

CS 485

Advanced Object Oriented Design

Design Patterns
Facade Pattern
Template Method Patten

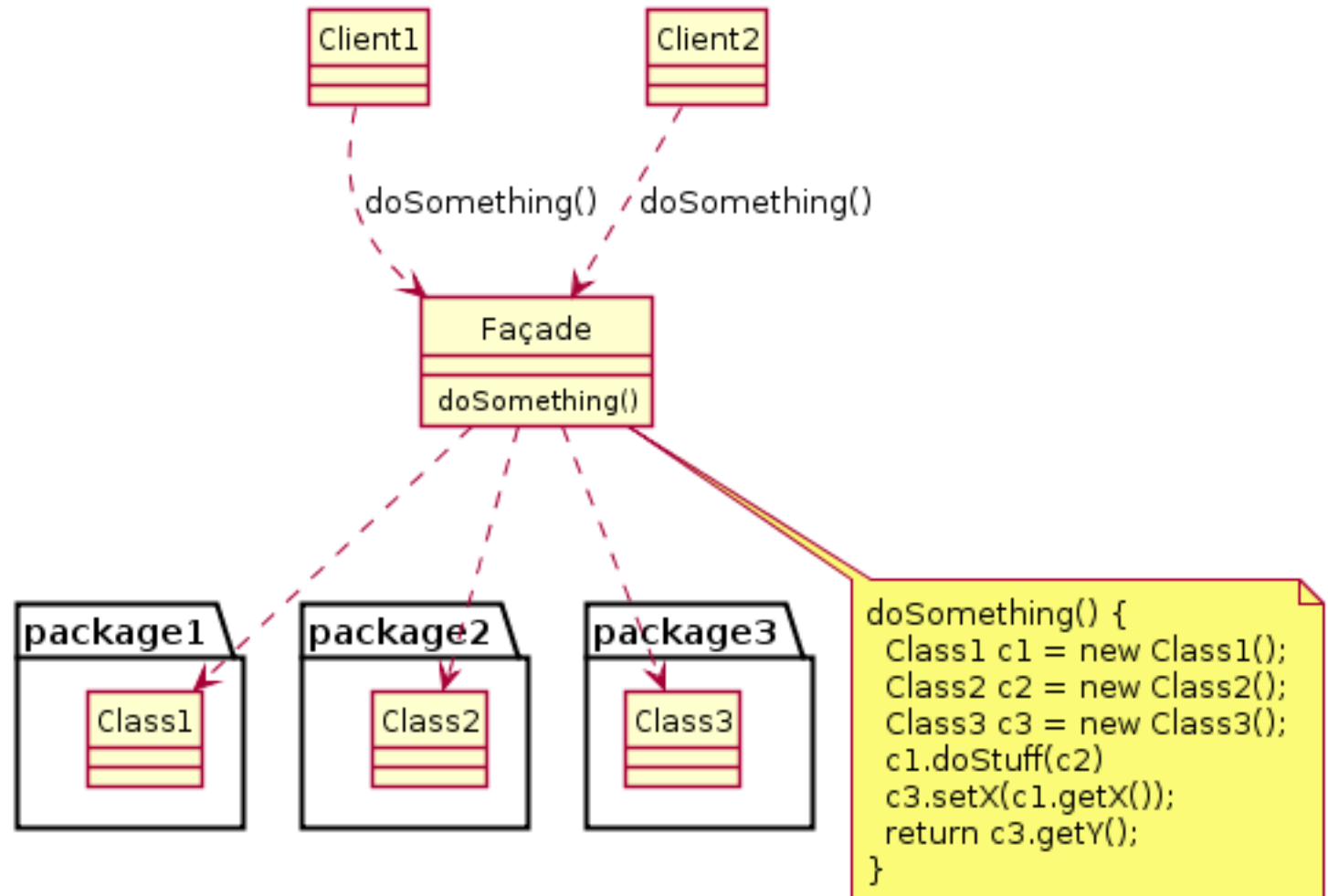
Spring 2017

<http://www.netobjectives.com/PatternRepository/index.php?title=PatternsByAlphabet>

<http://www.netobjectives.com/files/books/dpe/design-patterns-matrix.pdf>

Design Patterns

Facade Pattern



```
class SDLApp
{
public:
    bool createWindow (std::string windowName, ...);
    void setWindowBackgroundColor (Color cRgba);

    void cleanup ();

    void setColor (Color cRgba);
    void drawLine (int x1, int y1, int x2, int y2);
    void drawCircle (int x, int y, int radius);

    int loadFont (std::string fontname, int size);
    bool displayText (int x, int y, std::string msg, Color cRgba, int font);

    void gameLoop (unsigned int framesPerSecond = 60);

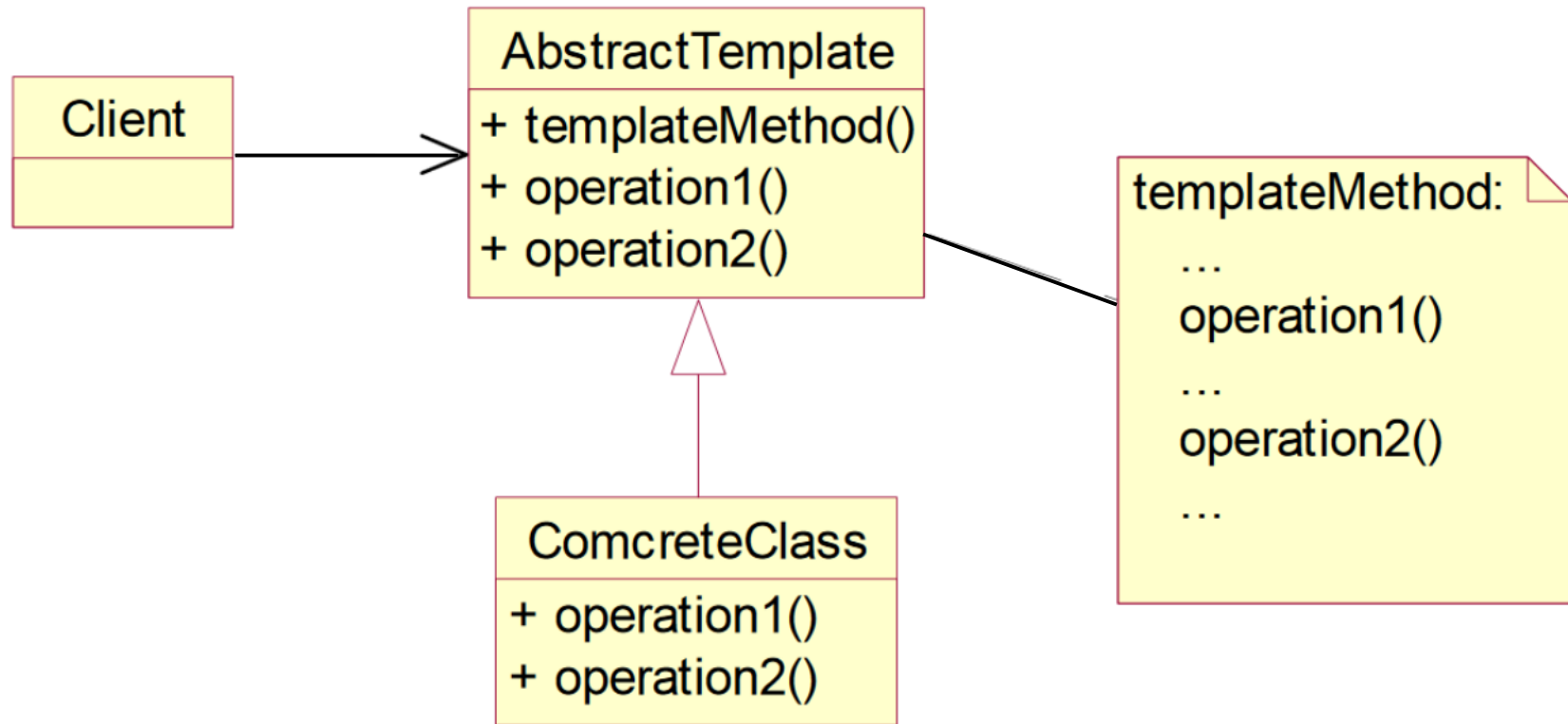
private:

    SDL_Window *mpWindow = nullptr;
    SDL_Renderer *mpRenderer = nullptr;

    // Colors
    Color mcCurrentColor;
    Color mcBackgroundColor;

    std::vector<TTF_Font *> mpFonts;
```

Template Method Pattern



```
void SDLApp::gameLoop (unsigned int framesPerSecond)
{
    while (bIsRunning)
    {
        update ();

        // Clear the window
        SDL_RenderClear (mpRenderer);

        render ();

        // Render the changes above
        SDL_RenderPresent (mpRenderer);

        uSdlEvent.type = SDL_FIRSTEVENT;

        if (SDL_PollEvent (&uSdlEvent) != 0)
        {
            if (uSdlEvent.type == SDL_QUIT)
            {
                bIsRunning = false;
            }
            handleEvent (uSdlEvent);
        }
    }
}
```

Usage

```
// BoomshineApp.cpp
```

```
void update()
{
    // move all MovingCircles
    // expand all ExpandingCircles
    // check for collisions
    //   handle collisions
    // check for win or lose
}

void render()
{
    // draw all MovingCircles
    // draw all ExpandingCircles
}

void handleEvent(SDL_Event uSDL_Event)
{
    // on mouse click
    //   drop ExpandingCircle

    // on space bar, restart level
}
```

```
// UntimedMinesweeperApp.cpp
```

```
void update()
{
    // do nothing
}

void render()
{
    // draw all tiles
}

void handleEvent(SDL_Event uSDL_Event)
{
    // on right mouse click
    //   set flag
    // on left mouse click
    //   check square
    //   lose, or replace tile
    //           and expand open space
    // check win
}
```