

Object-Technology

Lego Construction Approach

The RPN Calculator Example application

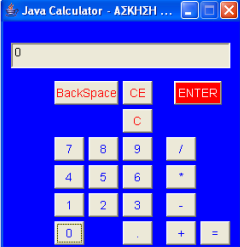
Kleanthis Thramboulidis
 Software and System Engineering
 University of Patras, Greece
<https://sites.google.com/site/thramboulidiskleanthis/>

1

H RPN Calculator example application

- Develop a program for the system to accept expressions based on the Polish reverse notation and calculate their values.

3 6 + 8 6 - * (3+6) * (8-6)
- Design Constraint
 - Stack has to be used to store operands.



©Κλεάνθης Θραμβουλίδης "Lego Construction" Approach

2

Ο Στόχος

- Η άσκηση έχει ως στόχο την εξοικείωση:
 - Με την διαδικασία ανάπτυξης εφαρμογής σε Java
 - με το περιβάλλον JDK της SUN το οποίο υποστηρίζει την ανάπτυξη εφαρμογών σε Java
 - με την εκπαιδευτική εφαρμογή BlueJ και το Eclipse
 - με την τεκμηρίωση της βασικής βιβλιοθήκης της Java
 - με την αξιοποίηση της βασικής βιβλιοθήκης της Java

©Κλεάνθης Θραμβουλίδης "Lego Construction" Approach

4

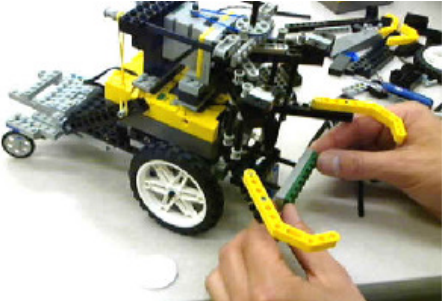
Agenda

- "Lego construction" approach
- Repository of components
- Program as aggregation of existing components
 - Mechanisms to integrate components
- The need for new objects
 - Mechanisms to Build new components
- The RPN Calculator example application

©Κλεάνθης Θραμβουλίδης "Lego Construction" Approach

5

"Lego construction" approach




©Κλεάνθης Θραμβουλίδης "Lego Construction" Approach

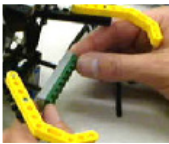
6

"Lego construction" approach

Requirements from the implementation environment



Repository of components



Mechanisms to Integrate components

Mechanisms to Build new components

©Κλεάνθης Θραμβουλίδης "Lego Construction" Approach

7

Σύνθεση Κλάσης

```
class Student {  
    int am;  
    static int numOfInstances;  
    int yearOfStudy();  
    public Student(...);  
    static int getNumOfInstances();  
}
```

Data member στιγμιότυπου
Προσδιορίζουν δομή στιγμιότυπου

Data member κλάσης
Προσδιορίζουν δομή κλάσης

Μέθοδος στιγμιότυπου
Προσδιορίζουν συμπεριφορά στιγμιότυπου

Δημιουργός
Μέθοδος για την δημιουργία στιγμιότυπων

Μέθοδος κλάσης
Προσδιορίζουν συμπεριφορά κλάσης

©Κλεάνθης Θεραπειουλίδης

"Lego Construction" Approach

14

14

Χρήση της κλάσης Double (1)

- Μέσα από το BlueJ
 - Δημιουργία στιγμιότυπου
 - Αποστολή μηνύματος σε στιγμιότυπο
 - Αποστολή μηνύματος στην κλάση

©Κλεάνθης Θεραπειουλίδης

"Lego Construction" Approach

15

15

Δραστηριότητα στο BlueJ

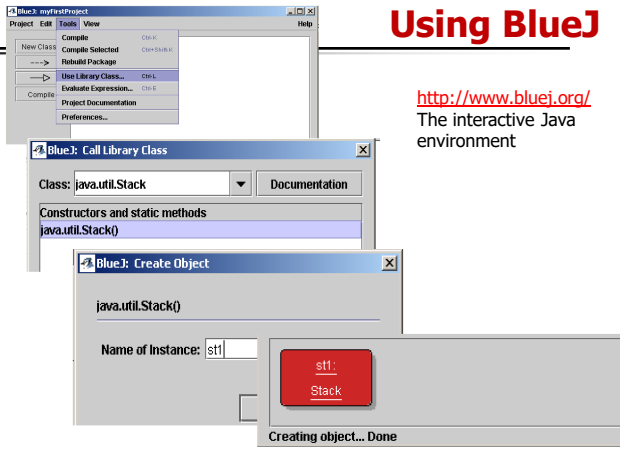
- Δημιουργήστε ένα **στιγμιότυπο της κλάσης Double** και αποδώστε του μια αρχική τιμή της αρεσκείας σας.
- Δημιουργήστε ένα δεύτερο στιγμιότυπο της κλάσης Double και αποδώστε του μια αρχική τιμή της αρεσκείας σας.
- Δημιουργήστε **μια στοιβα** και βάλτε τα παραπάνω στιγμιότυπα μέσα σε αυτήν.
- Βγάλτε τα δύο στιγμιότυπα από την στοιβα και κάντε τα Get στον χώρο στιγμιότυπων του BlueJ.
- Τέλος δημιουργήστε ένα νέο στιγμιότυπο της κλάσης Double και αποδώστε ως τιμή **το άθροισμα** των τιμών των δύο στιγμιότυπων που βγάλατε από την στοιβα.

©Κλεάνθης Θεραπειουλίδης

"Lego Construction" Approach

16

16



Using BlueJ

<http://www.bluej.org/>
The interactive Java environment

©Κλεάνθης Θεραπειουλίδης

"Lego Construction" Approach

17

17

API vs Command line

- Μπορώ να αποστείλω μήνυμα στην κλάση Double και τα στιγμιότυπα της μόνο μέσα από κώδικα (χρήση του API της κλάσης)
`Double.compare(10.2, 12.1);`
`myDouble1.compareTo(myDouble2);`
- Το BlueJ μου επιτρέπει να στείλω μήνυμα σε αντικείμενο διαμέσου γραφικής διεπαφής

©Κλεάνθης Θεραπειουλίδης

"Lego Construction" Approach

18

18

Χρήση της κλάσης Double (2)

Στα πλαίσια δημιουργίας νέου αντικειμένου

- Δημιουργία στιγμιότυπου και στη συνέχεια αποστολή μηνύματος σε αυτό
`Double myDouble1 = new Double(24.0);`
`myDouble1.doubleValue();`
- Αποστολή μηνύματος στην κλάση
`Double.valueOf("12.0");`

©Κλεάνθης Θεραπειουλίδης

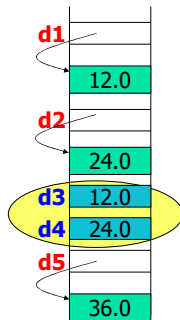
"Lego Construction" Approach

19

19

Χρήση της κλάσης Double (3)

```
Double d1 = new Double(12.0);
Double d2 = new Double(24.0);
double d3 = d1.doubleValue();
double d4 = d2.doubleValue();
Double d5 = new Double(d3+d4);
```



©Κλεάνθης Θερμοπούλιδης

"Lego Construction" Approach

20

20

Agenda

- "Lego construction" approach
- Repository of components
- **Program as an aggregation of existing components**
 - Mechanisms to integrate components
- The need for new objects
 - Mechanisms to Build new components
- Using the standard library

©Κλεάνθης Θερμοπούλιδης

"Lego Construction" Approach

21

21

1st iteration

Αναπτύξτε ένα πρόγραμμα σε Java σύμφωνα με το οποίο το σύστημα θα προσθέτει τους αριθμούς 12.0 και 24.0 και θα δίνει στην κύρια έξοδο το αποτέλεσμα της άθροισης τους.

- **Design Constraint**
 - Stack must be used to store operands.

©Κλεάνθης Θερμοπούλιδης

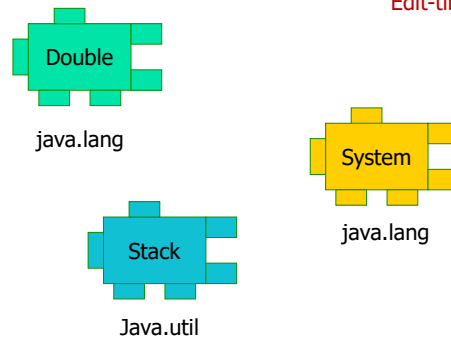
"Lego Construction" Approach

22

22

The constituent components

Edit-time vs. run-time



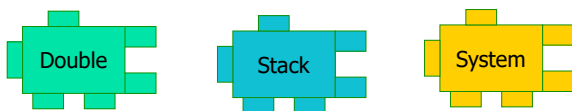
©Κλεάνθης Θερμοπούλιδης

"Lego Construction" Approach

23

23

Integrate existing components



Ένα περιβάλλον ανάπτυξης αντικειμενοστρεφών εφαρμογών πρέπει να παρέχει

- Μηχανισμούς για χρήση των έτοιμων αντικειμένων.
 - Δημιουργία στιγμιοτύπων
 - αποστολή μηνυμάτων σε αντικείμενα

©Κλεάνθης Θερμοπούλιδης

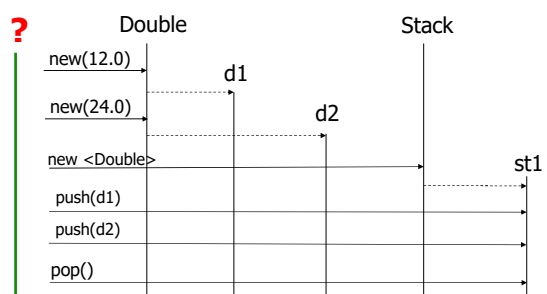
"Lego Construction" Approach

24

24

Διάγραμμα Αλληλεπίδρασης Αντικειμένων

UML sequence diagram



Ανάγκη για δημιουργία νέου αντικειμένου

©Κλεάνθης Θερμοπούλιδης

"Lego Construction" Approach

25

25

Agenda

- "Lego construction" approach
- Repository of components
- Program as an aggregation of existing components
 - Mechanisms to integrate components
- **The need for new objects**
 - Mechanisms to Build new components
- Using the standard library

©Κλεάνθης Θραμπουλιδής

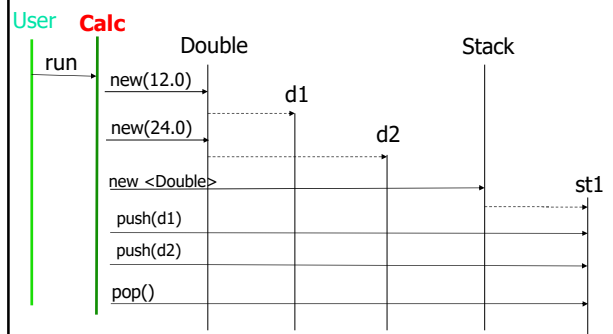
"Lego Construction" Approach

26

26

Διάγραμμα Αλληλεπίδρασης Αντικειμένων

UML sequence diagram



©Κλεάνθης Θραμπουλιδής

"Lego Construction" Approach

27

27

Δημιουργία αντικειμένου

επιβάλλεται για την κατασκευή προγράμματος

- Απαιτείται ένα αντικείμενο το οποίο όταν το πρόγραμμα ενεργοποιείται θα αναλαμβάνει να δημιουργήσει τα απαραίτητα στιγμιότυπα και τις σχέσεις μεταξύ των αντικειμένων που απαρτίζουν το πρόγραμμα.
- Το αντικείμενο αυτό είναι το αντικείμενο που αναπαριστά το πρόγραμμα μας.

©Κλεάνθης Θραμπουλιδής

"Lego Construction" Approach

28

28

Build new components

Ένα περιβάλλον ανάπτυξης αντικειμενοστρεφών εφαρμογών πρέπει να παρέχει

- Μηχανισμούς για δημιουργία νέων τύπων αντικειμένων
 - Ορισμό κλάσης δηλαδή ορισμό δομής (structure) και συμπεριφοράς (behavior) στιγμιότυπων
 - Ορισμό διεπιφάνειας και υλοποίησης των αντικειμένων
- Μηχανισμό για ορισμό αντικειμένου σαν συνάθροιση των μερών του
- Μηχανισμό για ορισμό αντικειμένου σαν εξειδίκευση άλλου αντικειμένου

©Κλεάνθης Θραμπουλιδής

"Lego Construction" Approach

29

29

Αναπαράσταση προγράμματος

- Αναπαριστούμε το πρόγραμμα με μία κλάση
- Η κλάση θα πρέπει να έχει συμπεριφορά στο μήνυμα **run** που στέλνει ο χρήστης διαμέσου του λειτουργικού συστήματος
- Μέθοδος **main()**

©Κλεάνθης Θραμπουλιδής

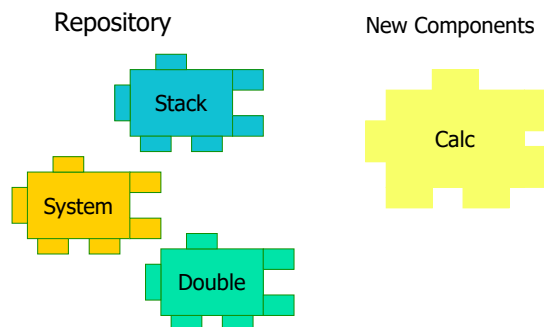
"Lego Construction" Approach

30

30

The constituent components

Edit-time vs. run-time



©Κλεάνθης Θραμπουλιδής

"Lego Construction" Approach

31

31

class Calc



32

Ορισμός της κλάσης Calc

```
class Calc {  
    //data members ?  
    // methods ?  
    public static void main() {  
        ?  
    }  
}
```

33

class Calc (1st version)

```
public class Calc{  
    static Double d1,d2;  
    static Stack<Double> st;  
  
    public static void main(){  
        d1 = new Double(20);  
        d2 = new Double(40);  
        st = new Stack<Double>();  
        st.push(d1);  
        st.push(d2);  
        Double d3 = st.pop();  
        Double d4 = st.pop();  
        st.push(d3+d4);  
        System.out.println("sum is="+ st.pop());  
    }  
}
```

Compare it with
the object
interaction
diagram

34

Generics

- enable *types* (classes and interfaces) to be parameters when defining classes, interfaces and methods.
- Much like the more familiar *formal parameters* used in method declarations, type parameters provide a way for you to re-use the same code with different inputs. The difference is that the inputs to formal parameters are values, while the inputs to type parameters are types.

35

Generics

- Code that uses generics has many benefits over non-generic code:
 - Stronger type checks at compile time.
A Java compiler applies strong type checking to generic code and issues errors if the code violates type safety. Fixing compile-time errors is easier than fixing runtime errors, which can be difficult to find.
- Elimination of casts.
 - Enabling programmers to implement generic algorithms.
By using generics, programmers can implement generic algorithms that work on collections of different types, can be customized, and are type safe and easier to read.

36

Αναγνώριση αντικειμένων

Χρήση της έννοιας του use case

- Χρησιμοποιούμε την έννοια των **use cases** (περιπτώσεις χρήσης) για να εντοπίσουμε τα αντικείμενα από τα οποία θα αποτελείται το σύστημα μας.
- Πόσες και ποιες υπηρεσίες παρέχει το πρόγραμμα μας;

37

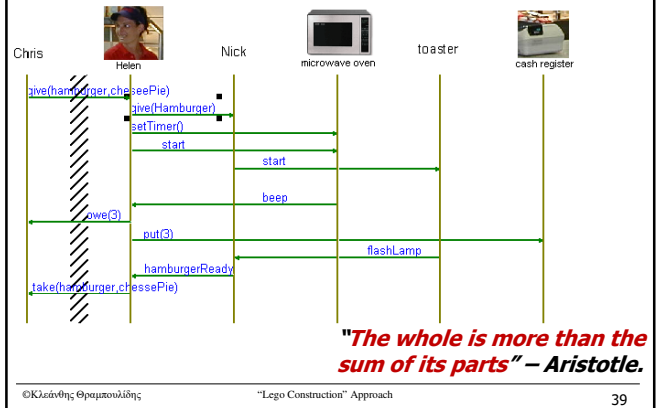
Περίπτωση χρήσης συστήματος

Actor: πελάτης

Περιγραφή: Ο πελάτης ενεργοποιεί το πρόγραμμα. Το σύστημα δημιουργεί ένα στιγμιότυπο τύπου Double με τιμή 12.0. Το σύστημα δημιουργεί ένα στιγμιότυπο τύπου Double με τιμή 24.0. Το σύστημα βρίσκει το άθροισμα τους. Τυπώνει το αποτέλεσμα. Η περίπτωση χρήσης ολοκληρώθηκε.

Ειδικές περιπτώσεις:

Object Interaction Diagram



"The whole is more than the sum of its parts" – Aristotle.

2nd iteration - Increase reusability

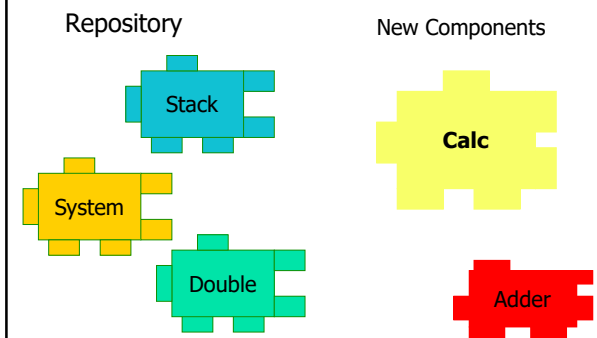
```
public class Calc{
    ....
    public static void main(){
        d1 = new Double(20);
        d2 = new Double(40);

        st = new Stack<Double>();
        st.push(d1);
        st.push(d2);

        Double d3 = st.pop();
        Double d4 = st.pop();
        st.push(d3+d4);
        System.out.println("sum is="+st.pop());
    }
}
```



2nd Version – Constituent Components

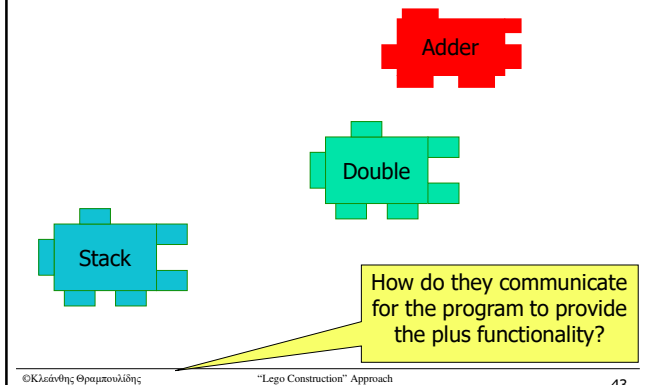


Build new Component

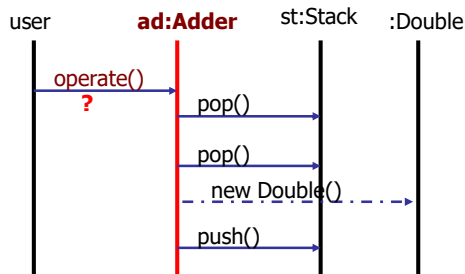


```
class Adder{
    public static void operate(){
        Double d3 = st.pop();
        Double d4 = st.pop();
        st.push(new Double(d3+d4));
    }
}
```

The components for the + operation



Object Interaction Diagram for +



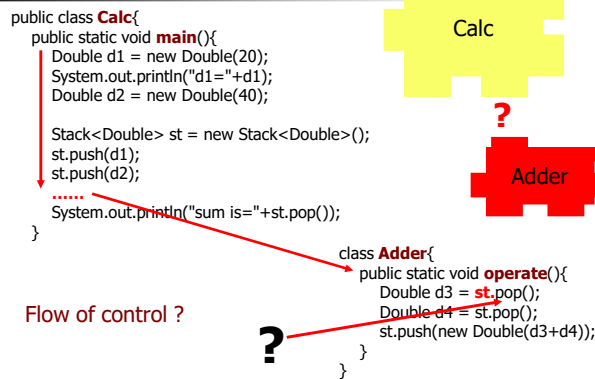
©Κλεάνθης Θραμμουλίδης

"Lego Construction" Approach

44

44

Integrate components



©Κλεάνθης Θραμμουλίδης

"Lego Construction" Approach

45

45

Source code (2nd version)

```
public class Calc{
    public static Stack<Double> st;
    public static void main(){
        Double d1 = new Double(20);
        System.out.println("d1="+d1);
        Double d2 = new Double(40);

        st = new Stack<Double>();
        st.push(d1);
        st.push(d2);

        Adder.operate();
        System.out.println("sum is="+st.pop());
    }
}

class Adder{
    public static void operate(){
        Double d3 = Calc.st.pop();
        Double d4 = Calc.st.pop();
        st.push(new Double(d1+d2));
    }
}
```

Identify alternative Implementations for addition

©Κλεάνθης Θραμμουλίδης

"Lego Construction" Approach

46

46

Kleanthis Thramboulidis
 Διαχειριστής · 25 Οκτωβρίου στις 5:03 μ.μ. · Exercise

Εναλλακτικές Υλοποιήσεις #Ασκηση #RPNCalculator #Reusability #Activity3

Το 2ο iteration της δραστηριότητας 3 έχει ως στόχο την αύξηση του modularity του κώδικα του 1ου iteration.

Επιλέξαμε το παρακάτω τμήμα τους κώδικα της main που υλοποιεί μια συγκεκριμένη λειτουργικότητα (functionality).

```
Double d3 = st.pop();
Double d4 = st.pop();
st.push(d3+d4);
```

Το iteration αυτό έχει ως στόχο την αναζήτηση εναλλακτικών υλοποιήσεων που θα μας επιτρέψουν την επαναχρησιμοποίηση της λειτουργικότητας αυτής.

Δώστε με απάντηση στην ανάρτηση αυτή, μια πρόταση υλοποίησης αφού πρώτα έχετε ελέγξει πως αυτή δεν έχει περιγραφεί από κάποιον άλλον πριν από εσάς.

Περιγράψτε με λόγια την προτεινόμενη υλοποίηση και τον τρόπο που η λειτουργικότητα αυτή μπορεί να αξιοποιηθεί στα πλαίσια της main.

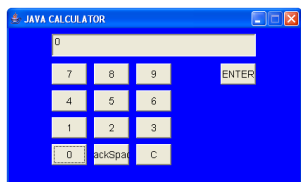
Δώστε στο τέλος της περιγραφής σας το URL του pdf αρχείου με τον κώδικα της πρότασης σας.

Στόχος της ανάρτησης αυτής είναι η καταγραφή των εναλλακτικών προτάσεων υλοποίησης.

13 1 κοινοποίηση 47

47

3rd Iteration – Adding a Gui



30 61 + 8 6 - * =

CalcGuiV1Exer6.jar

- Ποια αντικείμενα απαρτίζουν το σύστημα;
- Πως αυτά πρέπει να συνεργαστούν ώστε το πρόγραμμα μας να παρέχει τη ζητούμενη συμπεριφορά;

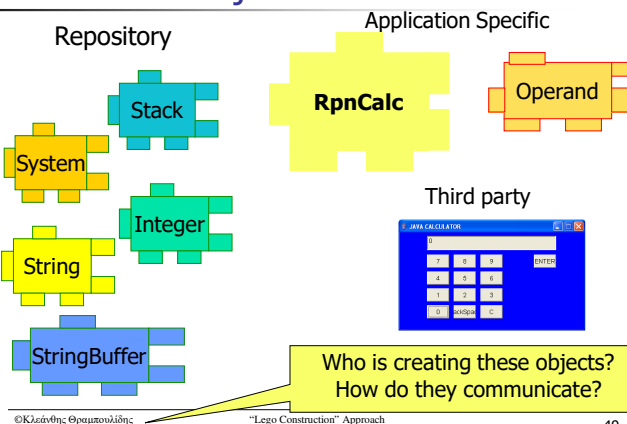
©Κλεάνθης Θραμμουλίδης

"Lego Construction" Approach

48

48

Constituent objects



©Κλεάνθης Θραμμουλίδης

"Lego Construction" Approach

49

49