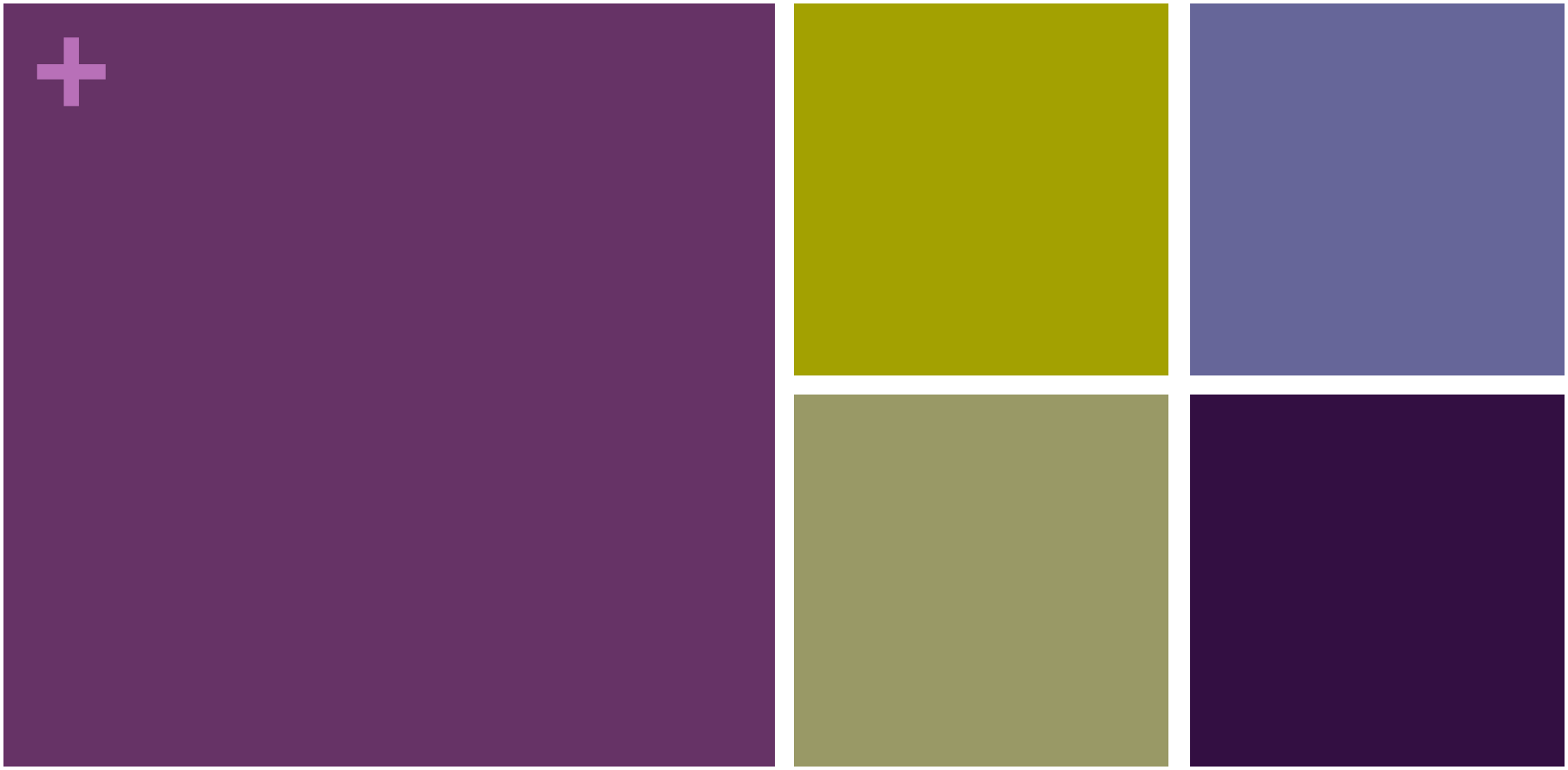


# Building New Classes

CS 1324

Deborah A. Trytten

# Organizing Data

- We've started abstracting data
  - Separating things in the world into abstract categories: int, double, char, String
  - Grouping homogeneous data together into arrays

- We also need to be able to group things that are not homogeneous together
  - Objects and Classes

# General Structure

- Each object contains different data
- Data describes the state of the objects
  - Properties are represented as attributes
- Each object follows the behavior of the class
  - Actions are represented as methods
- What is the State of the String? Behavior?
- How is a StringBuilder different from a String?
  - Properties or Actions?

# iClicker Question

- Which of the following is not a property of the ArrayList class?

a) A sequence of objects

b) The current size of the sequence

c) The largest value stored in the sequence

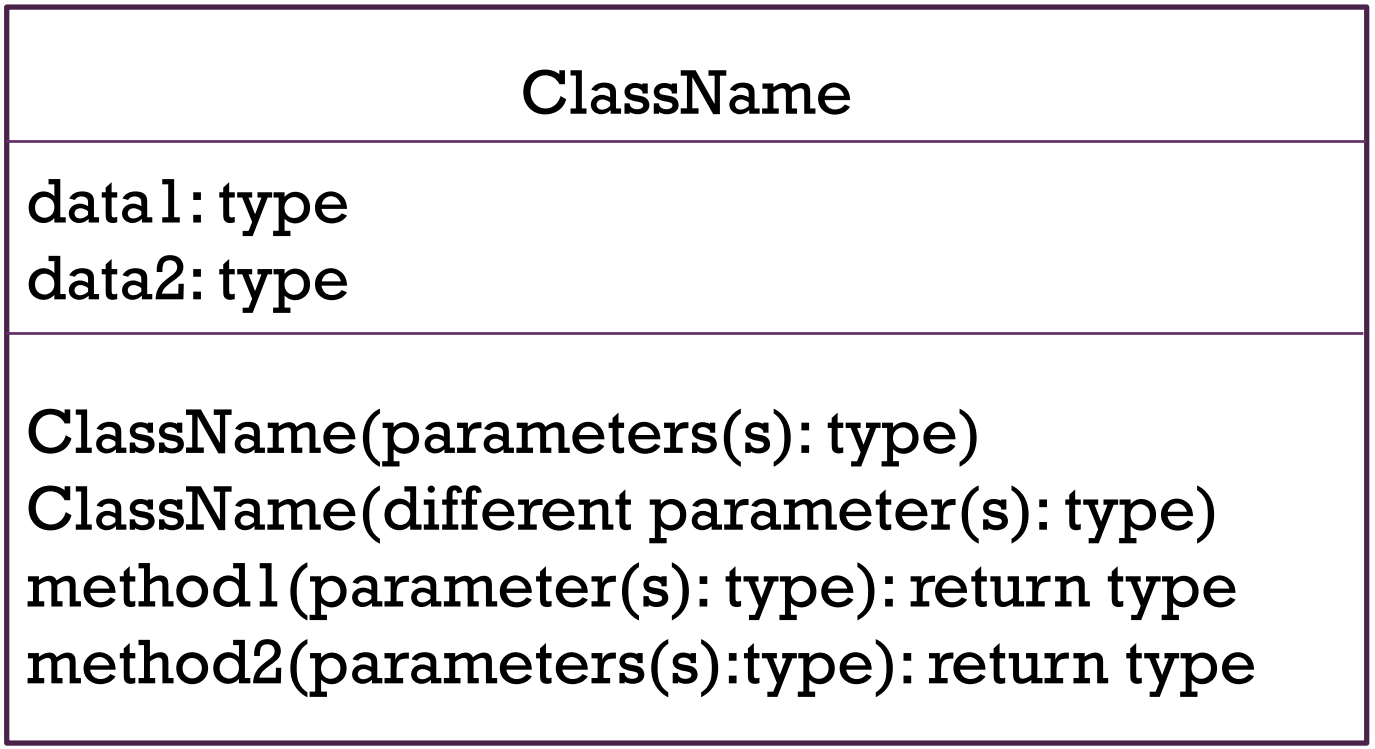
d) The current capacity of the sequence

# Example: Point

- Data: x and y (public unusual)
- Constructors
- Methods
  - getLocation()
  - equals()
  - toString()
  - setLocation()
  - translate()
- What is the state? Behavior?

# UML Class Diagram

- Unified Modeling Language (UML)
- Expresses the design of a class/objects graphically



# Defining Classes-I

- Data
  - Properties of the objects in the class
    - Each object gets its own copy
- Constructors
  - Instructions for building objects
  - Looks like a method, but no return type
  - May or may not have parameters
  - After constructor executes, all data is initialized

# Defining Classes-II

- Methods
  - Describe the behavior of the object
  - Accessors
    - Provide a copy of class/instance data to someone outside the class
  - Mutators
    - Change the state of the object

# Example: Contact Class

- Write a simple contact class for a cell phone directory
  - Keep it simple!
- Design first in UML
  - Name?
  - Data?
  - Methods?
  - Constructors?
- Implement

# Construct Objects in Another Class

- Create a second class for the main program
  - Often called the "driver" class
- Create an object reference in the main method
  - The reference is initially null
- Construct the object
- Constructors initialize the data
- What if we want to store a group of them?
  - Works exactly like classes from the API

# iClicker Question

- Suppose we want to create a class to store an individual task to do. Which of the fields below would not be included?

a) Name of task

b) Due date

c) Name of responsible parties

d) Number of tasks to be done

e) Priority (high, usual, low)

# Debugging Problem

- Remove the constructor
- Create a constructor with no parameters, but give it a return type
  - This makes it a method instead of a constructor
- Construct an object

# Default Constructor

- If you don't create a constructor, the compiler creates a default constructor for you
  - This can cause debugging problems
  - Compiler really doesn't know how to initialize objects
- If you forget that constructors don't have return values, compiler interprets constructor as just another method
  - But you may not notice since it can still call default constructor