

Evidence for Implementation and Testing Unit.

Gabriela Lewandowska

E15

18 September 2017

I.T 1- Demonstrate one example of encapsulation that you have written in a program.

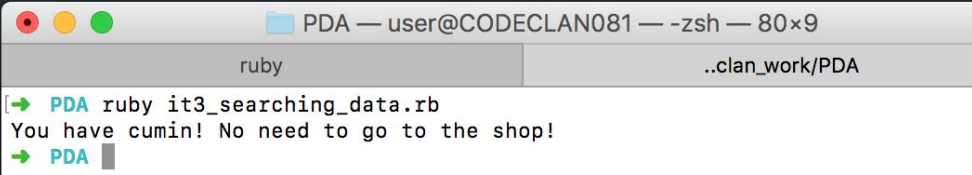
```
public class Tile {  
    int number;  
    ArrayList<Player> players;  
  
    public Tile(int number) {  
        this.number = number;  
        this.players = new ArrayList<>();  
    }  
  
    public int getNumber() { return number; }  
  
    public ArrayList<Player> getPlayers() { return players; }
```

I.T 2 - Inheritance in a program.

```
1  class Animal
2  |  def walk
3      return "I'm walking!"
4  |  end
5  end
6
7  class Cat < Animal
8  |  def meow
9      return "Meow!"
10 |  end
11 end
12
13 maurice = new Cat
14 maurice.walk
15 maurice.meow
```

I.T 3 - Example of searching and sorting data.

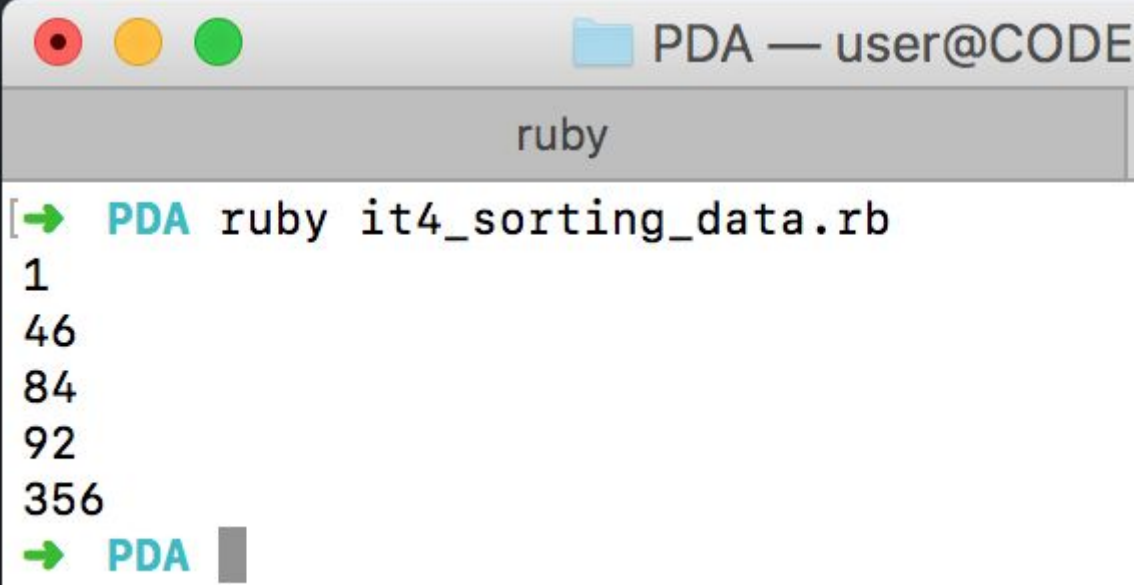
```
it3_searching_data.rb
1  cupboard = ["chilli", "thyme", "oregano", "basil", "cumin"]
2  needed_ingredient = "cumin"
3
4  if cupboard.include?(needed_ingredient)
5    puts "You have " + needed_ingredient + "! No need to go to the shop!"
6  else
7    puts "You don't have " + needed_ingredient + "! You need to buy it!"
8  end
9
```



```
PDA — user@CODECLAN081 — -zsh — 80x9
ruby
..clan_work/PDA
[→ PDA ruby it3_searching_data.rb
You have cumin! No need to go to the shop!
→ PDA
```

I.T 4 - Example of an array, a function that uses an array and the result

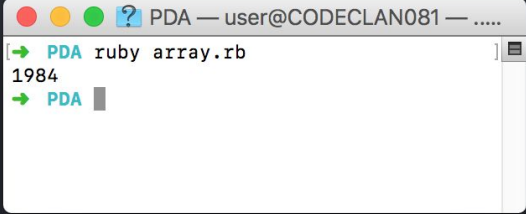
```
it4_sorting_data.rb
1  array = [1, 46, 84, 92, 356]
2  sorted_array = array.sort
3  puts sorted_array
4
```



```
PDA — user@CODE
ruby
[→ PDA ruby it4_sorting_data.rb
1
46
84
92
356
→ PDA
```

I.T 5 - Example of a hash, a function that uses a hash and the result

```
array.rb
1 books = ["1984", "War and Peace", "Crime and Punishment", "Pride and Prejudice"]
2
3 def find_the_odd_one_out(array)
4   for book in array
5     return book if book.include?("and") == false
6   end
7 end
8
9 puts find_the_odd_one_out(books)
10
```



The terminal window shows the command `PDA ruby array.rb` being executed, resulting in the output `1984`.

I.T 7 - Demonstrate the use of polymorphism in a program

```
public class Event {
    Sport sportType;
    int maximumNumberOfCompetitors;
    ArrayList<Competitor> competitors;
    ArrayList<Competitor> rankedCompetitors;
    MedalTable medalTable;

    public Event(Sport sportType, int maximumNumberOfCompetitors) {
        this.sportType = sportType;
        this.medalTable = new MedalTable();
        this.maximumNumberOfCompetitors = maximumNumberOfCompetitors;
        this.competitors = new ArrayList<>();
        rankedCompetitors = new ArrayList<>();
    }

    public Sport getEventType() { return sportType; }

    public int getMaximumNumberOfCompetitors() { return maximumNumberOfCompetitors; }

    public ArrayList<Competitor> getCompetitors() { return competitors; }

    public MedalTable getMedalTable() { return medalTable; }

    public ArrayList<Competitor> getRankedCompetitors() { return rankedCompetitors; }

    public void addCompetitor(Competitor competitor){
        if(this.competitors.size() < this.maximumNumberOfCompetitors) {
            this.competitors.add(competitor);
        }
    }

    public void assignScoreToCompetitors(){
        Random random = new Random();
        for(int i = 0; i < this.competitors.size(); i++){
            this.competitors.get(i).setScore(random.nextInt(100));
        }
    }
}
```

```
public abstract class Competitor implements Comparable<Competitor> {
    private Country country;
    private int score;
    private HashMap<MedalType,Integer> medal;

    public Competitor(Country country) {
        this.country = country;
        this.score = 0;
        this.medal = new HashMap();
        this.medal.put(MedalType.GOLD, 0);
        this.medal.put(MedalType.SILVER, 0);
        this.medal.put(MedalType.BRONZE, 0);
    }
}
```

```
public class Athlete extends Competitor {  
    private String name;  
  
    public Athlete(String name, Country country) {  
        super(country);  
        this.name = name;  
    }  
  
    public String getName() { return name; }  
}
```

```
public class Team extends Competitor {  
    private ArrayList<Athlete> teamMembers;  
  
    public Team(Country country) {  
        super(country);  
        teamMembers = new ArrayList<>();  
    }  
  
    public ArrayList<Athlete> getTeamMembers() {  
        return teamMembers;  
    }  
}
```