

NAME: DAYANNA SILVA

**INTRODUCTION TO OBJECTS**

You do not necessarily have to know C, but if you feel unprepared you can read thinking in C.

**THE PROGRESS OF ABSTRACTION**

All languages ​​provide abstractions the programmer must establish the association between the model of the machine and the model of the problem prolog converts all problems into decision chains the object-oriented approach tries to go one step further by providing tools to the programmer to represent the elements in the space of the problem

The goal is that when you read the code that describes the solution, you are reading words that also express the problem.

Alan kay, summed up the smalltalk, the first java language to succeed

1. everything is an object
2. A program is a lot of objects that tell each other what to do, sending messages
3. Each object has its own memory formed by other objects
4. Every object has an associated type
5. All objects of a particular type can receive the same messages

Each object has an exclusive memory address.

**EVERY OBJECT HAS AN INTERFACE**

Remember type = class and vice versa

The programmer defines a class to adapt a problem instead of forcing the use of an existing data type that was designedto represent a storage unit in a machine Once you have defined a class, you can create as many objects of that class as you want and these objects can be manipulated as if they were the elements of the problem that is trying to solve a great challenge in POO is to create a one-to-one correspondence between the elements of the problem space and the objects of the problem space of the solution.

The requests that can be made to an object are defined through its interface

**AN OBJECT PROVIDES SERVICES**

One of the best ways to think about objects is as if they were "service providers", before creating an object Questions should be asked such as: What services does this object provide ?, if this approach is made the programmer could say "This object is simple enough to write it myself" A common problem among programmers is assigning too many functionalities to the object

**THE HIDDEN IMPLEMENTATION**

The objective of the client programmer is to reuse a complete toolbox to be used for the development of applications The objective of the class creator is to expose the client only what is necessary and keep everything else hidden In any relationship it is important to maintain limits + Java access modifiers: public, private, protected

**IMPLEMENTATION REUSE**

The simplest way to reitulize a class is to directly use an object of that class = create a member object Defining a new class from an existing class is called composition

**HERITAGE**

Using inheritance you can build a hierarchy of types that expresses the problem you are trying to solve in terms of their types all messages that can be sent to the objects of the base class can also be sent to the objects of the derived class you do not necessarily have to add new methods to the interface to substitute a method it is enough to create a new definition for it in the derived class

**RELATIONSHIPS IS A AND IT IS**

the relationship between the base class and the derived classes is said to be "a" add new interface elements to a derived type, expanding the interface "is like a" interchangeable objects with polymorphism.

If it is not known which code fragment will be executed then a new subtype will be added and the code that is executed can be different without it being necessary to perform

changes in the method that calls it

**THE SINGLE ROOT HIERARCHY**

All classes of the last instance should be inherited from a single base class = yes

the flexibility of c ++, only if you have a lot of code in C

it is possible that all the objects of a single root hierarchy have a certain functionality containers

you do not need to know how much to store a container, just create a container object

two basic types of arraylist and linkedl containers

**Parameterized types**

A parameterized type is a class that the compiler can automatically customize to

that works with each specific type

**Creation and life of objects**

you must know whether or not it is necessary to eliminate an object ttan facilmete

if a new object is needed it is created in the memory cumulu java has a memory debugger, if an object is not in use, it destroys it,

Treatment of exceptions an exception is a generated object where the error has occurred

**Concurrent programming**

Tthe program running separately is called threads and the general set is called occurrence

Java and internet.

Java results in programs that emerge from the www.

In the information repository is centralized.

The software, repository etc is called server.

The one that shows the machine itself is called customer.