

# Question Sheet 1

Name: \_\_\_\_\_ ID: \_\_\_\_\_

These questions do not have a formal, definitive answer. They are meant to be food for thoughts. Feel free to seek answers on browsing the Internet, talking to other software developers or reading books.

## 1 UML

1. UML is a graphical language. It relies on diagrams to describe software. There are two kinds of diagrams: static and dynamic. Which are the static diagrams, and which are the dynamic ones?
2. There are a number of criticisms to UML. List a number of these criticisms.

## 2 Use Cases

1. An *actor* is an entity that has a role in the system, but that exists outside the system. An actor may be a human being, but it may also be an electronic device, or a external software. For instance, in a on-line shop we may have the following actors:
  - The customer who buys something through the web.
  - The employee who will process the order.
  - The banking system, that receives the value of the product via a credit card transaction.

Who are the actors in an ATM system? Describe the actors using the notation of use case diagrams.

2. Use cases are descriptions of the behavior of a system, given an external stimulus. Use cases are normally described textually. For instance, the use case *check-out item* enumerates the steps that a salesperson must follow to sell a product to a customer:
  - (a) Customer brings item to the counter.
  - (b) Salesperson scans item's bar code.
  - (c) System finds item's informations given the bar code.
  - (d) System sounds a beep indicating that the item was found.
  - (e) Etc, etc, . . .

What are the use cases that you would use to describe an ATM?

3. What are the actors involved in the ATM's use case *withdraw money*?

4. Describe the use case *withdraw money* textually.

5. Are there exceptional situations that may happen in the use case *withdraw money*? How these exceptional cases can be handled?

6. There are two ways to reuse notation in the use cases diagram:

- A use case  $c_1$  includes another use case  $c_2$  if the behavior of  $c_2$  is part of the behavior of  $c_1$ .
- A use case  $c_1$  extends a use case  $c_2$  if  $c_1$  adds a new sequence of steps onto  $c_2$ . We normally use extends to model exceptional conditions.

What situations in the list of use cases for the ATM system can be modeled via extension? Which situations are better modeled using inclusion?

7. Use case diagrams are important documentation tools, but they are not really design tools. Do you agree or disagree with this sentence and why?

### 3 Class Diagrams

1. Class diagrams use some symbols to specify the visibility of object fields and methods. Which levels of visibility are defined, and how they correspond to Java?
2. Classes and interfaces have different representations. What is the notational difference between these two entities?
3. Classes may associate to each other via inheritances or associations. A class may also implement an interface. Associations are further divided into aggregations and compositions. All these relations are represented via arrows. What is the notation used for each kind of arrow?

4. What is the difference between aggregation and composition?

5. Consider the program below, implemented in C++. The two objects are related through an aggregation. How to change this implementation, so that the objects are related through a simple association?

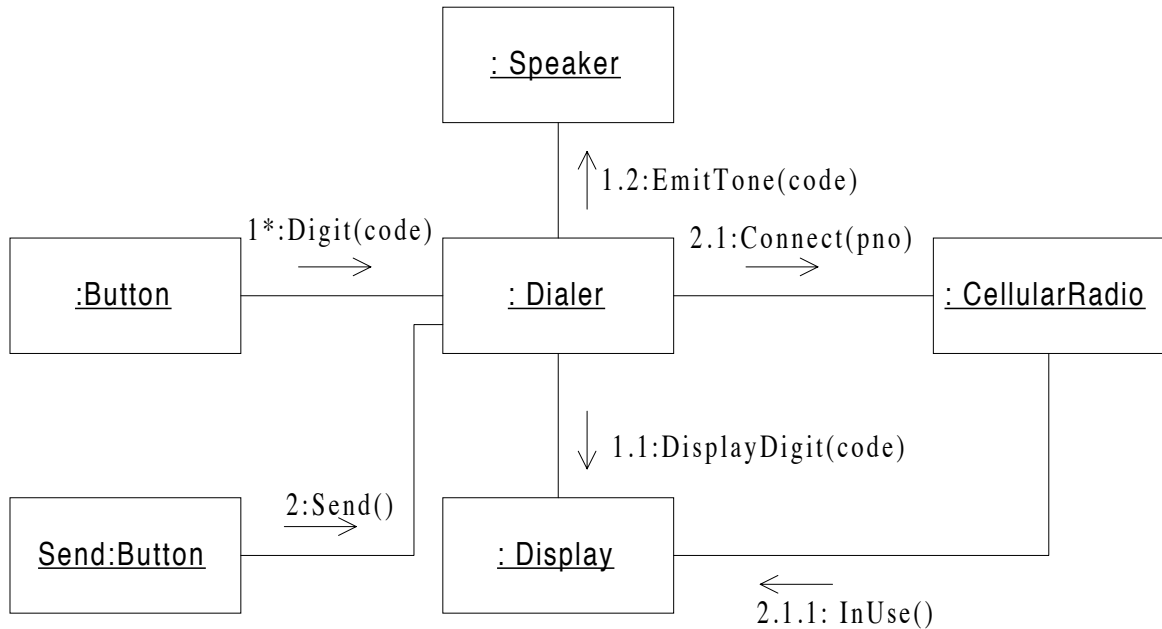
```
#include <iostream>
class Obj1 {
public:
    Obj1() {std::cout << "Creating obj1\n";}
    ~Obj1() {std::cout << "Deleting obj1\n";}
    int p;
};
class Obj2 {
public:
    Obj2() {std::cout << "Creating obj2\n";}
    ~Obj2() {std::cout << "Deleting obj2\n";}
    Obj1 o;
};
int main() {
    Obj2 *o = new Obj2();
    delete o;
}
```

6. How to distinguish associations from compositions in Java?

## 4 Collaboration and Sequence Diagrams

1. Consider the software that controls a cell phone system. This program has buttons to deal call numbers, and a special button send, to start the call. The software has a

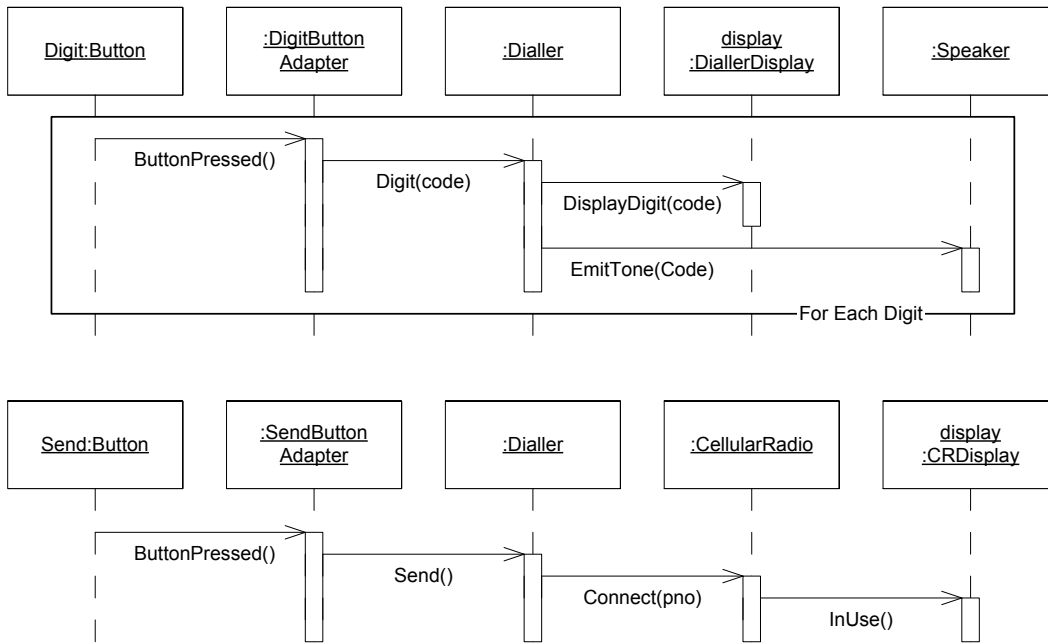
special component called *Dialer*, which receives the digits to call, and, for each one, issues a call sound. This program controls a radio device, that is in charge of sending and receiving information. Moreover, the software controls a microphone, a speaker and a small screen. The iterations between the different objects in this system are described by the following collaboration diagram:



- (a) What is the meaning of the boxes, and what is the meaning of the links between the boxes?
  
- (b) Why are the names in the boxes underlined?
  
- (c) What is the meaning of the colon's?
  
- (d) What is the role of the arrows in this diagram?



4. An expanded version of the collaboration diagram seen before is represented below, as a sequence diagram:



(a) What are the boxes, the underlined names, and the colon's?

(b) What is the meaning of the horizontal arrows?

(c) What about the vertical dashed lines?

(d) What is the meaning of the small white rectangles on the dashed lines?

(e) What is the role of the large rectangle in the upper diagram?