```
package FallschirmBibliothek9
  model Fallschirmsprung9
    //Konstanten
    import Modelica.Constants.G;
    //Newton'sche Gravitationskonstante
    //Parameter
    //Variablen
    Modelica.Units.SI.Force F_friction;
    //Resultierente Kraft durch Widerstand bei Bewegung durch
Umgebungsgas!
    Modelica.Units.SI.Force F_g;
    //resultierende Graft durch die Gravitation des Planeten!
    //Klassen
    FallschirmBibliothek9.Person person(name = "Gary", mass = 85,
position(start = flugzeug.height), area_front = 1.0, cW = 0.78);
    FallschirmBibliothek9.Flugzeug flugzeug(height = 2000);
    FallschirmBibliothek9.Umgebung luft(h = person.position);
    FallschirmBibliothek9.Fallschirm fallschirm(area_open = 5.0,
area_closed = person.area_front, cW_closed = person.cW);
    FallschirmBibliothek9.Planet erde(name = "Erde", mass = 5.972e24,
radius = 6371e3);
    //Modelica-Block
  equation
    if person.position > 500 then
      F_friction = 0.5 * person.cW * person.area_front * luft.rho *
person.velocity ^ 2;
      fallschirm.reisleine_gezogen = false;
    else
      F_friction = 0.5 * fallschirm.cW * fallschirm.area * luft.rho *
person.velocity ^ 2;
      fallschirm.reisleine_gezogen = true;
    end if;
    person.acceleration * person.mass = (-F_g) + F_friction;
    F_g = G * (erde.mass * person.mass / (erde.radius +
person.position) ^ 2);
  algorithm
    when person.position < 0 then
      terminate("The person landed!");
    end when;
    annotation(
      experiment(StartTime = 0, StopTime = 150, Tolerance = 1e-06,
Interval = 0.01));
  end Fallschirmsprung9;

  class Person
    //Konstanten
    import Modelica.Constants.inf;
    //Parameter
    parameter Modelica.Units.SI.Mass mass = 80 "Masse der Person";
    parameter String name = "Max MustermPerson" "Name der Person";
```

```
    parameter Modelica.Units.SI.Area area_front = 1 "prokizierte Fläche
der Person bei Frontalansicht";
    parameter Real cW = 0.78 "strömungswiderstand einer Person bei
frontaler Angriffsfäche";
    //Variablen
    Modelica.Units.SI.Position position;
    //(vertikale) Position (des Schwerpunktes) der Person über dem
Erdboden
    Modelica.Units.SI.Velocity velocity;
    //(vertikale) Geschwindigkeit (des Schwerpunktes) der Person
gegenüber dem Erdboden
    Modelica.Units.SI.Acceleration acceleration;
    //(vertikale) Beschleunigung (des Schwerpunktes) der Person über
dem Erdboden
    //Klassen
    //Modelica-Blöcke
  equation
    der(position) = velocity;
    der(velocity) = acceleration;
  end Person;

  class Flugzeug
    //Konstanten
    //Parameter
    parameter Modelica.Units.SI.Height height = 2000;
    //Variablen
    //Klassen
    //Modelica-Blöcke
  end Flugzeug;

  class Umgebung
    constant Real R = Modelica.Constants.R;
    //Universelle Gaskonstante
    constant Modelica.Units.SI.AbsolutePressure p_0 = 101325;
    //Luftdruck auf Meereshöhe
    parameter Modelica.Units.SI.MolarMass M = 0.028949 "Molare Masse
des Gases";
    parameter Modelica.Units.SI.Height H_0 = 7800 "Skalenhöhe im
erdnahen Bereich";
    Modelica.Units.SI.Density rho "Dichte des Umgebungsgases";
    Modelica.Units.SI.Temperature T "Temperatur der Umgebungsluft";
    Modelica.Units.SI.AbsolutePressure p "Luftdruck der Umgebungsluft";
    Modelica.Units.SI.Height h "aktuelle Höhe über dem Meeresspiegel";
  equation
    rho = p * M / (T * R);
    T = 294.15 - 7.5 / 1000 * h;
    p = p_0 * exp(-h / H_0);
  end Umgebung;

  class Fallschirm
```

```
    //Konstanten
    import Modelica.Math.exp;
    import Modelica.Math.log;
    //Parameter
    parameter Modelica.Units.SI.Area area_open = 5 "Projektionsfläche
eines geöffneten Fallschirms";
    parameter Modelica.Units.SI.Area area_closed = 0
"Projektionsflächte eines geschlossenen Fallschirms";
    parameter Modelica.Units.SI.Time opening_duration = 3 "Dauer die
der Fallschirm zum Öffnen benötigten";
    parameter Real cW_closed "Strömungswiderstandskoeffizient eines
geschlossenen Fallschirms";
    parameter Real cW_open = 1.33 "Strömungswiderstandskoeffizient
eines geöfneten Fallschirms";
    //Variablen
    Modelica.Units.SI.Area area "Projektionsfläche des Fallschirms zum
Zeitpunkt t";
    Real cW "Strömungswiderstand des Fallschirms zum Zeitpunkt t";
    Boolean reisleine_gezogen "gibt an ob die Reisleine des Fallschirms
gezogen wurde";
    Modelica.Units.SI.Time opening_time "Dauer die der Fallschirm zum
Öffnen benötigten";
    //Klassen
    //Modelica-Blöcke
    Modelica.Blocks.Continuous.CriticalDamping damper_area(f = 20, n =
3, y_start = area_closed);
    Modelica.Blocks.Continuous.CriticalDamping damper_cW(f = 20, n = 3,
y_start = cW_closed);
  equation
    when reisleine_gezogen then
      opening_time = time;
    end when;
    if not reisleine_gezogen then
      damper_cW.u = cW_closed;
      damper_area.u = area_closed;
    else
      damper_cW.u = cW_open;
      damper_area.u = (area_open - area_closed) / (1 + exp(-(10 /
opening_duration * (time - opening_time) - log((area_open -
area_closed) ^ 2 / 2) - 1 / (area_open - area_closed)))) + area_closed;
    end if;
    area = damper_area.y;
    cW = damper_cW.y;
  end Fallschirm;

  class Planet
    //Konstanten
    //Parameter
    parameter String name = "Erde" "Name des Planeten";
    parameter Modelica.Units.SI.Mass mass "Masse des Planeten";
```

```
    parameter Modelica.Units.SI.Length radius "Radius des Planeten";
    //Variablen
    //Klassen
    //Modelica-Blöcke
  end Planet;
  annotation(
    uses(Modelica(version = "4.0.0")));
end FallschirmBibliothek9;
```