ВИРТУАЛНИ ДЕСТРУКТОРИ.

гл.ас., д-р. Нора Ангелова

ВИРТУАЛНИ ДЕСТРУКТОРИ

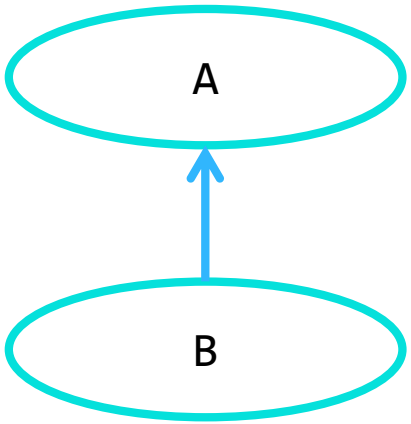
ВИРТУАЛНИ ДЕСТРУКТОРИ

```
A * ptr = new B("1234", “stringB");

delete ptr; // какво се извиква, какво се разрушава?
```

- Извиква се конструктор на A;
- Извиква се конструктор на B;
- Извиква се деструктор на A;
  ptr е от тип A*.
  Обръщението delete ptr;
  ще изпълни деструктора на A.

stringB не се разрушава

```
A::A(char* s) {
  x = new char[strlen(s)+1];
  assert(x != NULL);
  strcpy(x, s);
}
```




```
B::B(char* a, char* b) : A(a) {
  x = new char[strlen(b)+1];
  assert(x != NULL);
  strcpy(x, b);
}
```

ВИРТУАЛНИ ДЕСТРУКТОРИ

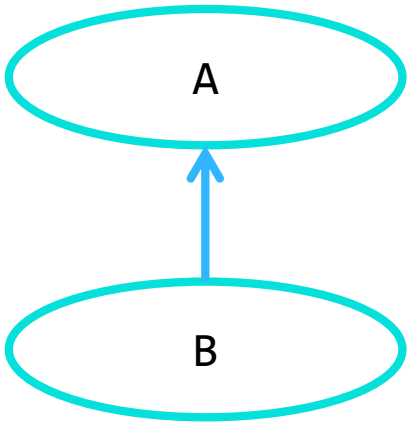
```
A * ptr = new B("1234", “stringB");

delete ptr; // какво се извиква, какво се разрушава?
```

Решение:

- Деструкторът на базовия клас A на йерархията се обяви за виртуален.

Обявяването на деструктор на клас на йерархия за виртуален причинява деструкторите на всички класове в наследствената за този основен клас йерархия да са виртуални.

```
A::A(char* s) {
  x = new char[strlen(s)+1];
  assert(x != NULL);
  strcpy(x, s);
}
```




```
B::B(char* a, char* b) : A(a) {
  x = new char[strlen(b)+1];
  assert(x != NULL);
  strcpy(x, b);
}
```

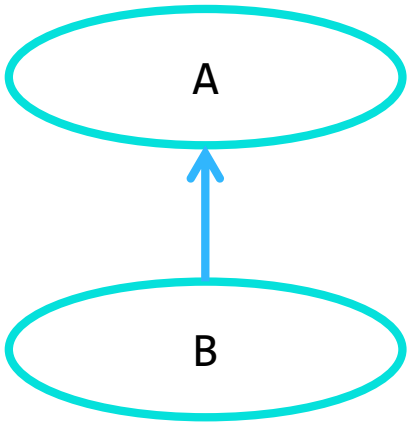
ВИРТУАЛНИ ДЕСТРУКТОРИ

```
A * ptr = new B("1234", “stringB");

delete ptr; // какво се извиква, какво се разрушава?
```

Решение:

- Обявяваме деструктора на класа A за виртуален.
  `virtual ~A();`
  `A::~A() {…}`
- Деструкторът на класа B автоматично става виртуален.
- delete ptr; ще изпълни:
  - деструктора на класа B,
  - деструктора на класа A,
  - ще разруши връзката на ptr с ДП.

```
A::A(char* s) {
  x = new char[strlen(s)+1];
  assert(x != NULL);
  strcpy(x, s);
}
```




```
B::B(char* a, char* b) : A(a) {
  x = new char[strlen(b)+1];
  assert(x != NULL);
  strcpy(x, b);
}
```

КРАЙ