КОПИ КОНСТРУКТОР

доц. д-р Нора Ангелова

(Копи) конструктор за присвояване

- Създаване обект като копие на друг вече съществуващ обект

Пример:

```
Point2D point(1,3);
Point2D secondPoint = point;
```

или

```
Point2D secondPoint(point);
```

Тази инициализация се създава от специален конструктор, наречен **конструктор за присвояване (копи конструктор)**.

(КОПИ) КОНСТРУКТОР

- Конструкторът за копиране е конструктор, поддържащ формален параметър от тип: <име_на_клас> const & (десният операнд от присвояването)

Пример:

```
class Point2D {
  public:
    Point2D(double xValue = 0, double yValue = 0);
    Point2D(Point2D const& pointObj);
    void print() const;
  private:
    double x;
    double y;
};
// ...
Point2D::Point2D(Point2D const& pointObj) {
    x = pointObj.x;
    y = pointObj.y;
}
```

(КОПИ) КОНСТРУКТОР

- Конструкторът за копиране е конструктор, поддържащ формален параметър от тип: <име_на_клас> const & (десният операнд от присвояването или обекта в скобите)
  - Параметърът НЕ се предава по стойност и няма да се копира
  - Запазената дума const забранява промяната на обекта

Пример:

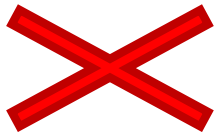
```
class Point2D {
  public:
    Point2D(double xValue = 0, double yValue = 0);
    Point2D(Point2D const& pointObj);
    void print() const;
  private:
    double x;
    double y;
};
// ...
Point2D::Point2D(Point2D const& pointObj) {
    x = pointObj.x;
    y = pointObj.y;
}
```

(Копи) конструктор

- Конструкторът за копиране е конструктор, поддържащ формален параметър от тип: <име_на_клас> const&

Пример:

```
// Ами ако пропуснем псевдонима?!?
Point2D::Point2D(Point2D const pointObj) {
  x = pointObj.x;
  y = pointObj.y;
}
…
Point2D point(1,3);
Point2D secondPoint = point;
```



Предаването на параметъра става по стойност - трябва point да се копира в стековата рамка?!?

(Копи) конструктор за копиране

Конструктор за копиране се използва при:

- създаване на обект на даден клас като копие на друг обект на същия клас;
- предаване на обект (по стойност) като аргумент на функция – създава се обект в стековата рамка, който е копие на фактическия параметър;
- връщане на обект (по стойност) като резултат от изпълнение на функция;

Изключение правят параметрите, които се подават по референция.

(КОПИ) КОНСТРУКТОР

Специфики

- **В клас явно НЕ е дефиниран копи конструктор**
  - Ако в един клас явно не е дефиниран конструктор за копиране, компилаторът автоматично създава такъв, в момента когато новосъздаден обект се инициализира с обект, намиращ се от дясната страна на знака за присвояване или в кръглите скоби – копи конструктор.
    Присвояването на член-данните става директно.
  - Последното важи и в случай, че вече имаме конструктор с параметри.

(Копи) конструктор за копиране

Специфики

- **В клас явно е дефиниран копи конструктор**

```
class Point2D {
  public:
    Point2D(double xValue = 0, double yValue = 0);
    Point2D(Point2D const& pointObj);
    void print() const;
  private:
    double x;
    double y;
};
// ...
Point2D::Point2D(Point2D const& pointObj) {
    x = pointObj.x;
    y = pointObj.y;
}
...

Point2D defaultPoint;                // defaultPoint се инициализира с (0, 0)
Point2D secondPoint = defaultPoint; // secondPoint се инициализира с (0, 0)
```

(Копи) конструктор за копиране

Специфики

- **В клас явно е дефиниран копи конструктор**

```
class Point2D {
  public:
    Point2D(double xValue = 0, double yValue = 0);
    Point2D(Point2D const& pointObj);
    void print() const;
  private:
    double x;
    double y;
};
// ...
Point2D::Point2D(Point2D const& pointObj) {
    x = pointObj.x + 1;
    y = pointObj.y + 1;
}
...

Point2D defaultPoint;                // defaultPoint се инициализира с (0, 0)
Point2D secondPoint = defaultPoint; // secondPoint се инициализира с (1, 1)
```

(Копи) конструктор за копиране

- Инициализация на член-данните
  - Член-данни на клас, които са обекти

```
class Rect {
  public:
    Rect(double x1 = 0, double y1 = 0, double x2 = 0, double y2 = 0);
    Rect(Rect const& rectObj);
  private:
    Point2D topLeft;
    Point2D bottomRight;
};
```

```
// Има член-данни от тип Point2D;
// Извиква се конструктор по
// подразбиране за двете точки;
```

```
Rect::Rect(Rect const& rectObj) {
 // …
}
...

Rect defaultRect; // defaultRect се инициализира с (0,0), (0,0)
Rect secondRect = defaultRect;
```

(КОПИ) КОНСТРУКТОР ЗА КОПИРАНЕ

- Инициализация на член-данните
  - Член-данни на клас, които са обекти

```
class Rect {
  public:
    Rect(double x1 = 0, double y1 = 0, double x2 = 0, double y2 = 0);
    Rect(Rect const& r);
  private:
    Point2D topLeft;
    Point2D bottomRight;
};
Rect::Rect(double x1, double y1, double x2, double y2) :
topLeft(x1, y1), bottomRight(x2, y2)
{}

Rect::Rect(Rect const& rectObj) : topLeft(rectObj.topLeft),
bottomRight(rectObj.bottomRight)
{}

Rect defaultRect; // defaultRect се инициализира с (0,0), (0,0)
Rect secondRect = defaultRect;
```

(Копи) конструктор за присвояване

- Кога да дефинираме копи конструктор?
  - Ако не е дефиниран конструктор за копиране - извършва се автоматично чрез директно присвояване.

Когато обект не може да се създаде чрез директно присвояване на член-данните

Пример – динамично заделяне на памет

(Копи) конструктор за присвояване

- Как дефинираме конструктора при динамично заделяне на памет?

ВРЕМЕ ЗА ВАШИТЕ ВЪПРОСИ