1. Software has to be expandable. By using self-contained classes, new functionality can be easily added by new classes

Software has to be changeable. A functionality is encapsulated in a class and needs to be changed only once.

Software has to be portable. By using interfaces, software can support multiple hardware, e.g. different keyboards.

2. The challenges that are involved by creators of engineering software are:

1) Employ efficient solution procedures.

2) Use the internet.

3) Provide comprehensible evaluation methods.

These can be solved by:

a) Object oriented programming:

This is because object-oriented programming provides a natural and powerful way to model actual engineering problems. Simplifies development of software. Let’s developers split software up into separate, reusable parts. Help developers to build reliable software systems. Simplifies programming in a team of programmers.

3. A class is a collection of program code(java), which defines the set of operations which can be carried out by an object, specifies what these operations do and defines the attributes which are used to store the state of an object.

An object is characterized by:

a) a set of methods

which can be invoked for this object in order to carry operations. In the example, we changed color and radius of spheres or let a Viewer object show up on the screen

b) attributes

defining the state of the object. In the example, spheres have a radius and a color.

Difference between them:

1) Classes exist only in the software text:

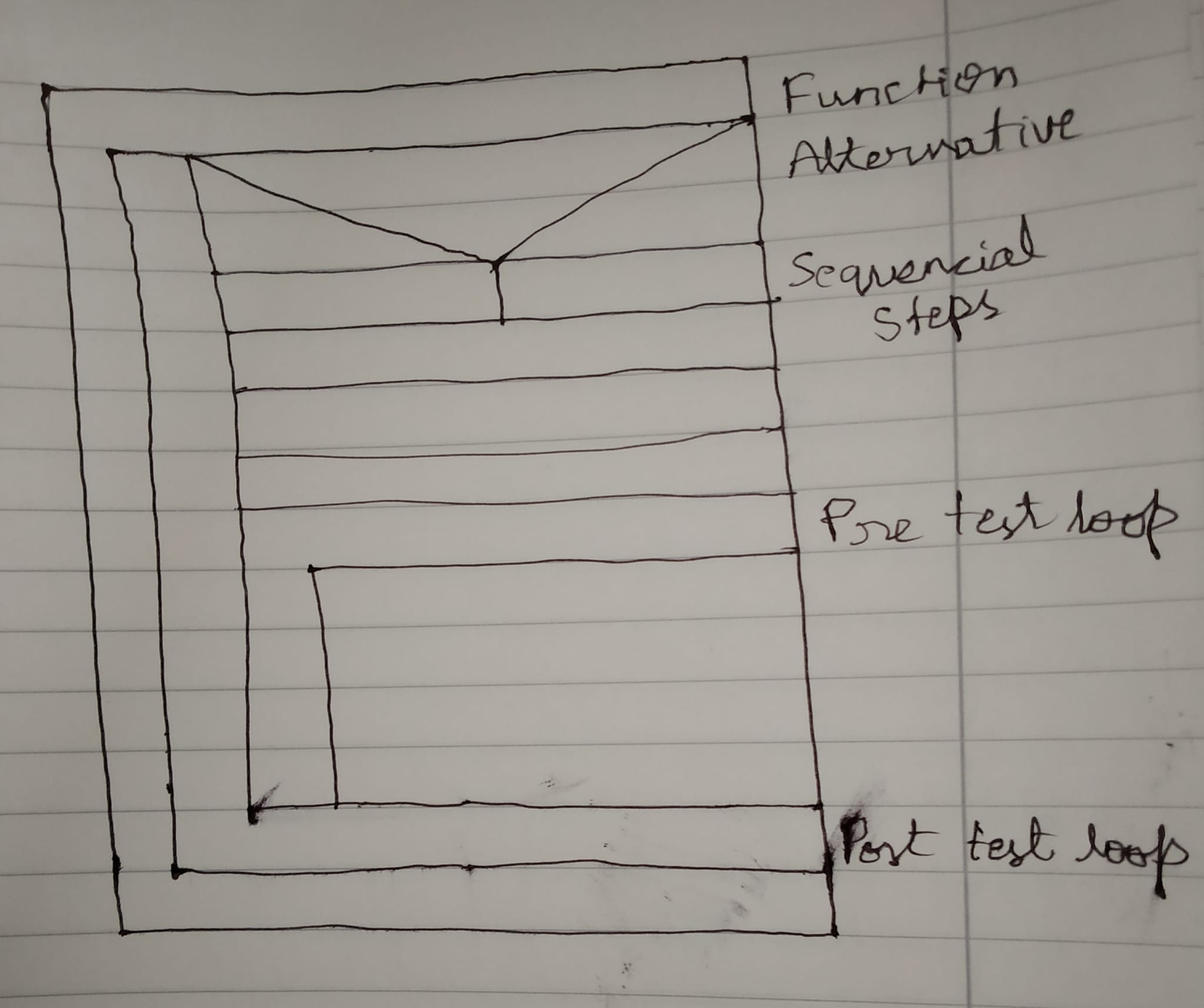
Defined by class text in the Java programming language

Describe properties of associated objects. Such objects are also called instances of a class

2) Objects exist only during program execution:

Visible in program text through variables pointing to run-time objects

4. Nassi Schneidermann Diagram can be used to model the algorithmic control flow in a program.



Function methods are written in Function.

When there is an option like if else statement, alternative step is used.

A series of sequential steps are written here.

Pre test loops is used for for loop and while loop.

Post test loop is used for do while loop.

5.

Table

Description automatically generated with medium confidence

6.

Diagram

Description automatically generated

7. 1. boolean data type can be used here because it is a decision if a book is lent in library or not.

2. The year of the publishing book can use int data type because years are integer values.

3. The price of the book can be given a data type of double as double can store more precise values for integer and decimals.

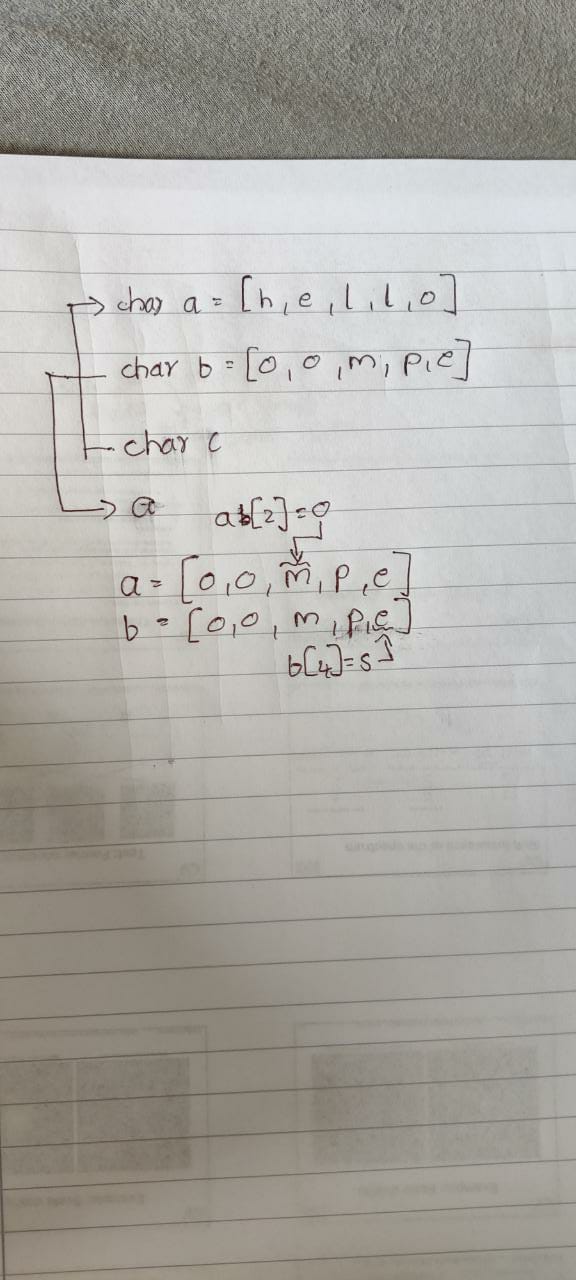
4. The single letter of the category can be given a char data type.

8.

The result of a is OOOPS

The result of b is OOOPS

The result of c is HELLO



10.

Class:

A class is a collection of program code (Java), which

• defines the set of operations which can be carried out by an object

• specifies what these operations do2

• defines the attributes which are used to store the state of an object

Association:

• An object is connected to other objects

Cardinalities:

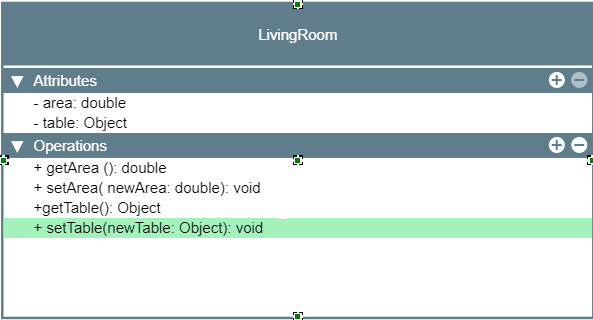
It specifies possible numbers of associated objects

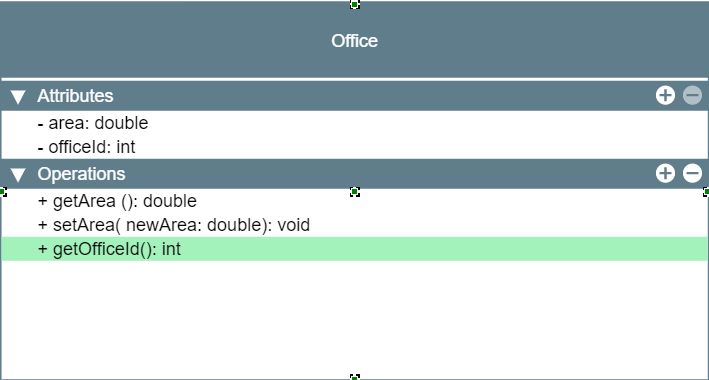
• Cardinalities make it easier for the reader to understand the idea behind the software design.

• Always include cardinalities in your diagrams.

• Replace m, n with numbers for concrete cases.

11.

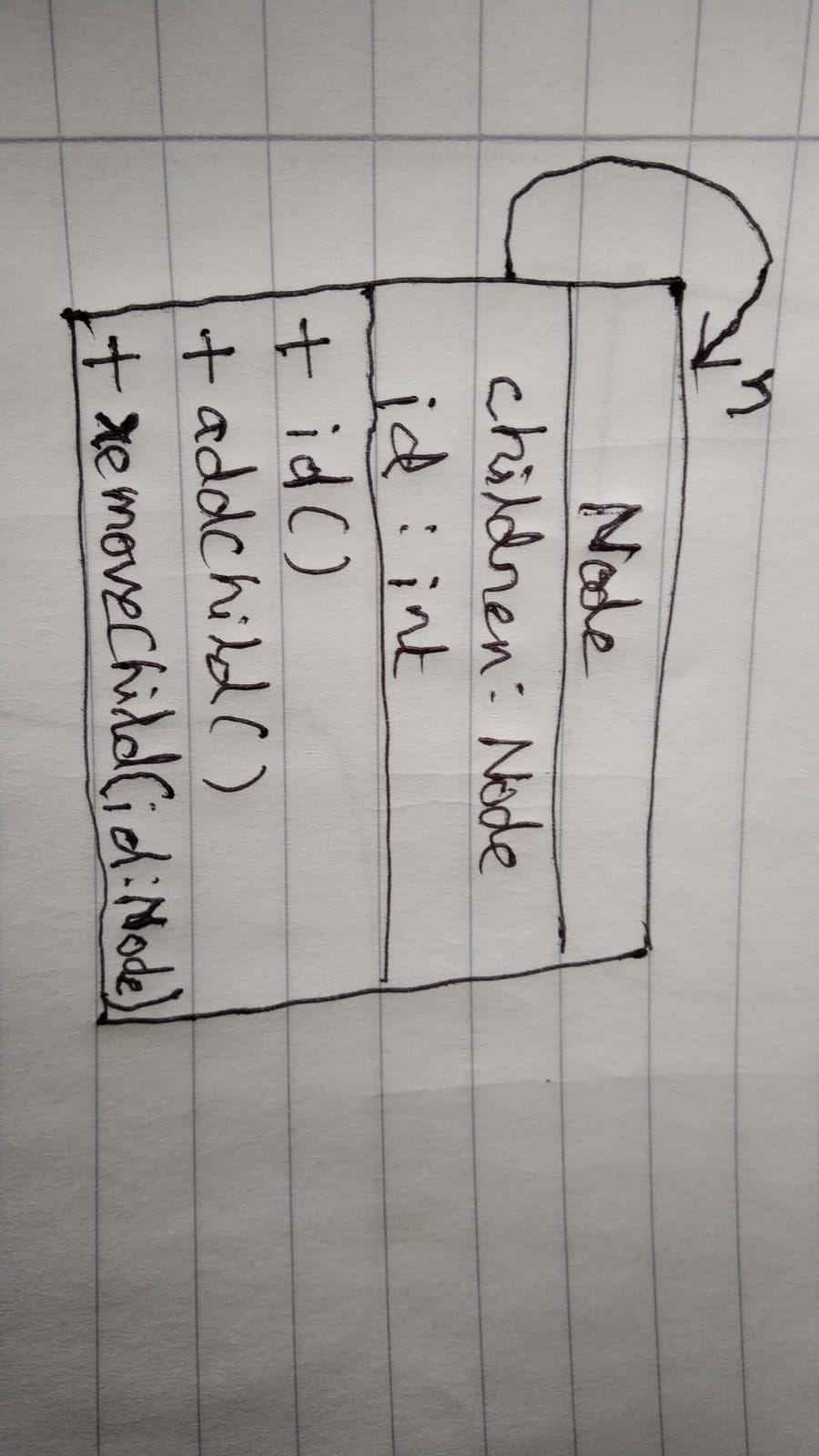




Inheritance helps to organize classes (office and LivingRoom) such that common properties and behaviours (area, getArea(), setArea()) have to be implemented only once.



12.



13.

public class Sensor {

private int id;

public Sensor(int id) {

this.id=id;

}

public int getid(){

return 0;

}

}

class LightSensor{

private double illumination;

public LightSensor(int id) {

}

public int getid(){

return 0;

}

public double getIllumination(){

return 8.9;

}

}

class TemperatureHumiditySensor{

private double temperature;

private double humidity;

public TemperatureHumiditySensor(int id){

}

public double getTemperature(){

return 6.4;

}

public double getHumidity(){

return 11.6;

}

}

14. public class main {

public int calculation(int a, int b) {

if (b == 0) {

return 1;

} else if (b < 0) {

throw new IllegalArgumentException("");

} else {

int result = a;

for (int i = 2; i < b; i++) {

result = result \* result;

i = i + 1;

}

return result;

}

}

}