
Subject: Overriding Display methods too complicated due to high amount of arguments

Posted by [Klugier](#) on Sat, 17 Oct 2020 22:01:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello,

I would like to tell you what I don't like about Display class (CtrlLib). It's two virtual methods are complicated to override due to the need to explicitly repeat the parameter list. It is exactly 6 parameters. Let's take a look at what we have now:

```
class Display {
public:
...
virtual void PaintBackground(Draw& w, const Rect& r, const Value& q,
                             Color ink, Color paper, dword style) const;
virtual void Paint(Draw& w, const Rect& r, const Value& q,
                  Color ink, Color paper, dword style) const;
```

The solution for that is to introduce simply DisplayPaintContext that will contain all parameters variables:

```
class Display {
public:
...
virtual void PaintBackground(const DisplayPaintContext& ctx) const;
virtual void Paint(const DisplayPaintContext& ctx) const;
```

```
// Or
virtual void PaintBackground(Draw& w, const DisplayPaintContext& ctx) const;
virtual void Paint(Draw& w, const DisplayPaintContext& ctx) const;
```

Of course, it can be implemented in alternative way for example by providing first Draw parameter and rest as a context. Of course we can not simply change the declaration of this method. To fix the problem we should introduce the new one and deprecate the old.

Sample replacment in Display and Array examplex:

```
struct FontFaceDisplay : Display {
    virtual void Paint(Draw& w, const DisplayPaintContext& ctx) const override {
        auto r = ctx.GetRect();

        Font fnt = Font(ctx.GetValue(), r.Height() - 2);
        String txt = Font::GetFaceName(ctx.GetValue());
        w.DrawRect(r, ctx.Paper());
        w.DrawText(r.left + 2, r.top + (r.Height() - GetTextSize(txt, fnt).cy) / 2, txt, fnt,
```

```
ctx.GetInk());  
}  
};
```

```
struct MyDisplay : public Display {  
    virtual void Paint(Draw& w, const DisplayPaintContext& ctx) const override  
        w.DrawRect(ctx.GetRect(), paper);  
        w.DrawEllipse(ctx.GetRect(), ctx.GetValue());  
}  
};
```

Reference:

- Coding Revolution - Long Parameter List
- Refactoring Guru - Long Parameter List
- SonarSource - Static analyzer - Functions should not have too many parameters (4 is maximum limit)

Klugier
