```
/*
 *  This file is part of DynamoPlus.
 *
 *  Copyright (c) 2014-2015 Technische Universitaet Muenchen,
 *  Chair of Computational Modeling and Simulation (https://www.cms.bgu.tum.de/)
 *  LEONHARD OBERMEYER CENTER (http://www.loc.tum.de)
 *
 *  Developed by Fabian Ritter (contact: mailto:mail@redinkinc.de) and Florian
   Englberger
 *
 *  DynamoPlus is free software: you can redistribute it and/or modify
 *  it under the terms of the GNU General Public License as published by
 *  the Free Software Foundation, either version 3 of the License, or
 *  (at your option) any later version.
 *
 *  DynamoPlus is distributed in the hope that it will be useful,
 *  but WITHOUT ANY WARRANTY; without even the implied warranty of
 *  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *  GNU General Public License for more details.
 *
 *  You should have received a copy of the GNU General Public License
 *  along with DynamoPlus. If not, see <http://www.gnu.org/licenses/>.
 */


using Autodesk.DesignScript.Runtime;
using DynamoPlus.Geometry;

namespace DynamoPlus.System
{
    /// <summary>
    ///
    /// </summary>
    public class IdealLoadsAirSystem
    {
        private string Name { get; set; }
        private string InletNode { get; set; }
        private string OutdoorInletNode { get; set; }
        private static double BasicAirFlowPerZone { get; set; }
        private static double MaxFlowRateHeat { get; set; }
        private static double MaxFlowRateCool { get; set; }

        /// <summary>
        ///
        /// </summary>
        /// <param name="zone"></param>
        /// <param name="basicAirFlowPerZone"></param>
        /// <param name="maxFlowRateHeat"></param>
        /// <param name="maxFlowRateCool"></param>
        public IdealLoadsAirSystem(Zone zone, double basicAirFlowPerZone, double
          maxFlowRateHeat, double maxFlowRateCool)
        {
            Name = zone.Name + " Purchased Air";
            InletNode = zone.Name + " Supply Inlet";
            BasicAirFlowPerZone = basicAirFlowPerZone;
            MaxFlowRateHeat = maxFlowRateHeat;
```

```
            MaxFlowRateCool = maxFlowRateCool;
        }

        /// <summary>
        /// Returns a string that represents the current object.
        /// </summary>
        /// <returns>
        /// A string that represents the current object.
        /// </returns>
        public override string ToString()
        {
            return Name;
        }

        /// <summary>
        ///
        /// </summary>
        /// <returns></returns>
        [IsVisibleInDynamoLibrary(false)]
        public string Write()
        {
            string temp = "\nZoneHVAC:IdealLoadsAirSystem,\n";
            temp += this.Name + ",  !- Name\n";
            temp += ",  !- Availability Schedule Name\n";
            temp += InletNode + ",   !- Zone Supply Air Node Name\n";
            temp += ", !- Zone Exhaust Air Node Name\n";
            temp += "50,                     !- Maximum Heating Supply Air
              Temperature {C}\n";
            temp += "13,                     !- Minimum Cooling Supply Air
              Temperature {C}\n";
            temp += "0.015,                  !- Maximum Heating Supply Air
              Humidity Ratio {kgWater/kgDryAir}\n";
            temp += "0.01,                   !- Minimum Cooling Supply Air
              Humidity Ratio {kgWater/kgDryAir}\n";
            temp += "LimitFlowRate,                 !- Heating Limit\n";
            temp += "" + MaxFlowRateHeat + ",                       !- Maximum
              Heating Air Flow Rate {m3/s}\n";
            temp += ",                       !- Maximum Sensible Heating Capacity
              {W}\n";
            temp += "LimitFlowRate,                 !- Cooling Limit\n";
            temp += "" + MaxFlowRateCool + ",                       !- Maximum
              Cooling Air Flow Rate {m3/s}\n";
            temp += ",                       !- Maximum Total Cooling Capacity {W}
              \n";
            temp += ",                       !- Heating Availability Schedule Name
              \n";
            temp += ",                       !- Cooling Availability Schedule Name
              \n";
            temp += "None,  !- Dehumidification Control Type\n";
            temp += ",                       !- Cooling Sensible Heat Ratio
              {dimensionless}\n";
            temp += "None,  !- Humidification Control Type\n";
            temp += "DesignSpecification:OutdoorAir,                        !-
              Design Specification Outdoor Air Object Name\n";
            temp += OutdoorInletNode + ",                       !- Outdoor Air
              Inlet Node Name\n";
```

```
            temp += "OccupancySchedule,                          !- Demand Controlled
               Ventilation Type\n";
            temp += "DifferentialDryBulb,                          !- Outdoor Air
              Economizer Type\n";
            temp += "Sensible,                          !- Heat Recovery Type\n";
            temp += "0.8,                           !- Sensible Heat Recovery
              Effectiveness {dimensionless}\n";
            temp += ",                              !- Latent Heat Recovery Effectiveness
               {dimensionless}\n";
            temp += ";                              !- Design Specification ZoneHVAC
              Sizing Object Name\n";

            temp += "DesignSpecification:OutdoorAir,\n";
            temp += "Specifikation af udeluft,          !- Name\n";
            temp += "Flow/Zone,                         !- Outdoor Air Method\n";
            temp += ",                                  !- Outdoor Air Flow per
              Person\n";
            temp += ",                                  !- Outdoor Air Flow per Zone
               Floor Area\n";
            temp += " " + BasicAirFlowPerZone + ",      !- Outdoor Air Flow per Zone
              \n";
            temp += ",                                  !- Outdoor Air Flow Air
              Change per Hour\n";
            temp += "ON;                                !- Outdoor Air Flow Rate
              Fraction Schedule Name\n";

            return temp;
        }
    }
}
```