Lesson 13: PatternPainter

45 minutes

Overview

What other subclasses could I create?

Students write a new subclass and practice decomposition to develop algorithms. Students translate algorithms to write methods in their new subclass.

Standards

Full Course Alignment

CSA Conceptual Framework

▶ **MOD-3** - When mulitple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierachy. Modifications made at the highest level of the hierarchy apply to the subclasses.

Agenda

Warm Up (10 minutes)
- **CS Pyramid**

Activity (30 minutes)
- **Planning the PatternPainter Class**
- **Writing the PatternPainter Class**

Wrap Up (5 minutes)
- **Your PatternPainter's Story**
- **Assessment: Check for Understanding**
- **AP Classroom Topic Questions**

Objectives

Students will be able to:

- Translate algorithms to `void` methods to add new behaviors to an existing class
- Use decomposition to develop algorithms
- Write a subclass that extends a superclass

Preparation

- Create code review groups if you are not reusing the same groups
- Print copies of the PatternPainter handout (one for each student)
- Check the **Teacher's Lounge** for verified teachers on the CSA Forum to find additional strategies or resources shared by fellow teachers

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the students

- **PatternPainter** - Handout
- **U1L13 Extra Practice** - Handout

Vocabulary

- **inheritance hierarchy** - where a class serves as a superclass for more than one subclass

Teaching Guide

Warm Up (10 minutes)

CS Pyramid

Remarks

> We have learned a lot of new terms so far! Let's review some of these terms through a game of CS Pyramid.

Group: Place students in pairs.

Do This: Review the instructions for playing CS Pyramid.

Do This: Play the music clip to cue the CS Pyramid activity, and direct students to play CS Pyramid. Click through the animated slide to display each pyramid, and direct students to switch roles with each new pyramid.

Activity (30 minutes)

Planning the PatternPainter Class (15 minutes)

Remarks

> Our `PainterPlus` class allows us to navigate and paint The Neighborhood with additional methods, such as turning right. We currently have a one-to-one relationship between the `PainterPlus` and `Painter` classes, but we can expand this relationship to create an inheritance hierarchy with multiple subclasses.

Do This: Review the lesson objectives.

Do This: Define *inheritance hierarchy*.

Distribute: Give each student a copy of the PatternPainter handout.

Teaching Tip

Have students share a hobby or interest and ask how these might be represented as a symbol. Give an example of a hobby or interest and possible symbols that you might make into a pattern.

Discuss: Click through the animated slide to display the prompts. Use the Hold That Thought strategy to discuss the prompts.

- *Why should we make a new type of* `Painter` *instead of adding methods for this problem to* `PainterPlus`*?*
- *Which should this new class extend –* `Painter` *or* `PainterPlus`*?*

Discussion Goal: Students realize this class focuses on creating patterns, while the `PainterPlus` class provides additional behaviors for basic navigation and painting in The Neighborhood. Students suggest extending `PainterPlus`, so the new class has the same behaviors as the `Painter` and `PainterPlus` classes.

Teaching Tip

If students struggle to identify the need for a new type of class or to determine which class to extend, ask additional guiding questions and refer them to the UML diagrams on the Unit 1 Guide. For example:

- *What is the purpose of the* `Painter` *class? How about the* `PainterPlus` *class?*
- *Why do we create subclasses?*
- *Should the new subclass have the same behaviors as the* `Painter` *class or the* `PainterPlus` *class?*

Remarks

> We can create a `PatternPainter` class that extends the `PainterPlus` class. This class would have the same attributes and behaviors as both the `Painter` and `PainterPlus` classes, and we can specialize this type of `Painter` to focus on creating patterns.

Do This: Direct students to create a UML diagram for the `PatternPainter` class and write pseudocode for the methods to write.

Remarks

> When painting patterns and murals, we need a way to set the amount of paint a `Painter` object has so we don't have to collect a lot of paint from multiple paint buckets.

Do This: Introduce and explain the `setPaint()` method.

Writing the PatternPainter Class (15 minutes)

Remarks

> Use your UML diagram, pseudocode, and notes to create the `PatternPainter` class. After writing the class, create a mural of your pattern.

Do This: Direct students to Level 1 on Code Studio to complete Levels 1, 2, and 3. Students complete a Check for Understanding on Level 1, then continue to Level 2 to write the `PatternPainter` class. Students create a mural with their `PatternPainter` class on Level 3.

Remarks

> This is a good time to commit our code and save our `PatternPainter` class to the Backpack. Anytime we make changes to our programs, it is helpful to commit, or save, our work as a new version in case we need to revert to a previous version.

Do This: Play the music clip to cue committing their code and saving the `PatternPainter` class to the Backpack.

Remarks

> When we write new code, getting feedback from our peers is helpful to make sure we have met the

> When we write new code, getting feedback from our peers is helpful to make sure we have met the requirements of the problem efficiently.

Do This: Click through the animated slide to have students participate in the Code Review Call and Response.

Do This: Direct students to complete a code review on their program on Level 4.

4 **Code Review: PatternPainter**

Wrap Up (5 minutes)

Your PatternPainter's Story

Remarks

> We have seen how computer science is used as a form of creative expression throughout the unit.

Discuss: *What is the story behind your* `PatternPainter`*? What does the pattern it creates represent?*

Discussion Goal: Students share their patterns and explain what they represent or mean to them.

Do This: Review the concepts covered in this lesson.

Display: Key Vocabulary

Assessment: Check for Understanding

Check For Understanding Question(s) and solutions can be found in each lesson on Code Studio. You can use these questions as an exit ticket.

5 **Check for Understanding**

AP Classroom Topic Questions

To assign questions from the AP Classroom Question Bank that align with this lesson, create a custom quiz in AP Classroom by searching the Question Bank for the Essential Knowledge statements listed at the top of this lesson plan. You can find instructions and video demonstrations to do this on **AP Central**.