Inheritance and polymorphism — Solution

```plaintext
create pp
pp.fly
```

**Solution:** The code does not compile, because the creation instruction should include a call to the creation procedure make of class PARAGLIDER_PILOT.

```plaintext
create {PARAGLIDER.PILOT} p.make ("John")
```

**Solution:** The code compiles and it prints

```
New pilot
New paraglider pilot
```

```plaintext
create up.make ("Ben")
up.fly
```

**Solution:** The code does not compile because up is declared of a deferred type.

```plaintext
create {PARAGLIDER.PILOT} mp.make ("Steve")
mp.fly
```

**Solution:** The code does not compile because PARAGLIDER_PILOT does not conform to MOTORIZED_PILOT.

```plaintext
create ap.make ("Sammy")
ap.fly
```

**Solution:** The code does not compile because there is no feature called fly in class AIRCRAFT_PILOT.

```plaintext
create {AIRCRAFT.PILOT} mp.make ("Jimmy")
mp.fly
```

**Solution:** The code compiles and displays

```
New pilot
I'm flying my aircraft
```

```plaintext
create {PARAGLIDER.PILOT} up.make ("Danny")
print (up.paraglider_type)
```

**Solution:** The code does not compile because there is no feature called paraglider_type in class UNMOTORIZED_PILOT.

```plaintext
create hgp.make ("Bobby")
hgp.fly
```

**Solution:** The code does not compile because feature make is not declared as a creation procedure in class HANG_GLIDER_PILOT.