Pg1

Tuesday, April 14, 2020 8:13 AM

Polymorphism :

The ability of an object to exhibit more than one form is known as polymorphism.
(one name many forms is the actual meaning of polymorphism)

Polymorphism can be classified into 2 categories:

1. Compile Time Polymorphism
2. Run Time Polymorphism

Compile Time Polymorphism (static binding) :

IF the binding happens at the compile time we call it as compile time polymorphism(static binding).

(What compiler sees at the compile time same thing gets executed even at the runtime)

Examples for Compile Time Polymorphism / (static binding) :

1. Variable shadowing
2. Method Overloading & Constructor Overloading
3. Method Shadowing

Variable Shadowing :

If the super class and subclass is having static or non-static variable with the same name it is known as **variable shadowing.**

Note :

1. Is-A relationship is necessary.
2. Variable Shadowing exhibits Compile Time Polymorphism
3. Basically which variable is used in the runtime Depends on Compile time Binding :
    1. place of usage
    2. reference type and not the object created.

for ex: refer, **app21/src**

Method Overloading & Constructor Overloading :

If a class has more than one method or constructor with the same name but different signature is known as method Overloading / Constructor overloading.

Note :

1. To achieve method overloading Is-A relationship is not required.
2. It exhibits Compile Time Polymorphism.
3. Which method or constructor to be executed at the runtime is decided by the compiler at the compile time itself, based on the values passes in the call statement (actual arguments).

<u>**Method Shadowing / Method Hiding :**</u>

If the super class and sub class is having static method with the same signature is known as Method Shadowing .

Note:

1. Method Shadowing exhibits Compile time polymorphism.
2. Which method gets executed at the runtime is decided by the compiler at the compile time based on :
    i) place of usage
    ii) reference type
    iii) **It does not depend on Object created.**

    for examples refer, **app22/src**

Rules to achieve Method Shadowing:

1. Inheritance is mandatory (Is-A)
2. Method shadowing is possible only with static methods.
3. Super class and sub class must have static methods with same signature.
4. the return type of both the methods should be same (refer **app22/src/Driver3.java**)
5. The access modifier of subclass should be either same or more visibility compared to super class static method's access modifier. (refer **app22/src/Driver4.java,Driver5.java,Driver6.java**)

Pg2

Tuesday, April 14, 2020 9:24 AM

Run Time Polymorphism :

Binding happens during the execution of the program, based on the Object created.
(What compiler see does not get executed)

Note :

- We can achieve runtime polymorphism with the help of
  - ☐ Is-A relationship
  - ☐ Method overriding
  - ☐ Derived Type casting

Method Overriding :

The process of providing implementation to the super class method from the subclass method is known as method overriding.

Note:
1. The super class non-static method is known as Overridden method
2. the subclass non-static method is known as Overriding method.

Rules to be Followed for Method Overriding :
1. Inheritance is mandatory. (Is-A)
2. Method Overriding can be done only for non-static methods.
3. In subclass the signature of the overriding method should be same as the signature of the overridden method in super class.
4. The return types of both overriding and overridden methods should be same. (If not we get CTE) for ex, refer **app23/src/Driver3.java**
5. The access modifier of the overriding method should be either same or having more visibility compared to overridden method. (if not we get CTE) for ex, refer **app23/src/Driver4.java**

Note :

1. Method Overriding exhibits Run Time Polymorphism
2. The binding happens during the runtime based on the object created.

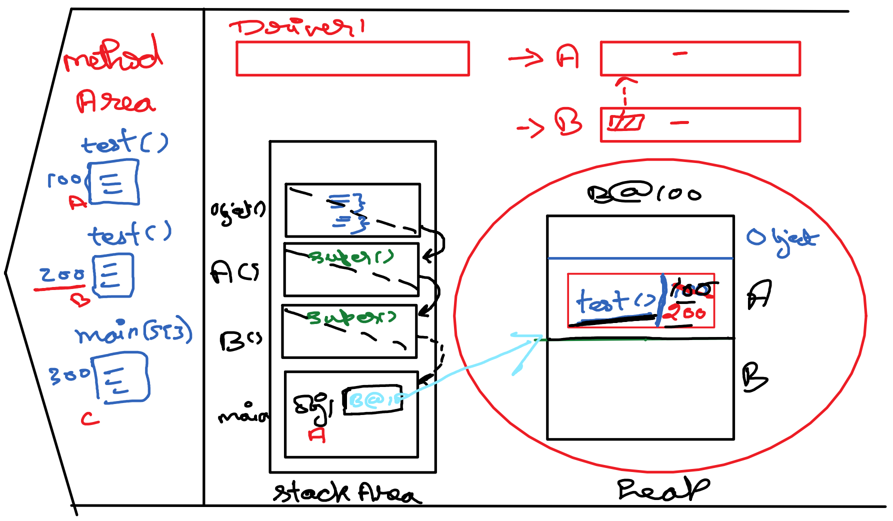
1. Method Overriding exhibits Run Time Polymorphism
2. The binding happens during the runtime based on the object created.

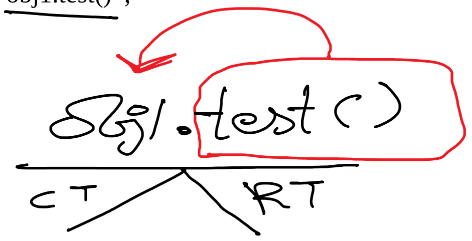
i.e. Execution of a method depends on Object created and not the reference used.

Pg 3

Wednesday, April 15, 2020 9:56 AM

```
class A  {
        void test(){
                System.out.println("Hello from class A ") ;
        }
}
class B extends A {
        void test(){
                System.out.println("Bye  from class B ") ;
        }
}
class  Driver1 {
        public static void main(String[] args) {
                A obj1  = new B() ;
                obj1.test()  ;
        }
}
```

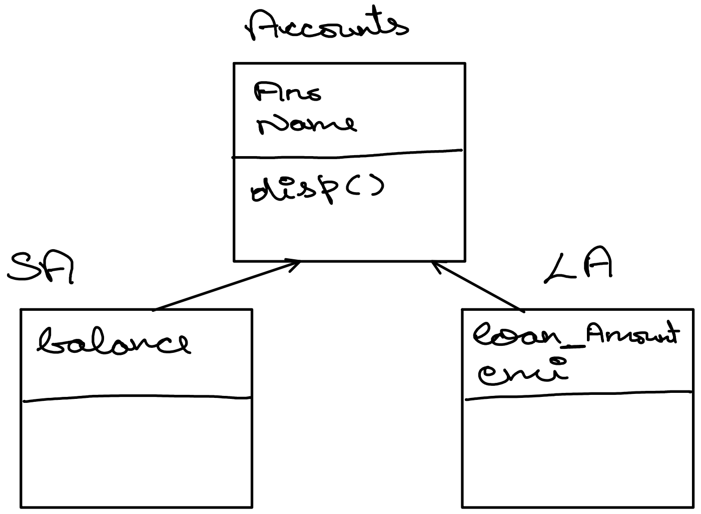




Binds test() with class

since class B Object was created, the class B test method overrides class A test method at the time of object creation There for class B method gets Executed

Task1 :



1. implement the following class diagram.
2. achieve data hiding for all the properties.
3. create suitable constructors.
4. Design the classes such that, if the user creates an object of LoanAccount and calls display method all the properties of LoanAccount and its super class must be displayed. If the user creates the an object of SavingsAccount all the properties of SavingsAccount including its superclass must be displayed.
5. Create a Driver class, to test the above design.
6. In Driver class create 1 object for each class and display their properties.

Refer programs in, **day9/app5/src**

Assignment

Wednesday, April 15, 2020 9:50 AM

1. What is Polymorphism ?
2. Explain the types of Polymorphism ?
3. What is variable shadowing? explain with an example
4. What is method Shadowing ? explain with an example.
5. What is method Overriding ? explain with an example.
6. What is the purpose of method overriding ? explain with an example.
7. What is Compile time polymorphism? give any one example .
8. What is Runtime polymorphism? explain with a program
9. What is the difference between Compile time and runtime polymorphism.
10. Explain the difference between method shadowing and method overriding.
11. What is overridden method ?
12. What is overriding method ?