: bool #controlAutomatico: bool #cargaActual: float +cCESSNA(string, float, float, string, short, eEstado, short, time_t, bool, float, bool) + ~cCESSNA() +getCargaActual(): const flloat +setCargaActual(float): void +despegar(): void +aterrizar(): void +estacionar(): void +operator>(cModelo): bool +to_string(): const string

Model::Main