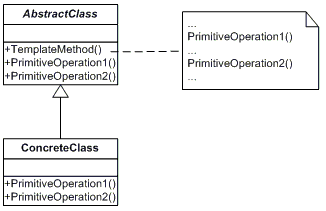
Design Patterns

Template Method Pattern

Collin Kemner

December 1st, 2016

Introduction



The purpose of this assignment is to create and demonstrate an application utilizing the Template Method Pattern. The application being used for this assignment is a textbox that showcases different “behaviors” exhibited at three different environments on a Saturday night: at home, at school, or at a concert.

UML Diagram

According to dofactory.com, the purpose of the Template Method Pattern is to “Define the skeleton of an algorithm in an operation, deferring some steps to subclasses.” In the diagram, the AbstractClass defines a few operations that will be done later on in the application as well as a “TemplateMethod()” operation. When this operation is called, it will collect all of the operations and fire them all at the same time. In the ConcreteClass, the same operations will be called, but will be redefined for each specific environment (in this application’s case).

First class is the NightlyActivies class, which is the AbstractClass in the UML diagram. Four abstract string methods are defined, which will be the four “behaviors”. Then, the InitiateTheNight() method is created, which is the TemplateMethod() on the UML diagram. The method creates a list of the behavior strings and then uses a for-loop to iterate through them to put them into a compact string, which is then returned.

Narrative of Code: NightlyActivities Class (AbstractClass)

public abstract class NightlyActivities //the AbstractClass

{

public abstract string DrinkingSubstances();

public abstract string Language();

public abstract string Clothes();

public abstract string Activities();

public string InitiateTheNight()

{

List<string> listOfActivities = new List<string>

{

DrinkingSubstances(), Language(), Clothes(), Activities()

};

string Altogether = "";

for (int i = 0; i < 4; i++)

{

Altogether += listOfActivities[i] + System.Environment.NewLine + System.Environment.NewLine;

}

return Altogether;

}

}

AtHomeConcrete, AtSchoolConcrete, AtConcertConcrete Classes (ConcreteClass)

public class AtHomeConcrete : NightlyActivities //the ConcreteClass

{

public override string DrinkingSubstances()

Next are the ConcreteClasses. They all work the same, so the AtHomeConcrete class will be explained. All of the abstract strings from the NightlyActivities class are overridden and then defined with specific behaviors. For example, normally at home, water and milk are common beverages as well as more conservative clothing and being bored. On top of that, at least at some households, the language will be restricted, unlike at a concert or a college party.

{

return "Drinks: Water, milk";

}

public override string Language()

{

return "Language: Highly restricted";

}

public override string Clothes()

{

return "Clothes: Conservative";

}

public override string Activities()

{

return "Activities: Shopping, cleaning, sleeping, eating, nothing fun";

}

}

public class AtCollegeConcrete : NightlyActivities //the ConcreteClass

{

public override string DrinkingSubstances()

{

return "Drinks: Water, Soda, drinks that contain C2H6O";

}

public override string Language()

{

return "Language: Unrestricted";

}

public override string Clothes()

{

return "Clothes: Provocative";

}

public override string Activities()

{

return "Activities: Parties, hood shenanigans, robbing banks, forming my own cult, ect.";

}

}

public class AtConcertConcrete : NightlyActivities //the ConcreteClass

{

public override string DrinkingSubstances()

{

return "Drinks: Lots of C2H6O";

}

public override string Language()

{

return "Language: What is PC?";

}

public override string Clothes()

{

return "Clothes: Clothes with holes";

}

public override string Activities()

{

return "Activities: Headbanging, stage-diving, crowd-surfing, getting hearing aids in about 30 years";

}

The form contains a button click which will display the information for the specific environment that someone may be in. The concrete class objects are defined at the top. The object to use is then determined by if-else statements which check what radio button is checked. The radio buttons correspond to the specific environments. When a radio button is selected, the text in the textbox will be set as the InitiateTheNight() string method as well as an image off to the side on a panel. There are specific images for each environment.

}

The Form

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void m\_btnNight\_Click(object sender, EventArgs e)

{

NightlyActivities ac = new AtCollegeConcrete();

NightlyActivities ah = new AtHomeConcrete();

NightlyActivities acon = new AtConcertConcrete();

if (m\_rbAtSchool.Checked)

{

m\_tbNightlyActivities.Text = ac.InitiateTheNight();

m\_pnlImage.BackgroundImage = System.Drawing.Bitmap.FromFile(@"C:\Users\c-kemner\Documents\201603\_0144\_ddaac\_sm.jpg");

}

else if (m\_rbAtHome.Checked)

{

m\_tbNightlyActivities.Text = ah.InitiateTheNight();

m\_pnlImage.BackgroundImage = System.Drawing.Bitmap.FromFile(@"C:\Users\c-kemner\Documents\angry-child-outbursts.jpg");

}

else if (m\_rbConcert.Checked)

{

m\_tbNightlyActivities.Text = acon.InitiateTheNight();

m\_pnlImage.BackgroundImage = System.Drawing.Bitmap.FromFile

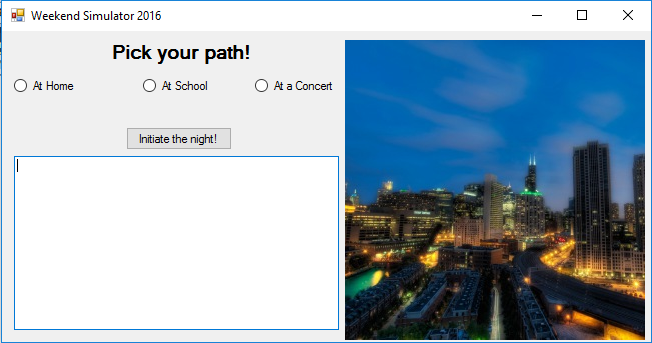
(@"C:\Users\c-kemner\Documents\concert-smoke-crowd-people-concert-music-youth-club-photos-crowd-cheering-the-mood-the-smoke-tools-136417-2560x14401-990x420-300x300.jpg");

}

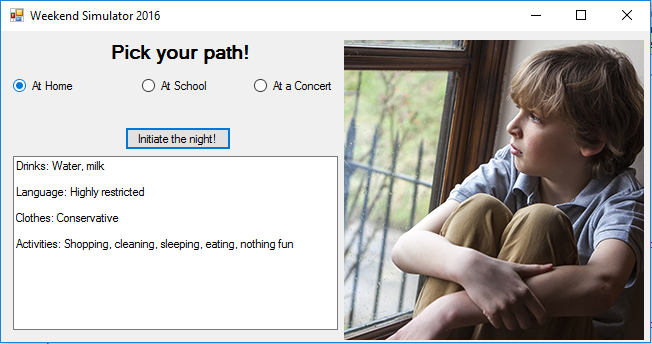
}

}

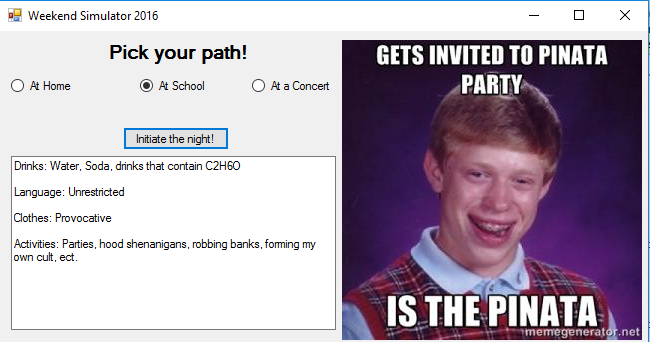
Screenshots of App



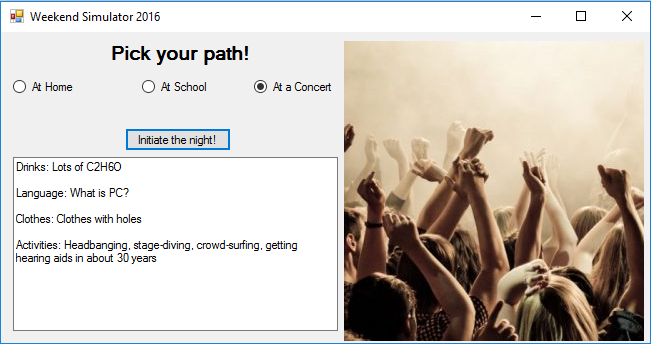
Application when first opened up.



Application when “At Home” is selected.



Application when “At School” is selected.



Application when “At a Concert” is selected.

Observations and Reflections

I’m going to be pretty blunt: this was an insanely simple and easy pattern, and my new favorite! Besides joking, I did have a good time doing this pattern as it allowed me to experiment a little bit with images and how I can change them if a certain action is performed. That was my favorite part of this assignment. Overall, I had fun and would make another one if I had to. 10/10 would recommend.