Inheritance and polymorphism

The following classes represent various kinds of pilots, grouped by the type of device that they can fly ("paraglider" = "Gleitschirm", "hang glider" = "Hängegleiter"). Figure ?? shows the class hierarchy. The listing below shows the source code of the classes.

![Class diagram for class PILOT and its descendants](image)

Figure 1: Class diagram for class PILOT and its descendants

Given the variable declarations

```java
p : PILOT
up : UNMOTORIZED_PILOT
mp: MOTORIZED_PILOT
ap : AIRCRAFT_PILOT
pp : PARAGLIDER_PILOT
hgp : HANG_GLIDER_PILOT
```

for each of the code fragments below:

- state if it compiles or not
- if it compiles, state what message (if any) will be printed to the console when it is executed
- if the code fragment does not compile, explain why it is invalid.

No other explanations are necessary.

```java
create pp
pp.fly
```

```java
create {PARAGLIDER_PILOT} p.make ("John")
```

```java
create up.make ("Ben")
up.fly
```
create {PARAGLIDER_PILOT} mp.make ("Steve")
mp.fly

create {AIRCRAFT_PILOT} mp.make ("Jimmy")
mp.fly

create {PARAGLIDER_PILOT} up.make ("Danny")
print (up.paraglider_type)

create hgp.make ("Bobby")
hgp.fly
deferred class PILOT

feature -- Initialization
make (a_name: STRING) is
  --- Create a new pilot whose name is 'a_name'.
  require
    a_name_valid: a_name /= Void and then not a_name.is_empty
do
  name := a_name
  print ("New pilot\%N")
ensure
  name_set: name = a_name
end

feature -- Basic operations
fly is
defered
end

feature -- State
name: STRING
  --- Name of the pilot

invARIANT
  valid_name: name /= Void and then not name.is_empty
end

defered class MOTORIZED_PILOT
inherit PILOT
end

defered class UNMOTORIZED_PILOT
inherit PILOT
end

class AIRCRAFT_PILOT
inherit
  MOTORIZED_PILOT
  rename fly as fly_aircraft
end

create
make

feature -- Basic operations
fly_aircraft is
do
  print ("I'm flying my aircraft\%N")
end
end
class PARAGLIDER_PILOT

inherit UNMOTORIZED_PILOT

redefine make

create

make

feature -- Initialization

make (a_name: STRING) is
  -- Create a new paraglider pilot called 'a_name' and with the default paraglider type.
  do
    Precursor (a_name)
    paraglider_type := "default"
    print ("New paraglider pilot\n"
  end

feature -- Basic operations

fly is
  do
    print ("I'm flying my paraglider\n"
  end

feature -- Paraglider

paraglider_type : STRING
  -- Type of paraglider

end

class HANG_Glider_PILOT

inherit UNMOTORIZED_PILOT

feature -- Basic operations

fly is
  do
    print ("I'm flying my hang glider\n"
  end

end