Ex.1

```
public class Paper implements Serializable { ..... /31
	private String title;
	private int nbWords;
	private Author arAuth[];  ..... 1
	private int nbA;          ..... 1
	public Paper(String title, int nbWords) { ..... /2
		this.title = title;
		this.nbWords = nbWords;
		nbA = 0;                  ..... 1
		arAuth = new Author[5];   ..... 1
	}

	public Author getSeniorFaculty(String p) { ..... /10
		int max = 0;  ..... 1
		Faculty oldest = null; ..... 1
		Faculty f;
		for(int i=0; i<nbA; i++) { ..... 1
			if( (arAuth[i].getAge() > max) && (arAuth[i] instanceof Faculty) ) { ...1 + 1
				f = (Faculty) arAuth[i]; ..... 1
				if(f.getPosition().equals(p)) { ..... 1
					max = arAuth[i].getAge(); ..... 1
					oldest = f; ..... 1
				}
			}
		}
		Return oldest; ..... 1
	}
	public int averageAgeOfStudents() { ..... /8
		int j=0,sum = 0; ..... 1 + 1
		for(inti=0; i<nbA; i++) { ..... 1
			if(arAuth[i] instanceof Student) { ..... 1
				sum += arAuth[i].getAge();..... 1
				j++; ..... 1
			}
		}
		if(j != 0) ..... 1
			return sum/j; ..... 1
		return 0;
	}
	public Student[] youngStudent() { ..... /8
		Student[] s = new Student[nbA]; ..... 1
		int j = 0; ..... 1
		int avg = averageAgeOfStudents();

		for(int i=0; i<nbA; i++) { ..... 1
			if(arAuth[i] instanceof Student && arAuth[i].getAge() <= avg) ..... 1 + 1
				s[j++]=(Student) arAuth[i]; ..... 1 + 1
		}
		Return s; ..... 1
	}
	public int getNbWords() { ..... 1
		Return nbWords;
	}
}
```

Ex.2

```
import java.io.*;
publicclass Conference { ..... /28
	private String name;
	private String location;
	private Paper arPap[];..... 1
	private int nbP; ..... 1

	public Conference(String name, String location) { ..... /2
		this.name = name;
		this.location = location;
		arPap = new Paper[50]; ..... 1
		nbP = 0; ..... 1
	}

	public int readPapersFromFile(String filename) throws IOException { ..... /12
		File f = new File(filename); ..... 1
		FileInputStream fs = newFileInputStream(f); ..... 1
		ObjectInputStream os = newObjectInputStream(fs); ..... 1
		int j = 0; ..... 1

		try { ..... 1
			while(true) { ..... 1

				arPap[nbP++] = (Paper) os.readObject();..... 1 + 1 + 1
				j++; ..... 1
			}
		}
		catch (EOFException e){}
		catch (IndexOutOfBoundException e){} ..... 1

		os.close();
		return j; ..... 1
	}

	public int readPapersFromFile(String filename) throws IOException { ..... /12
		File f = new File(filename); ..... 1
		FileInputStream fs = newFileInputStream(f); ..... 1
		ObjectInputStream os = newObjectInputStream(fs); ..... 1

		int j = 0; ..... 1
		Paper p;

		try { ..... 1
			while(true) { ..... 1
				p = (Paper) os.readObject();..... 1
				if (nbP < arPap.length) { ..... 1
					arPap[nbP++] = p; ..... 1 + 1
					j++; ..... 1
				}
			}
		}
		catch (EOFException e){os.close();}

		return j; ..... 1
	}
```

```
	public Paper[] savePapersIntoFile(String filename) throws IOException { ..... /12
		File f = newFile(filename); ..... 1
		FileOutputStream fs = newFileOutputStream(f); ..... 1
		ObjectOutputStream os = new ObjectOutputStream(fs); ..... 1

		Paper p[] = new Paper[50]; ..... 1
		int j=0; ..... 1

		for(int i=0; i<nbP; i++) { ..... 1
			if(arPap[i].getNbWords() >= 500) ..... 1
				os.writeObject(arPap[i]); ..... 1
			else
				p[j++] = arPap[i]; ..... 1 + 1
		}

		os.close();..... 1
		return p; ..... 1
	}
}
```

Ex.3

```
public class LinkedListOfPapers { ..... /22

	private Node head;

	public LinkedListOfPapers() { ..... /1
		head = null; ..... 1
	}

	public void insertAtBack(Paper p) { ..... /8
		Node newN = new Node(p); ..... 1

		if(head == null) { ..... 1
			head = newN; ..... 1
		}
		else { ..... 1
			Node current = head; ..... 1
			while(current.getNext() != null) ..... 1
				current = current.getNext();..... 1

			current.setNext(newN); ..... 1
		}
	}

	public int getTotalNbWords() { ..... /6
		int sum = 0; ..... 1
		Node current = head; ..... 1
		while(current != null) { ..... 1
			sum += current.getData().getNbWords();..... 1
			current = current.getNext();..... 1
		}

		return sum; ..... 1
	}

	public LinkedListOfPapersgetPapers(int n) { ..... /7
		LinkedListOfPapersresult = new LinkedListOfPapers(); ..... 1

		Node current = head; ..... 1
		while(current != null) ..... 1
		{
			if(current.getData().getNbWords() == n) ..... 1
				result.insertAtBack(current.getData());..... 1

			current = current.getNext();..... 1
		}
		return result; ..... 1
	}

}
```