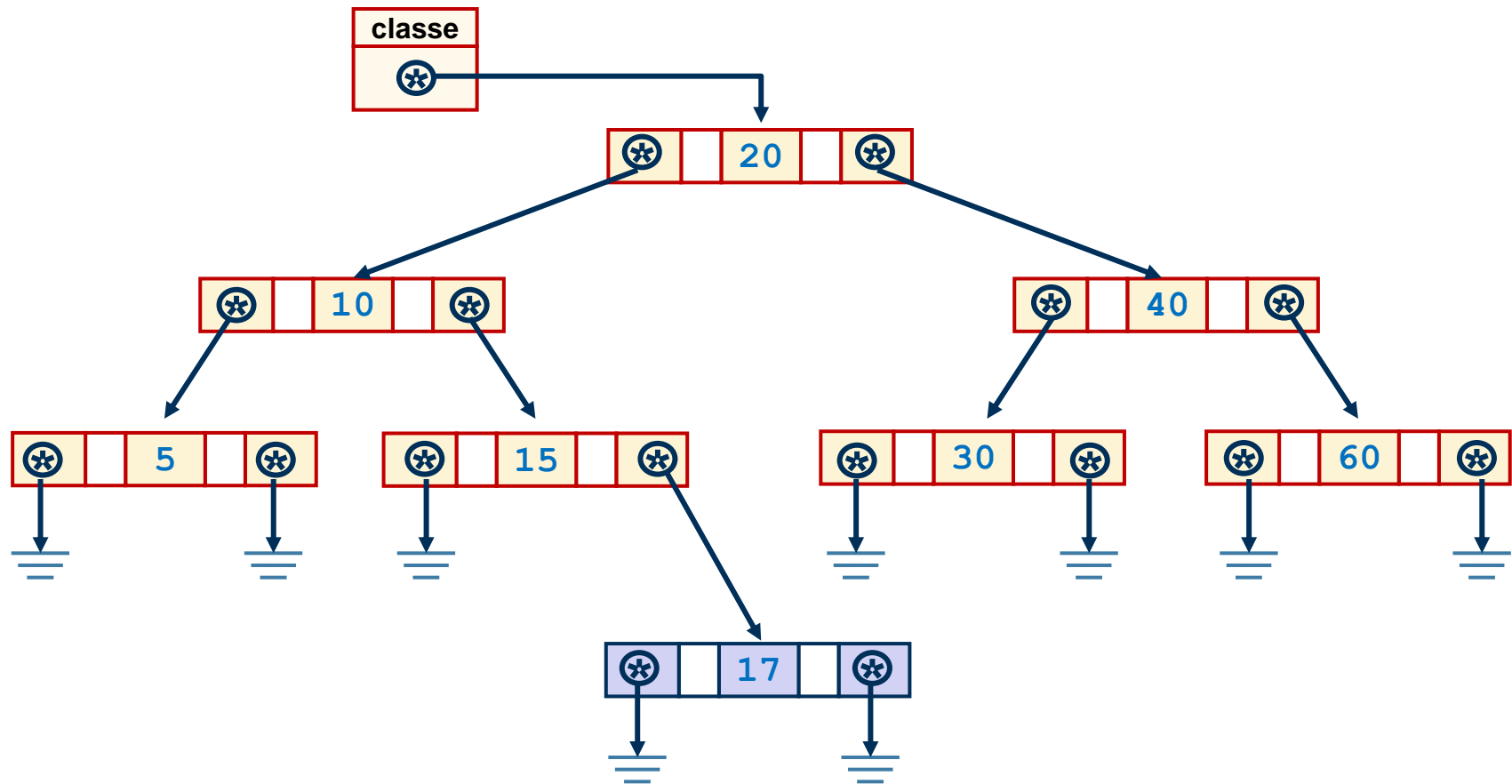


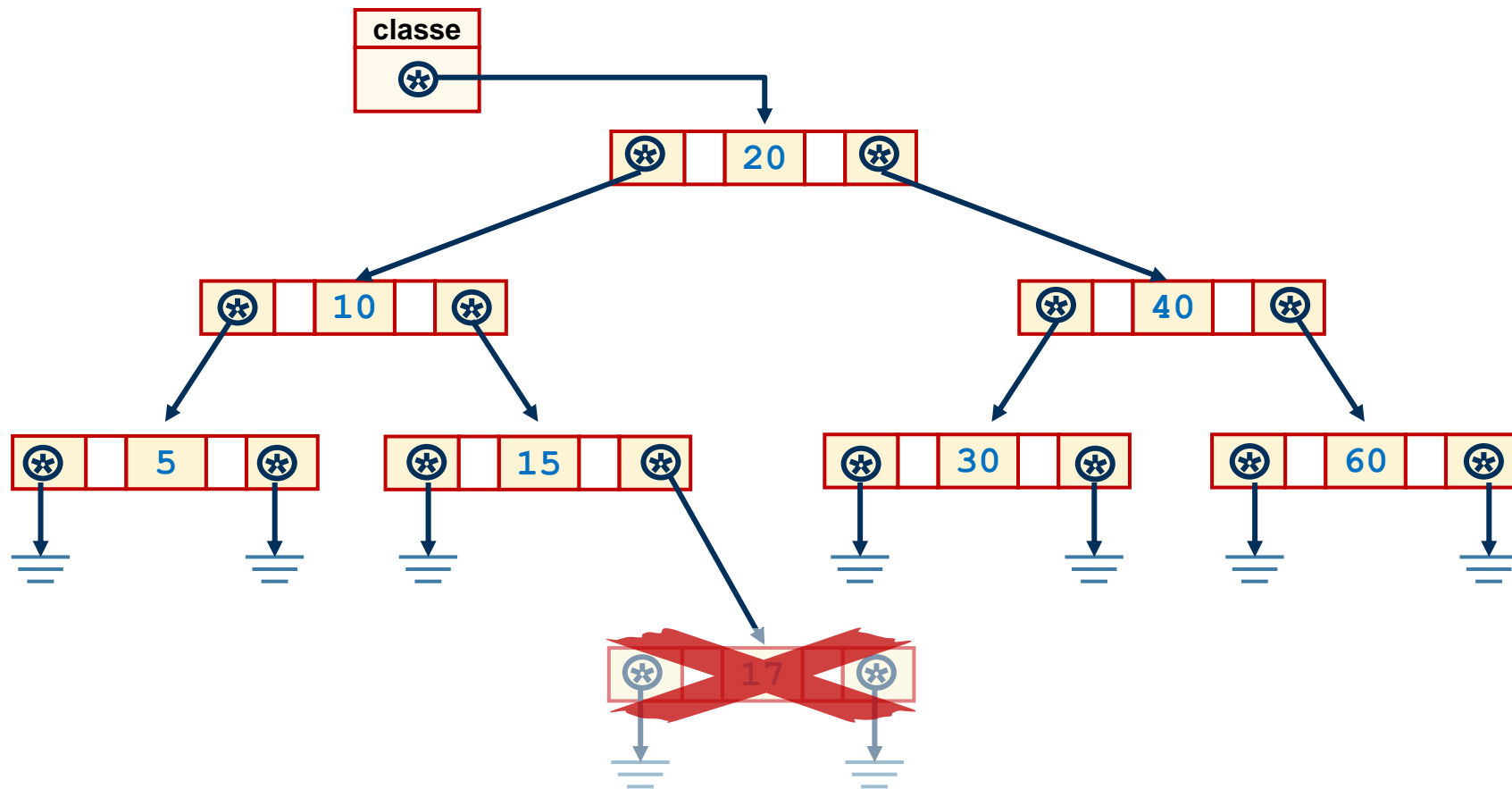


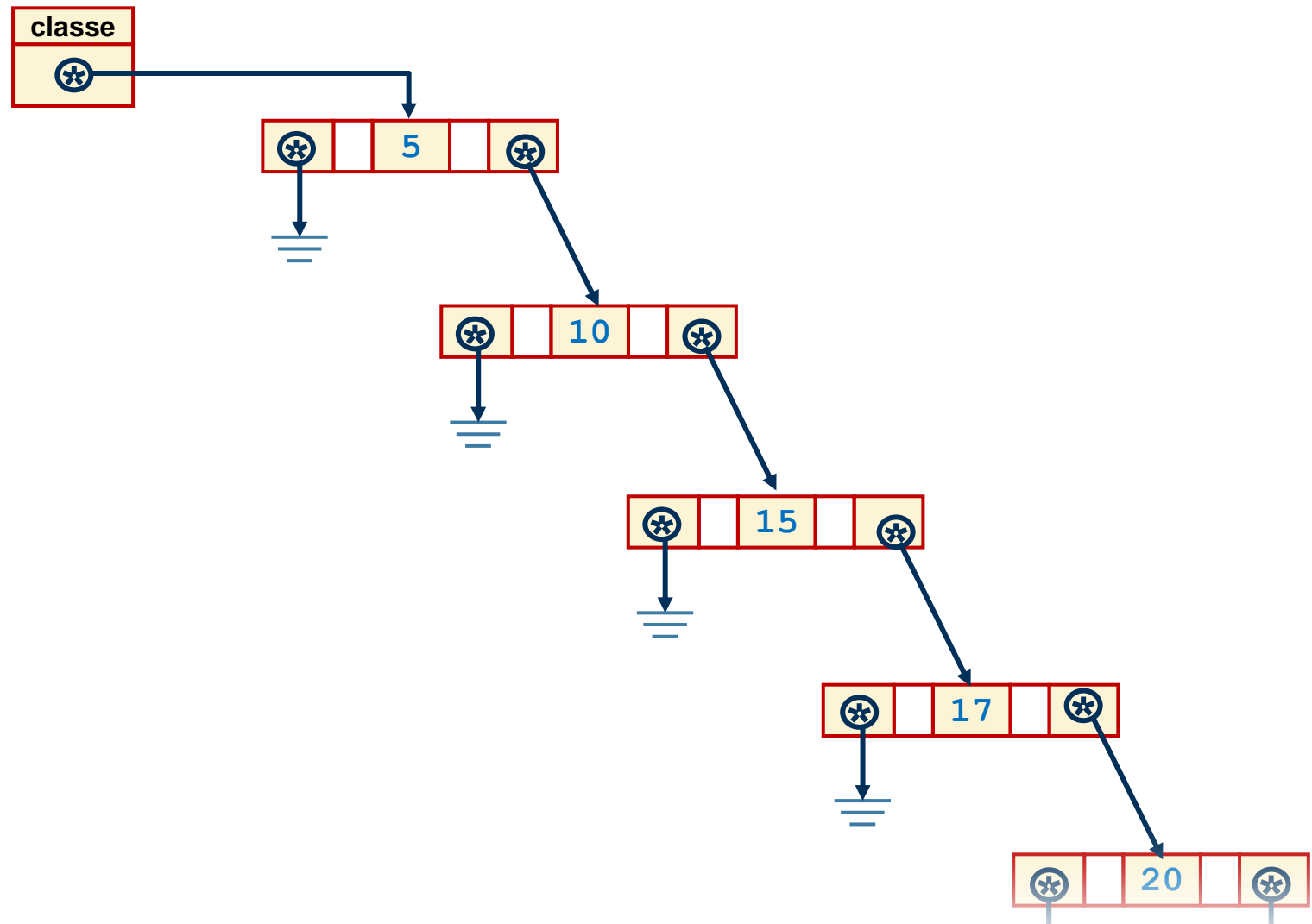
POLITECNICO  
DI MILANO

# INFORMATICA

Inserimento di nuovi  
elementi in alberi binari,  
con ricorsione.

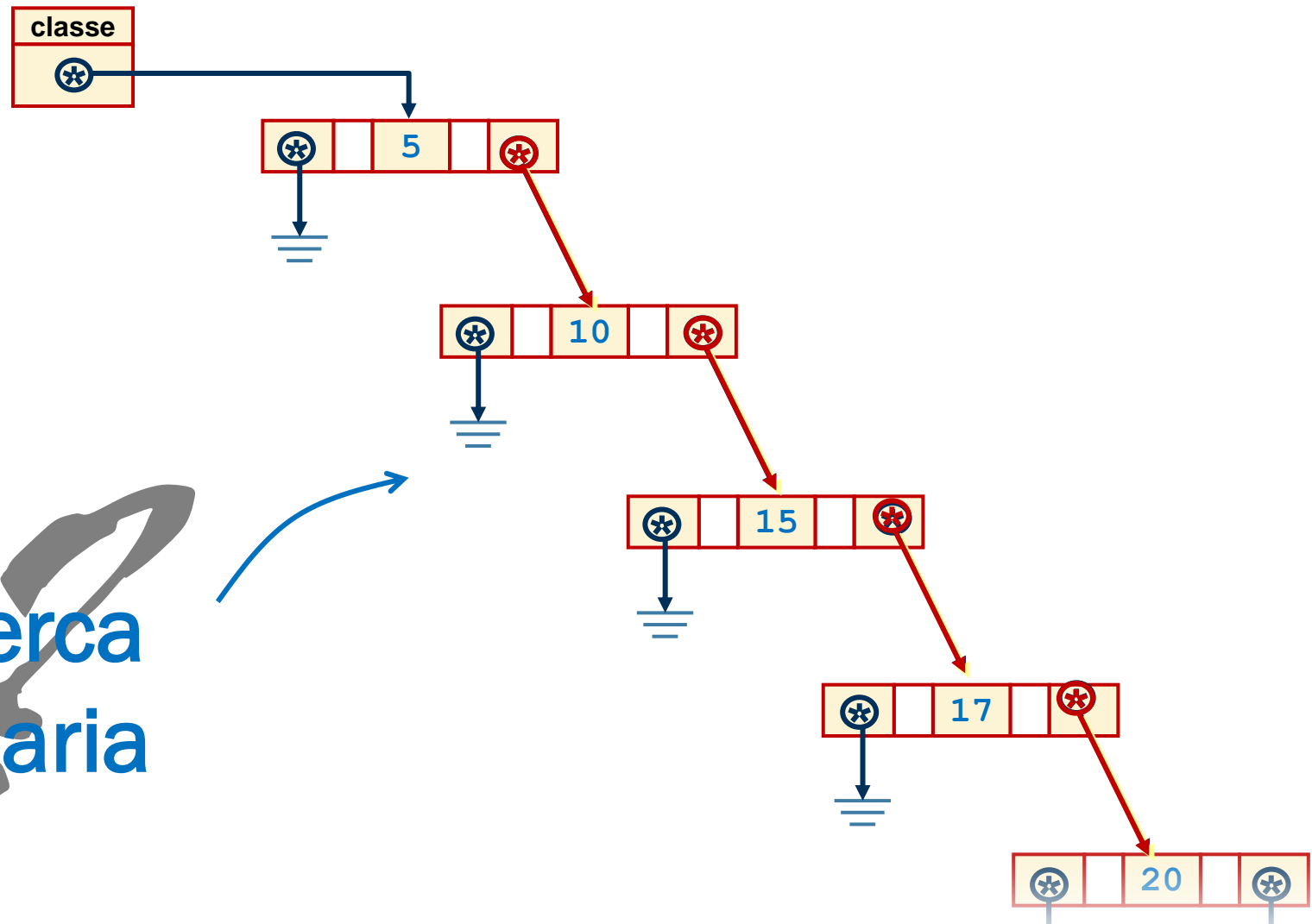




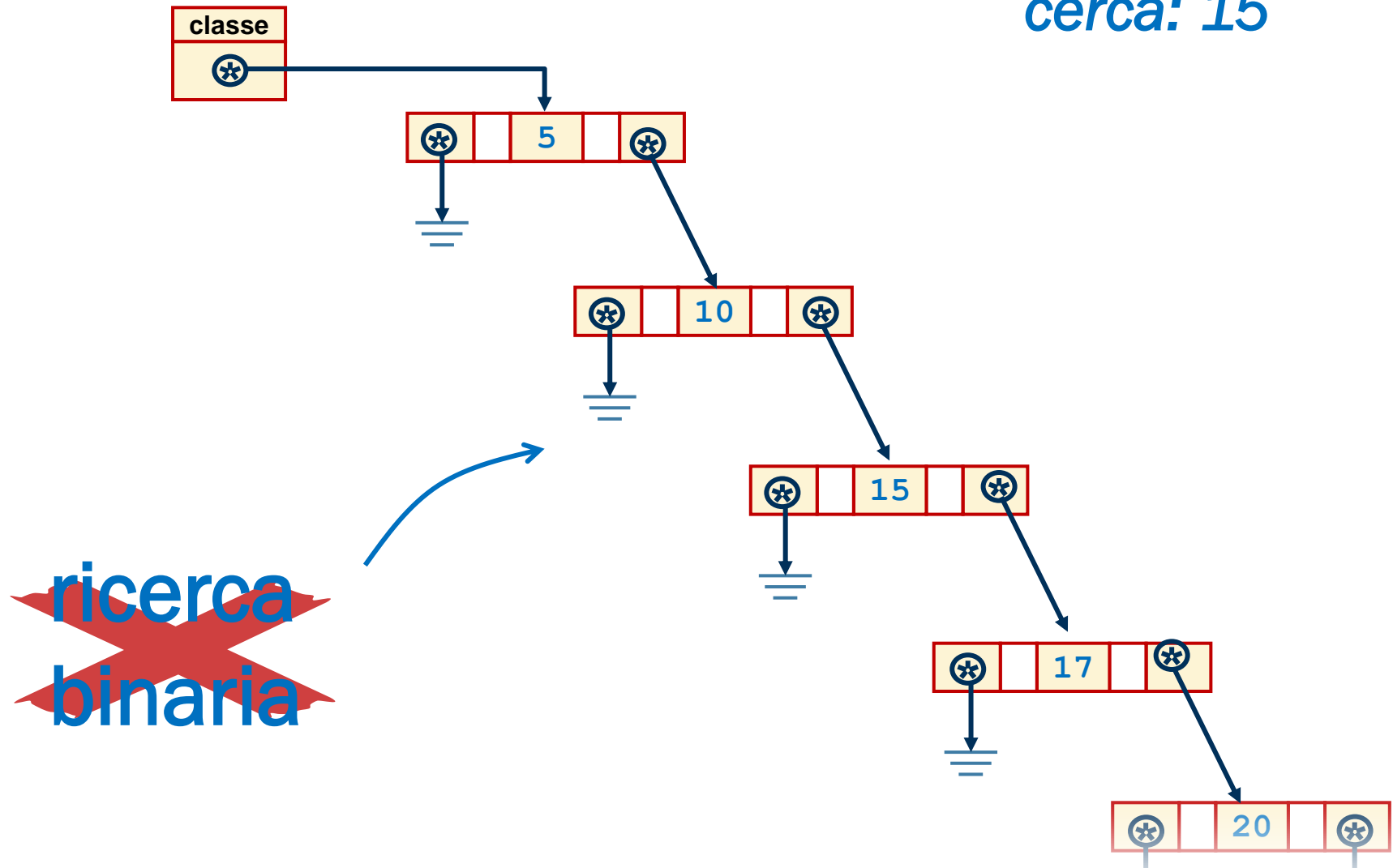


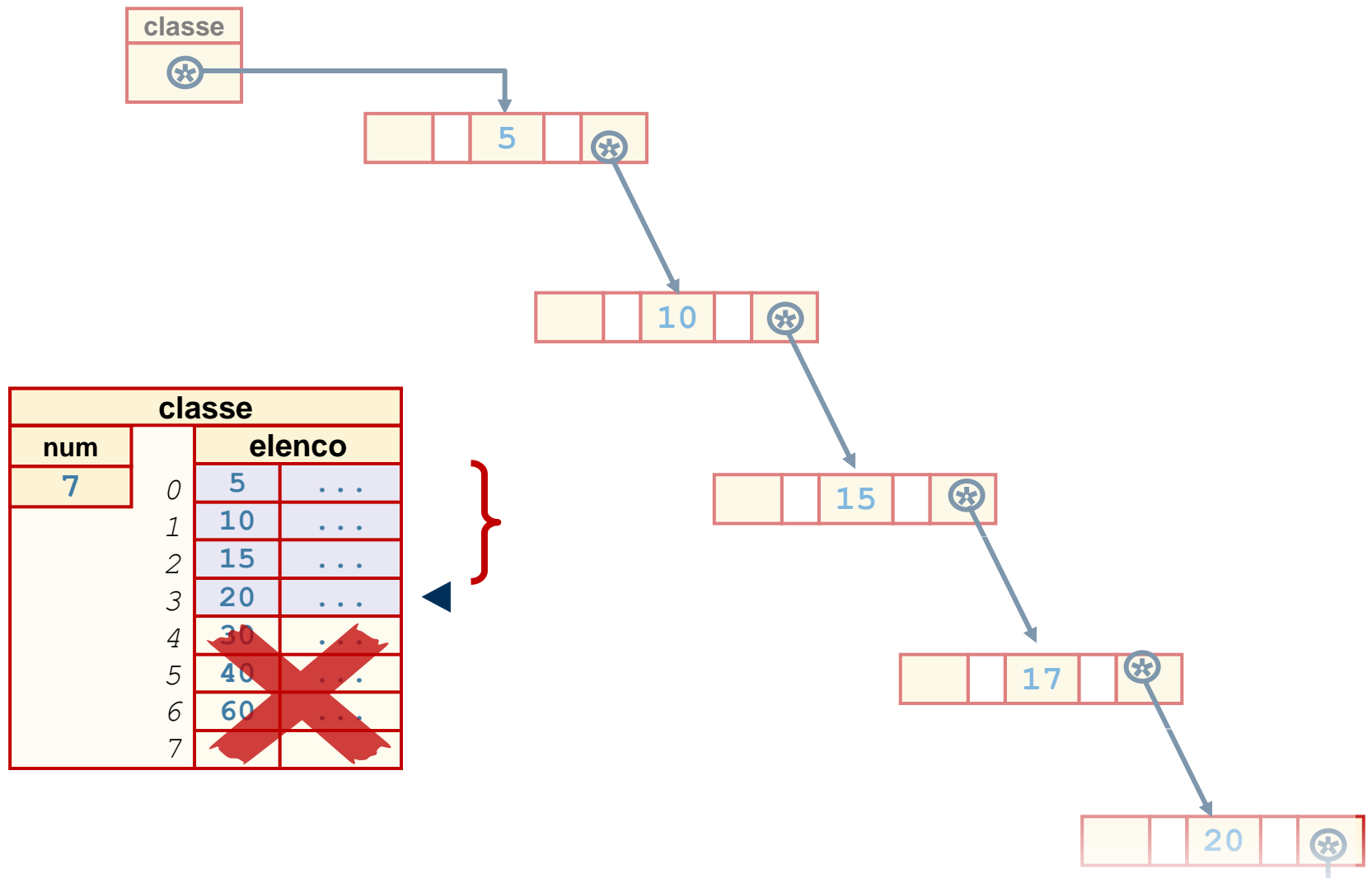
Lista lineare

ricerca  
binaria

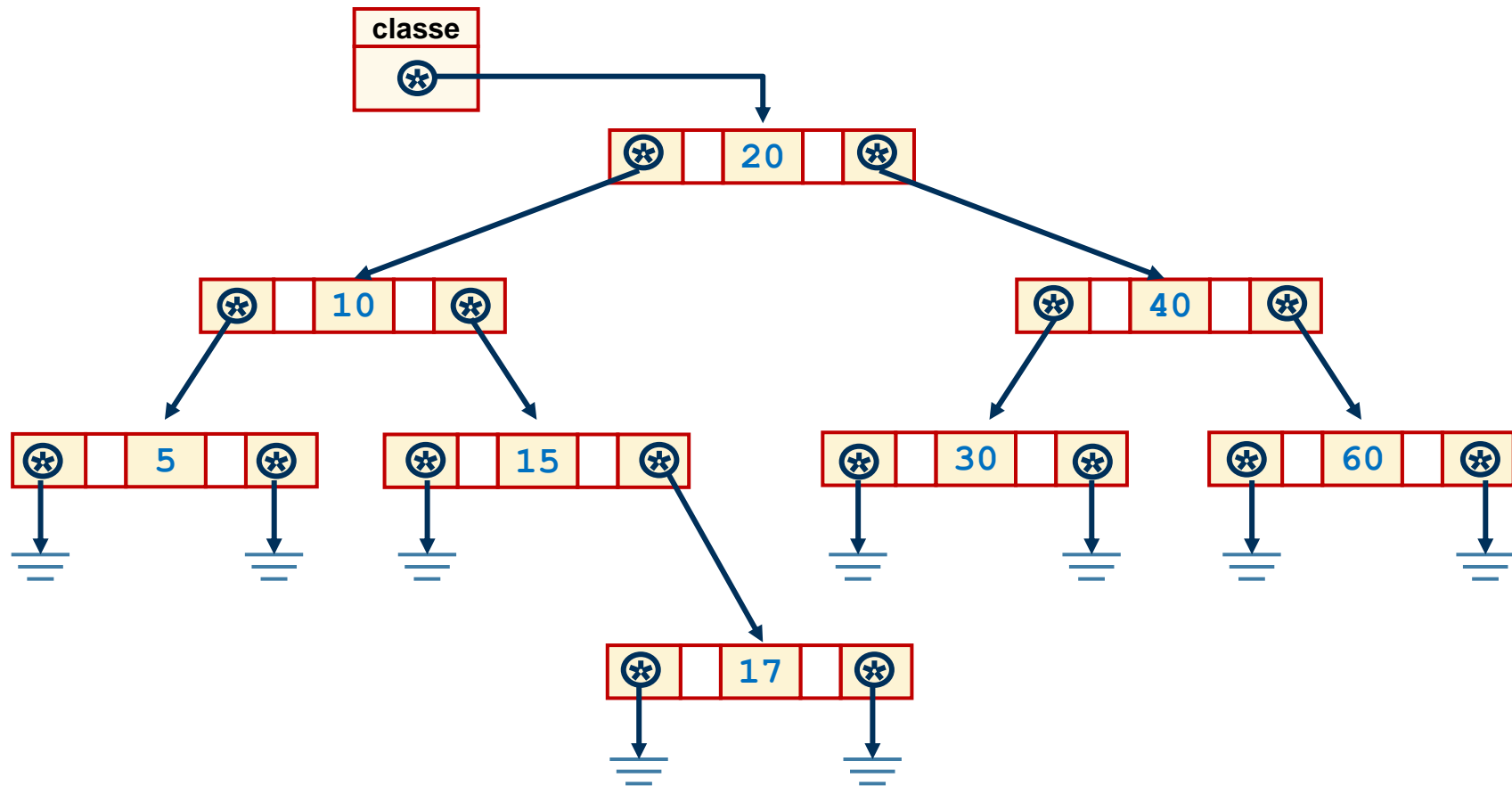


*cerca: 15*

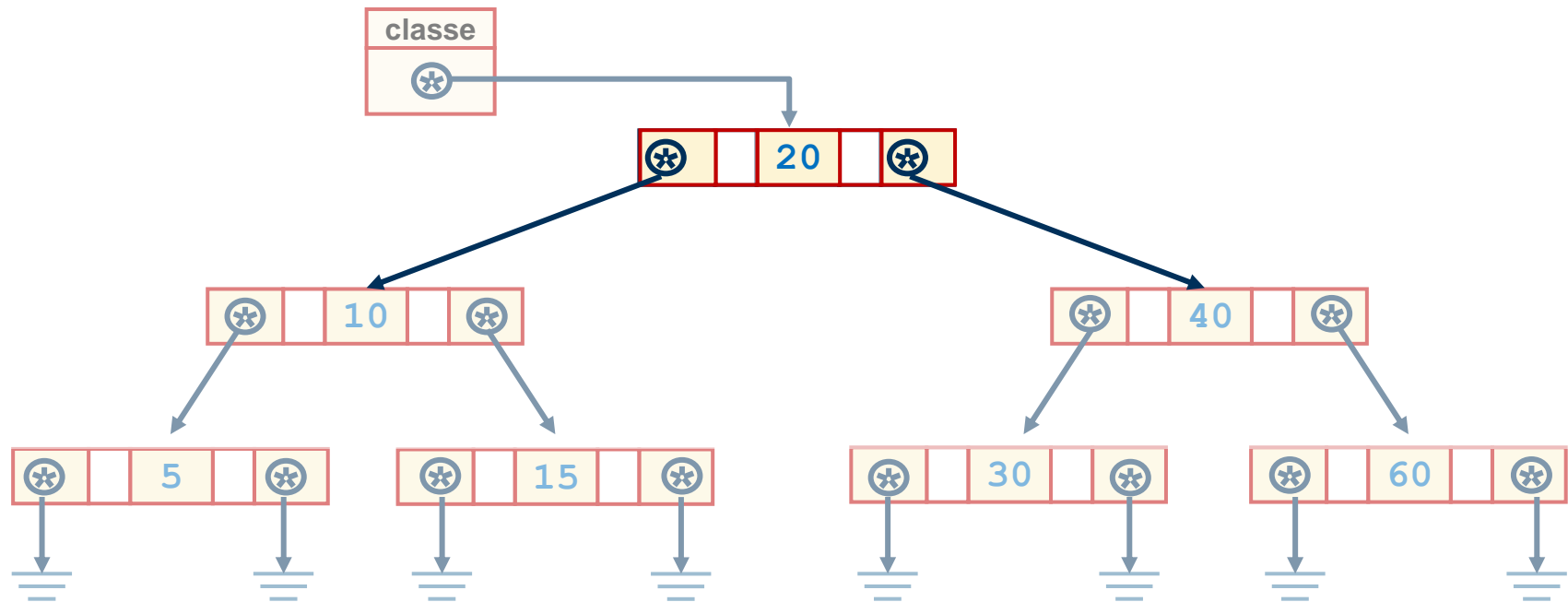


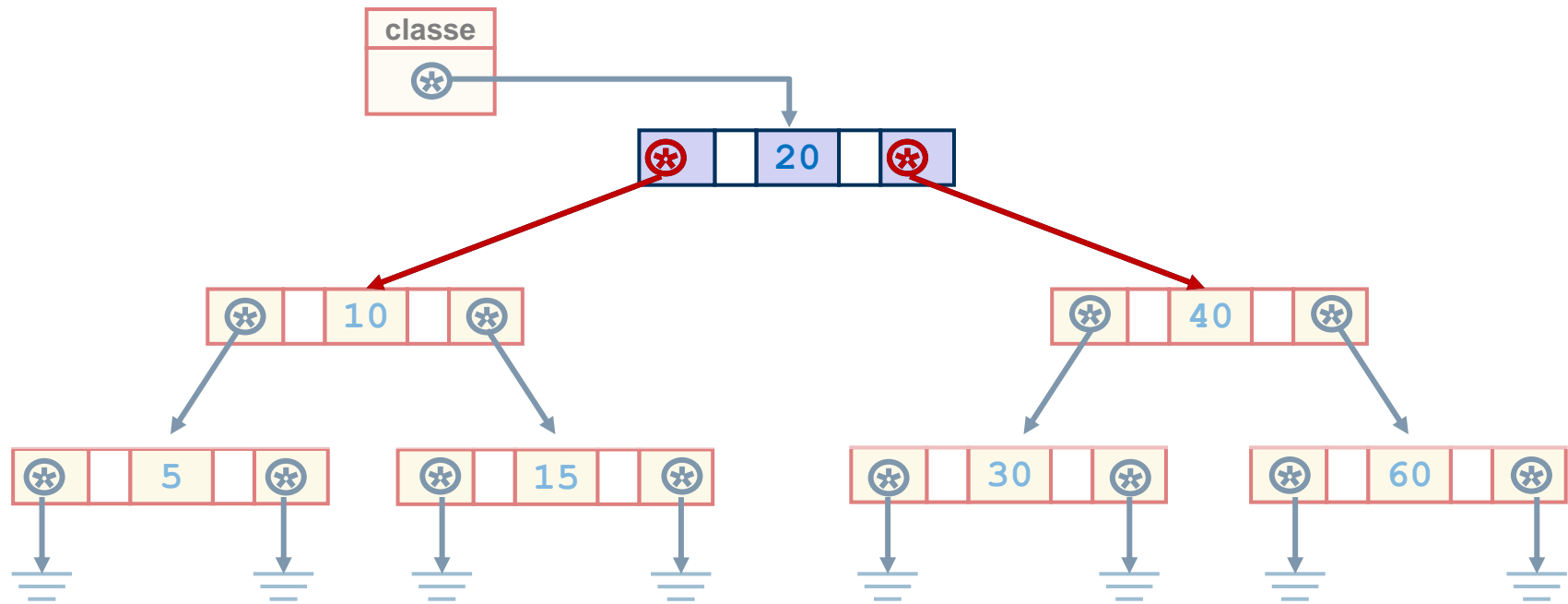


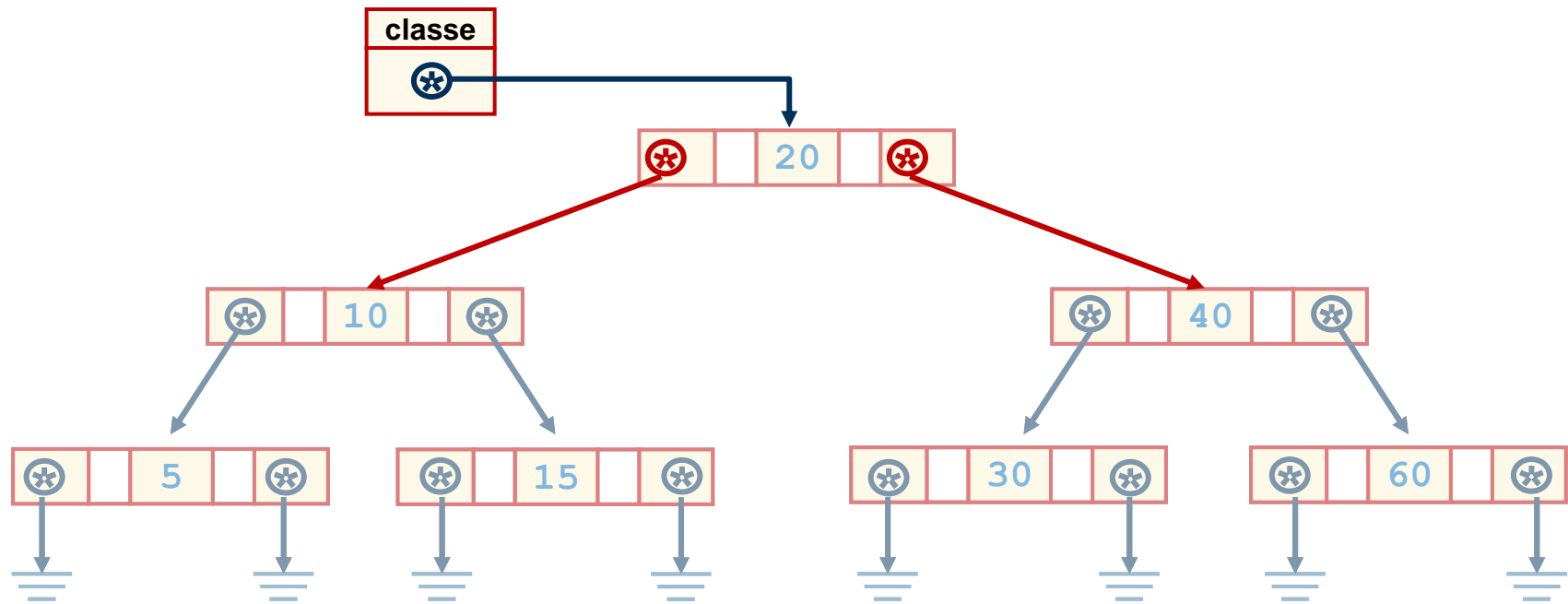
## Albero bilanciato

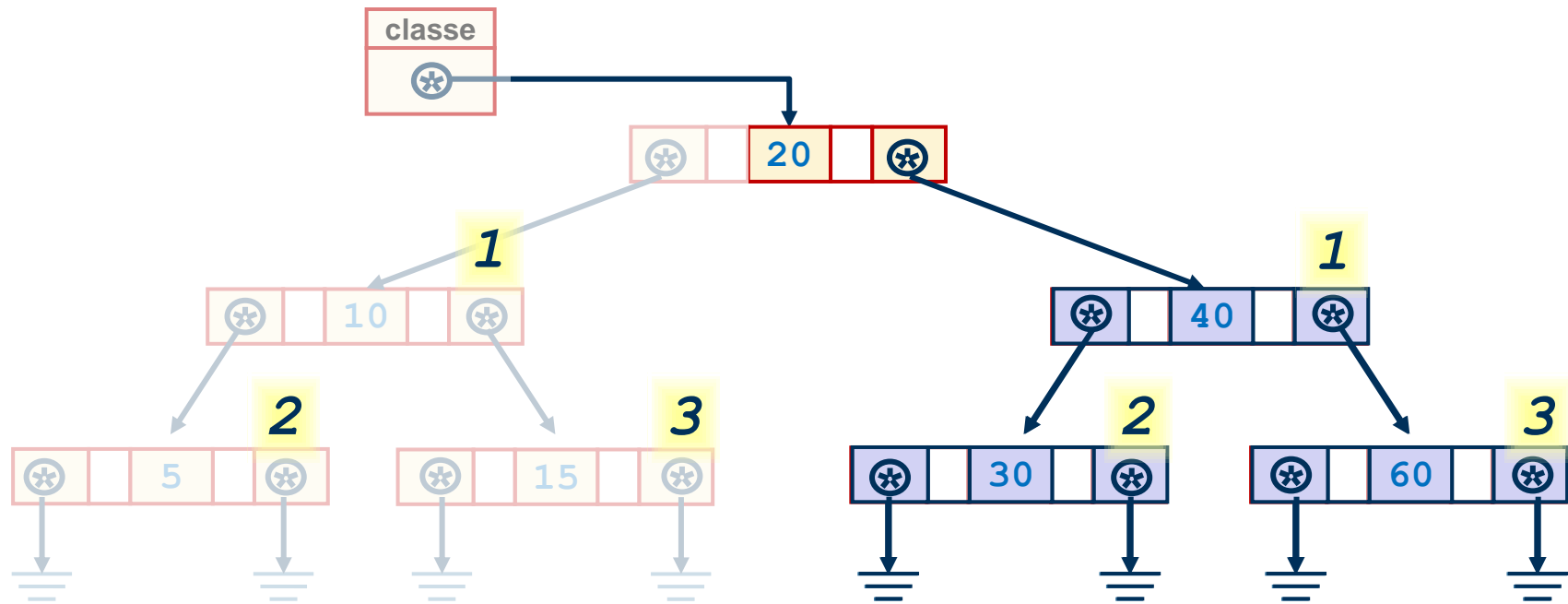


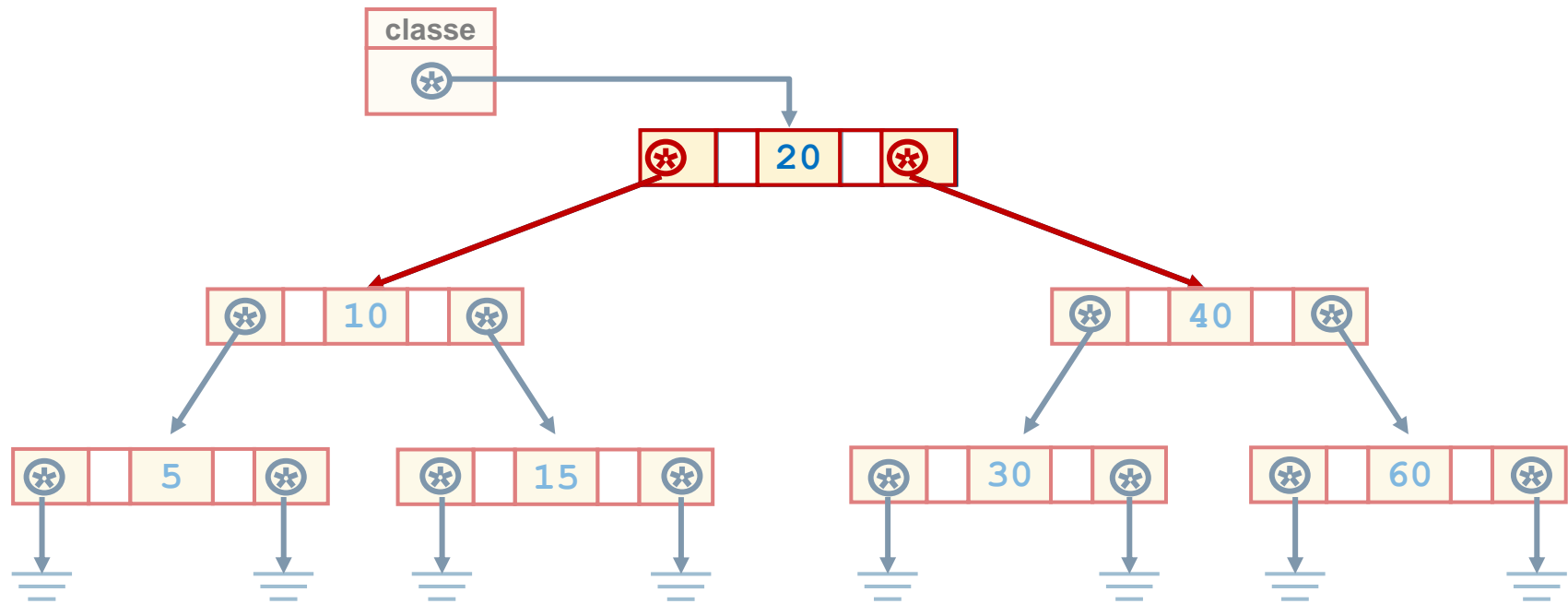


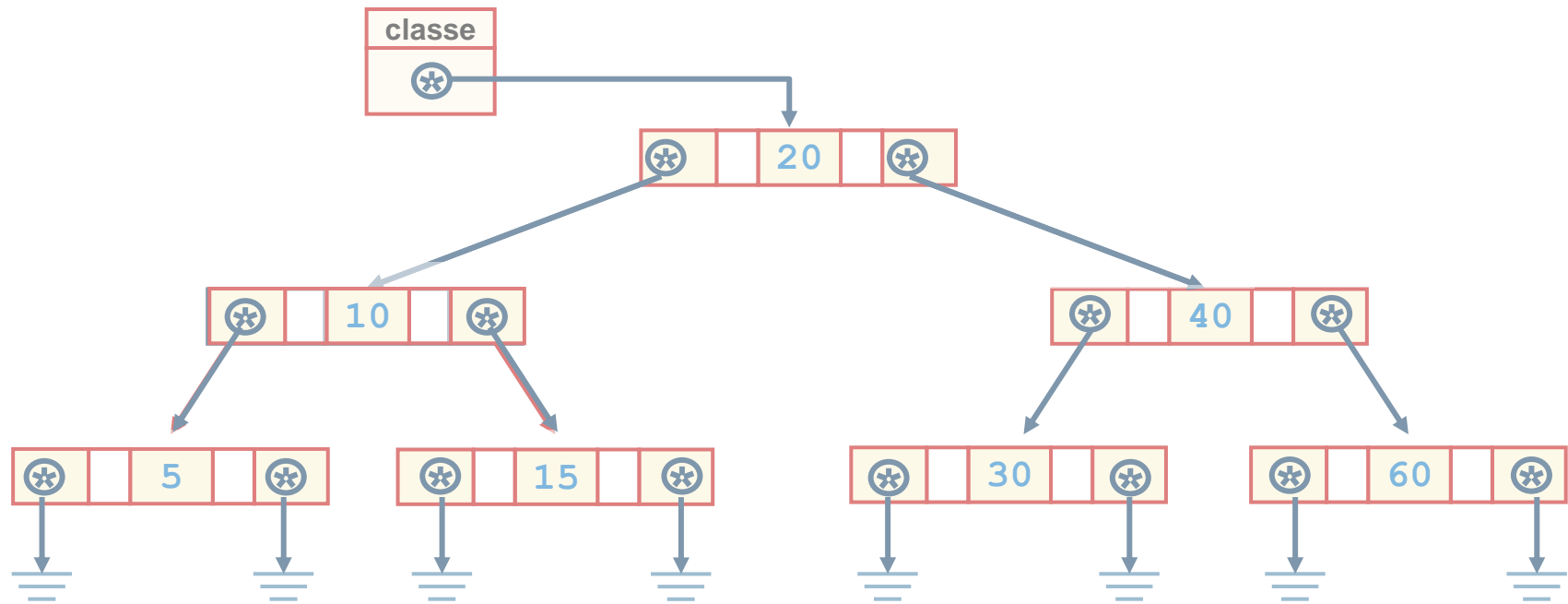


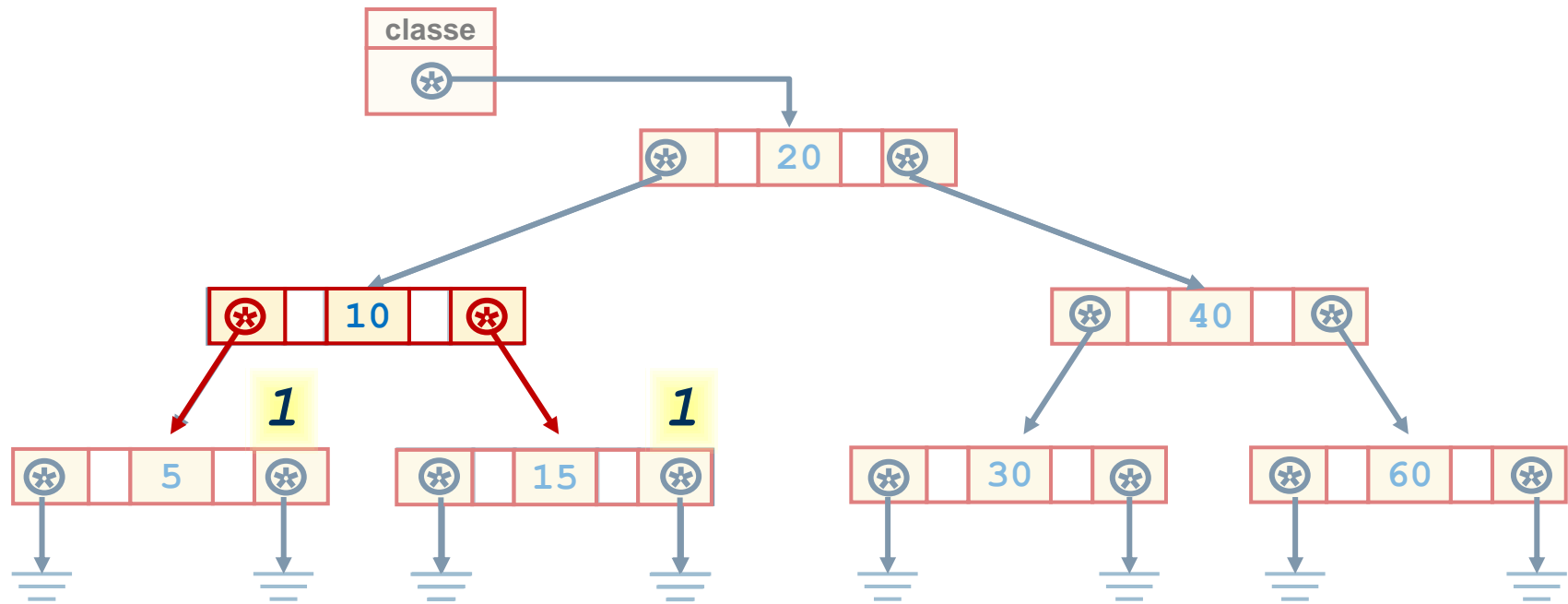


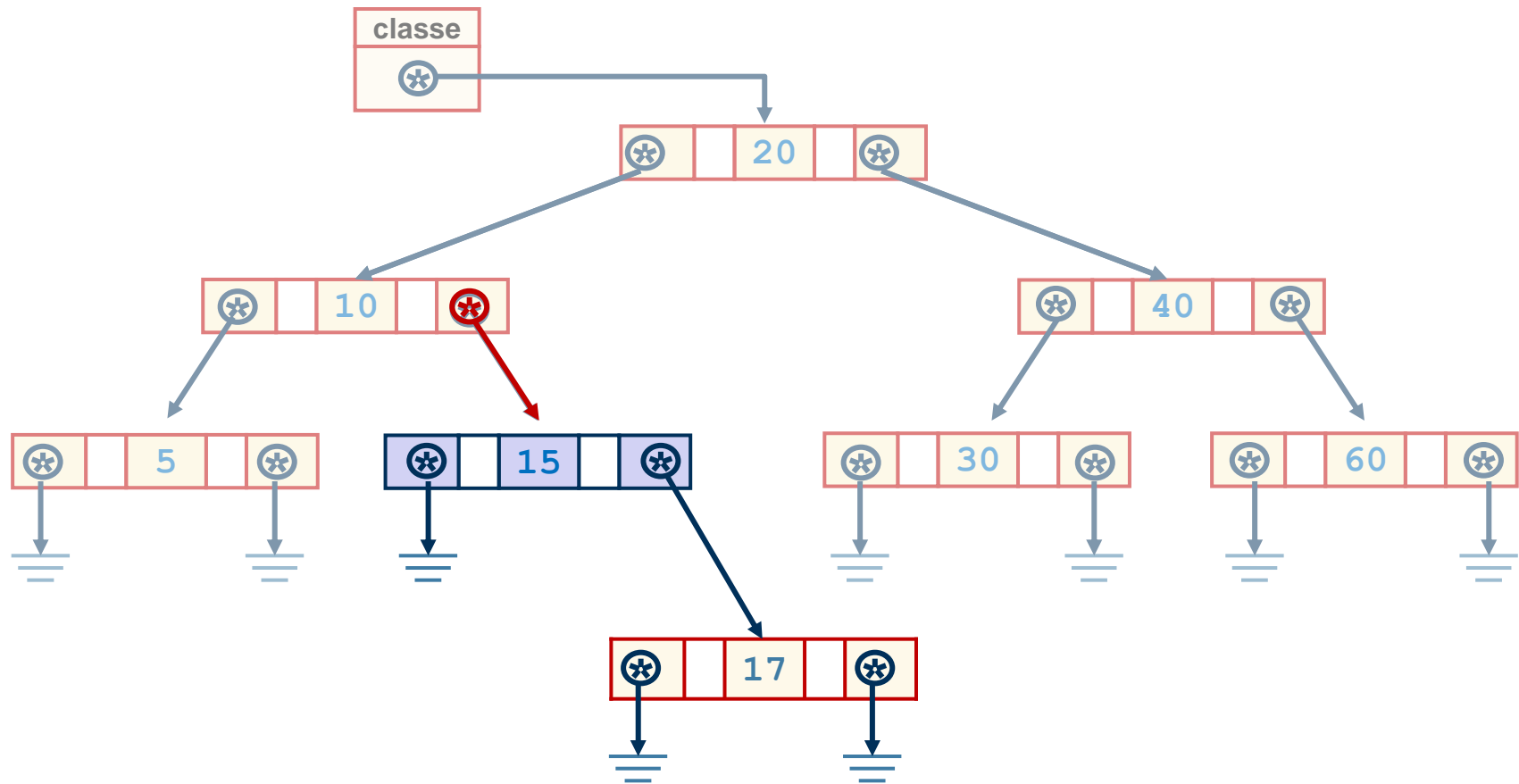




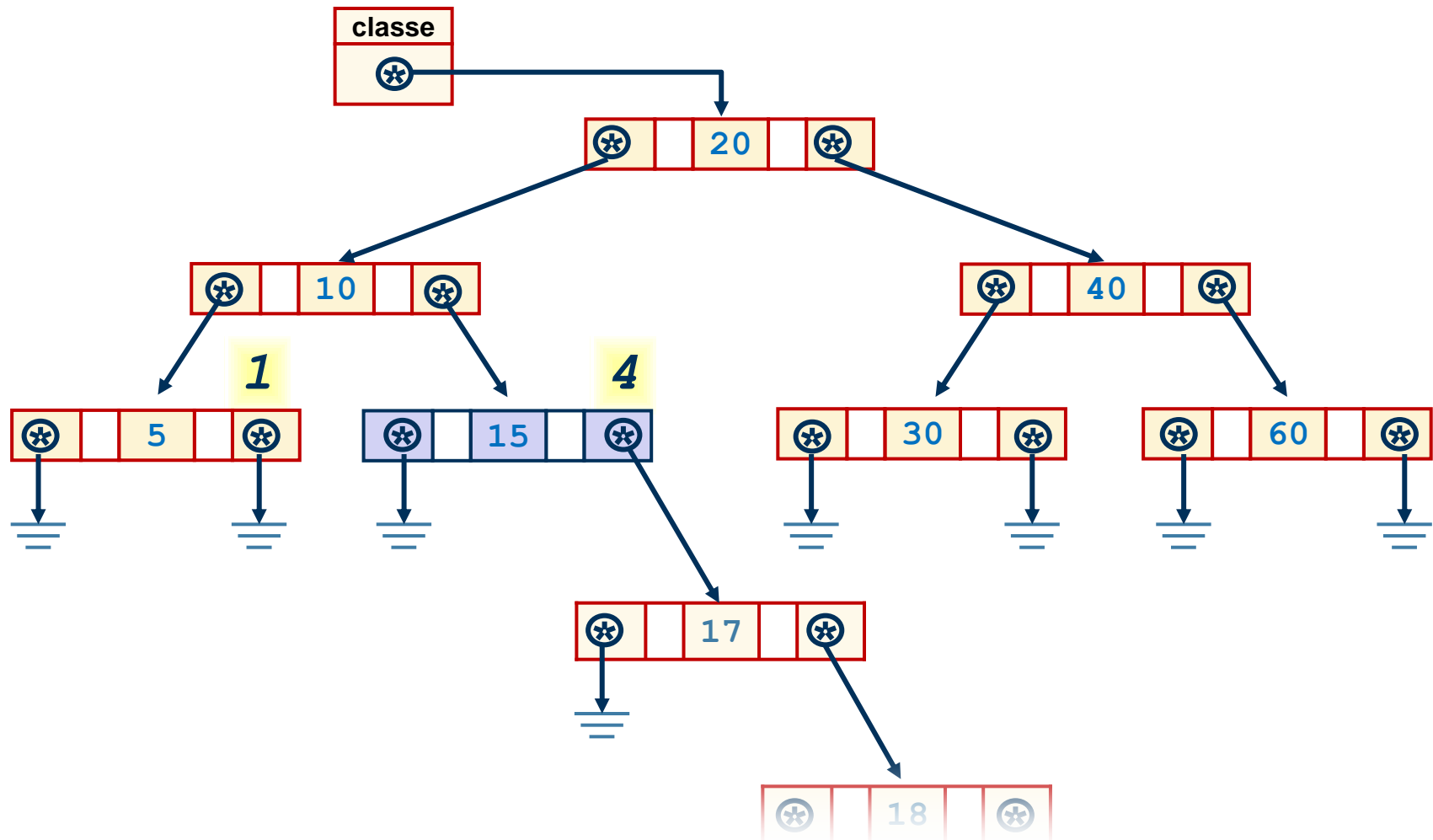






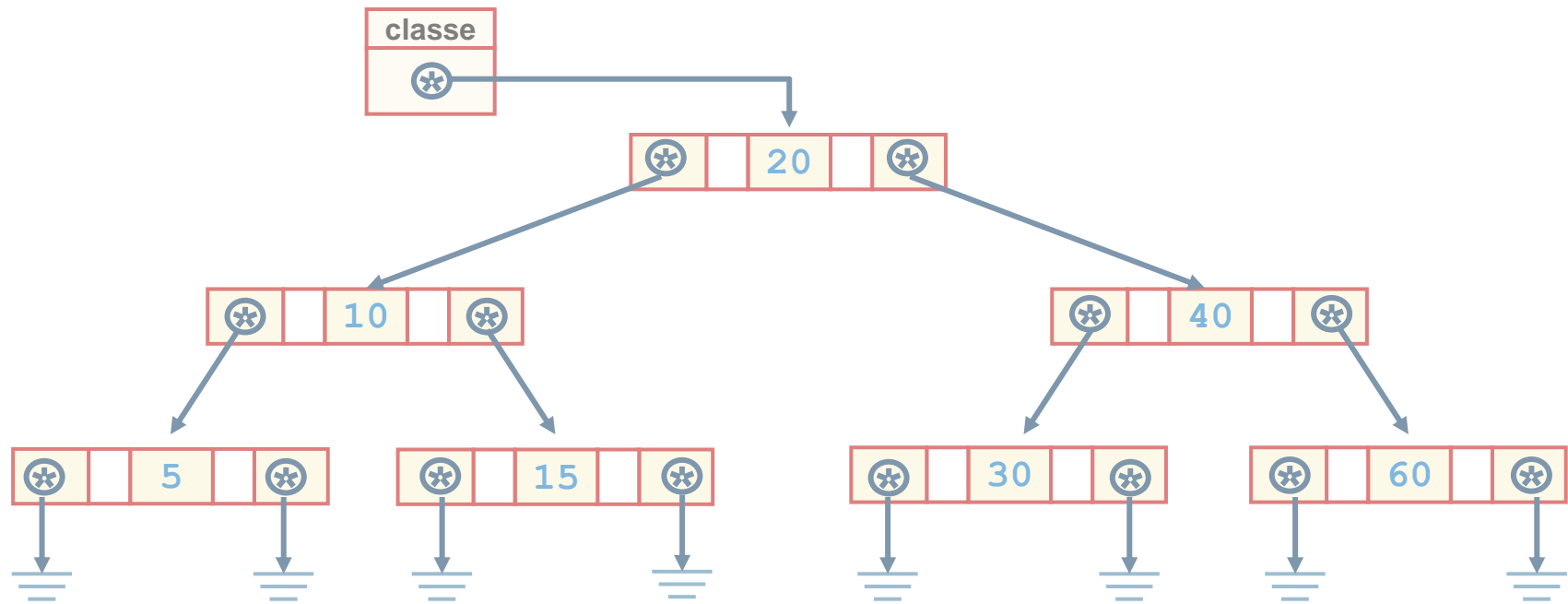


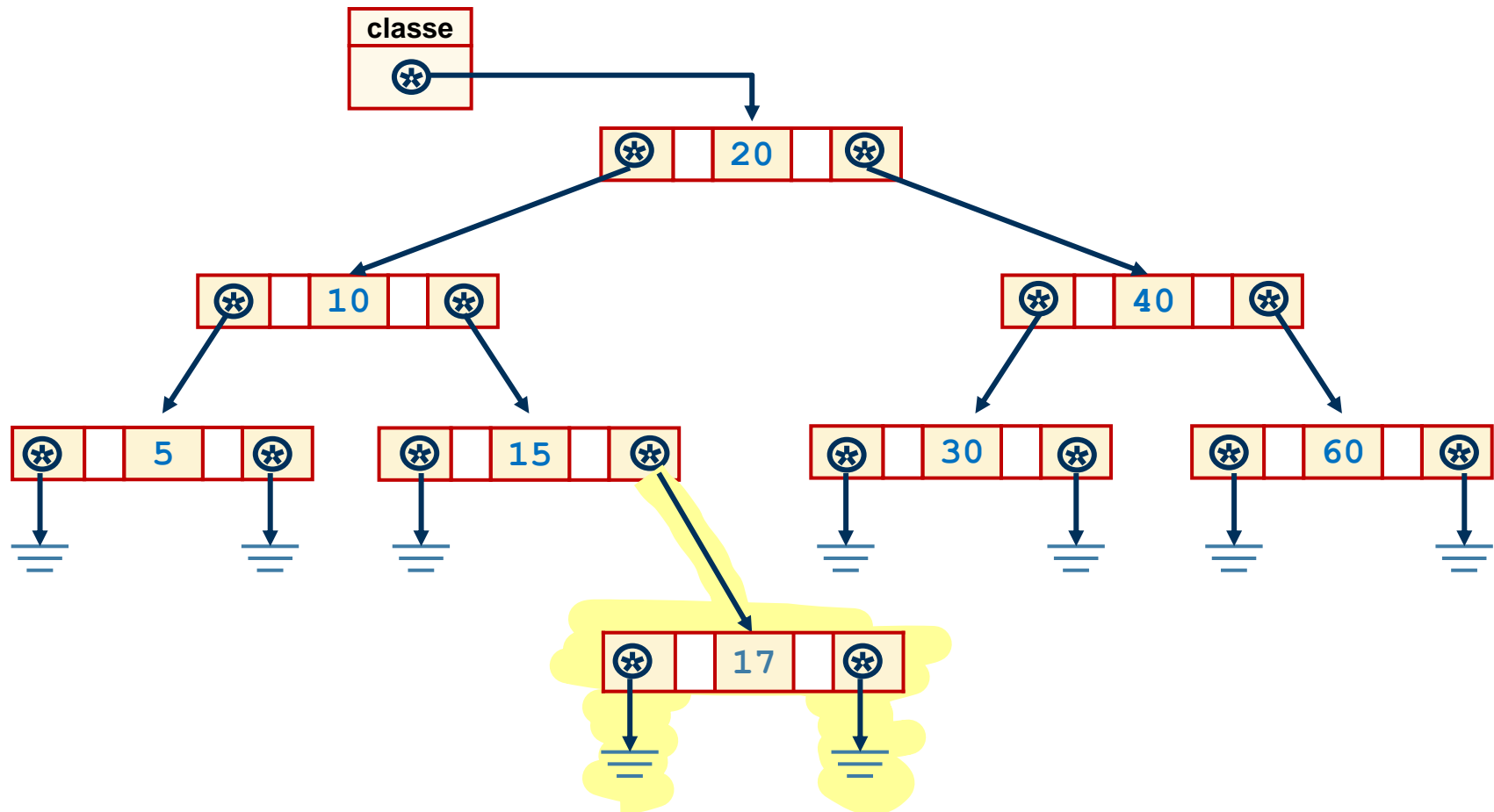




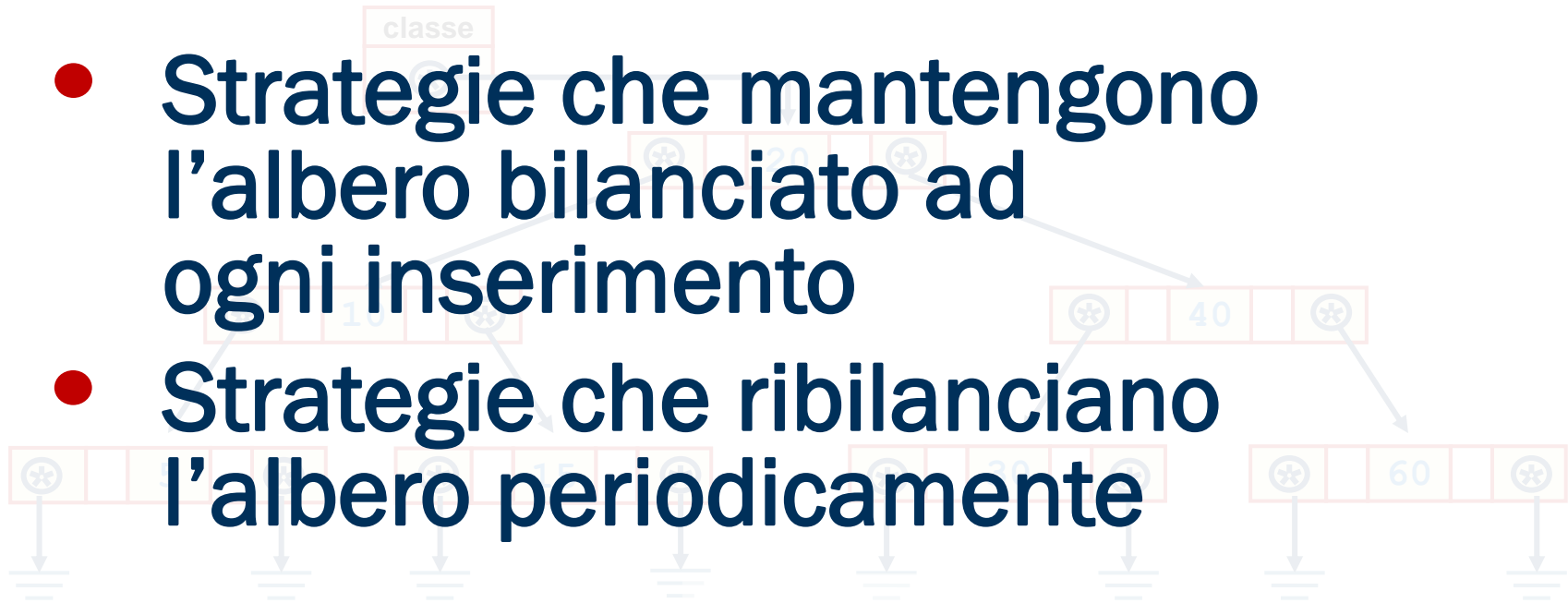
- Strategie che mantengono l'albero bilanciato ad ogni inserimento



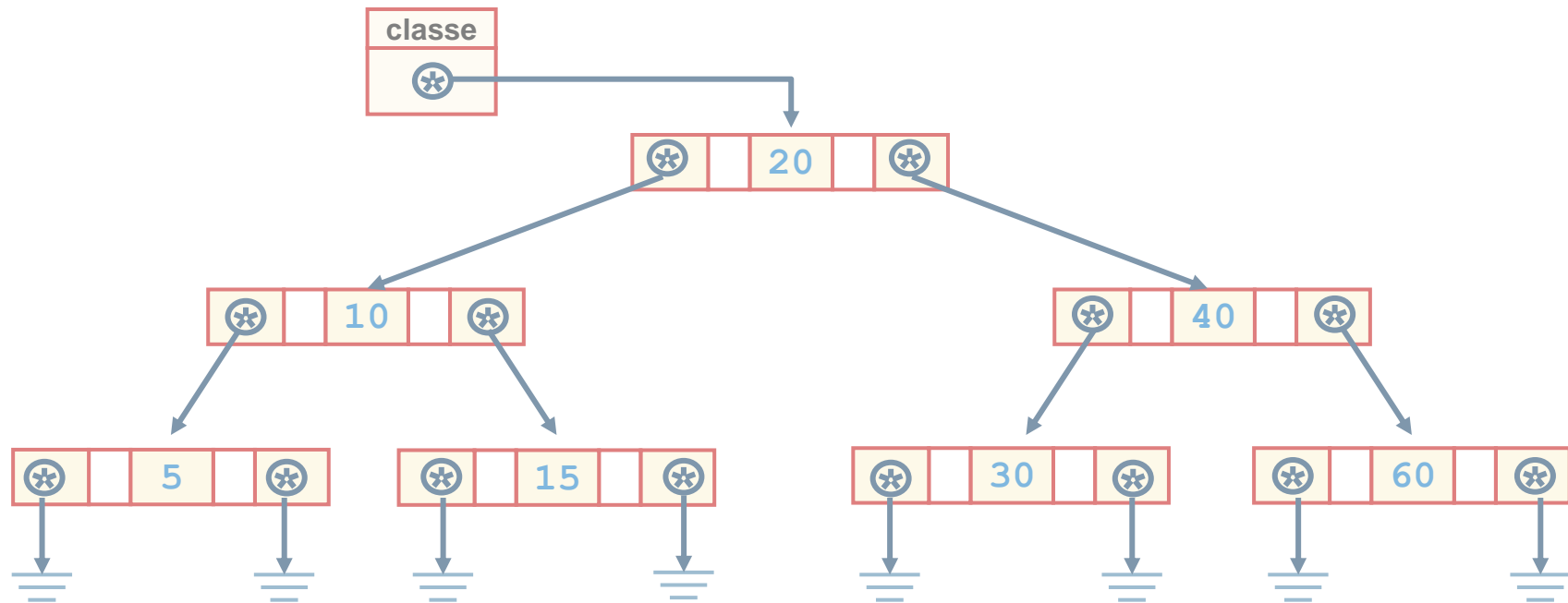




- Strategie che mantengono l'albero bilanciato ad ogni inserimento
- Strategie che ribilanciano l'albero periodicamente



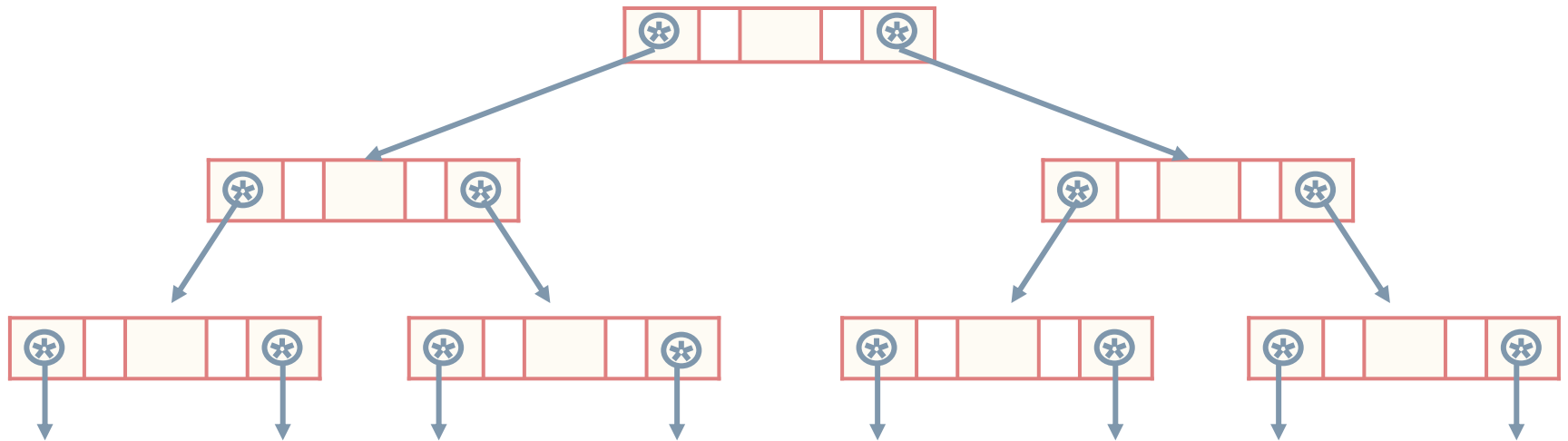
## Il codice



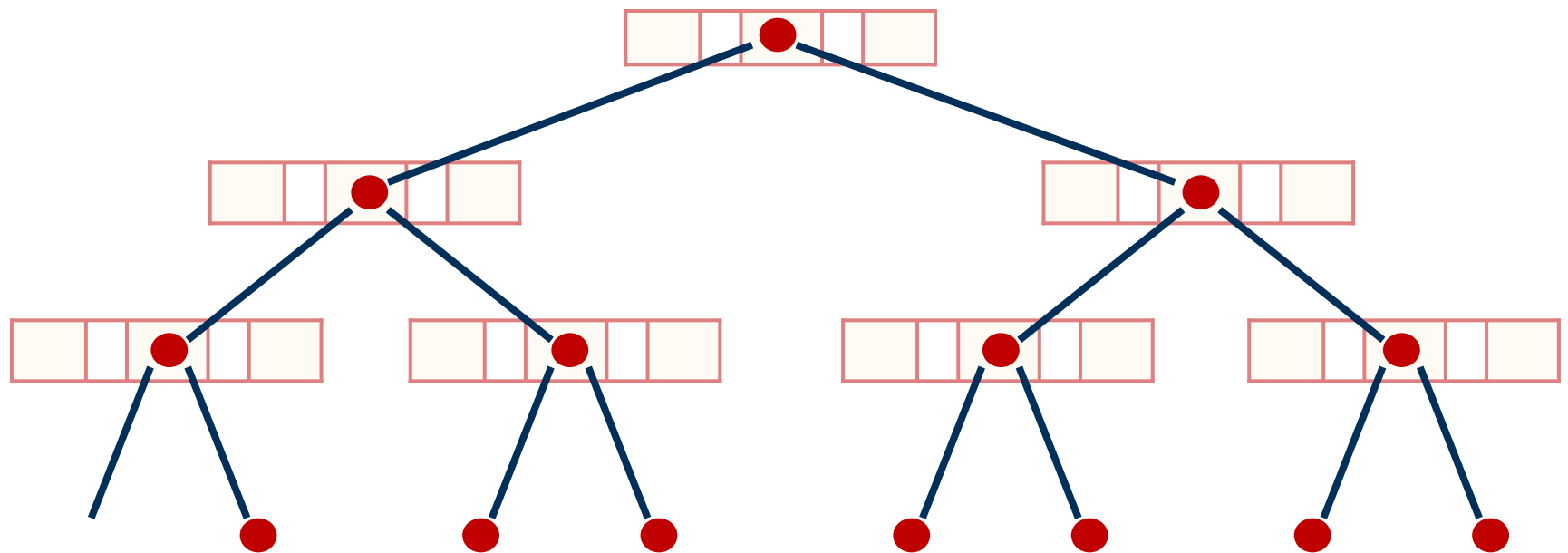
## Ricorsione

**inserisciSeNonEsiste**

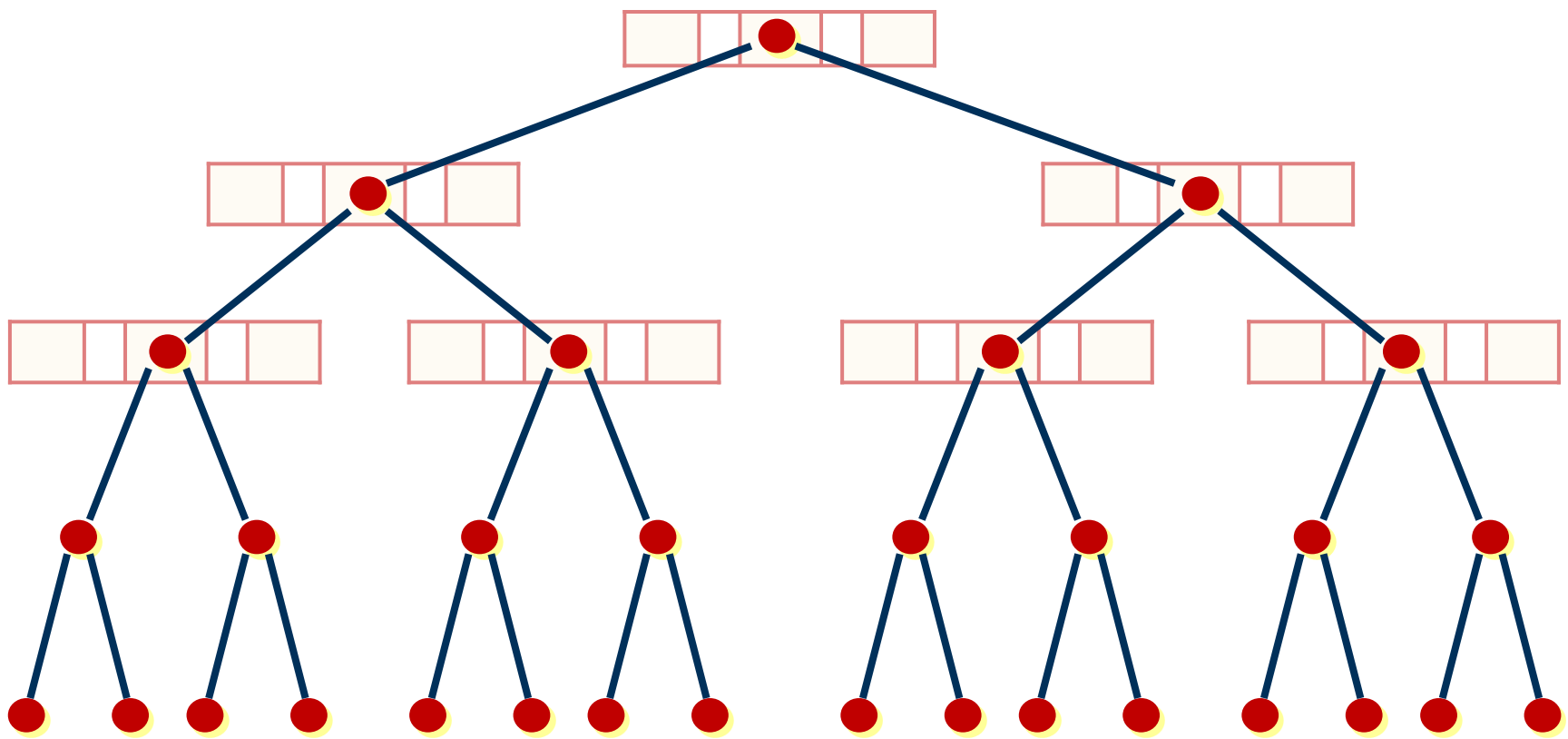
## Ricorsione



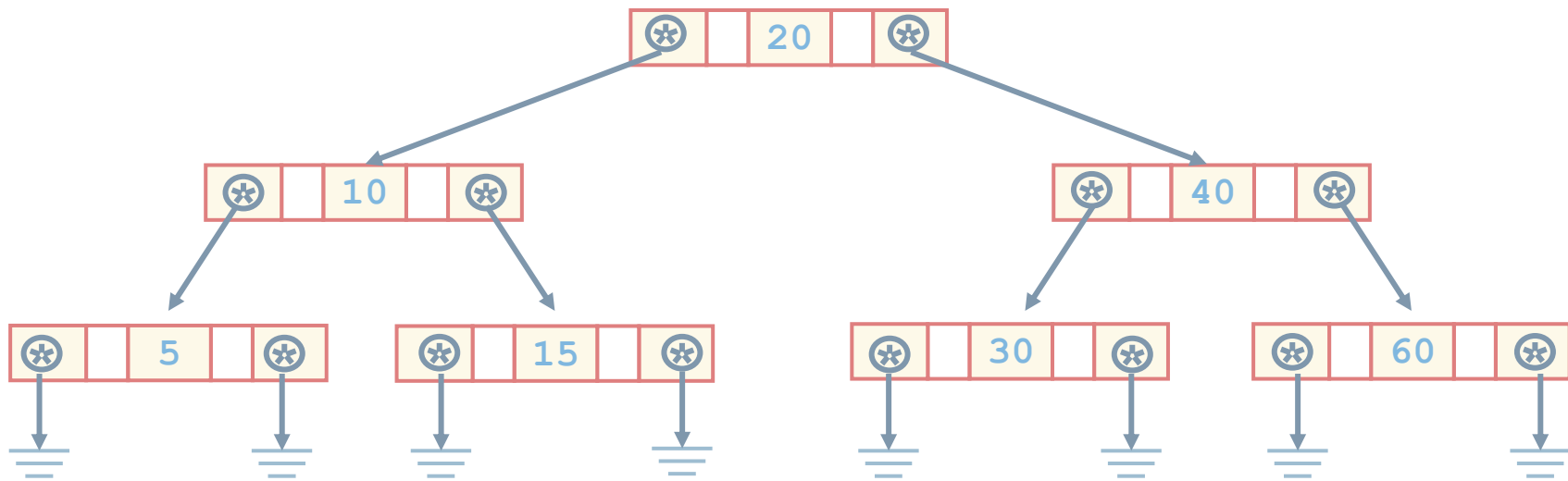




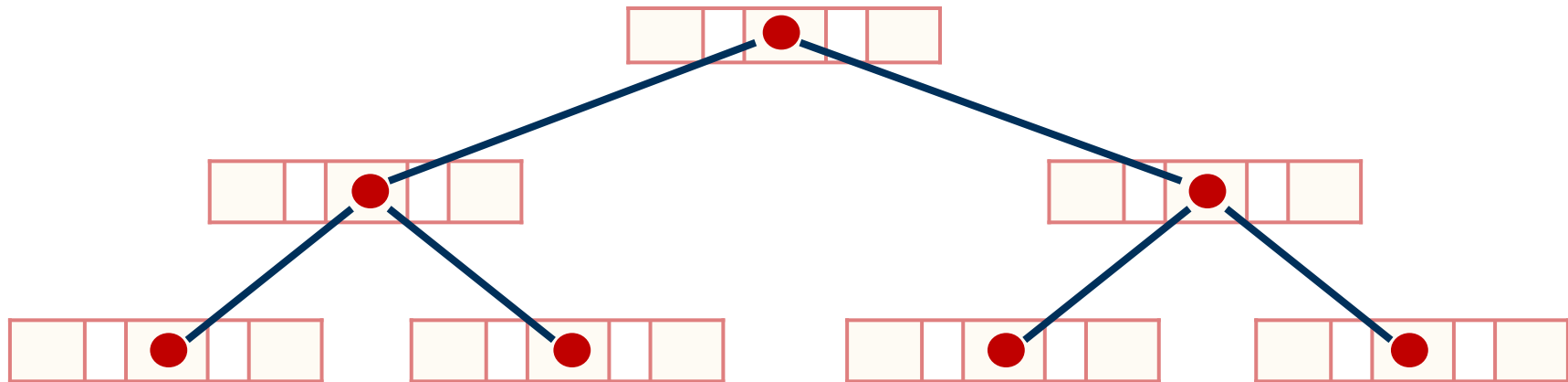
Stessa struttura



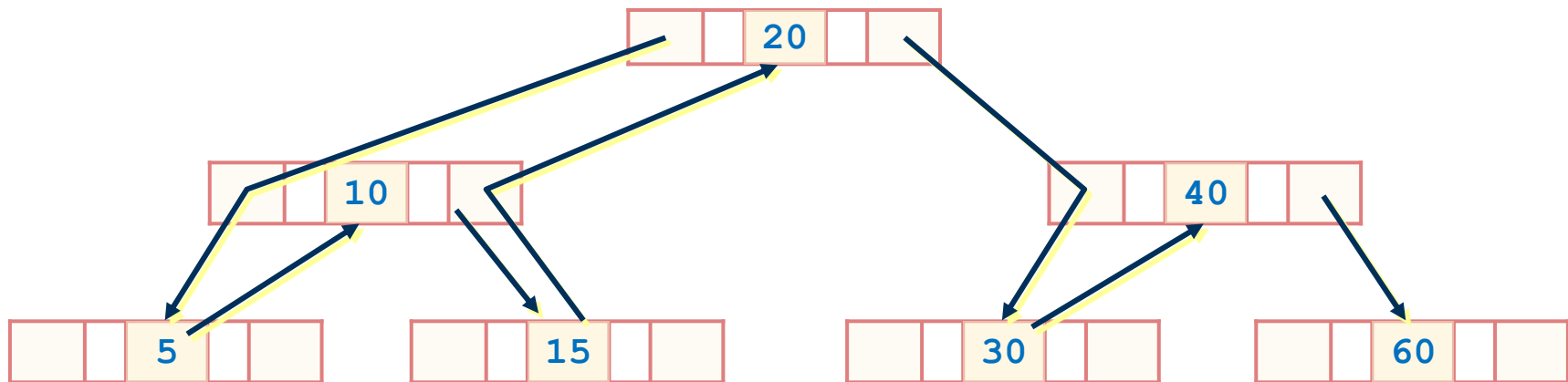
## Stampa ordinata per matricola

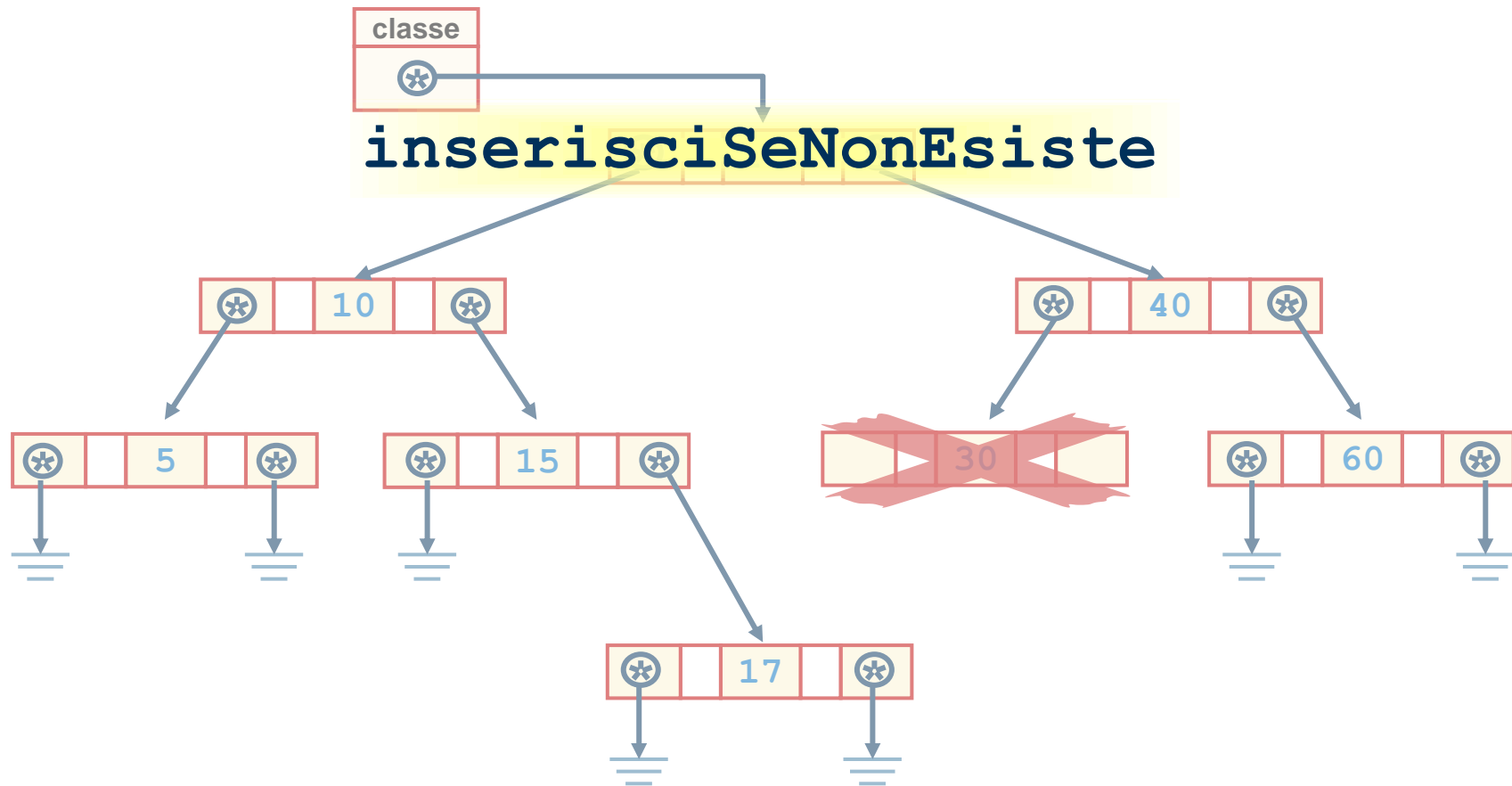


## Stampa ordinata per matricola



## Stampa ordinata per matricola



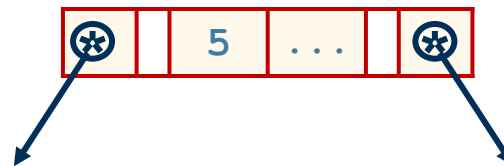


**inserisciSeNonEsiste**

- stato tabella
- dati studente

- aggiornamento tabella

```
struct TreeNode  
{  
  
};
```





```
struct TreeNode
{
    Studente datiStud;
    TreeNode *leftPtr;
    TreeNode *rightPtr;
};
```



```
struct TreeNode
{
    Studente datiStud;
    TreeNode *leftPtr;
    TreeNode *rightPtr;
};
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
```

```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
```

**inserisciSeNonEsiste**

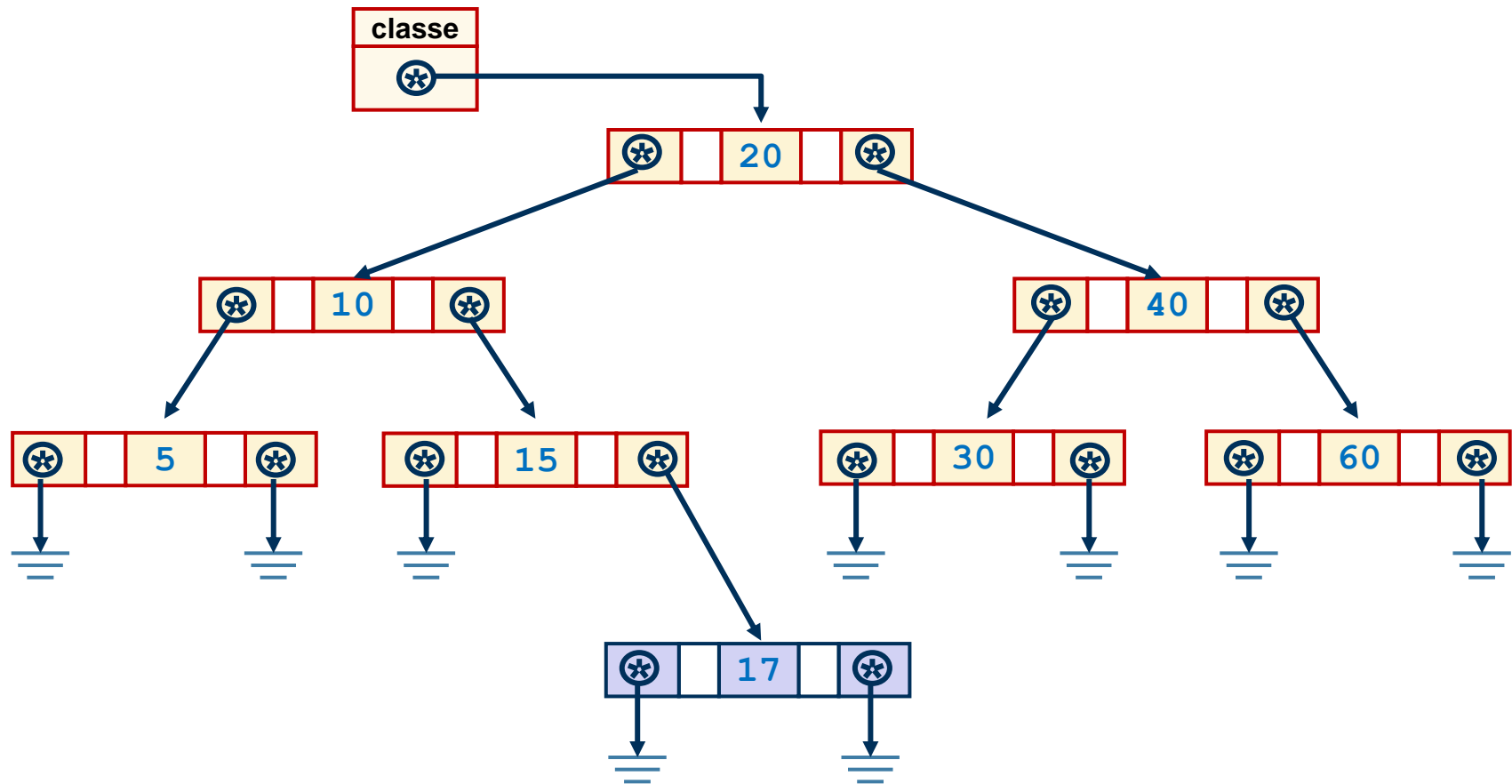
**cercaSeEsisteEPosizione**

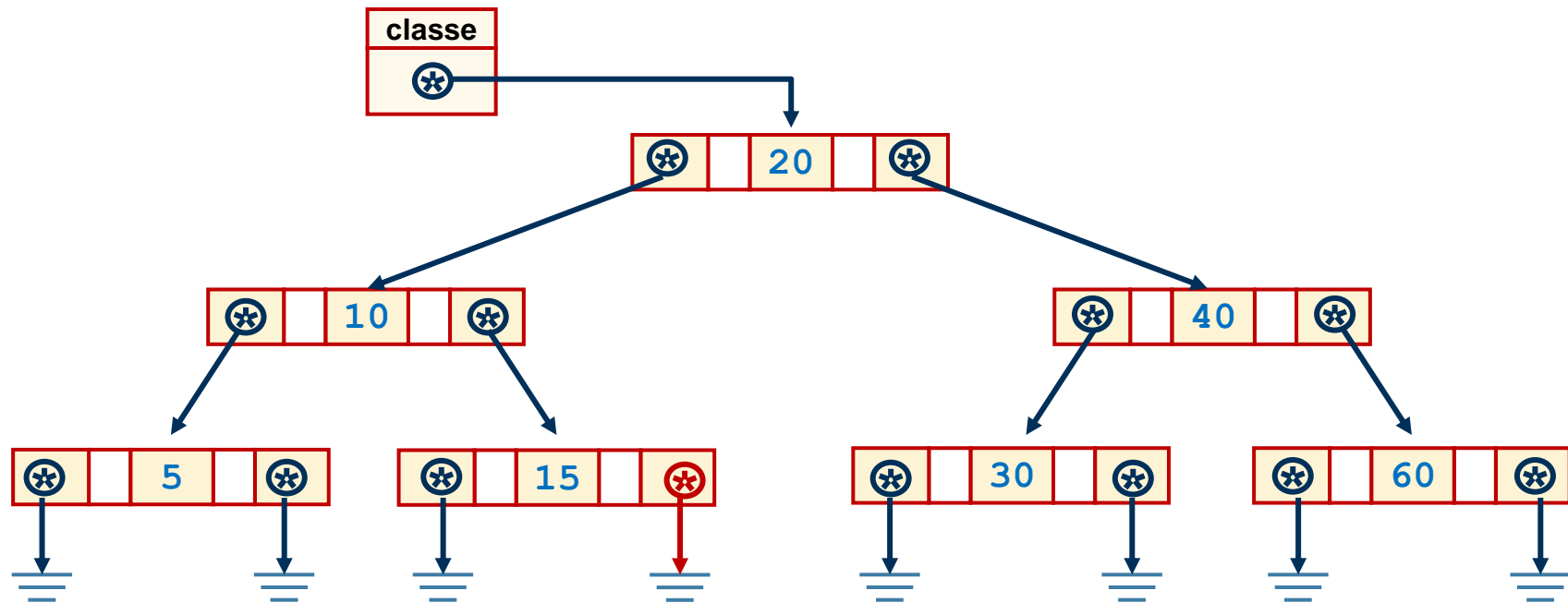


```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
```

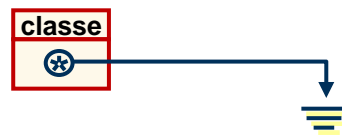
**inserisciSeNonEsiste**

Funzione ricorsiva



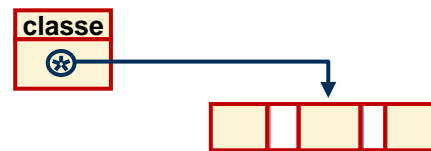


```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
```

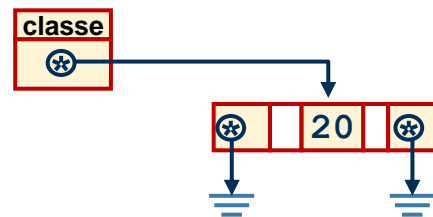




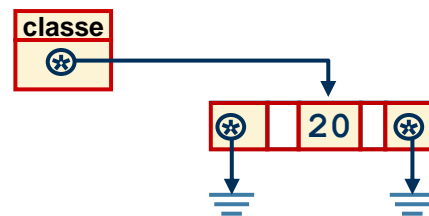
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
```



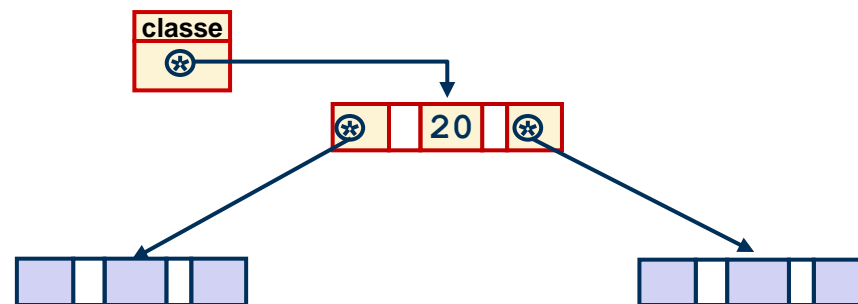
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
    }
}
```



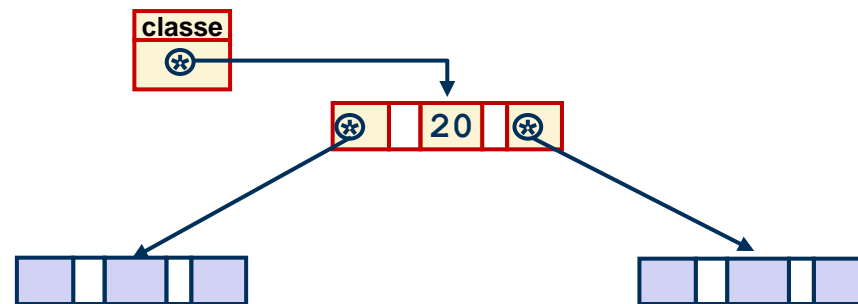
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
}
```

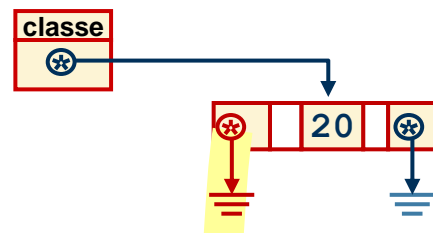


```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
}
```

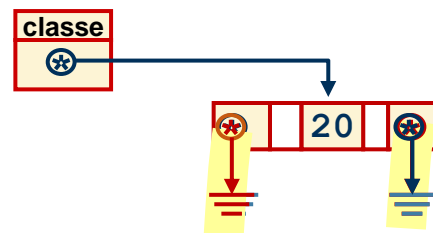
tab->leftPtr



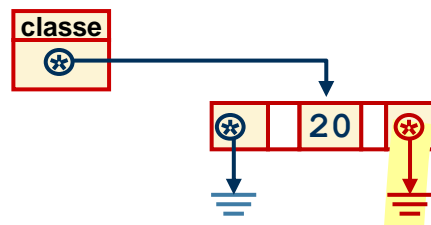
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
}
```

tab->leftPtr

tab->rightPtr

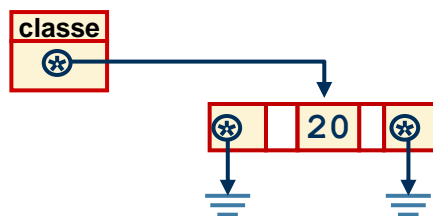


```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola
             tab->leftPtr
    else if (nuovoStudente.matricola
             tab->rightPtr
```

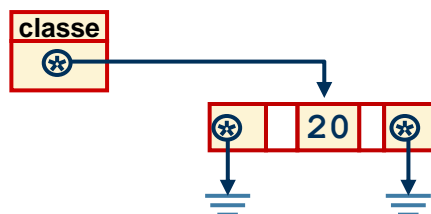




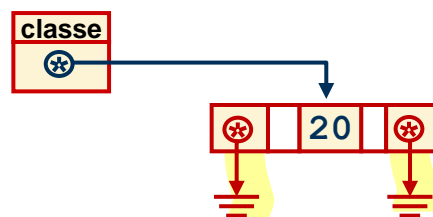
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



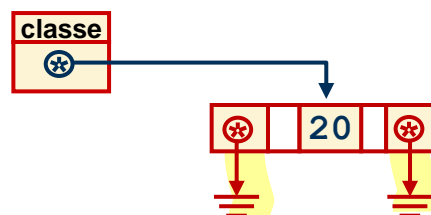
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste(tab->leftPtr, nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste(tab->rightPtr, nuovoStudente);
}
```



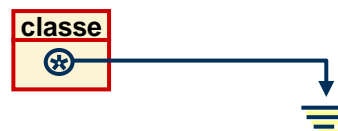
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



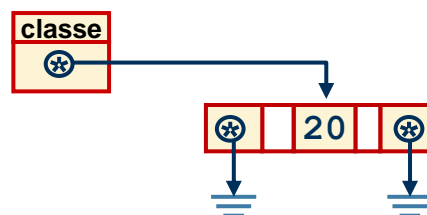
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr, nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr, nuovoStudente);
}
```



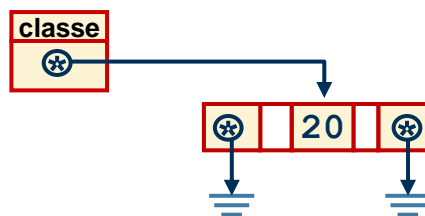
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



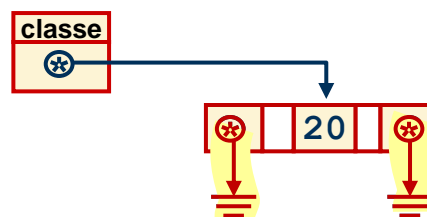
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```

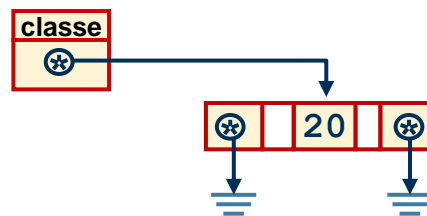


```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```

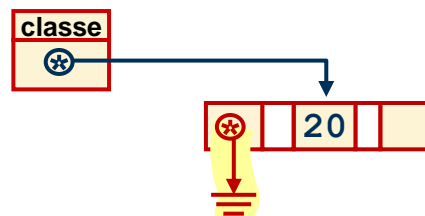




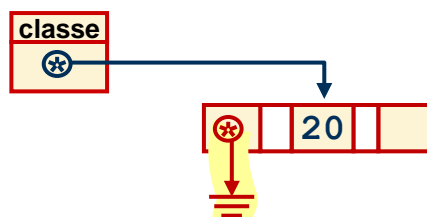
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



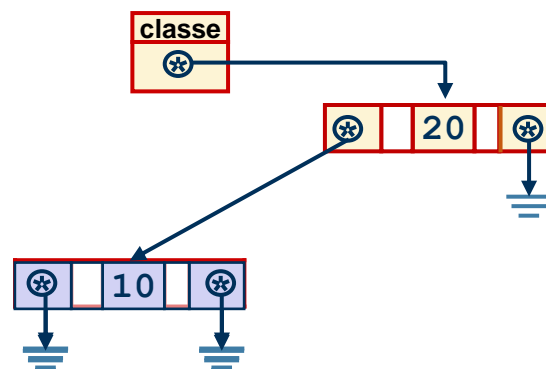
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



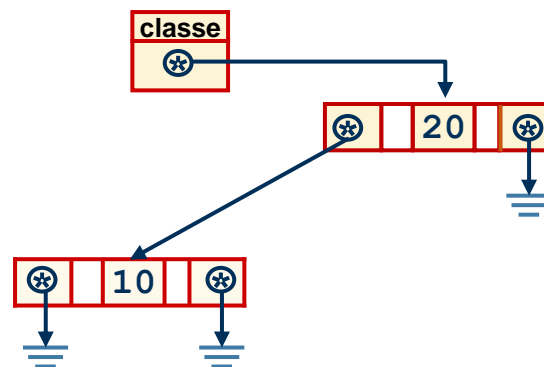
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



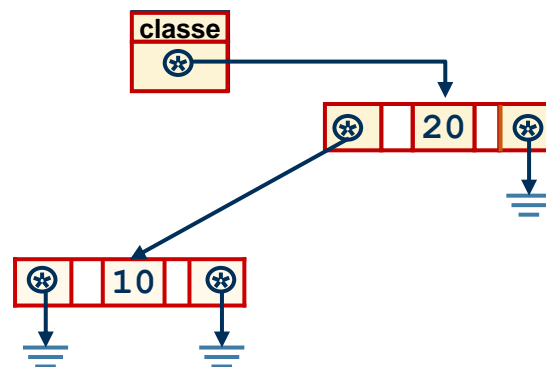
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



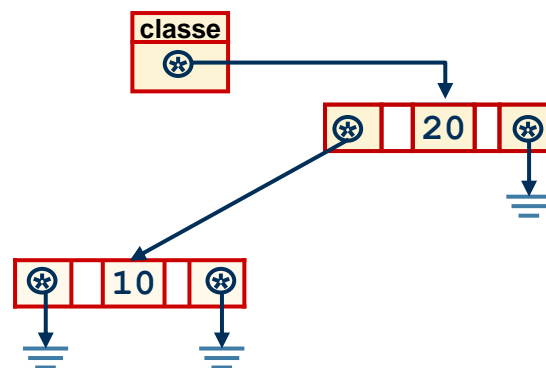
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



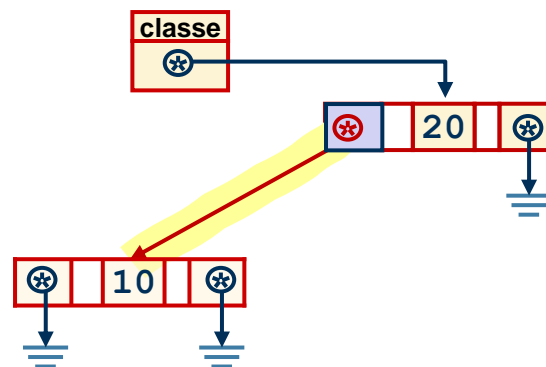
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```

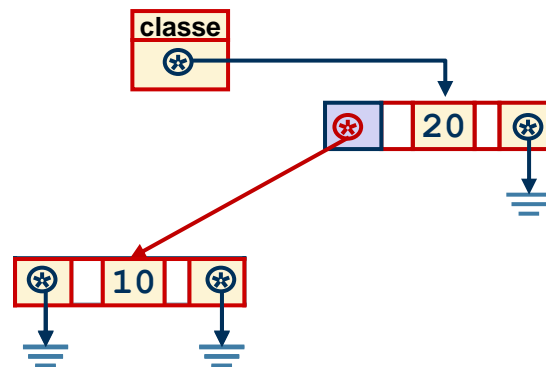


```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr, nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr, nuovoStudente);
}
```

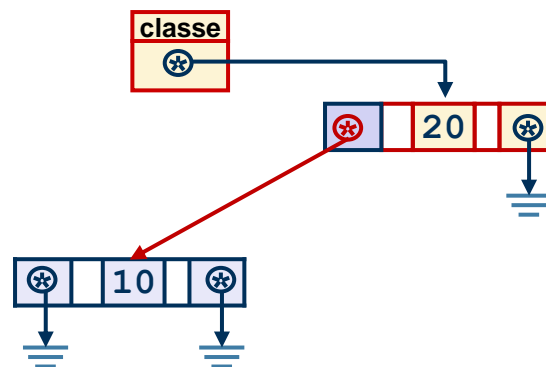




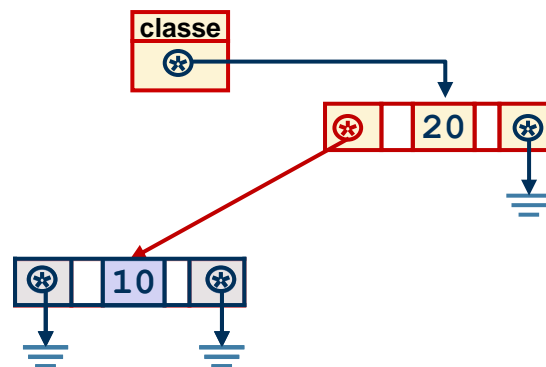
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



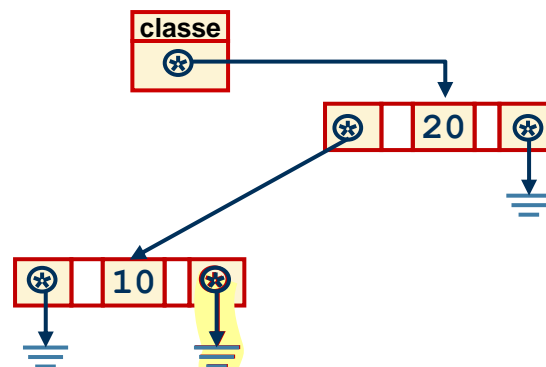
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



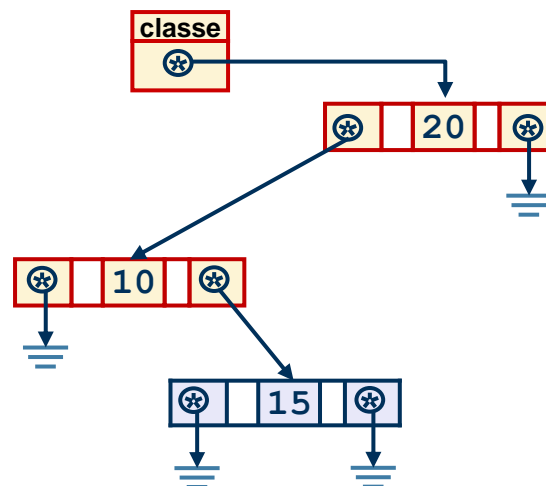
```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
}
```



```
inserisciSeNonEsiste(TreeNode *&tab, Studente nuovoStudente)
{
    if (tab == 0)
    {
        tab = new TreeNode;
        tab->datiStud = nuovoStudente;
        tab->leftPtr = 0;
        tab->rightPtr = 0;
    }
    else if (nuovoStudente.matricola < tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->leftPtr , nuovoStudente);
    else if (nuovoStudente.matricola > tab->datiStud.matricola)
        inserisciSeNonEsiste( tab->rightPtr , nuovoStudente);
    else
        cout << nuovoStudente.matricola << " duplicato" << endl;
}
```

