

СТАТИЧНО И ДИНАМИЧНО
СВЪРЗВАНЕ.
ПОЛИМОРФИЗЪМ.
АБСТРАКТЕН КЛАС.
ВИРТУАЛНИ ДЕСТРУКТОРИ.

гл.ас., д-р. Нора Ангелова

СТАТИЧНО И ДИНАМИЧНО СВЪРЗВАНЕ

СТАТИЧНО И ДИНАМИЧНО СВЪРЗВАНЕ

- Статично свързване - изборът на функцията, която трябва да се изпълни става по време на компилация.

```
point3 p3(1,2,3);  
colPoint3 p4(1,2,3,1);  
point2 * point2Ptr = &p3;  
point2Ptr->print(); // print на point2
```

```
point2 * point2Ptr2 = &p4;  
point2Ptr2->print(); // print на point2
```

- Динамино свързване - изборът на функцията, която трябва да се изпълни става по време на изпълнение на програмата.

СТАТИЧНО И ДИНАМИЧНО СВЪРЗВАНЕ

Динамично свързване

- Разширяването на йерархията не създава проблеми.
- Не се налага проверка на типа.
- Усложняване на кода и забавя процеса на изпълнение на програмата.
- Реализира се чрез специални член-функции на класове - **виртуални член-функции**.
- Виртуалните функции се декларират чрез поставяне на запазената дума **virtual**.

`virtual <тип_на_резултата> <име_на_метод>(<параметри>);`

СТАТИЧНО И ДИНАМИЧНО СВЪРЗВАНЕ

Динамино свързване

// Декларация на print в point2, point3, colPoint3

```
virtual void print();
```

```
point3 p3(1, 2, 3);
```

```
colPoint3 p4(1, 2, 3, 1);
```

```
point2 * ptr = &p3;
```

- Ще се определи по време на изпълнението на програмата
- Определянето е в зависимост от класа на ОБЕКТА

```
ptr->print(); // print на point3
```

```
point2 * ptr2 = &p4;
```

```
ptr2->print(); // print на colPoint3
```

СТАТИЧНО И ДИНАМИЧНО СВЪРЗВАНЕ

Динамично свързване

1. Само член-функциите на класовете могат да се декларират като виртуални.
2. Ако функция е обявена за **виртуална в основния клас**, декларираните член-функции в производните класове със същия прототип **също са виртуални** дори ако запазената дума бъде пропусната.
3. Ако в производен клас се дефинира виртуална функция, която има същия прототип като неавиртуална функция в основния клас, двете функции се интерпретират като различни член-функции.
4. Възможно е виртуална функция **да се дефинира** извън клас. Тогава не започва със запазената дума `virtual`.
5. Виртуалните член-функции се наследяват като останалите компоненти на класа.
6. Основният клас, в който член-функция е обявена за виртуална, трябва да е с атрибут **public** в производните от него класове.
7. Виртуалните член-функции се извикват чрез указател или псевдоним на обект на някакъв клас.
8. Виртуалната член-функция, която в действителност се изпълнява, зависи от класа на обекта, към който сочи указателят.

ПОЛИМОРФИЗЪМ

- ⦿ Едни и същи действия се реализират по различен начин в зависимост от обектите, върху които се прилагат.
- ⦿ Действията се наричат полиморфни.
- ⦿ Свойство на член-функциите на класовете.
- ⦿ Реализира се чрез виртуални функции.
- ⦿ Класовете, върху които ще се прилага, трябва да имат общ родител или прародител, т.е. да са производни на един и същ клас.
- ⦿ В класа се дефинира виртуален метод, съответстващ на полиморфното действие.
- ⦿ Всеки клас предефинира **или не** виртуалния метод.
- ⦿ Активирането става чрез **указател към базов клас**, на който може да се присвоят адресите на обекти на който и да е от производните класове от йерархията.
- ⦿ Ще се изпълни методът на съответния **обект**.

АБСТРАКТЕН КЛАС

- Ако класовете, в които трябва се дефинират виртуални методи, нямат общ родител, такъв може да бъде създаден **изкуствено** чрез т.нар. **абстрактен клас**.
- Клас, в който има поне една чисто виртуална функция.

virtual <тип_на_резултата> <име_на_метод>(<параметри>) = 0;

- Не могат да се създават обекти от тези класове, но могат да се дефинират указатели към такива класове.
- Чисто виртуалните функции **задължително** трябва да бъдат предефинирани в производните класове **или да бъдат обявени като чисто виртуални** в тях.

ВИРТУАЛНИ ДЕСТРУКТОРИ

```
base * b = new der(1,2);
```

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

КРАЙ