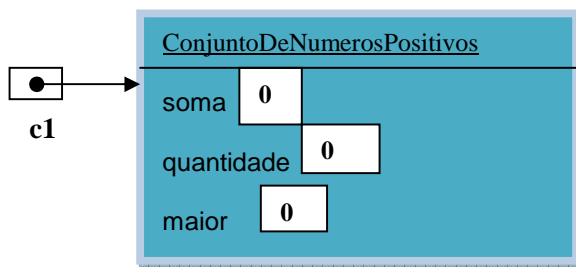


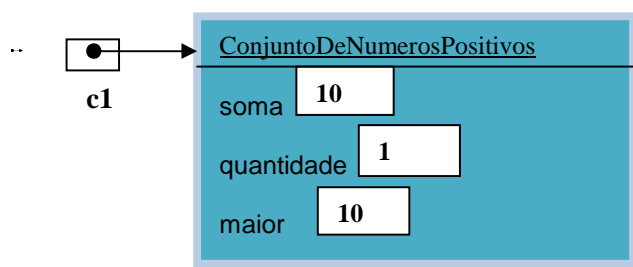
- 1) a) Programe a classe **ConjuntoDeNumerosPositivos** com atributos para armazenar: a soma de todos os números, a quantidade de números e o maior número. O construtor cria um conjunto de dados vazio (zerando os três atributos). A classe deve oferecer os seguintes métodos:
- **incluiUmNumero** – recebe, via parâmetro, um número inteiro e inclui somente se positivo. Atualiza, adequadamente, os atributos com a inclusão de mais um número. Observe que o número incluído não é armazenado no conjunto;
 - **getMedia** - calcula e retorna a média aritmética de todos os números incluídos;
 - **getMaior** - retorna o maior número incluído.

Abaixo, um exemplo passo a passo, visualizando o objeto na memória, a partir da sua criação, passando pelas diversas inclusões de números:

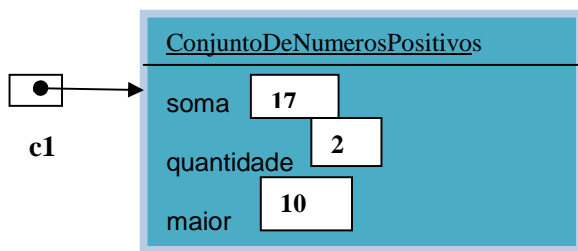
passo 1: criação do objeto:



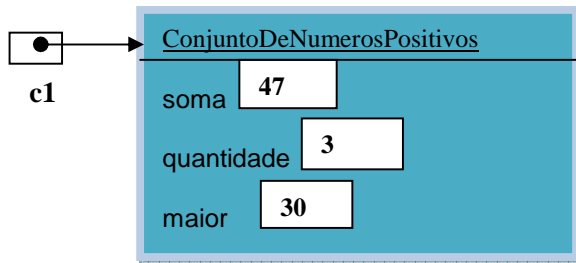
passo 2: inclusão do número **10**, pelo método incluiUmNumero:



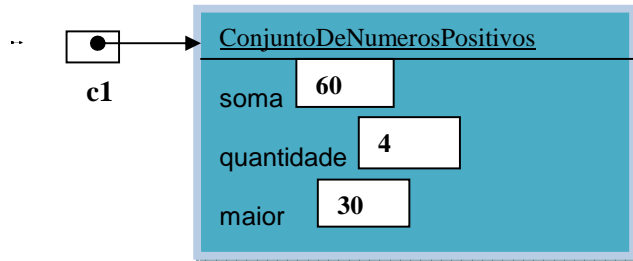
passo 3: inclusão do número **7** pelo método incluiUmNumero:



passo 4: inclusão do número **30**, pelo método incluiUmNumero:



passo 5: inclusão do número **13**, pelo método incluiUmNumero:



b) Programe uma classe de teste, criando um objeto **ConjuntoDeNumerosPositivos**, e adicionando números lidos via teclado. No final, exibir a média e o maior.

Obs. Encerrar a entrada de dados quando for lido o zero. Esta forma de encerrar a entrada de dados usando um valor fictício (que não faz parte da série de dados) é conhecida como usar um Flag.

- 2) a) Programe a classe **GeraExibeSequenciaInteiros** com um construtor e 2 atributos inteiros positivos (limiteUm, limiteDois). A classe deve oferecer os seguintes métodos:
- **Construtor:** recebe dois inteiros positivos e armazena o menor em limiteUm e o maior em limiteDois
 - **geraExibeInt:** gera e exibe os inteiros de limiteUm a limiteDois.
 - **geraExibePares:** gera e exibe os pares de limiteUm a limiteDois.
 - **geraExibeMultiplos:** gera e exibe os múltiplos comuns de 3 e 5, de limiteUm a limiteDois.
 - **geraExibeMultiplosCalcMed:** gera e exibe os múltiplos de **m** (recebido como parâmetro) de limiteUm a limiteDois. Devolve a média aritmética dos múltiplos gerados.
 - **geraExibeDivisores:** gera e exibe os divisores de limiteDois. (Ex. se limiteDois é 15; os divisores são: 1, 3, 5 e 15).
 - **geraExibePrimos:** gera e exibe os números primos, de limiteUm a limiteDois.

b) Programe uma classe de teste que ofereça um menu:

-
- 1 – Gera e exhibe inteiros
 - 2 - Gera e exhibe pares
 - 3 - Gera e exhibe múltiplos comuns de 3 e 5
 - 4 – Exhibe a média aritmética dos múltiplos
 - 5 - Gera e exhibe divisores
 - 6 - Gera e exhibe primos
 - 7 – Sair
-

Escolha a opção:

Obs. valide a opção. Sugestão: use do..while

3) Faça uma nova classe de teste para a classe **Funcionario**, da lista anterior:

Crie **n** (lido do teclado) funcionários com dados obtidos via teclado. Valide o código de reajuste, exibindo uma mensagem de erro e lendo novamente, cada vez que for digitado um código inválido, até ser digitado um código válido.

Após a leitura e processamento dos **n** funcionários, exhibir:

- O total pago pela empresa em salário, antes do reajuste.
- O maior salário da empresa após o reajuste.
- O salário médio do departamento 3 após o reajuste

4) Modifique o exercício anterior, para que não seja informada a quantidade de funcionários. Encerrar a entrada de dados quando for digitado "9" (flag) para o nome do funcionário.

`String n = t.leString("Entre com o nome do funcionário. [9 para encerrar]: ");`

5) Modifique o exercício anterior, para que não seja informada a quantidade de funcionários e nem seja utilizado flag. Cada vez perguntar se quer ou não entrar com dados, conforme o código abaixo:

```
char resp;  
do  
    resp= Character.toUpperCase (t.leChar("Quer entrar com dados [S-sim N-não] ? "));  
while (resp != 'S' && resp != 'N');
```