



Μάθημα: Προγραμματισμός Ι

3^η Συνεδρία Ηλεκτρονικής Υποστήριξης

Σοφοκλής Στουραΐτης

sofos@aueb.gr

Packages (Πακέτα)



Πακέτα (packages)

- Ένα **πακέτο** (**package**) κλάσεων (ή/και διεπαφών) είναι μία ομαδοποιημένη συλλογή ενός πλήθους κλάσεων. Τα πακέτα συνήθως αποτελούν τρόπο ομαδοποίησης κλάσεων συναφούς λειτουργικότητας και είναι ένας τρόπος να οργανώσουμε συλλογές κλάσεων Java σε βιβλιοθήκες, και να αποφύγουμε τη σύγκρουση ονομάτων.
- Ένα **package** μπορεί να περιέχει οποιοδήποτε αριθμό κλάσεων που σχετίζονται μεταξύ τους με βάση το **σκοπό**, την **εμβέλεια** ή την **κληρονομικότητά** τους.
 - Στη Java μέσω των **πακέτων** σχηματίζονται **βιβλιοθήκες** κλάσεων.
- Η δήλωση του πακέτου είναι το πρώτο πράγμα που γίνεται σε μια κλάση (1^η γραμμή)
Σύνταξη: **package** <packageName>;
Παράδειγμα: **package** lab3;
public class Test { ... Test class code ... }
- Οι κλάσεις του κάθε πακέτου τοποθετούνται σε δικό τους φάκελο (**folder**) που έχει το ίδιο όνομα με το πακέτο. Η δημιουργία του φακέλου γίνεται αυτόματα κατά τη δημιουργία του πακέτου σε ολοκληρωμένα περιβάλλοντα ανάπτυξης (IDE) όπως είναι το **Eclipse**.
- Οι κλάσεις του πακέτου μπορεί να χρησιμοποιηθούν από άλλη κλάση (εκτός πακέτου) που θα τις κάνει **import** (πχ **import** lab3.Test; ή **import** lab3.*;) .



Πακέτα (packages) και υπο-πακέτα (subpackages)

- Τα πακέτα μπορεί να οργανώνονται περαιτέρω σε **υποπακέτα** (subpackages) κ.ο
- Τα πακέτα χωρίζονται σε 2 κατηγορίες
 - **Build-in packages** (πακέτα που τα διαθέτει-προσφέρει η Java)
 - **User Defined packages** (Αυτά που δημιουργούμε εμείς)
- Οι κλάσεις από τις οποίες αποτελείται η java είναι ομαδοποιημένες σε τέτοιου είδους πακέτα (Build-in packages), ενώ κάθε πακέτο έχει το δικό του όνομα.

Παραδείγματα:

- Το πακέτο **java.lang** περιέχει τις θεμελιώδεις κλάσεις της java (όπως π.χ. την Κλάση Math για τις μαθηματικές συναρτήσεις ή την κλάση String).
- Το πακέτο **java.io** περιέχει κλάσεις οι οποίες αφορούν στην διαχείριση των λειτουργιών εισόδου/εξόδου.
- Το πακέτο **java.util** παρέχει κλάσεις οι οποίες καλύπτουν ένα εύρος από τυποποιημένες λειτουργίες (πχ η Scanner) όπως η διαχείριση δεδομένων, η πρόσβαση στην ώρα και στην ημερομηνία του συστήματος κτλ.
- Το πακέτο **java.sql** παρέχει κλάσεις για πρόσβαση και επεξεργασία δεδομένων που είναι αποθηκευμένα σε μια (συνήθως σχεσιακή) βάση δεδομένων.



Πακέτα (packages) - 2

- Παράδειγμα η κλάση Person

```
1 package lab3.example2;  
2  
3 public class Person {  
4  
5     private String name;  
6     private String surname;  
7     private int yearOfBirth;  
8  
9     public Person(String name, String surname, int yearOfBirth) {  
10         this.name = name;  
11         this.surname = surname;  
12         this.yearOfBirth = yearOfBirth;  
13     }  
14  
15     public String getName() {  
16         return name;  
17     }  
18  
19     public void setName(String name) {  
20         this.name = name;  
21     }  
22  
23     public String getSurname() {  
24         return surname;  
25     }  
26  
27     public void setSurname(String surname) {  
28         this.surname = surname;  
29     }  
30  
31     public int getYearOfBirth() {  
32         return yearOfBirth;  
33     }  
34  
35     public void setYearOfBirth(int yearOfBirth) {  
36         this.yearOfBirth = yearOfBirth;  
37     }  
38  
39     @Override  
40     public String toString() {  
41         return "Person [name=" + name + ", surname=" + surname + ", yearOfBirth=" + yearOfBirth + "];"  
42     }  
43 }
```

- Χρήση της κλάσης Person από άλλη κλάση (εκτός του ίδιου πακέτου)

Πακέτο

Υπο-πακέτο (subpackage)

Κλάση

```
1 import lab3.example2.Person;  
2  
3 public class TestPerson {  
4  
5     public static void main(String[] args) {  
6  
7         Person someone = new Person("John", "Doe", 1988);  
8  
9         System.out.println( someone );  
10        //the same as: System.out.println( someone.toString() );  
11  
12    }  
13 }
```

<terminated> TestPerson [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (30 Μαρ 2019, 7:39:38 μ.μ.)
Person [name=John, surname=Doe, yearOfBirth=1988]

Εκτέλεση (run) της TestPerson



Πακέτα (packages) - 3

- Χρησιμοποιούμε μικρά γράμματα για την ονοματολογία του πακέτου.
- Βασικά πλεονεκτήματα:
 - **Name Conflicts** (2 κλάσεις με το ίδιο όνομα)
Μπορούμε να ορίσουμε δύο κλάσεις με το ίδιο όνομα σε διαφορετικά πακέτα, ώστε να αποφευχθεί η σύγκρουση ονόματος. **Τα πακέτα είναι η λύση για την αποφυγή συγκρούσεων στην ονοματολογία των κλάσεων.**
 - **Reusability** (επαναχρησιμοποίηση κώδικα)
Κατά την ανάπτυξη ενός προγράμματος στη java, συχνά αισθανόμαστε ότι υπάρχουν λίγα πράγματα που γράφουμε ξανά και ξανά στον κώδικα μας. Χρησιμοποιώντας πακέτα, μπορείτε να βάλετε μέσα τις κλάσεις σας, και κάθε φορά που πρέπει να εκτελέσετε την ίδια εργασία, απλώς εισαγάγετε (**import**) εκείνο το πακέτο και χρησιμοποιήστε την κλάση.
 - **Access Control** (Έλεγχος πρόσβασης)
Απόκρυψη των κλάσεων (ή/και μεθόδων) ενός πακέτου, αν δεν θέλουμε άλλες κλάσεις (εκτός πακέτου) να έχουν πρόσβαση σε αυτές.
 - **Better Organization & efficiency** (Καλύτερη οργάνωση των κλάσεων & αποδοτικότητα)
Σε μεγάλες έργα (Java Projects) όπου έχουμε μεγάλο πλήθος κλάσεων, είναι καλό (πρέπει) πάντοτε να ομαδοποιούνται οι παρόμοιοι τύποι κλάσεων σε ένα «νόημα» με το όνομα του πακέτου, ώστε να μπορεί να οργανωθεί καλύτερα ο κώδικας και να γίνεται πιο εύκολος ο εντοπισμός και η χρήση των κλάσεων όταν χρειάζεται, πράγμα που βελτιώνει την αποδοτικότητα-αποτελεσματικότητα.



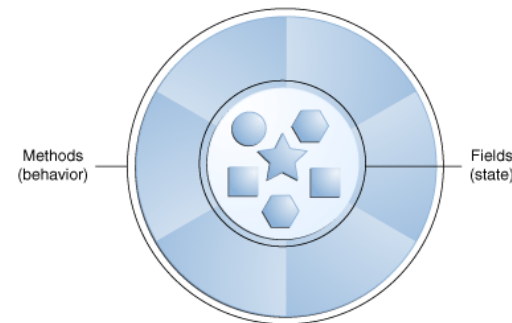
OOP

- **Object-oriented programming (OOP)** is a programming paradigm based on the concept of "objects", which can contain data, in the form of **fields** (often known as *attributes*), and code, in the form of procedures (often known as *methods*).
 - From Wikipedia



Αντικείμενο (Object)

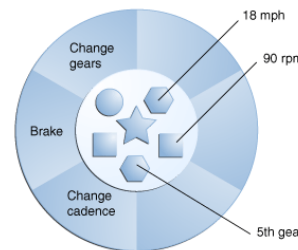
- Κεντρικό στοιχείο του Αντικειμενοστραφούς Προγραμματισμού είναι η χρήση αντικειμένων (Objects). Τα αντικείμενα αυτά, είναι παρόμοια με αντικείμενα πραγματικού κόσμου, όπως για παράδειγμα μπορεί να είναι το αυτοκίνητο, ο σκύλος, το ποδήλατο, η τηλεόραση κλπ.
- Τα αντικείμενα στον πραγματικό κόσμο μοιράζονται δύο χαρακτηριστικά: Όλα έχουν **κατάσταση** και **συμπεριφορά**. Τα σκυλιά έχουν κατάσταση (όνομα, χρώμα, ράτσα, ηλικία, εάν είναι πεινασμένα κτλ.) και συμπεριφορά (γαβγίζουν, τρέχουν κτλ.).
- Τα αντικείμενα λογισμικού (**software objects**) είναι εννοιολογικά παρόμοια με τα αντικείμενα πραγματικού κόσμου, δηλαδή και αυτά χαρακτηρίζονται από:
 - **κατάσταση (State)**
(στοιχεία/δεδομένα που υπαγορεύουν την εμφάνιση, την κατάσταση ή άλλα ποιοτικά χαρακτηριστικά του αντικειμένου)
 - **συμπεριφορά (Behavior)**
(Τα πράγματα που μπορεί να κάνει)



A software object

Αντικείμενο (Object) και Ενθυλάκωση (Encapsulation)

- Ένα αντικείμενο αποθηκεύει την κατάσταση του σε **πεδία** και εκθέτει τη συμπεριφορά του μέσω **μεθόδων**. Οι **μέθοδοι** λειτουργούν στην εσωτερική κατάσταση ενός αντικειμένου και χρησιμεύουν ως ο **κύριος μηχανισμός** για την επικοινωνία του αντικειμένου με άλλα αντικείμενα.
- Η απόκρυψη της εσωτερικής κατάστασης και η απαίτηση ότι όλες οι αλληλεπιδράσεις θα πραγματοποιούνται μέσω των **μεθόδων** ενός αντικειμένου είναι γνωστή ως **ενθυλάκωση (Encapsulation)** - μια θεμελιώδης αρχή του αντικειμενοστραφούς προγραμματισμού.
- Για παράδειγμα στο αντικείμενο ποδήλατο, με την απόδοση της κατάσταση του (*current speed, current pedal cadence, and current gear*) και παροχή μεθόδων για την αλλαγή αυτής της κατάστασης, το αντικείμενο παραμένει υπό τον έλεγχο του τρόπου με τον οποίο ο εξωτερικός κόσμος μπορεί να το χρησιμοποιήσει. Για παράδειγμα, αν το ποδήλατο έχει μόνο 6 ταχύτητες, μια μέθοδος αλλαγής ταχυτήτων μπορεί να απορρίψει οποιαδήποτε τιμή μικρότερη από 1 ή μεγαλύτερη από 6.



A bicycle modeled as a software object.

Ενθυλάκωση (Encapsulation)

- **Ενθυλάκωση δεδομένων (*data encapsulation*)** καλείται η ιδιότητα που προσφέρουν οι κλάσεις να «κρύβουν» τα ιδιωτικά δεδομένα (πεδία) τους από άλλες κλάσεις (δηλ. τον «έξω κόσμο») και να εξασφαλίζουν πως μόνο μέσω των δημόσιων μεθόδων τους θα μπορούν αυτά να προσπελαστούν/αλλάξουν.
- Η ενθυλάκωση επιτυγχάνεται με την χρήση **τροποποιητών πρόσβασης (Access Modifiers)** ως εξής:
 - κατασκευή ιδιωτικών (***private***) πεδίων σε μια κλάση
(Αν ένα πεδίο έχει δηλωθεί ιδιωτικό, δεν μπορεί να υπάρξει πρόσβαση σε αυτό έξω από την κλάση)
 - παροχή πρόσβασης στα πεδία αυτά μόνο μέσω δημόσιων (***public***) μεθόδων.
- **Απόλυτος έλεγχος στον τρόπο πρόσβασης και τροποποίησης των πεδίων-μεταβλητών της κλάσης (δηλ. δεδομένων).** Τα πεδία μπορούν να γίνουν μόνο για ανάγνωση (***read-only***) ή μόνο για εγγραφή (***write-only***).
 - Στην κλάση **User** (επόμενο Παράδειγμα) καταργώντας την μέθοδο **getPassword** το πεδίο **password** είναι μόνο για εγγραφή.



Παράδειγμα

```
1 public class User {
2
3     private String username;
4     private String password;
5     private String fullname;
6
7     public User(String username, String password, String fullname) {
8         this.username = username;
9         this.password = password;
10        this.fullname = fullname;
11    }
12
13    public void setUsername(String username) {
14        this.username = username;
15    }
16    public void setPassword(String password) {
17        this.password = password;
18    }
19    public void setFullname(String fullname) {
20        this.fullname = fullname;
21    }
22
23    public String getUsername() {
24        return username;
25    }
26    public String getPassword() {
27        return password;
28    }
29    public String getFullname() {
30        return fullname;
31    }
32
33    @Override
34    public String toString() {
35        return "User [username=" + username + ", password=" + password + ", fullname=" + fullname + "]";
36    }
37
38 }
```

Ιδιωτικά πεδία (μεταβλητές στιγμιοτύπου)

Κατασκευαστής

Μέθοδοι εκχώρησης τιμών στα πεδία (Setters ή Mutator Methods)

Μέθοδοι προσπέλασης τιμών των πεδίων (Getters ή Accessor Methods)

Μέθοδος (Ερώτηση: το όνομα της είναι τυχαίο;)



Μερικά οφέλη

Ο συνδυασμός/δομή κώδικα σε μεμονωμένα αντικείμενα λογισμικού (java κλάσεις) παρέχει πολλά οφέλη, μεταξύ των οποίων:

- **Modularity**

Ο πηγαίος κώδικας μπορεί να γραφτεί και να διατηρηθεί ανεξάρτητα από τον πηγαίο κώδικα άλλων αντικειμένων (κλάσεων).

- **Information-Hiding**

Με την αλληλεπίδραση μόνο με τις (δημόσιες) μεθόδους ενός αντικειμένου, τα πεδία (δεδομένα) και οι λεπτομέρειες υλοποίησης του παραμένουν κρυμμένες από τον έξω κόσμο.

- **Code re-use**

Εάν ένα αντικείμενο (java κλάση) υπάρχει ήδη (πχ γραμμένο από άλλο προγραμματιστή), μπορείτε να χρησιμοποιήσετε στο πρόγραμμά σας. Αυτό επιτρέπει σε ειδικούς στον εκάστοτε τομέα να υλοποιούν (implement/test/debug) κώδικα τον οποίο μπορείτε να εμπιστευτείτε και να χρησιμοποιήσετε στον κώδικά σας (πχ η κλάση Scanner).

- **Pluggability and Debugging Ease**

Εάν ένα συγκεκριμένο αντικείμενο αποδειχθεί προβληματικό, μπορείτε απλά να το αφαιρέσετε από την εφαρμογή σας και να συνδέσετε ένα διαφορετικό αντικείμενο ως αντικατάστασή του. Αυτό είναι ανάλογο με τον καθορισμό μηχανικών προβλημάτων στον πραγματικό κόσμο. Αν σπάσει ένας ιμάντας, τον αντικαθιστούμε, όχι ολόκληρη τη μηχανή.



Παράδειγμα – Χρήση της User

```
User.java  TestUser.java ✕
1 public class TestUser {
2
3     public static void main(String args[]) {
4         User user1 = new User("jdoe", "12345", "John Doe");
5         System.out.println("User: " + user1.toString());
6         System.out.println("My password is: " + user1.getPassword());
7         System.out.println("I will change password now...");
8         user1.setPassword("777757");
9         System.out.println("User: " + user1.toString());
10    }
11
12 }
13 |
```

Problems @ Javadoc Declaration Console ✕

<terminated> TestUser [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Apr 23, 2018, 4:10:26 AM)

```
User: User [username=jdoe, password=12345, fullname=John Doe]
My password is: 12345
I will change password now...
User: User [username=jdoe, password=777757, fullname=John Doe]
```



Ασκήσεις



Άσκηση 1

1. Δίνεται η κλάση **Student**. Να γράψετε ένα πρόγραμμα java με όνομα **Lab3Exercise1** το οποίο θα ζητάει από τον χρήστη να εισάγει με την σειρά τα παρακάτω στοιχεία:

1. τον αριθμό μητρώου του
2. το όνομα του
3. το επώνυμο του
4. το έτος γέννησης

Στη συνέχεια, θα χρησιμοποιεί τα στοιχεία που εισήγαγε ο χρήστης για να δημιουργήσει ένα αντικείμενο της κλάσης **Student** και τέλος θα τυπώνει τα στοιχεία μέσω της μεθόδου **toString()** του αντικειμένου **Student** που δημιούργησε.

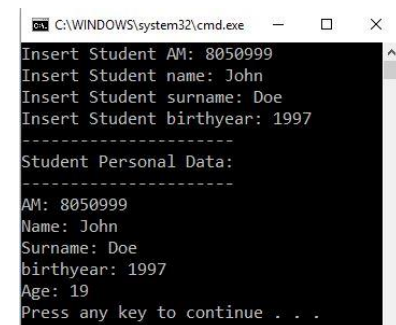
Σημείωση: Να δοκιμάσετε να λύσετε την άσκηση με δυο τρόπους:

- α) Χρησιμοποιώντας τον κατασκευαστή που δέχεται ορίσματα.
- β) Χρησιμοποιώντας τον κατασκευαστή χωρίς ορίσματα.

Παράδειγμα Εκτέλεσης:

Για είσοδο των παρακάτω στοιχείων:

1. 8050999
2. John
3. Doe
4. 1997



```
C:\WINDOWS\system32\cmd.exe
Insert Student AM: 8050999
Insert Student name: John
Insert Student surname: Doe
Insert Student birthyear: 1997
-----
Student Personal Data:
-----
AM: 8050999
Name: John
Surname: Doe
birthyear: 1997
Age: 19
Press any key to continue . . .
```



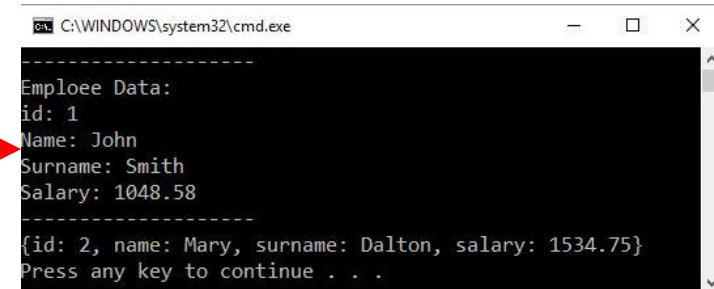
Άσκηση 2

2. Δίνεται η παρακάτω κλάση **TestEmployee**

```
public class TestEmployee {  
    public static void main(String args[]) {  
  
        Employee employee1 = new Employee (1, "John", "Smith", 1048.58);  
        Employee employee2 = new Employee();  
  
        employee2.setId(2);  
        employee2.setName("Mary");  
        employee2.setSurname("Dalton");  
        employee2.setSalary(1534.75);  
  
        System.out.println(employee1.toString());  
        System.out.println("{id: " + employee2.getId() + ", "  
                                + "name: " + employee2.getName() + ", "  
                                + "surname: " + employee2.getSurname() + ", "  
                                + "salary: " + employee2.getSalary() + "}");  
    }  
}
```

Ζητείται να γράψετε την κλάση **Employee** έτσι ώστε να μπορεί να εκτελεστεί σωστά το παραπάνω πρόγραμμα.

Εκτέλεση προγράμματος **TestEmployee**



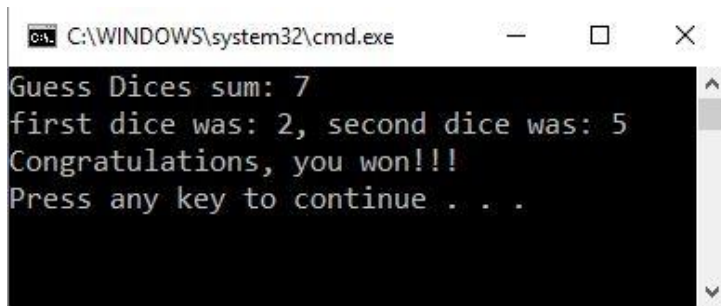
```
C:\WINDOWS\system32\cmd.exe  
-----  
Employee Data:  
id: 1  
Name: John  
Surname: Smith  
Salary: 1048.58  
-----  
{id: 2, name: Mary, surname: Dalton, salary: 1534.75}  
Press any key to continue . . .
```



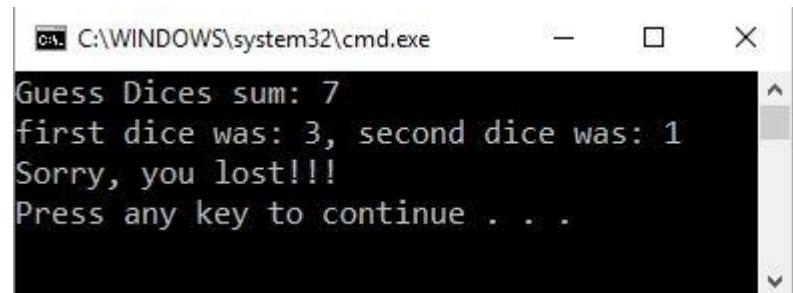
Άσκηση 3

3. Δίνεται η κλάση **Dice**. Να γράψετε ένα πρόγραμμα java με όνομα **Lab3Exercise3** το οποίο θα ζητάει από τον χρήστη να μαντέψει το άθροισμα του αποτελέσματος μιας ζαριάς (1 ζαριά = 2 ζάρια). Πιο συγκεκριμένα, θα ζητάει από τον χρήστη να εισάγει έναν ακέραιο αριθμό και στην συνέχεια θα χρησιμοποιεί την μέθοδο **roll** (2 φορές) για «ρίξει» το ζάρι και θα υπολογίζει το άθροισμα των 2 ζαριών. Στην συνέχεια θα κάνει τα εξής:
1. Θα τυπώνει τα αποτελέσματα του 1^{ου} και του 2^{ου} ζαριού
 2. Σε περίπτωση που το άθροισμα είναι ίσο με τον ακέραιο που εισήγαγε ο χρήστης τότε θα τυπώνει το μήνυμα: **Congratulations, you won!!!**. Διαφορετικά, θα τυπώνει το μήνυμα: **Sorry, you lost!!!**

Παραδείγματα Εκτέλεσης:



```
C:\WINDOWS\system32\cmd.exe
Guess Dices sum: 7
first dice was: 2, second dice was: 5
Congratulations, you won!!!
Press any key to continue . . .
```



```
C:\WINDOWS\system32\cmd.exe
Guess Dices sum: 7
first dice was: 3, second dice was: 1
Sorry, you lost!!!
Press any key to continue . . .
```



Άσκηση 4

4. Δίνεται η κλάση **Dice**. Να γράψετε ένα πρόγραμμα java με όνομα **Lab3Exercise4** το οποίο θα ζητάει από τον χρήστη να μαντέψει το άθροισμα του αποτελέσματος μιας ζαριάς (1 ζαριά = 2 ζάρια). Πιο συγκεκριμένα, ο χρήστης θα ξεκινάει το παιχνίδι με 5 πόντους, κάθε φορά που μαντεύει σωστά θα κερδίζει 1 πόντο, ενώ για κάθε λάθος που κάνει θα του αφαιρείται 1 πόντος. Το παιχνίδι θα τελειώνει όταν ο χρήστης έχει 0 πόντους ή όταν εισάγει (ακέραιο) αριθμό που δεν ανήκει στο διάστημα [2, 12].

Παράδειγμα Εκτέλεσης:

```
C:\WINDOWS\system32\cmd.exe
Guess Dices sum: 7
first dice was: 1, second dice was: 6
you won 1 point, now you have: 6
Guess Dices sum: 8
first dice was: 2, second dice was: 2
Sorry, you lost 1 point, now you have: 5
Guess Dices sum: 4
first dice was: 4, second dice was: 3
Sorry, you lost 1 point, now you have: 4
Guess Dices sum: 5
first dice was: 1, second dice was: 2
Sorry, you lost 1 point, now you have: 3
Guess Dices sum: 7
first dice was: 1, second dice was: 2
Sorry, you lost 1 point, now you have: 2
Guess Dices sum: 2
first dice was: 4, second dice was: 5
Sorry, you lost 1 point, now you have: 1
Guess Dices sum: 12
first dice was: 2, second dice was: 3
Sorry, you lost 1 point, now you have: 0
Sorry, you lost all points, bye!
Press any key to continue . . .
```

