```
pragma SPARK_Mode (On);

with SPARK.Text_IO;use  SPARK.Text_IO;


package fuel is

   -- Constants representing fuel capacities for different regions
   Ocean : constant Integer := 5000;
   Europe : constant Integer := 3500;
   Asia : constant Integer := 2000;

   -- Critical fuel level threshold
   FRC : constant Integer := 1000;

   -- temp_ range = fuel_R
   --type Fuel_R is new Integer range 0.. Asia; -- correct to get
maxximum is 2000
   -- type Fuel_R2 is new Integer range 0.. Europe; --maxximumis 3500


   -- Define fuel range types for different flight segments
   type Fuel_R is new Integer range 0.. Asia;
   type Fuel_R2 is new Integer range 0.. Europe;
   type Fuel_R3 is new Integer range 0.. Ocean;

   -- Enumeration for fuel system status
   type Status_Fuel_T is (ON,  OFF);


   -- Structure to hold fuel status for different flights
   type SST  is
      record
         FM : Fuel_R;
         FM2: Fuel_R2;
         FM3 : Fuel_R3;
         SFS : Status_Fuel_T;
      end record;

   -- Fuel system status instance
   FFS : SST;

   -- Flight procedure, taking inputs and updating fuel status
   procedure RF with
     Global => (In_Out => (Standard_Input, Standard_Output,FFS)),
     Depends => (Standard_Output => (Standard_Output,Standard_Input),
                 Standard_Input  => Standard_Input,
                 FFS  => (FFS, Standard_Input));

   -- Function to convert fuel system status to string
   function SFSTS (SFS   : Status_Fuel_T) return String;




   -- Print system status procedure
   procedure PS with
     Global => (In_Out => Standard_Output,
                Input  =>  FFS),
     Depends => (Standard_Output => (Standard_Output, FFS));


   -- Functions to check if fuel levels are above thresholds
   function ISS (Status : SST) return Boolean is ---
     (if Integer(Status.FM)>  FRC
      then Status.SFS = ON
      else Status.SFS= OFF);

   function ISS2 (Status : SST) return Boolean is
     (if Integer(Status.FM2) >  Integer(Status.FM)
      then Status.SFS = ON
      else Status.SFS= OFF);

   function ISS3 (Status : SST) return Boolean is
     (if Integer(Status.FM3) >  Integer(Status.FM2)
      then Status.SFS = ON
      else Status.SFS= OFF);



   -- Monitor fuel status and update system status
   procedure MFS  with
     Global  => (In_Out  =>  FFS),
     Depends => ( FFS =>  FFS),
     Post    => ISS( FFS) AND ISS2(FFS) AND ISS3 (FFS);

   -- Initialize the system with default values
   procedure Init with
     Global => (Output => (Standard_Output,Standard_Input, FFS)),
     Depends => ((Standard_Output,Standard_Input, FFS) => null),
     Post    => ISS( FFS) AND ISS2(FFS) AND ISS3 (FFS);




end  fuel;
```