Programming 2
Exercises 10

Kenny Štorgel

UP FAMNIT

Koper, May 2020

Inherit

Inherit is a keyword used to declare that an object inherits variables, methods, etc. from another object. We need to specify which object and apply the corresponding parameters.

```
class class_name class_parameters =
   object
   inherit class2_name class2_parameters

   end;;
```

We can also name the object to access its methods. We can use any name, but the name **super** is most common established.

```
inherit class2_name class2_parameters as super
```

Overriding methods

When we **override** a method, we can simply rewrite it, but we need to be carefull that the new method has the same type (the same arguments types, the same return type).

Or we can rewrite it by using the method from superclass and adding something more. We call the method of the superclass the same way as calling the methods of the class itself, only this time with the name of the superclass.

Multiple inheritance

We can **inherit from multiple objects** at the same time. When two methods have the same name, then only the last method (in terms of declaration) is kept.

We can however access all the methods of the superclasses.

Abstract classes

Abstract classes cannot be instantiated. In abstract classes, we can declare methods without implementation by using only the type signature and the keyword **virtual**.

```
class virtual abstract_class_example =
   object (self)
      method virtual get_info : unit -> string
      method virtual get_info2 : string
      method print_hello = Printf.printf "Hello world!\n"
   end;;
```

Inheriting abstract class

When we inherit from the abstract class, we need to define all the virtual methods according to the type signatures.

```
class example_subclass =
   object (self)
   inherit abstract_class_example
      method get_info () = "Example"
      method get_info2 = "Example2"
   end;;
```