

CS 485

Advanced Object Oriented Design

Design Patterns
Facade Pattern
Template Method Patten

Spring 2019

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

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

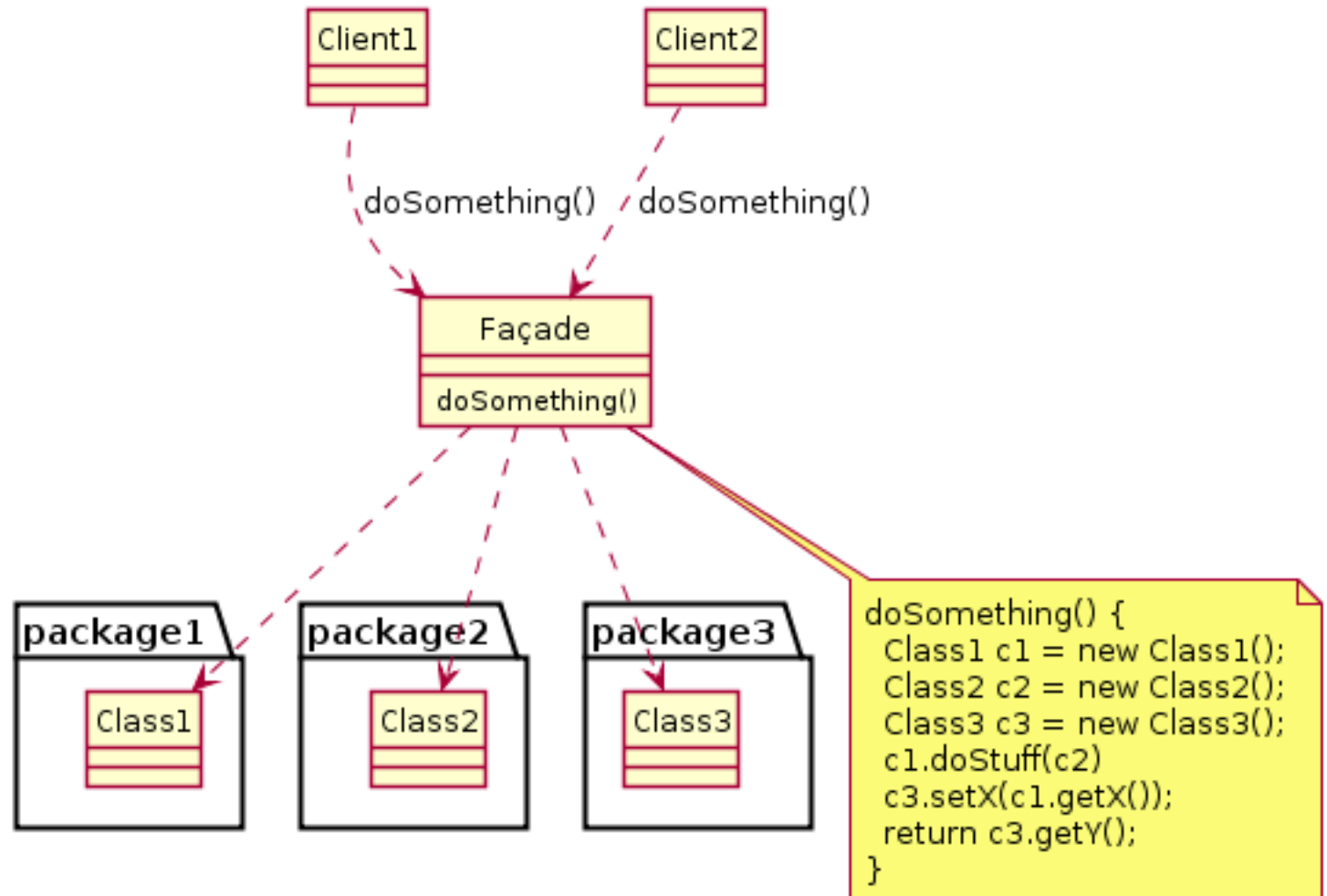
https://sourcemaking.com/design_patterns

Design Patterns

Some Patterns

- Creational
 - Factories
- Behavioral
 - Command
 - Strategy
 - **Template Method**
- Structural
 - **Facade**

Facade Pattern



```
class SDLApp
{
public:
    SDLApp ();

    virtual ~SDLApp ();

    bool createWindow (std::string windowName, .... );
    void setWindowBackgroundColor (Color cRgba);

    void cleanup ();

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

    void displayText (int x, int y, std::string msg, Color cRgba);

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

    bool loadSprite (std::string name, int &w, int &h);
    void displaySprite (const SDLSprite &rcSprite);
```

```
SDL_Window *pWindow = nullptr;
SDL_Renderer *pRenderer = nullptr;

SDL_Event sdlEvent;

SDL_Init(SDL_INIT_EVERYTHING);

pWindow = SDL_CreateWindow("Boomshine!",..., SDL_WINDOW_RESIZABLE |
    SDL_RENDERER_PRESENTVSYNC);

pRenderer = SDL_CreateRenderer(pWindow, -1, SDL_RENDERER_ACCELERATED);

while (blsRunning)
{
    startTime = SDL_GetTicks();

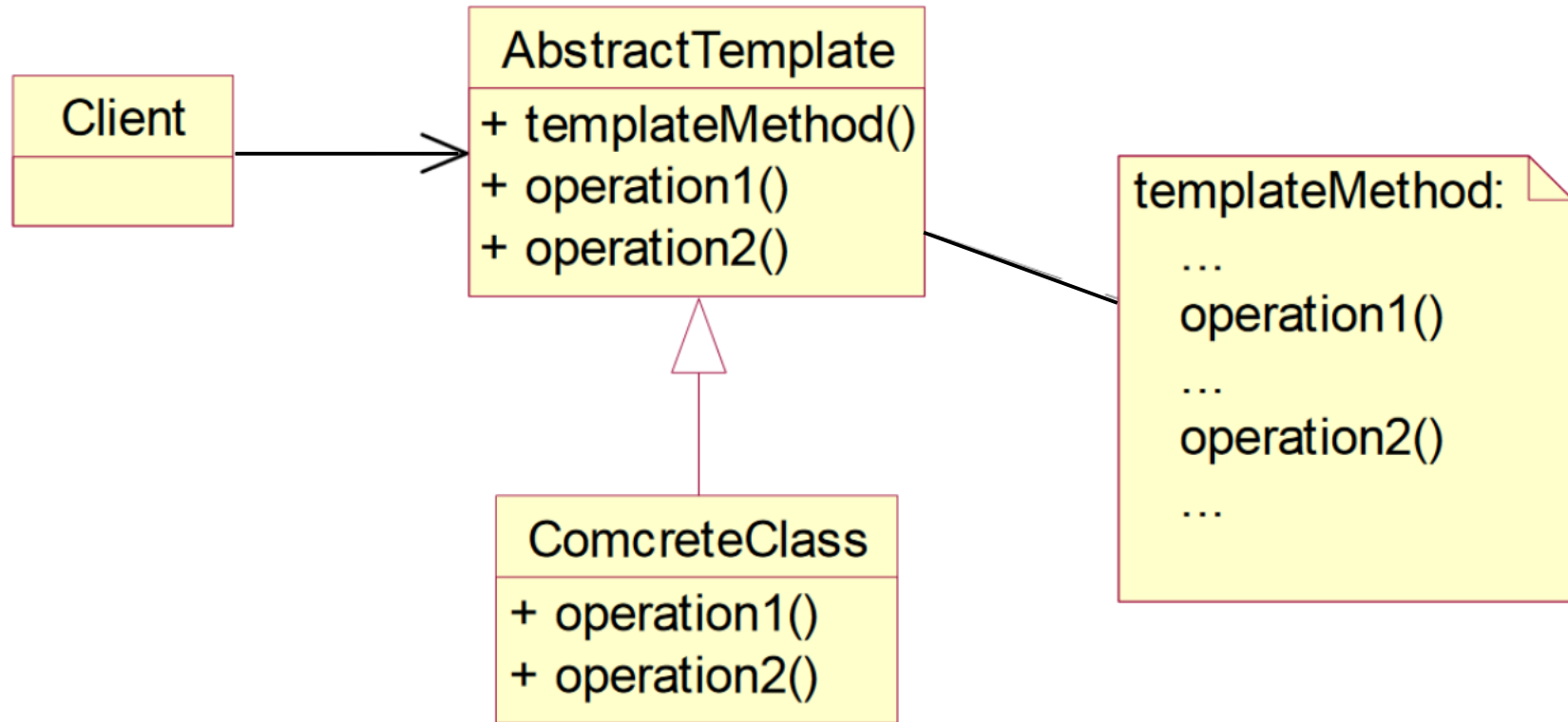
    SDL_SetRenderDrawColor(pRenderer, 255, 255, 255, 255);

    SDL_RenderClear(pRenderer);

    SDL_RenderPresent(pRenderer);

    // decide how fast to update the screen.
    if (-1 != FRAMES_PER_SECOND)
    {
        // force a particular number of frames per second
        endTime = SDL_GetTicks();
        if (1000 / FRAMES_PER_SECOND > endTime - startTime)
        {
            delay = 1000 / FRAMES_PER_SECOND - (endTime - startTime);
            SDL_Delay(delay);
        }
    }
}
```

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
}
```