

Questão 1:

Abstração: é uma forma de conseguir representar um objeto real dentro de um sistema, de forma que apenas seja declarado apenas o necessário para o sistema em específico. No geral basta saber o que o sistema deseja e abstrair coisas irrelevantes do objeto real para o sistema. Utilizamos para isso a identidade(nome do objeto), as propriedades(características do objeto) e os métodos(ações do objeto) são importantes para esse processo.

Exemplo: precisamos fazer um cachorro latir no sistema logo a identidade é o próprio cachorro, as propriedades pode ser colocada o tamanho do cachorro pois influencia no latido do mesmo, e o método é a ação de latir

Encapsulamento: é uma forma de proteger a aplicação e evitar manutenções em arquivos já prontos, sendo assim se baseia em proteger uma classe através os métodos getters e setters, que irão retornar o valor da propriedade da classe, não expondo diretamente a classe as outras, fazendo ela ficar não vulnerável.

Exemplos: é a mesma situação de usarmos interruptores e os seus “espelhos”, para que não possa alterar os filamentos(como proteção das pessoas e funcionamento de uma lâmpada).

Em uma aplicação podemos ter uma classe chamada Cachorro e colocarmos o nome dele como privado onde só podemos acessar da classe, e criar métodos gets para buscar o nome e sets para trocar ou adicionar o nome ao cachorro.

Herança: É uma forma de aproveitar código de outras classes a grosso modo, mas se baseia em uma Classe ter Subclasses que compartilham dos mesmos métodos e dependendo do tipo da propriedade ela também, dessa forma é herdado esse conjunto de informações para se aplicar novas informações na subclasse.

Exemplo: Animal é uma classe logo cachorro é uma subclasse de Animal pois herda todas as funções de um animal, como andar, entoar sons e etc. Assim como um gato também é uma subclasse de Animal. Logo todo cachorro é um animal mas nem todo animal é um cachorro.

Polimorfismo: Consiste em mudar a forma que um método de uma herança passada para subclasse, isso é, alterar o funcionamento do método para cada subclasse que atua com o método de formas diferentes.

Exemplo: A classe Jogador de futebol tem como método jogar bola, mas a forma que a subclasse atacante joga bola é diferente de como o zagueiro joga bola, mas ainda sim eles jogam bola.

Questão 2:

A primeira coisa a ser mostrada na linha de comando é a frase:

“Maooeee, quanto que vai dar nessa conta? É 10, é 50 reais? Quem quer pontoooo??”

Enquanto X=1 e Y=1(sem impressão)

Isso ocorre pois é a primeira coisa escrita na main através do comando
System.out.println()

Enquanto nada ocorre com x e y e continuam acumulando o valor original

A segunda linha mostra a frase:

"Ritmoaaa, ritmooooo de contrutorrrrr!!"

Enquanto X=2 e Y=0(sem impressão)

Após o comando passado no main(explicado na primeira linha) é passado

"C2 v = new C3(2)"

Isto é uma variável do tipo C2 é criada e chamada de v , ela recebe o construtor

"C3(2)"

OBS: isso é permitido pois C3 é uma herança de C2, ou seja, uma extensão, logo variáveis do tipo C2 podem ser um construtor de C3, pois C3 ainda é um C2

Isso acarreta a vermos o construtor de C3(int f)

OBS2: É o único construtor de C3 que passa parâmetros int, logo é esse que o construtor o referenciado

A primeira linha ele passa:

super(5)

Nesse momento ele chama o construtor C2(int f), pois se passa um parâmetro (Mesma razão de Obs2), esse construtor não passa nenhum super() no código, mas como C2 é uma Herança de C1 o super() fica implícito na primeira linha chamando assim C1() (pois o super() não passa parâmetros, mesma explicação de OBS2)

Sendo assim imprime a frase, com o comando System.out.println() passada na primeira linha de C1()

Em baixo da impressão é feito os comandos:

x++;

y--;

Que simboliza:

x=x+1

y=y-1

Que resulta em x=2 e y=0

finalizando o construtor C1

A terceira linha de comando apresenta:

"Passa ou repassa esse contrutor Silvio?"

Enquanto x=7 e y=0(sem impressão)

Ao finalizar o construtor C1(), volta super() implicito em C2(int f) (como foi explicado na linha anterior)

Após isso se imprime a frase através do comando System.out.println()

Logo depois tem os comandos:

y*=f

x+=f

Sendo que f é o parâmetro passado em C2(int f) que é chamado no super(5) de C3(int f), como explicado anteriormente, sendo que o super() é passado o 5 como parâmetro o f de de C2(int f) é igual ao mesmo.

Y=f*y=0*5=0

X=f+x=5+2=7

A quarta linha de comando apresenta:

“Maooeee, mas será que essa é a resposta? O Roque já trouxe o super!”

Enquanto x=7 e y=0(sem impressão)

Ao finalizar C2(int f) é retornado ao construtor C3(int f), abaixo a linha de super(5)

Onde há um comando System.out.println() que irá imprimir a frase

O resultado de x e y não se alteram

A quinta linha de comando apresenta:

“7”

Enquanto x=7 e y=0(sem impressão)

Ao finalizar C3(int f) é retornado a linha do main a baixo que passava

v.opC2(2)

Isso é chamando a função de opC2(int f) de C2, que dentro da o comando System.out.println() da soma de x+y e 7+0=7.