

Summary 1 - Variables, Types, Classes and Objects

- What is this?

In Java, **every variable has a declared type**, either:

- **primitive type** (e.g. int, double), or
- **object type** (e.g. Point, JFrame, Eye)

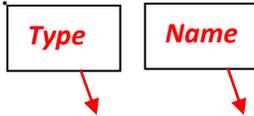
Variables are declared by using the

Type Name pattern, e.g.

```
Eye redEye;
```

```
int x;
```

```
public void mouseMoved(MouseEvent event)
```



In addition, every object has an *actual* type – see the Summary on **polymorphism**.

Variables can also be given an initial value when they are declared, e.g.

```
Eye redEye = new Eye(Color.red, Color.white);
```

```
int x = 20;
```

A **class definition** describes the type of the same name, e.g. the class definition for **Eye** describes the type **Eye**. Hence we use **class** and **type** somewhat synonymously. A class description specifies:

- **Fields** (aka **member variables** or **instance variables**) – the data associated with instances of the class
- **Constructors** – the code that runs when an instance of the class is constructed
- **Methods** (aka **functions** or **operations**) – the operations that an instance of the class can do

Instances of a class are called **objects** and are **constructed** by using the **new** operator, e.g.

```
new Eye(Color.red, Color.white);
```

Objects can be stored in variables: `p = new Point(4, 29);`

- Example (from JavaEyes) of a **class definition**

```
public class StartButton extends JButton
    implements ActionListener {
    private static final Color DEFAULT_COLOR = Color.GREEN;
    private AnimatedPanel squarePanel;
    private boolean hasBeenPressed;

    public StartButton(String title, AnimatedPanel panel) {
        super(title);
        this.squarePanel = panel;
        this.hasBeenPressed = false;
        this.setBackground(StartButton.DEFAULT_COLOR);
        this.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        (new Thread(this.squarePanel)).start();
        this.hasBeenPressed = true;
    }

    public boolean hasBeenPressed() {
        return this.hasBeenPressed;
    }
}
```

- For further study:

- *Big Java*, chapter 2 *Using Objects*
- *Big Java*, chapter 3 *Implementing Classes*
- This summary's *author*: David Mutchler
- See also the Summaries on:
 - *Constructing and Using Objects*
 - *Variables: Fields vs. Parameters vs. Local Variables*
 - *Variables: Primitive Type vs. Object Type*