

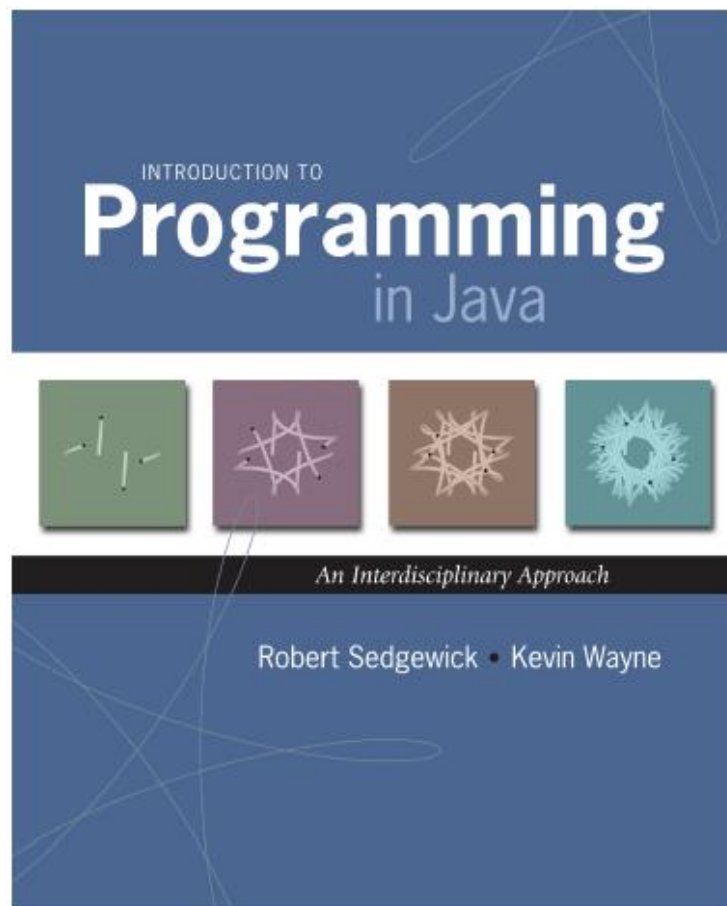
# برنامه نویسی شی گرا: طراحی انواع داده‌ای جدید

سید ناصر رضوی [www.snrazavi.ir](http://www.snrazavi.ir)

۱۳۹۶

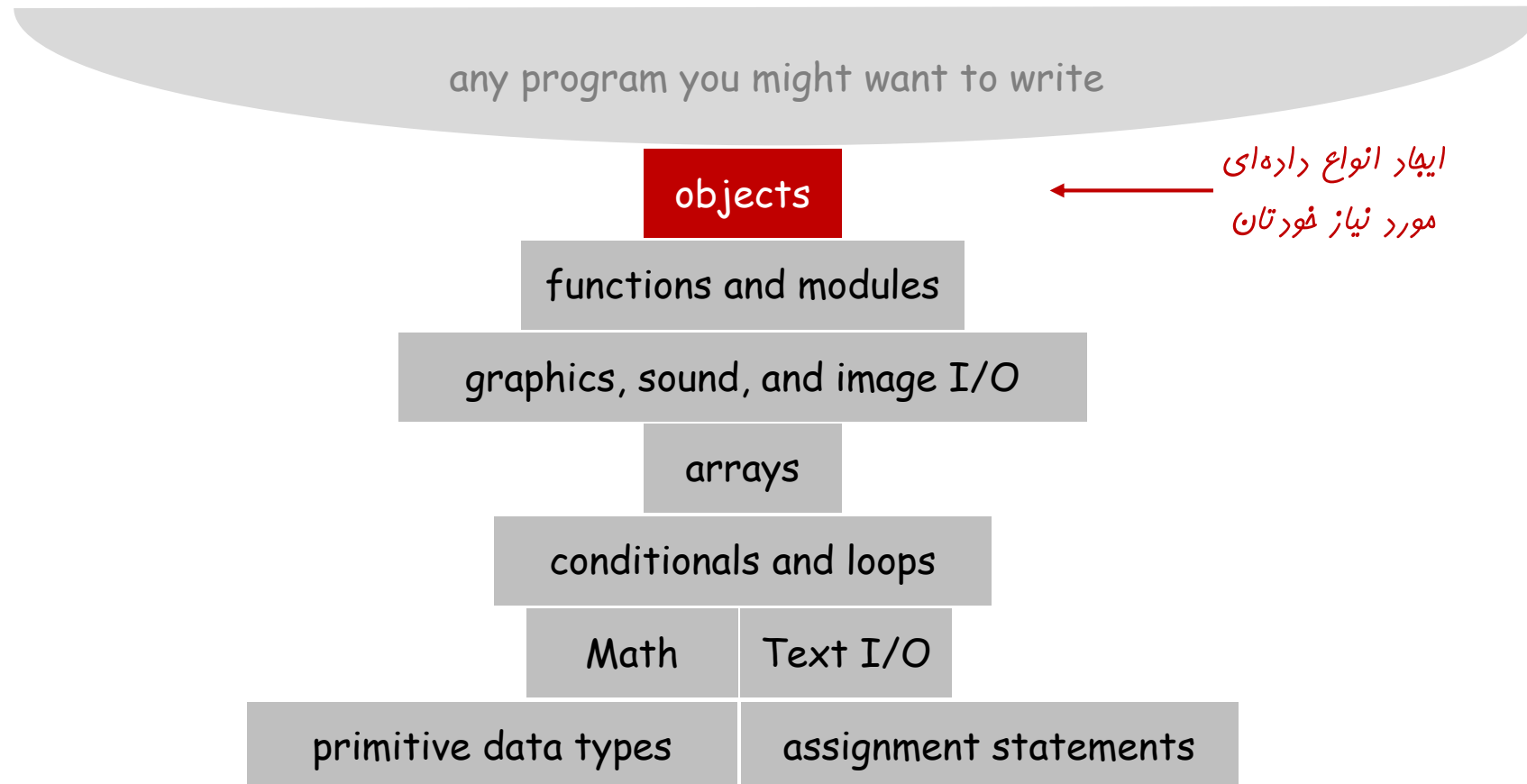
# ۳-۳ طراحی انواع داده‌ای

۲



# اجزای برنامه نویسی

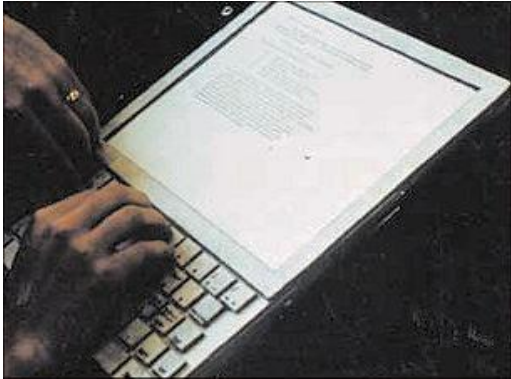
۳



# برنامه‌نویسی شی‌گرا

- برنامه‌نویسی رویه‌ای. [فعل‌گرا]
  - به کامپیوتر بگو این کار را انجام دهد.
  - به کامپیوتر بگو آن کار را انجام دهد.
  
- فلسفه ال‌کی. نرم‌افزار یک **شبیه‌سازی** از دنیای واقعی است.
  - ما (تقریباً) می‌دانیم دنیای واقعی چگونه کار می‌کند.
  - طراحی نرم‌افزار برای مدل‌سازی دنیای واقعی.
  
- برنامه‌نویسی شی‌گرا. [اسم‌گرا]
  - یک الگوی برنامه‌نویسی بر محور انواع داده‌ای.
  - مشخص کردن چیزهایی که بخشی از دامنه مسئله یا راه‌حل هستند.
  - اشیا در دنیا چیزهایی را می‌دانند: متغیرهای نمونه.
  - اشیا در دنیا کارهایی را انجام می‌دهند: متدها.

# الن کی



الن کی  
جائزہ تورینگ ۲۰۰۳

□ الن کی. [زیراکس، دهه ۷۰]

□ پدید آورنده زبان برنامه‌نویسی smalltalk.

□ ایده‌ها: لپ‌تاپ، واسط‌های گرافیکی کاربر امروزی، برنامه‌نویسی شی‌گرا.

« انقلاب کامپیوتری هنوز شروع نشده است. »

« بهترین روش برای پیش‌بینی آینده ابداع آن است. »

« اگر حداقل ۹۰ درصد از مواقع شکست نمی‌خوری، به اندازه کافی هدف‌دار نیستی. »

-- آلن کی

# پنهان سازی

- نوع داده‌ای. مجموعه‌ای از مقادیر و عملیات قابل انجام بر روی آن‌ها.
- مثال. رشته، عدد مختلط، بردار، ماتریس، سیم گیتار، ...
- نوع داده‌ای پنهان سازی شده. پنهان سازی بازنمایی درونی نوع داده‌ای.
- جداسازی پیاده سازی از مشخصات طراحی.
- یک کلاس فراهم کننده بازنمایی داده‌ها و کدهایی برای عمل کردن بر روی آن داده‌ها است.
- یک مشتری از نوع داده‌ای به صورت یک «جعبه سیاه» استفاده می کند.
- یک API قرارداد میان مشتری و کلاس را مشخص می کند.
- سخن آخر. برای استفاده از یک نوع داده‌ای لازم نیست بدانید چگونه پیاده سازی شده است.

# معنای شهودی

۷



## پیاژه سازی

- لامپ پرتوی کاتدی
- تفنگ الکترونی
- وزن ۲۴۱ پوند

پیاژه سازی باید براند چه واسطی را  
پیاژه سازی کند



## واسط برنامه نویسی

- تنظیم صدا
- تغییر کانال
- تنظیم تصویر

مشتری و پیاژه سازی باید از قبل بر روی واسط توافق کنند.



## مشتری

مشتری باید براند چگونه  
از واسط استفاده کند

# معنای شهودی



## پیاده‌سازی

- صفحه نمایش پلاسما
- قابل نصب روی دیوار
- عمق ۴ اینچ

پیاده‌سازی باید براند چه واسطی را پیاده‌سازی کند



## واسط برنامه‌نویسی

- تنظیم صدا
- تغییر کانال
- تنظیم تصویر

باید بتوان یک پیاده‌سازی بهتر را بدون نیاز به تغییر مشتری جایگزین کرد.



## مشتری

مشتری باید براند چگونه از واسط استفاده کند



# نوع داده‌ای شمارنده

۹

□ شمارنده. یک نوع داده‌ای برای شمارش آرای الکترونیکی.

```
public class Counter {  
    public int count;  
    public final String name;  
  
    public Counter(String id) { name = id; }  
    public void increment() { count++; }  
    public int value() { return count; }  
}
```

□ یک برنامه مشتری معتبر.

```
Counter c = new Counter("Volusia County");  
c.count = -16022;
```

□ س. مشکل چیست؟ تعداد آرای منفی!

# نوع داده‌ای شمارنده

□ شمارنده. یک نوع داده‌ای پنهان‌سازی شده برای شمارش آرای الکترونیکی.

```
public class Counter {  
    private int count;  
    private final String name;  
  
    public Counter(String id) { name = id; }  
    public void increment() { count++; }  
    public int value() { return count; }  
}
```

□ برنامه زیر کامپایل نمی‌شود.

```
Counter c = new Counter("Volusia County");  
c.count = -16022;
```

□ مزیت. تضمین این که مقدار هر متغیر همواره در یک حالت سازگار باقی می‌ماند.

# تغییر بازنمایی داخلی

۱۱

□ پنهان سازی.

□ بازنمایی داده را با استفاده از اصلاح گر دستیابی **private** پنهان کن.

□ واسط را با استفاده از اصلاح گر دستیابی **public** در معرض دید مشتری ها قرار بده.

```
public class Complex {  
    private final double re, im;  
  
    public Complex(double re, double im) { ... }  
    public double abs() { ... }  
    public Complex plus(Complex b) { ... }  
    public Complex times(Complex b) { ... }  
    public String toString() { ... }  
}
```

□ مزیت. تغییر بازنمایی داخلی بدون نیاز به تغییر برنامه های مشتری.

مثلاً به مفصّات قطبی

# بخواه ولی دست زن!

۱۲



ادل گلدبرگ  
رئیس پیشین ACM  
از توسعه‌دهندگان Smalltalk

□ انواع داده‌ای پنهان‌سازی شده.

- بدون دست زدن به داده‌ها هر کاری می‌خواهی انجام بده.
- در عوض، از شی مربوطه بخواه داده‌هایش را دست‌کاری کند.

« بخواه ولی دست زن. »

□ درس. محدود کردن حوزه، نگهداری و درک برنامه‌ها را ساده‌تر می‌کند.

# تغییر ناپذیری

□ داده‌های تغییر ناپذیر. مقدار شی پس از ایجاد شدن دیگر نمی‌تواند تغییر کند.

## *mutable*

Picture

Histogram

Turtle

StockAccount

Counter

Arrays

## *immutable*

Charge

Color

Stopwatch

Complex

String

primitive types

# تغییر ناپذیری: مزایا و معایب

- داده‌های تغییر ناپذیر. مقدار شی پس از ایجاد شدن دیگر نمی‌تواند تغییر کند.
- مزایا.
  - اجتناب از خطاهای مربوط به نام مستعار.
  - ساده‌تر کردن اشکال‌زدایی از برنامه.
  - محدود کردن حوزه کد که می‌تواند مقادیر را تغییر دهد.
  - ارسال اشیا به متدها بدون نگرانی در مورد تغییر کردن آنها.
- ایراد. به ازای هر مقدار، باید یک شی جدید ایجاد شود.

# اصلاح‌گر دستیابی final

۱۵

□ اصلاح‌گر final. اعلان یک متغیر نمونه به عنوان final به این معنا است که شما تنها می‌توانید یک بار آن متغیر را مقداردهی کنید. (مثلاً در یک سازنده)

```
public class Counter {  
    private final String name;  
    private int count;  
    ...  
}
```

پس از ایجاد شی، مقدار این  
متغیر دیگر تغییر نخواهد کرد.

مقدار این متغیر می‌تواند با فراخوانی  
متدهای نمونه تغییر کند.

□ مزایا.

- کمک به تضمین تغییر ناپذیری.
- جلوگیری از تغییرات تصادفی.
- ساده‌تر کردن اشکال‌زدایی.

# نوع داده‌ای بردار

□ مجموعه مقادیر. دنباله‌ای از اعداد حقیقی. [مختصات کارتیزین]

```
public class Vector
```

<code>Vector(double[] a)</code>	ایجاد یک بردار با مختصات داده شده
<code>Vector plus(Vector b)</code>	مماسبه مجموع این بردار و بردار <b>b</b>
<code>Vector minus(Vector b)</code>	مماسبه تفریق این بردار و بردار <b>b</b>
<code>Vector times(double t)</code>	ضرب اسکالر این بردار در عدد <b>t</b>
<code>double dot(Vector b)</code>	ضرب نقطه‌ای این بردار و بردار <b>b</b>
<code>double magnitude()</code>	مماسبه اندازه این بردار
<code>Vector direction()</code>	برداری واحد هم جهت با این بردار
<code>double cartesian(int i)</code>	مختصات کارتیزین <b>i</b> ام از این بردار
<code>String toString()</code>	بازنمایی رشته‌ای

$$x = (0, 3, 4, 0), y = (0, -3, 1, -4)$$



# نوع داده‌ای بردار: کاربردها

□ اهمیت. یک مدل ریاضی بسیار پر کاربرد.

□ کاربردها.

□ آمار.

□ جبر خطی.

□ خوشه‌بندی و جستجوی شباهت.

□ نمایش نیرو، سرعت، شتاب، اندازه حرکت، گشتاور.

□ ...

# نوع داده‌ای بردار: پیاده‌سازی

۱۸

```
public class Vector {
```

```
    private int N;
```

```
    private double[] coords;
```

متغیرهای عفو

```
    public Vector(double[] a) {
```

```
        N = a.length;
```

```
        coords = new double[N];
```

```
        for (int i = 0; i < N; i++)
```

```
            coords[i] = a[i];
```

```
    }
```

سازنده

```
    public double dot(Vector b) {
```

```
        double sum = 0.0;
```

```
        for (int i = 0; i < N; i++)
```

```
            sum += (coords[i] * b.coords[i]);
```

```
        return sum;
```

```
    }
```

```
    public Vector plus(Vector b) {
```

```
        double[] c = new double[N];
```

```
        for (int i = 0; i < N; i++)
```

```
            c[i] = coords[i] + b.coords[i];
```

```
        return new Vector(c);
```

```
    }
```

مترها

# نوع داده‌ای بردار: پیاده‌سازی

۱۹

```
public Vector times(double t) {
    double[] c = new double[N];
    for (int i = 0; i < N; i++)
        c[i] = t * coords[i];
    return new Vector(c);
}

public double magnitude() {
    return Math.sqrt(this.dot(this));
}

public Vector direction() {
    return this.times(1.0 / this.magnitude());
}

...
}
```

مترها

- اشاره‌گر **this**. یک اشاره‌گر به شی فراخوانی کننده متد.
- مثال. وقتی متد **a.magnitude()** فراخوانی می‌شود، اشاره‌گر **this** یک نام دیگر برای شی ارجاع شده به وسیله **a** است.