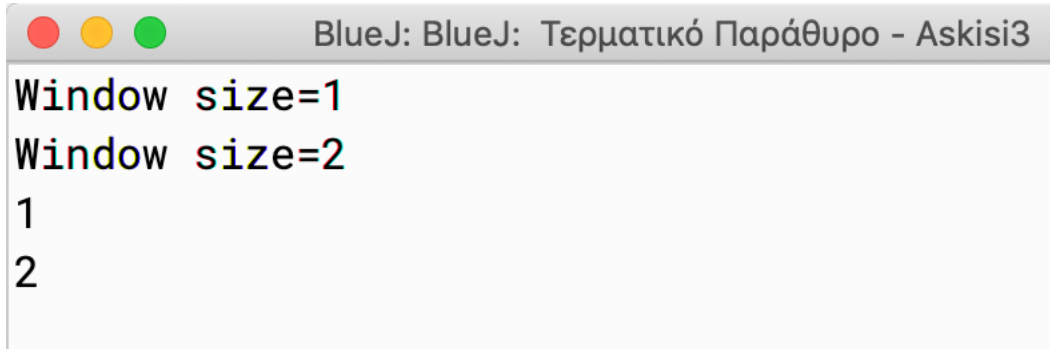


ΑΣΚΗΣΗ 3

2^α) Τρέχοντας το πρόγραμμα παίρνουμε το παρακάτω αποτέλεσμα:



```
Window size=1
Window size=2
1
2
```

Η κλάση RunWindow δημιουργεί δύο αντικείμενα(w1,w2) τα οποία εντάσσονται στην κλάση Window και μετέπειτα ,το ένα δίχως όρισμα και το άλλο με όρισμα 2.Στη συνέχεια καλείται να εμφανιστεί το στυγμιότυπο size από τους δημιουργούς Window και Window(int size).Επειδή στον πρώτο δημιουργό το στυγμιότυπο size αρχικοποιείται (size=1) και στον δεύτερο το size παίρνει την τιμή του ορίσματος,προκύπτει το παραπάνω αποτέλεσμα.(Σημειώνεται ότι εκτελούνται και οι εντολές `System.out.println("Window size="+size);`

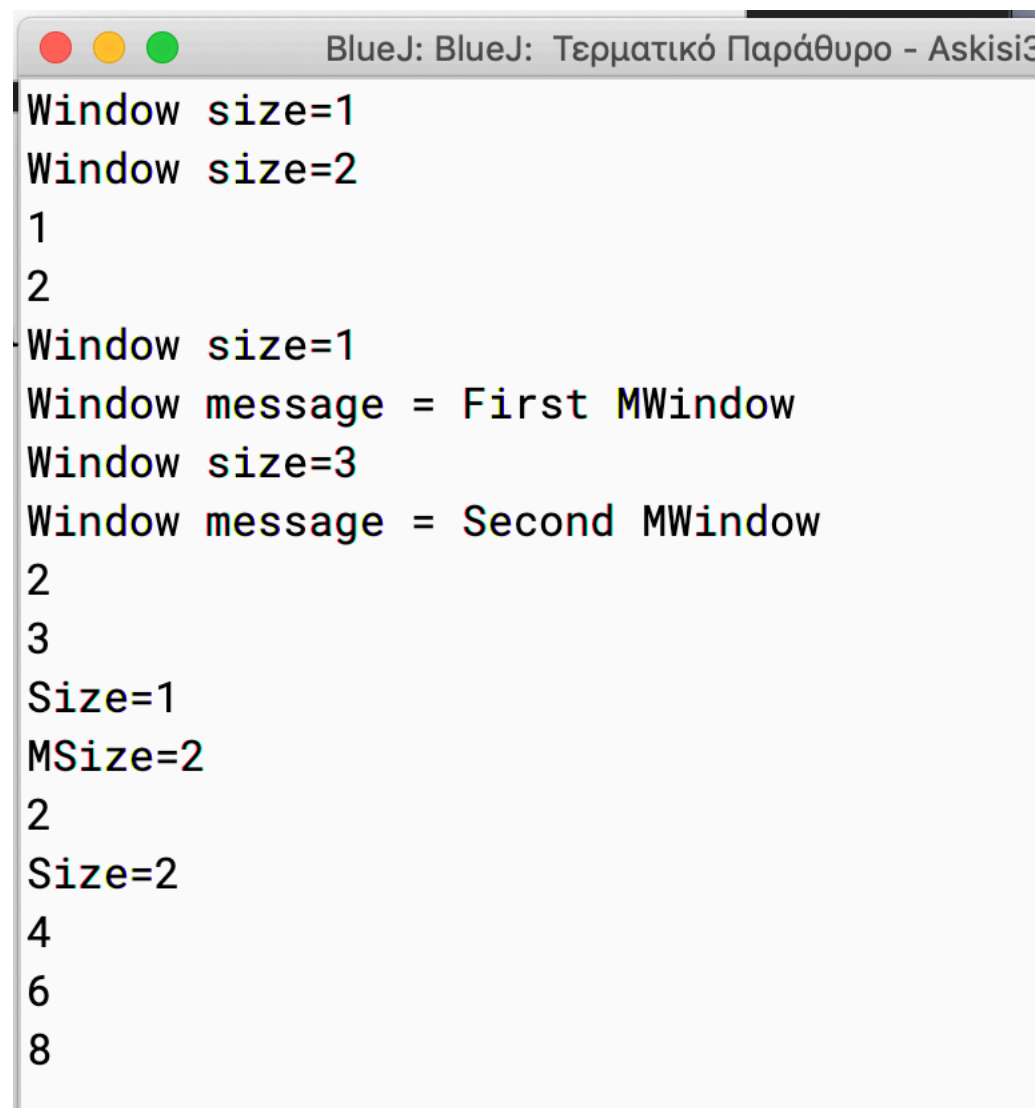
οι οποίες βρίσκονται εντός των δημιουργών επειδή η RunWindow αποτελεί υποκλάση της Window(έμεσσο super).

2^β) Τρέχοντας το πρόγραμμα με τις επιπρόσθετες εντολές έχουμε ως αποτέλεσμα:

```
BlueJ: BlueJ: Τερματικό Παράθυρο - Askisi3
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
3
```

Πρακτικά η RunWindow δημιουργεί 2 αντικείμενα τα οποία εντάσσονται στους δημιουργούς MWindow(String message) και MWindow(int size, String message) αντίστοιχα, της κλάσης MWindow η οποία αποτελεί υποκλάση της Window. Το πρώτο αντικείμενο αποκτά size=2 και το δεύτερο αποκτά size=3=με το όρισμα που έχει δοθεί(όπως φαίνεται από τον κώδικα της MWindow). Τέλος όταν η RunWindow εκτυπώνει τις size των αντικειμένων w1 και w2 και όπως και πριν, επιδή όταν καλούμε προς εκτέλεση την μέθοδο μιας υποκλάσης εκτελείται και ο κώδικας της κύριας κλάσης(έμεσσο super), καταλήγουμε το παραπάνω αποτέλεσμα.

2γ) Εκτελώνοντας το πρόγραμμα με τις νέες εντολές έχουμε το αποτέλεσμα:

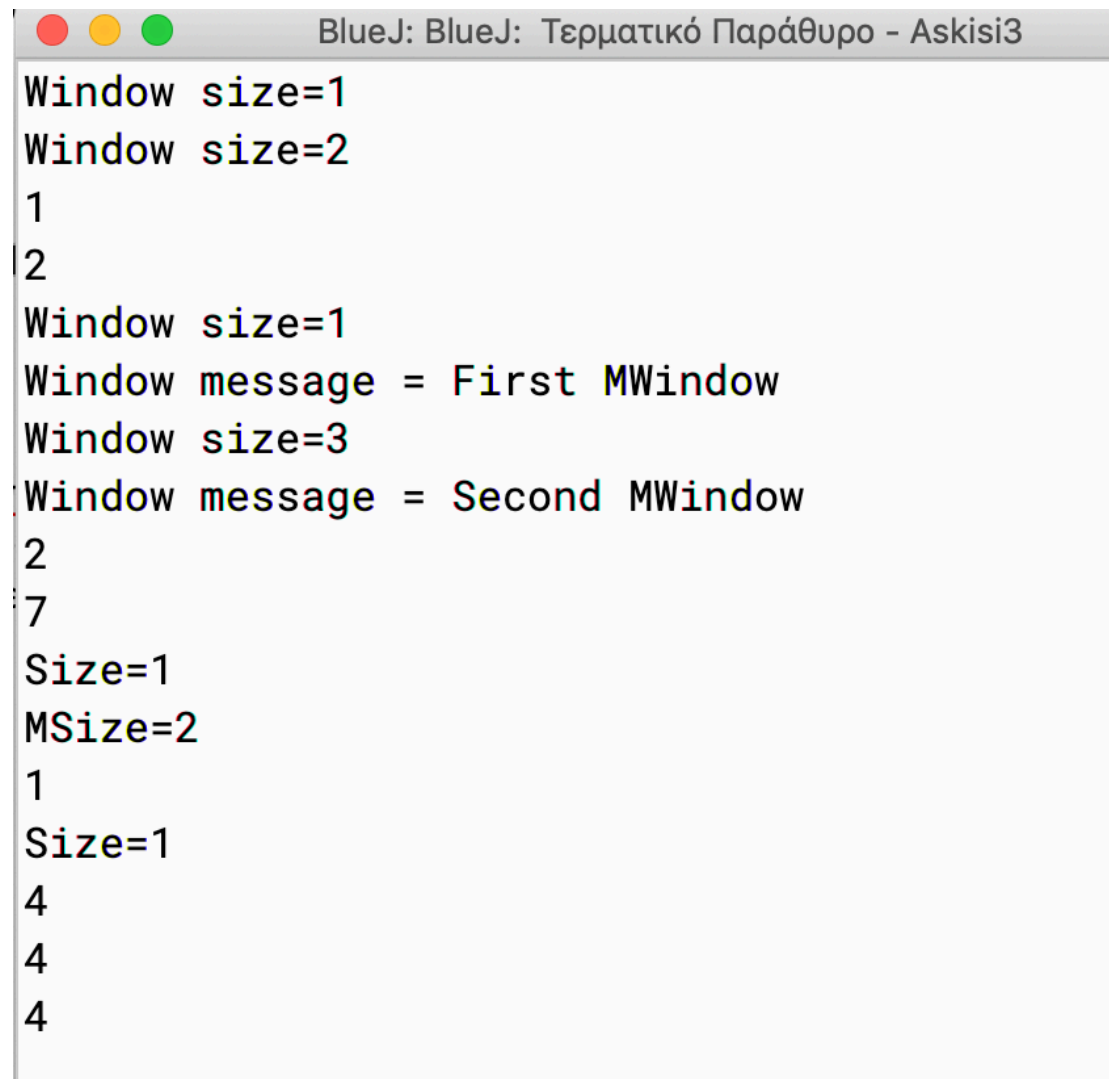


```
BlueJ: BlueJ: Τερματικό Παράθυρο - Askisi3
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
3
Size=1
MSize=2
2
Size=2
4
6
8
```

Τα επιπρόσθετα αποτελέσματα οφείλονται στην κλήση των μεθόδων `setSize1`, `setSize2`, `setSize` από την `main`. Οι μέθοδοι `setSize1`, `setSize2` εντάσσονται στην υποκλάση `MWindow` και εντάσσεται στη `Window`. Η `setSize1` δέχεται ένα όρισμα και το θέτει το στιγμιότυπό της `size` ίσο με το όρισμα, η `setSize2` δέχεται ένα όρισμα αλλά ωστόσο λόγω της εντολής `super.setSize(z)`; Θέτει το στιγμιότυπο `size` της `setSize()` ίσο με την επικαλυπτόμενη μεταβλητή `size` από προηγουμένως + το όρισμα που έχουμε δώσει (γιαυτό και $4 + 2 = 6$ το οποίο είναι το αποτέλεσμα μας). Τέλος η `setSize` από την κλάση `Window`

παίρνει το ήδη υπάρχον στιγμότυπο(στην περίπτωση μας size=6) και του προσθέτει το όρισμα που έχουμε δώσει στη μέθοδο(γιαυτό size += 8 στην περίπτωση μας).

3) Τρέχοντας τώρα το πρόγραμμα παίρνουμε τα εξής αποτελέσματα.

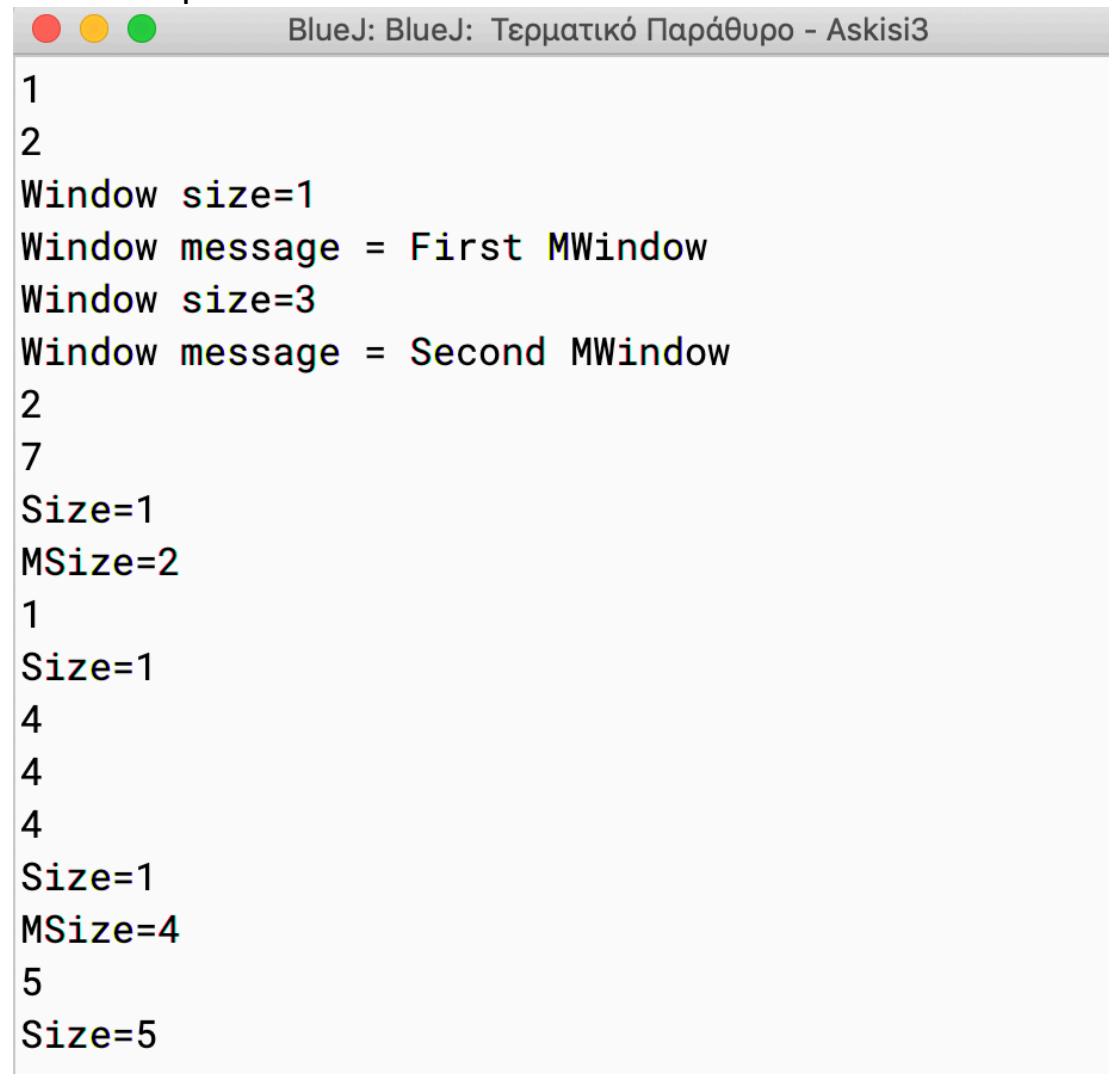


```
Window size=1
Window size=2
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
Size=1
MSize=2
1
Size=1
4
4
4
```

Τα αποτελέσματα διαφέρουν αρκετά από αυτά της 2). Αυτό συμβαίνει αποκλειστικά και μόνο λόγω της ύπαρξης της εντολής super στις μεθόδους Window, setSize2, setSize και printSize1,2. Πρακτικά η super όταν χρησιμοποιείται για κλήση μεταβλητών, καλεί τις μεταβλητές από την υπερκλάση της κλάσης στην οποία βρισκόμαστε. Ακριβώς αυτό συμβαίνει και

εδώ: με την χρήση της super καλούμε τις επικαλυπτόμενες μεταβλητές οι οποίες βρίσκονται πιο «πάνω» στην ιεραρχία απότι οι εσωτερικές των μεθόδων. Συγκεκριμένα ο συνδιασμός της εντολής `protected int size = 7;` με την εντολή `super(size);` μέσα στην `MWindow` διακηρύσσει την ύπαρξη του αριθμού 7 στο αποτέλεσμα (αφού η `protected int size` δρα ως επικαλυπτόμενη) ενώ στην `setSize 2` καλείται η `size=2` από την `MWindow`, όπως και στην `setSize` καλείται η ήδη υπάρχουσα `size=2` από την `MWindow` και πάλι. Στην `setSize1` δεν υπάρχει εντολή `super` επομένως δεν μεταβάλλεται το αποτέλεσμα.

4) Τρέχοντας το πρόγραμμα προκύπτουν τα εξής αποτελέσματα:



```
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
Size=1
MSize=2
1
Size=1
4
4
4
Size=1
MSize=4
5
Size=5
```

Τα αποτελέσματα αυτά είναι λογικά αφού η `printSize()` της κλάσης `Window` δεν μεταβάλλεται από τις αλλαγές στις υποκλάσεις. Επιπλέον η `printSize` της `MWindow` αφού δεν εμπεριέχει τη `super`, αντίστοιχα δεν επιρεάζεται το αποτέλεσμα της. Ωστόσο η `printSize1` όπως και η `printSize2`, λόγω της ύπαρξης των εντολών `System.out.println (super.size);` και `super.printSize();` καλούν τις μεταβλητές `size` και την μέθοδο `setSize` από την υπερκλάση `Window`, με αποτέλεσμα την αλλαγή της τιμής των αποτελεσμάτων σε σχέση με προηγουμένως.

5) Με τις νέες εντολές τα αποτελέσματα δεν αλλάζουν:



```
1
2
Window size=1
Window message = First MWindow
Window size=3
Window message = Second MWindow
2
7
Size=1
MSize=2
1
MSize=2
4
4
4
Size=1
MSize=4
5
MSize=4
```

Αυτό συμβαίνει επειδή οι εντολές :

```
1. System.out.println ( (Window) this ).size;
```

Και

```
2. System.out.println (super.size);
```

Είναι ισοδύναμες, αφού και οι δύο παραπέμπουν στην ίδια μεταβλητή size της υπερκλάσης Window.