Полиморфизъм. Абстрактен клас.

гл.ас., д-р. Нора Ангелова

ПОЛИМОРФИЗЪМ

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.

ПОЛИМОРФИЗЪМ

- Реализира се чрез виртуални функции.
- За да се реализира полиморфно действие, 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.

ПОЛИМОРФИЗЪМ

Да разгледаме пример за полиморфно действие с класове от животни:

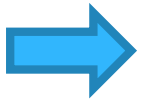
- Класове за котка, зайче и мишка;
- Животните могат да издават звук;
- Класовете притежават метод за извеждане на звук, който се издава от съответното животно;

- Извикване на методите?

- Искаме да създадем множество от различни животни, които издават звуци.

ПОЛИМОРФИЗЪМ

```
class zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n";
    cout << "Address:\n Sofia, Bulgaria\n";
  }
};
class cat : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Cat\n";
  }
};
class mouse : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Mouse\n";
  }
};
class rabbit : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Rabbit \n";
  }
};
```



```
class zooAnimal {
  public:
  virtual void print() const {
    cout << "ZooAnimal\n";
    cout << "Address:\n Sofia, Bulgaria\n";
  }
};
class cat : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Cat\n";
  }
};
class mouse : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Mouse\n";
  }
};
class rabbit : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Rabbit \n";
  }
};
```

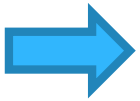
ПОЛИМОРФИЗЪМ

Друг пример за полиморфно действие:

- Класове с член-функции с еднакви прототипи;
- Член-функциите извършват еднотипни действия;
- Член-функциите на производните класове обикновено извършват редица общи действия.
- Метод за извеждане на името на зоологическата градина и животното.
- В този случай в основния клас може да се реализира една невиртуална функция, която извършва общите действия и след (или преди) това извиква виртуалната функция, извършваща специфичните действия за класовете.

ПОЛИМОРФИЗЪМ

```
class zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n";
    cout << "Address:\n Sofia, Bulgaria\n";
  }
};
class cat : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Cat\n";
  }
};
class mouse : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Mouse\n";
  }
};
class rabbit : public zooAnimal {
  public:
  void print() const {
    cout << "ZooAnimal\n" << "Rabbit \n";
  }
};
```



```
class zooAnimal {
  public:
  virtual void spec() const {
    cout << "Address:\nSofia, Bulgaria\n";
  }

  void print() const {
    cout << "ZooAnimal\n";
    spec(); // разрешава се динамично
  }
};
class cat : public zooAnimal {
  public:
  virtual void spec() const {
    cout << "Cat\n";
  }
};
class mouse : public zooAnimal {
  public:
  virtual void spec() const {
    cout << "Mouse\n";
  }
};
class rabbit : public zooAnimal {
  public:
  virtual void spec() const {
    cout << "Rabbit\n";
  }
};
```

ПОЛИМОРФИЗЪМ

Съществуват три случая, при които обръщение към виртуална член-функция се разрешава статично:

- Виртуалната функция се извиква чрез обект на класа, в който е дефинирана.

```
cat myCat;
myCat.spec();
```

- Виртуалната член-функция се активира чрез указател към или чрез псевдоним на обект, но явно, чрез оператора ::, е посочена конкретната виртуална член-функция.

```
zooAnimal *animalPtr = &myCat;
animalPtr->spec();              // динамично свързване
animalPtr->zooAnimal::spec(); // статично свързване
```

- Виртуалната член-функция се активира в тялото на конструктор или деструктор на основен клас.

  В този случай се изпълнява виртуалната член-функция на основния клас. Това е така, защото виртуалната функция в конструктора или деструктора на основния клас се извиква когато обектът от производния клас още не е създаден или вече е разрушен.

Полиморфизъм

- Ако класовете, над които ще се реализира полиморфно действие, нямат общ родител, такъв може да бъде създаден изкуствено чрез дефиниране на т.н. абстрактен клас.

АБСТРАКТЕН КЛАС

Полиморфизъм

- Ако класовете, в които трябва се дефинират виртуални методи, нямат общ родител, такъв може да бъде създаден изкуствено чрез т.нар. абстрактен клас.

Чисто виртуална функция

- Възможно е виртуална член-функция да не е дефинирана, а само декларирана в клас. Такава виртуална член- функция се нарича чисто виртуална.

```
virtual [<тип_на_резултата>] <име_на_метод> (<параметри>) [const] = 0;
```

Абстрактен клас

- Клас, в който има поне една чисто виртуална функция се нарича абстрактен.

```
virtual [<тип_на_резултата>] <име_на_метод> (<параметри>) [const] = 0;
```

- Не могат да се създават обекти от тези класове, но могат да се дефинират указатели от такива класове.

- Чисто виртуалните функции задължително трябва да бъдат предефинирани в производните класове или да бъдат обявени като чисто виртуални в тях.

Абстрактен клас

- Предназначение

Абстрактните класове са предназначени да са базови на други класове.
Чрез тях се обединяват в обща структура различни йерархии.

АБСТРАКТЕН КЛАС

- Хетерогенна структура - съставна структура от данни, компонентите на която са от различни типове се нарича хетерогенна.

- Полиморфизмът, с помощта на абстрактните класове, се позволява създаването на хетерогенни (полиморфни) структури от данни.

Пример:
Стек, елементите на който са от различен тип:
- точки в равнината и в пространството;
- домашни животни и горски животни;

КРАЙ