UECS2344 Software Design: Lecture 2

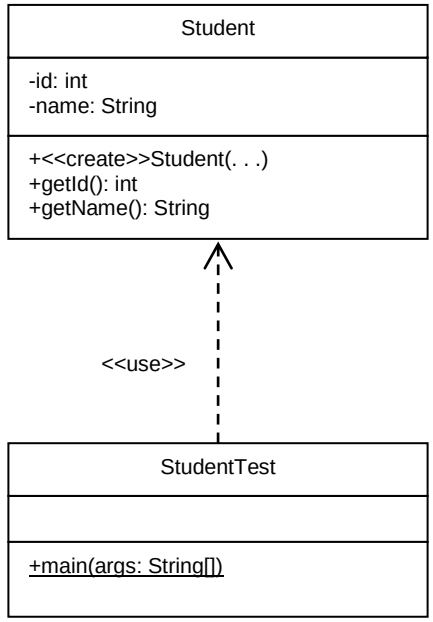
Object-Oriented Paradigm / Approach

Class Diagram and Sequence Diagram

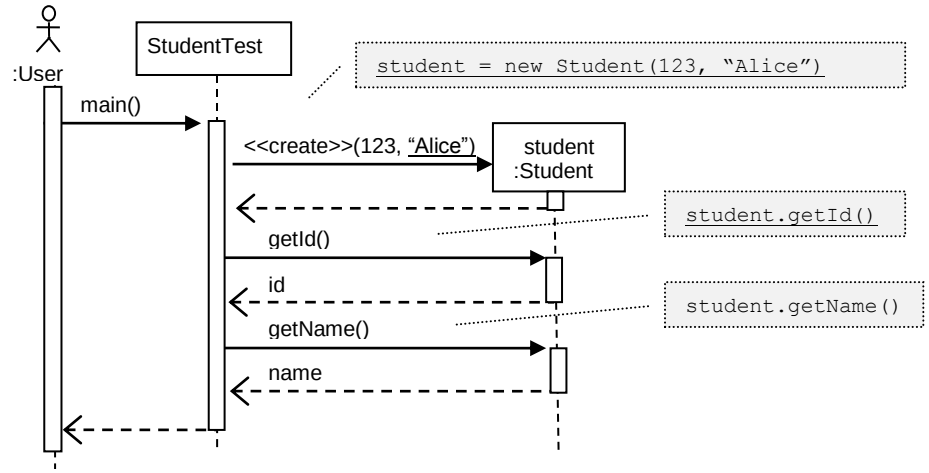
Code Example 1

```
public class Student {
  // instance variables
  private int id;
  private String name;

  public Student(int id, String name) {
    this.id = id;
    this.name = name;
  }
  public int getId() {
    return id;
  }
  public String getName() {
    return name;
  }
}

public class StudentTest {

  public static void main(String[] args) {
    // local variable
    Student student;

    student = new Student(123, "Alice");
    System.out.println("ID: "
                       + student.getId());
    System.out.println("Name: "
                       + student.getName());
  }
}
```

Design Class Diagram



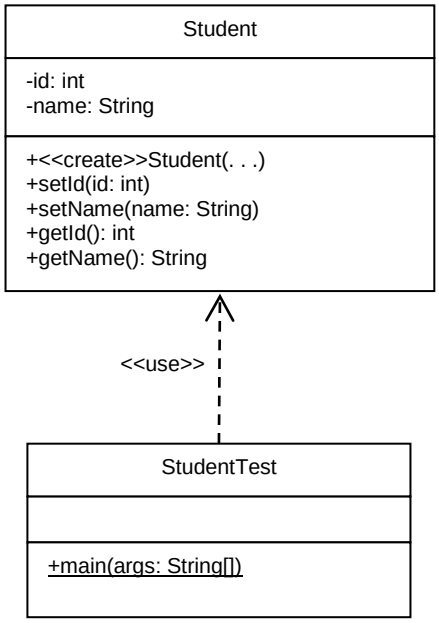
Sequence Diagram



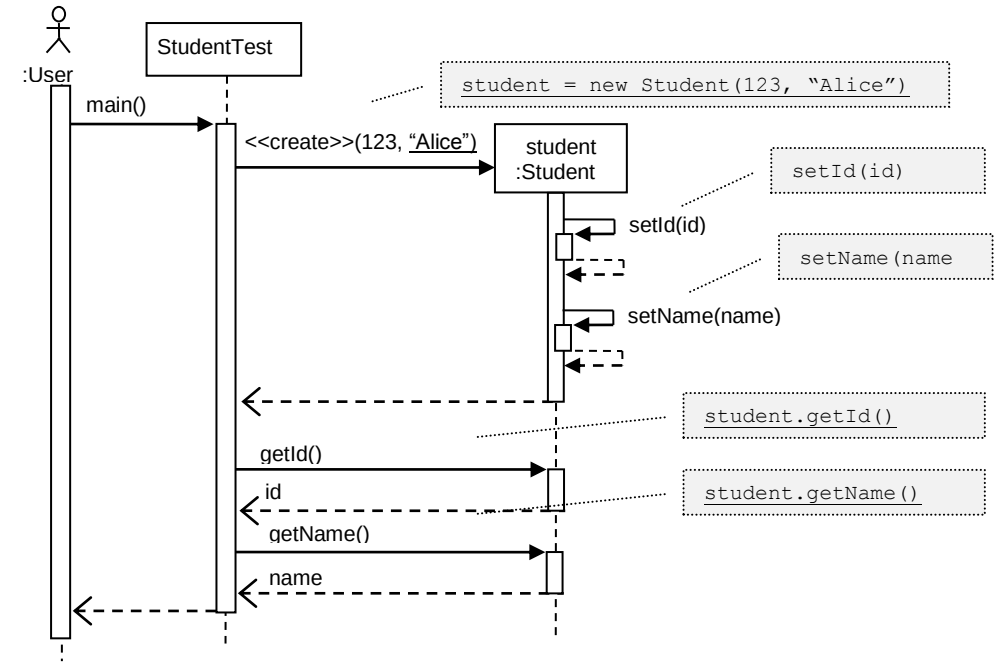
Code Example 2

```
public class Student {
  // instance variables
  private int id;
  private String name;

  public Student(int id, String name) {
    setId(id);
    setName(name);
  }
  public void setId(int id) {
    this.id = id;
  }
  public void setName(String name) {
    this.name = name;
  }
  public int getId() {
    return id;
  }
  public String getName() {
    return name;
  }
}

public class StudentTest {

  public static void main(String[] args) {
    // local variable
    Student student;
    student = new Student(123, "Alice");
    System.out.println("ID: "
                       + student.getId());
    System.out.println("Name: "
                       + student.getName());
  }
}
```

Design Class Diagram



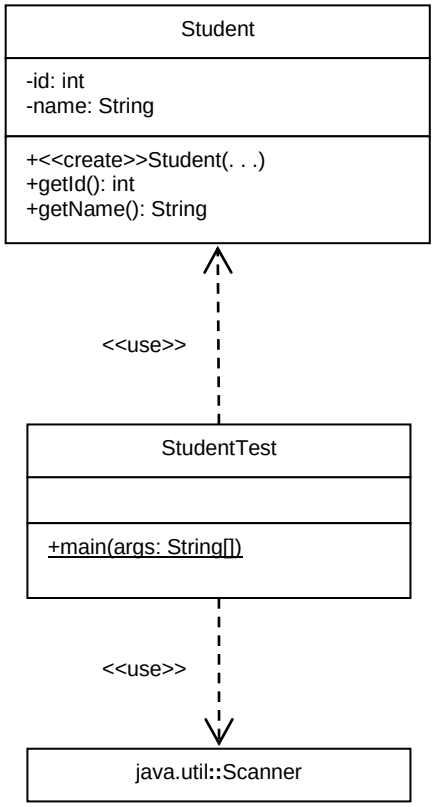
Sequence Diagram



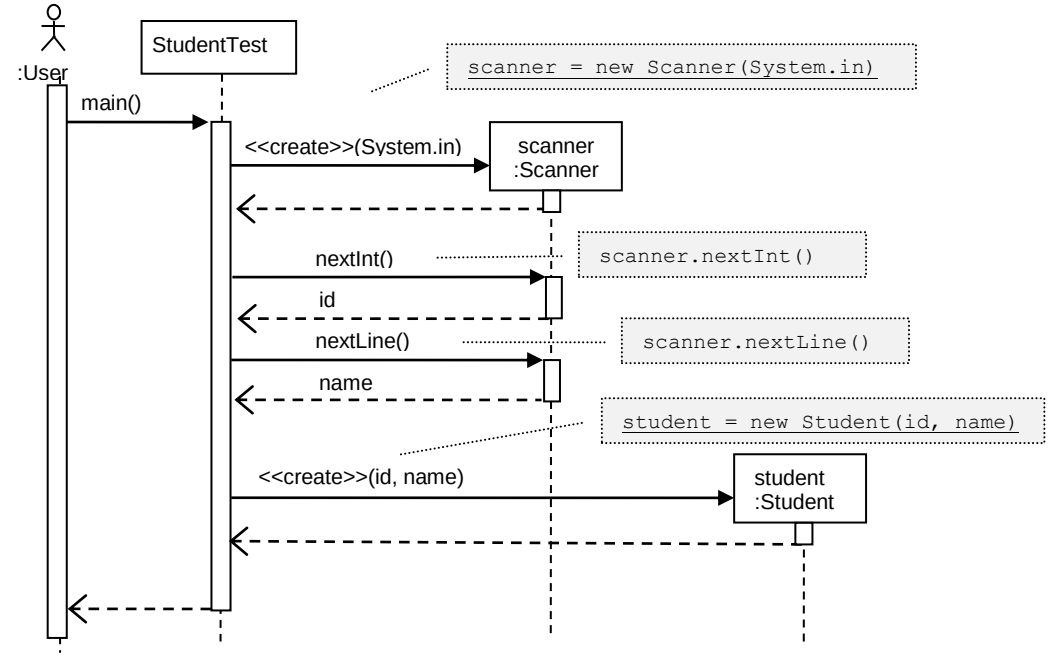
Code Example 3

```
public class Student {
  // instance variables
  private int id;
  private String name;

  public Student(int id, String name) {
    this.id = id;
    this.name = name;
  }
  public int getId() {
    return id;
  }
  public String getName() {
    return name;
  }
}

public class StudentTest {

  public static void main(String[] args) {
    // local variables
    Scanner scanner;
    Student student;

    scanner = new Scanner(System.in);
    System.out.print("Enter id: ");
    int id = scanner.nextInt();
    System.out.print("Enter name: ");
    String name = scanner.nextLine();

    student = new Student(id, name);
  }
}
```

Design Class Diagram

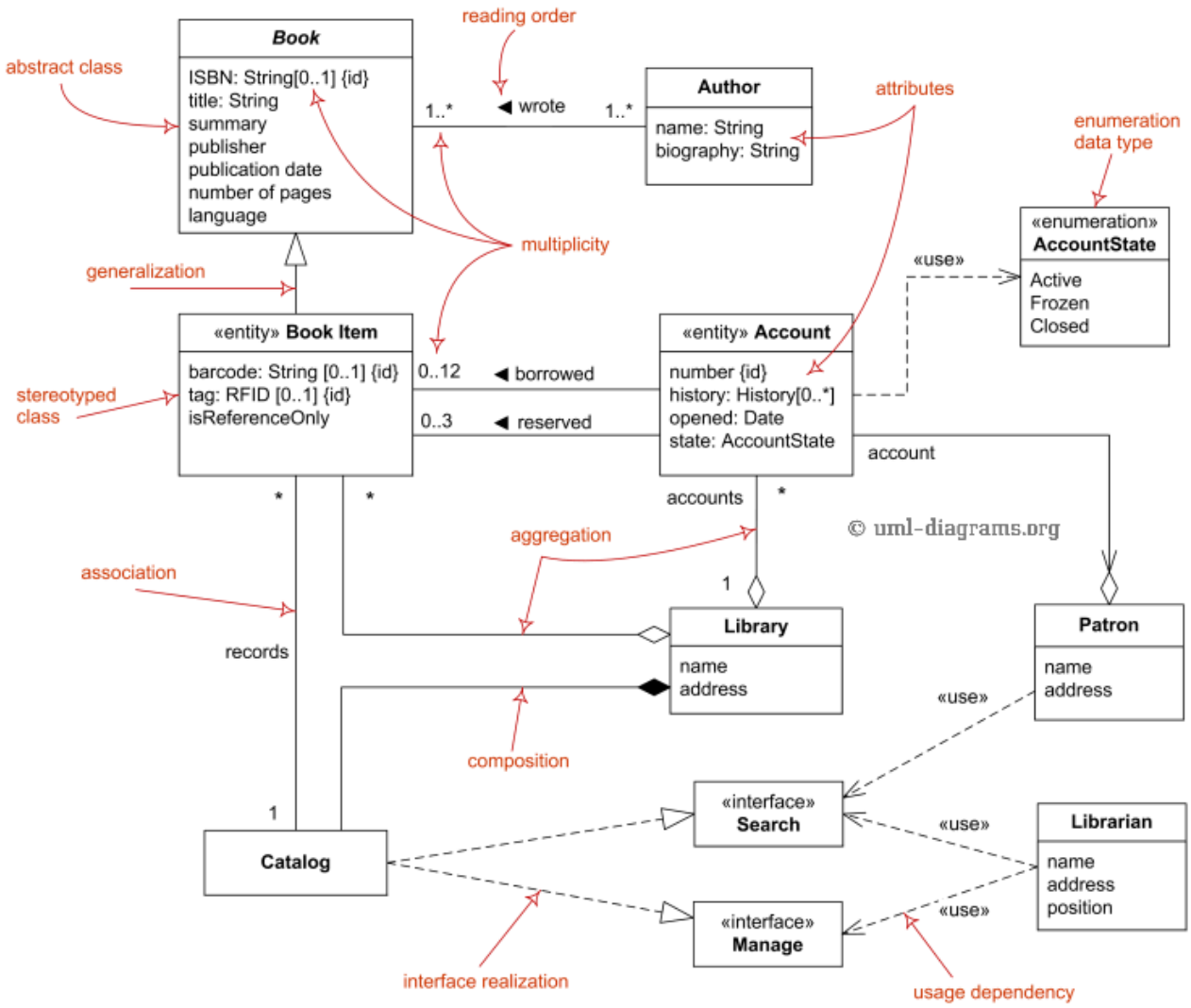


Sequence Diagram



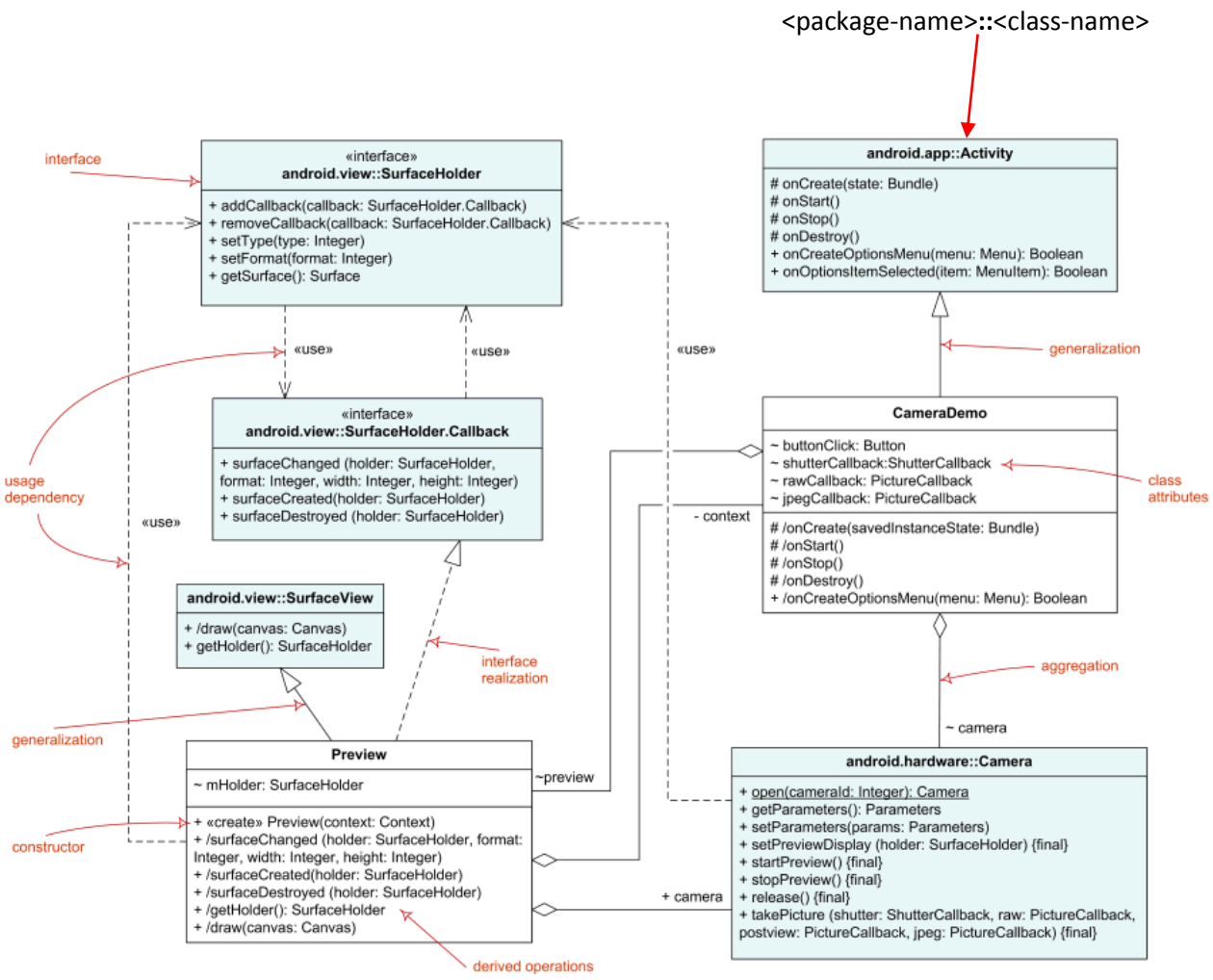
UML 2 Class Diagram Example

(from http://www.uml-diagrams.org/class-diagrams-overview.html)



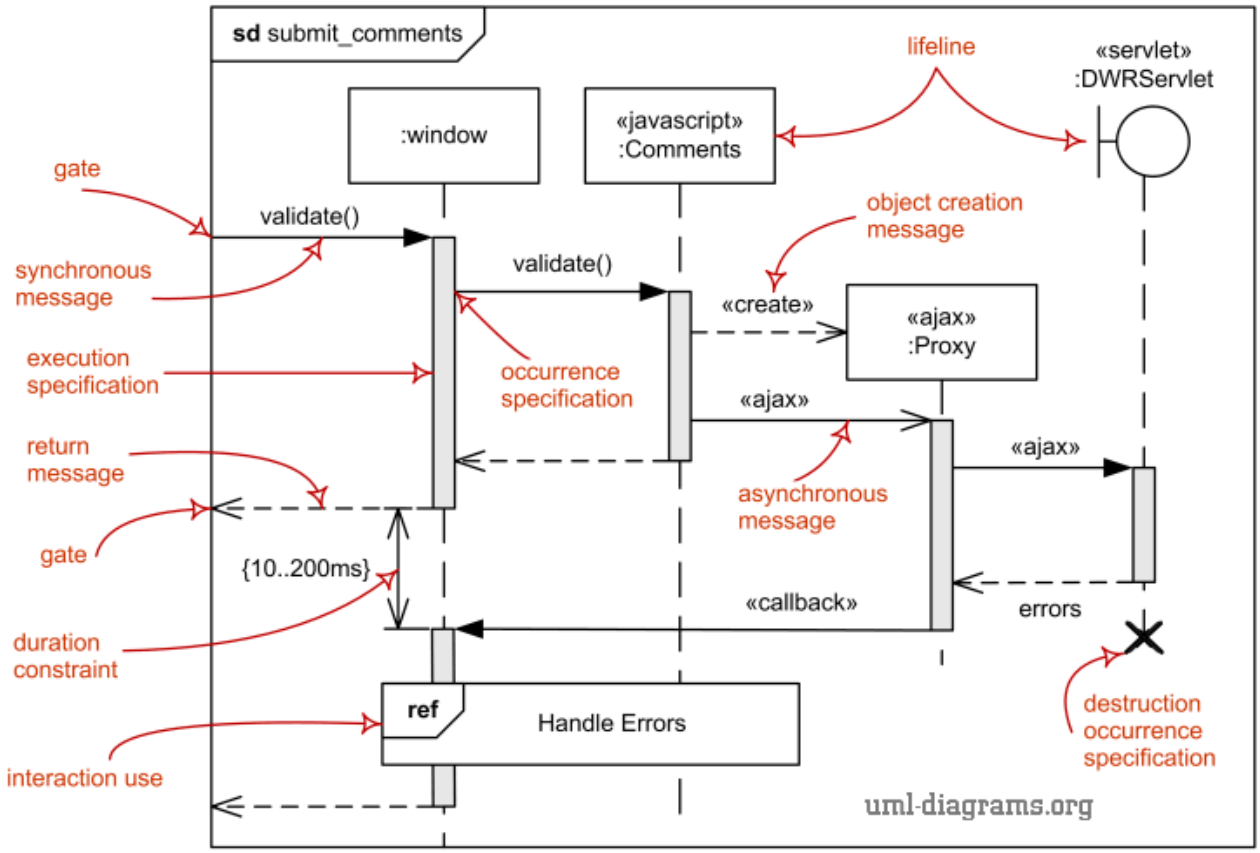
UML 2 Class Diagram Example (detailed)

(from http://www.uml-diagrams.org/class-diagrams-overview.html)

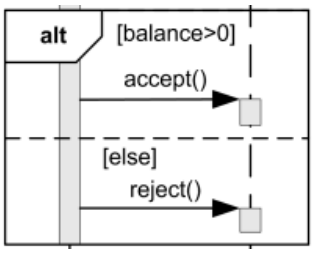


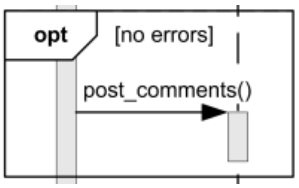
UML 2 Sequence Diagram Examples

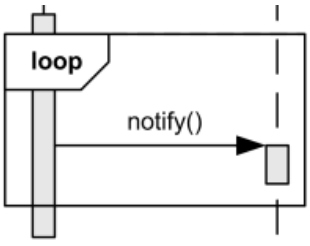
(from http://www.uml-diagrams.org/sequence-diagrams.html)



(From http://www.uml-diagrams.org/sequence-diagrams-combined-fragment.html)

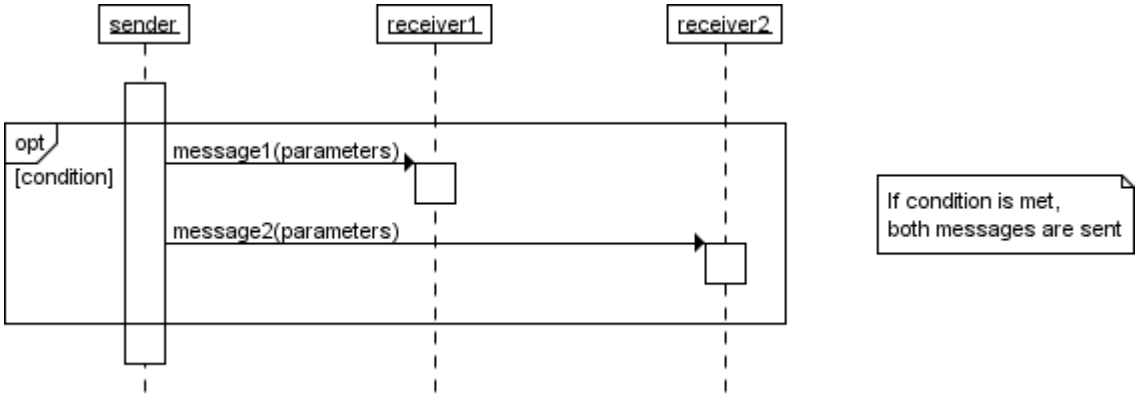




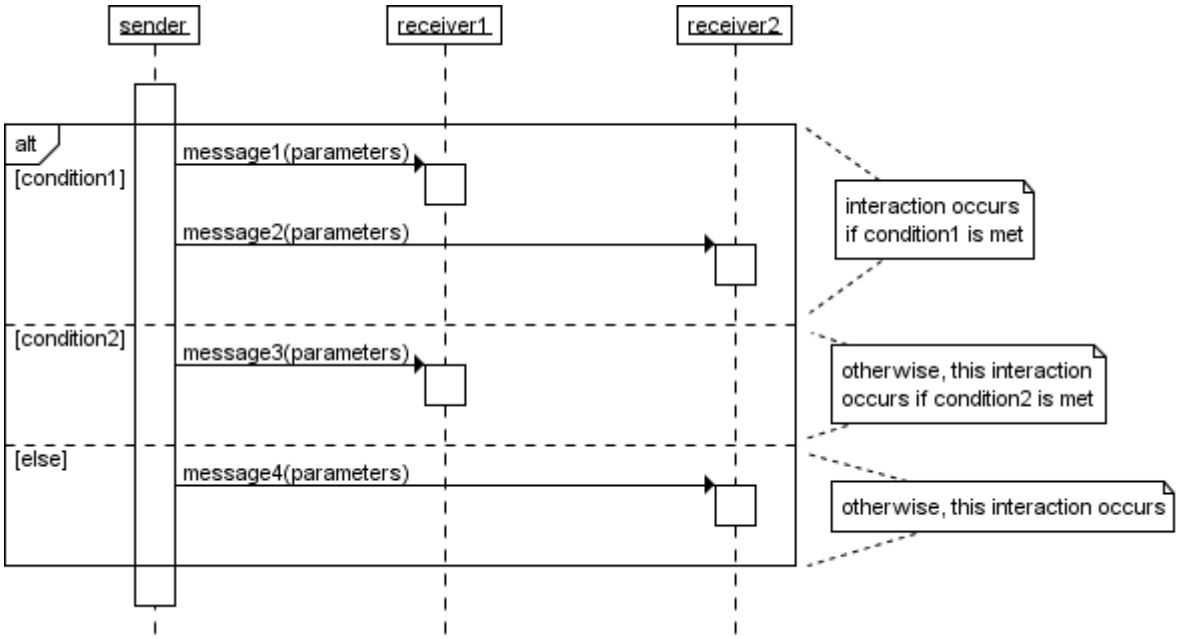


(from http://www.tracemodeler.com/articles/a_quick_introduction_to_uml_sequence_diagrams/)

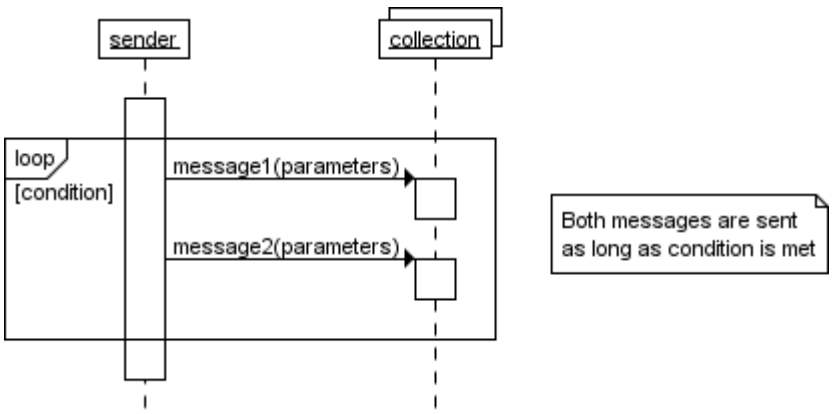
Example - opt



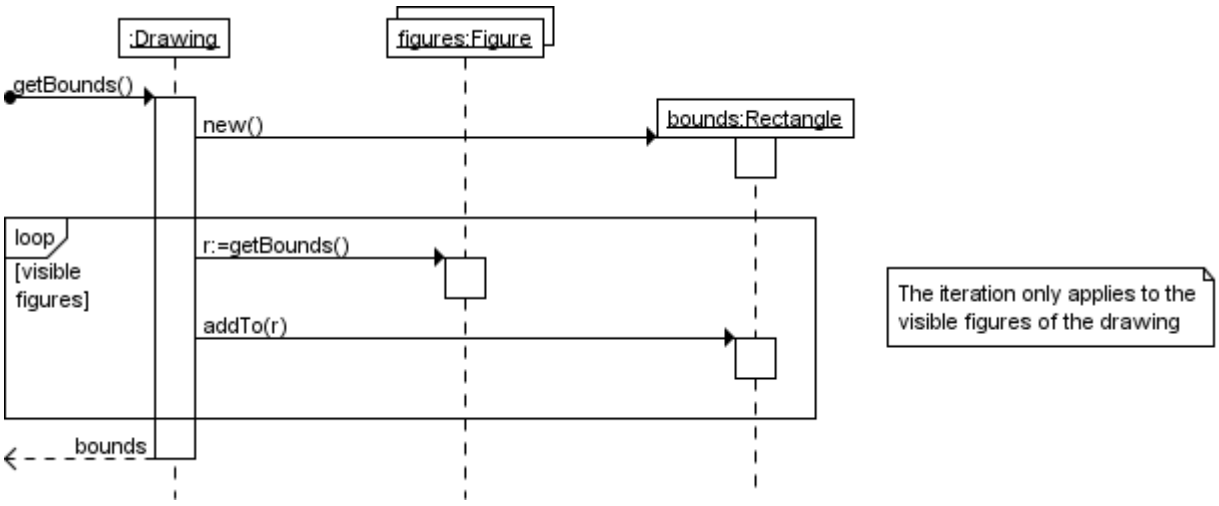
Example - alt



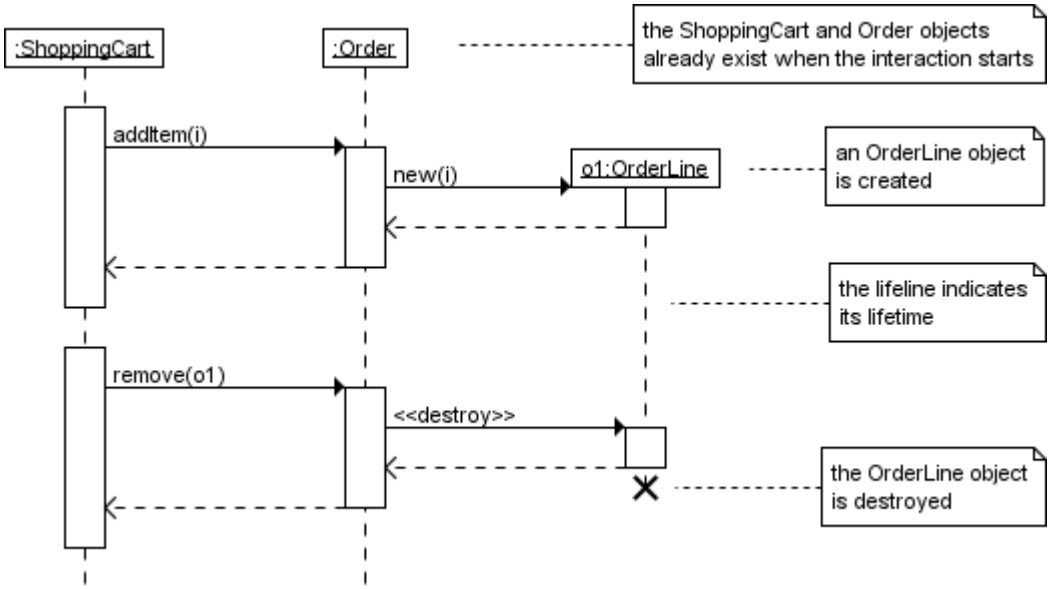
Example - loop



Example - loop



Example – create and destroy



Exercise:

Consider the Java program below which uses the object-oriented approach.

1. Draw a **Design Class Diagram** which shows class names and details of attributes and operations.
2. Draw a **Sequence Diagram** which shows the objects and the messages sent to these objects (showing return messages as well).

Java program using ArrayList of objects

```
// in file Student.java
public class Student {

   private String name;

   public Student() {                  // constructor
          setName("");
   }

   public void setName(String n) {     // setter method
          name = n;
   }

   public String getName() {           // getter method
          return name;
   }
}

-----------------------------------------
// in file StudentApp.java
import java.util.Scanner;
import java.util.ArrayList;

public class StudentApp {

    public static void main(String[] args) {

        ArrayList<Student> studentList = new ArrayList<Student>();
        Scanner scanner = new Scanner(System.in);

        int choice;

        do {
             System.out.println("Do you want to:");
             System.out.println("1. Create new Student object");
             System.out.println("2. Search for a Student object");
             System.out.println("3. Exit");
             System.out.print("Enter your choice (1-3): ");

             choice = scanner.nextInt();
             while (choice < 1 || choice > 3) {
               System.out.println("Invalid choice. Please enter again: ");
               choice = scanner.nextInt();
               // read the enter key after integer input
               String skip = scanner.nextLine();
             }
```

```
            switch(choice) {
                  case 1: addStudent(studentList, scanner); break;
                  case 2: searchStudent(studentList, scanner); break;
                  default: break;
            }

            System.out.println();

        } while (choice != 3);
    }

    public static void addStudent(ArrayList<Student> studentList, Scanner scanner) {

      System.out.print("Enter student name: ");
      String theName = scanner.nextLine();

      Student aStudent = new Student();
      aStudent.setName(theName);
      studentList.add(aStudent);
      System.out.println("Student " + theName + " added");
      System.out.println();
    }

    public static void searchStudent(ArrayList<Student> studentList, Scanner scanner) {
      if (studentList.size() == 0) {
            System.out.println("The list is empty");
      } else {

            System.out.print("Enter name of student: ");
            String theName = scanner.nextLine();

            boolean found = false;
            int i = 0;
            Student theStudent;
            while (i<studentList.size() && !found) {
                  theStudent = studentList.get(i);
                  if (theStudent.getName().equals(theName)) {
                        found = true;
                  } else {
                        i++;
                  }
            }

            if (found)
                  System.out.println("The student is in the list");
            else
                  System.out.println("No student with that name found");

            System.out.println();
      }
    }
}
```