This is a brief document which explains certain aspects of the implementation document “paramstyle8.ml”.

Used a list data structure to hold the various passing styles and used a record to represent specific passing styles which consist of passing style name and factors. Also used a record to hold the various factors of the passing styles.

Determining or defining a passing style depends on the values of these factors that are assumed to effect parameter passing style. When the code is first run, values of these factors (Entity type, Context, Evaluation and typing) are demanded. If these values corresponds to values of known passing styles then the passing style is return alongside it properties (name and factors) and the effects induced by these values. If values do not corresponds to values for known passing style then a new passing style is created with an input name for the passing style and is added to the list of passing styles. This is illustrated below.

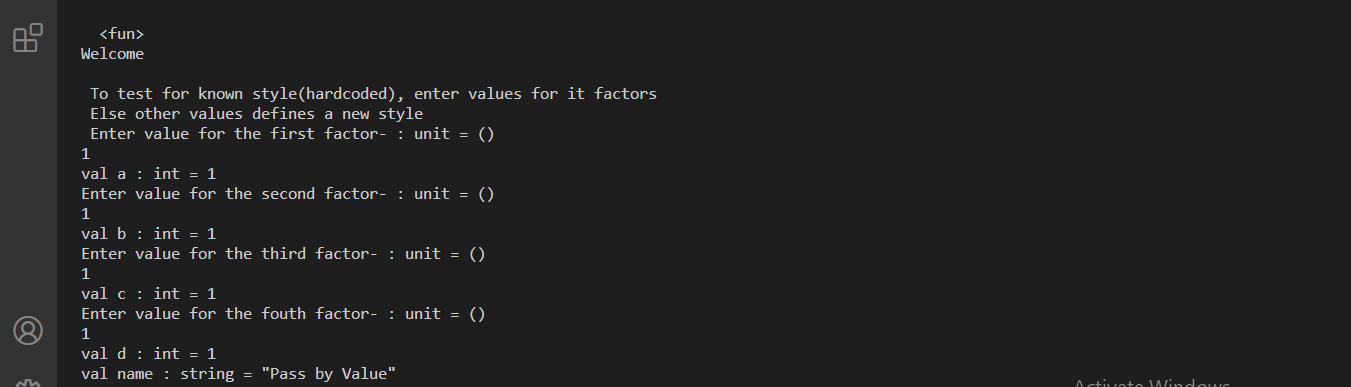


Fig.1

For values of known passing style (1, 1, 1, 1), this will display the name of the passing style, properties (name and factors) after which it has display the list of passing styles. This can be seen in Fig.2 below.

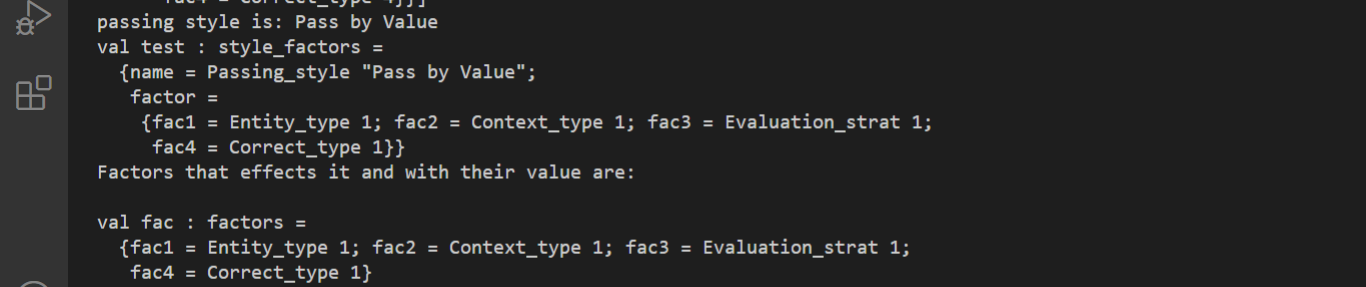


Fig.2

For values of factors that doesn’t correspond to known passing styles (1, 5, 7, 3), this will ask for a name to give to the passing style with values of factors (1, 5, 7, 3), constructs the passing style then adds it to the list of passing style. This can be seen in Fig.3 below.

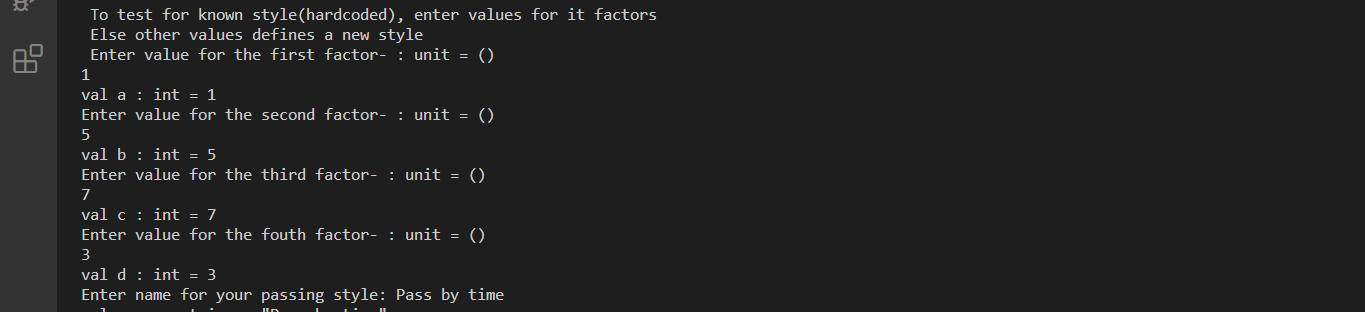


Fig.3

The new passing style is then added to the list of passing style and the style is displayed (its name and factors) after displaying the list of passing style which has the new style defined. This can be seen as indicated in the figure below.

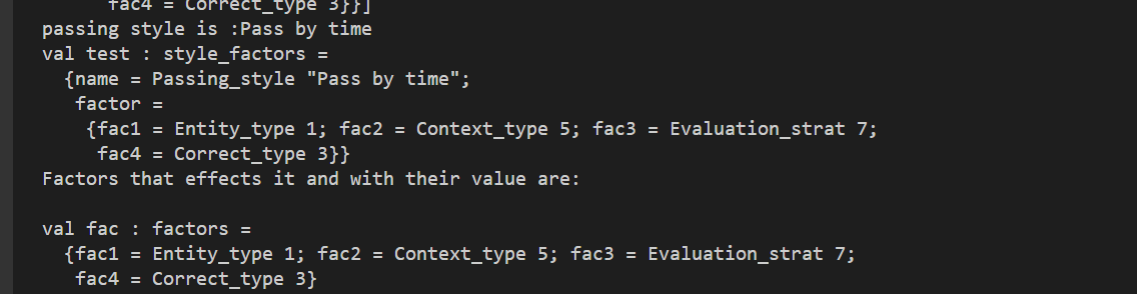


Fig.4

All the above mentioned is produced by the piece of code below

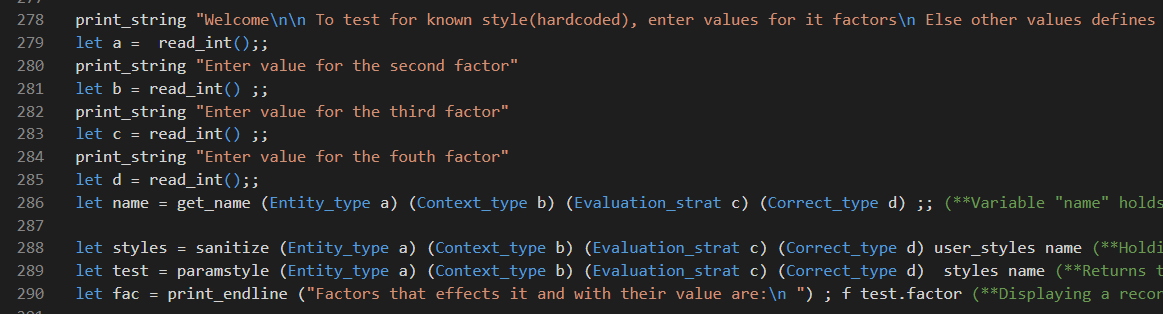


Fig.5

For the known passing styles, effects induced by specific values of factors are displayed as indicated below. This case is used to showcase (explain) the effects of call by value.

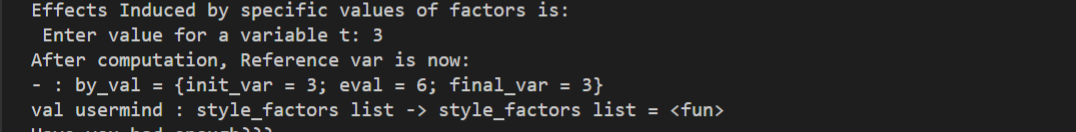


Fig.6

As seen in the figure above, by\_val is a record type that holds the effect that the specific values of factors induced and is used to showcase this effects. It ask for a number which is used to initialize a variable (v), init\_var is the initial value of the variable, eval is the result of a computation and final\_var is the final value of the variable after the computation. For call by value, we see that the value of the variable (v) used as argument in the method f is not modified after the computation. The piece of code that produces this effect can be seen in the figure below.

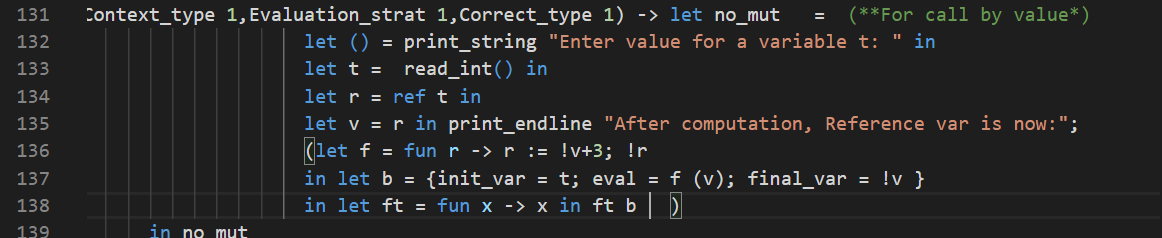


Fig.7

This effect is only for call by value. Other effects can be seen in code.

After this effect induced by specific values of factors has been displayed, a select menu is displayed asking if satisfaction has been achieved. If user goes for no (by selecting 2), a menu is displayed giving options on what they may wish to do next. If yes, then the list containing all passing styles (be it user defined or known styles) is displayed.

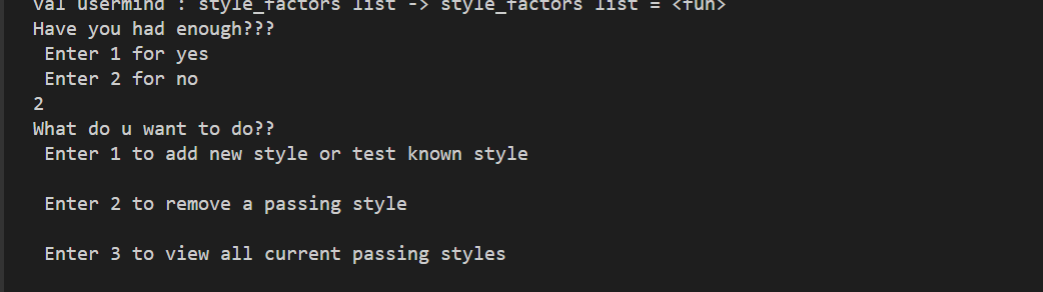


Fig.8

If option 2 is selected in the first menu, then a list of operations will be displayed for one to select from, and when selected, the operation is immediately initiated. This can be seen as shown below.

Selecting 1 for add new passing style, the values of the factors known to effect the passing style are entered and also a name for the passing style. The style is then added to the list of passing styles and the menu asking if satisfaction reached is displayed again. To see the list of styles, the selected option this time should be 1. This can be seen as shown on the figure below.

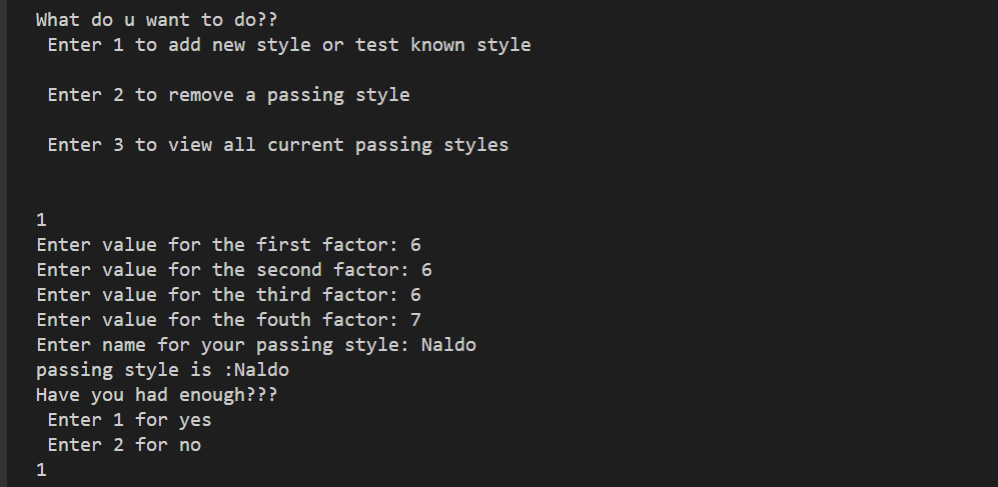


Fig.9

Selecting or entering 1 displays the list of passing styles included the recently add style or the result of the previous operation carried out if the option was entered. The list of passing style displayed when 1 is entered can be seen on the below. It includes the properties of the recently added passing style.

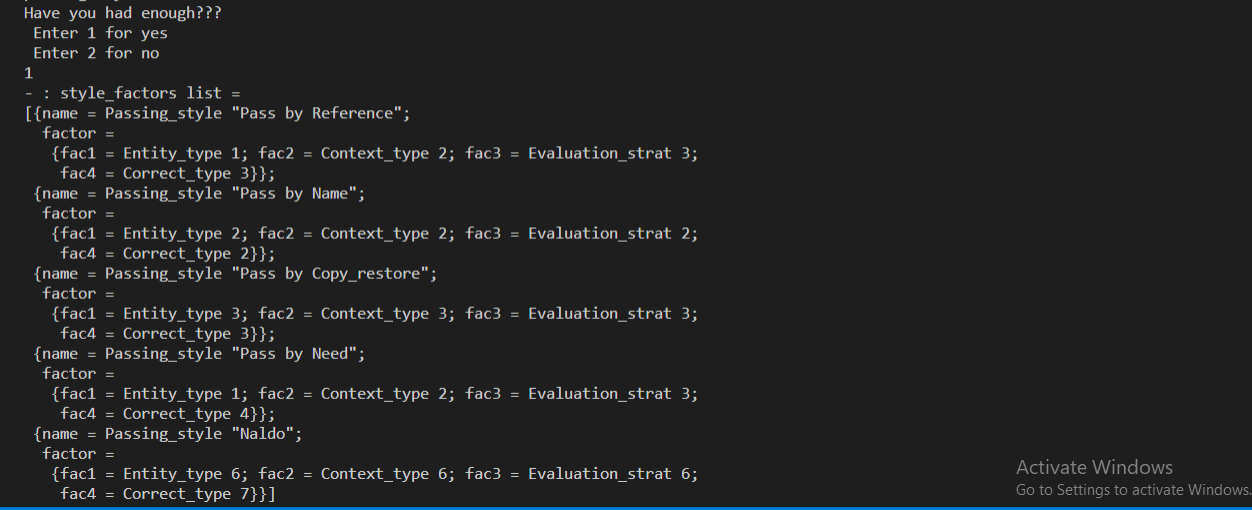


Fig.10

The piece of code used to perform the above relating to Fig 8, 9, and 10 can be seen on the figure below.

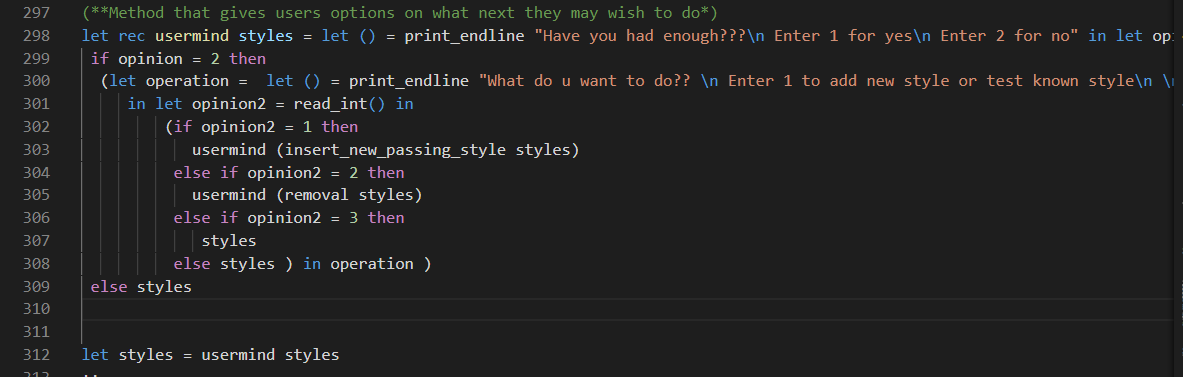


Fig.11

The methods “insert\_new\_passing\_style” and “removal” carries out operations for inserting and removing a passing style from the list of passing styles respectively.

After this process, if users wants to still carry out operations like add or remove a passing style, the method is invoked by typing the command (calling the method) “usermind styles” where usermind is the method invoked and styles refers to the current list of passing styles.