

پروژه‌سازی پیشرفته



مفاهیم پایه

مهدی مصطفی زاده

سرفصل مطالب

- دستورات ورودی و خروجی
- متغیرها و انواع داده‌ی اولیه
- عملگرها
- کنترل جریان برنامه
- متد
- آرایه
- رشته



دستورات ورودی و خروجی

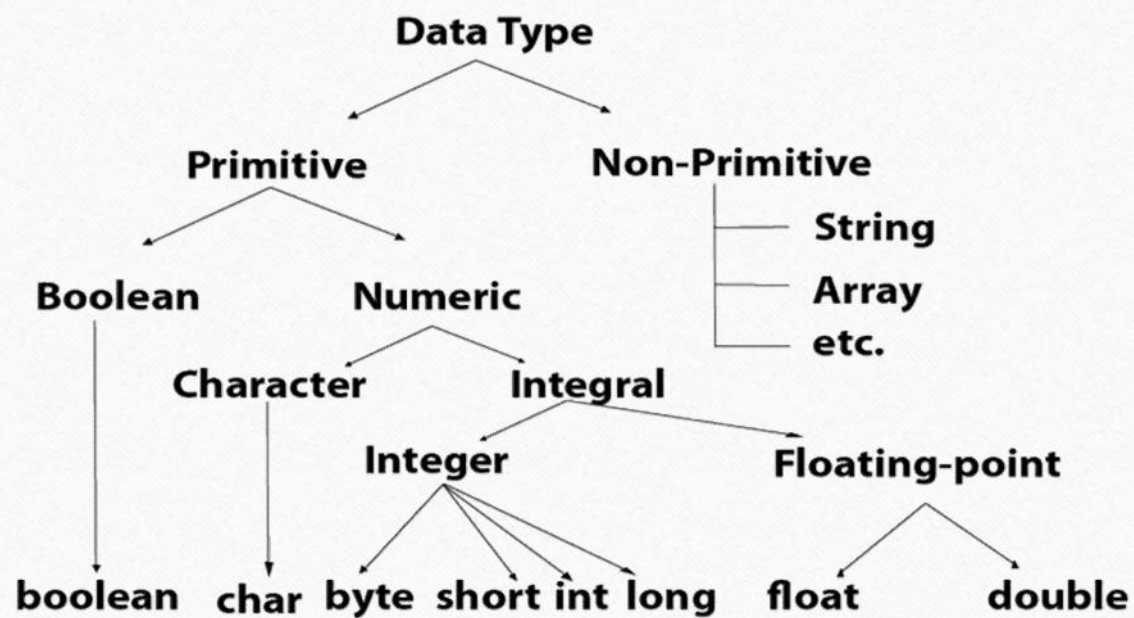
- مثال:

برنامه‌ای بنویسید که دو عدد صحیح را از کاربر دریافت کرده و حاصل جمع آن‌ها را در خروجی نمایش دهد.

- مفهوم بسته (package)
- رده‌ی `java.util.Scanner` برای خواندن ورودی
- استفاده از `System.out.println()` برای نمایش خروجی



انواع داده‌ی اولیه (دسته‌بندی)



انواع داده‌ی اولیه (دامنه)

Primitive Type	Size	Minimum Value	Maximum Value
char	16-bit	Unicode 0	Unicode $2^{16}-1$
byte	8-bit	-128	+127
short	16-bit	-2^{15} (-32,768)	$+2^{15}-1$ (32,767)
int	32-bit	-2^{31} (-2,147,483,648)	$+2^{31}-1$ (2,147,483,647)
long	64-bit	-2^{63} (-9,223,372,036,854,775,808)	$+2^{63}-1$ (9,223,372,036,854,775,807)
float	32-bit	32-bit IEEE 754 floating-point numbers	
double	64-bit	64-bit IEEE 754 floating-point numbers	
boolean	1-bit	true OR false	

نوع char در واقع یک عدد صحیح ۱۶ بیتی است که نویسه‌ای را در قالب سیستم یونیکد (Unicode-encoded) نمایش می‌دهد.

کمینه: '\u0000'

پیشینه: '\uffff'

جدول یونیکد:

unicode-table.com



انواع داده‌ی اولیه (تبدیل نوع)

Java Primitive Type Conversion rules

L→R	boolean	byte	short	char	int	long	float	double
boolean	-	N	N	N	N	N	N	N
byte	Not allowed	-	Y	C	Yes	Y	Y	Y
short	N	Cast	-	C	Y	Y	Y	Y
char	N	C	C	-	Y	Y	Y	Y
int	N	C	C	C	-	Y	Y* (loss)	Y
long	N	C	C	C	C	-	Y*	Y*
float	N	C	C	C	C	C	-	Y
double	N	C	C	C	C	C	C	-

Chapter 1 - 32



انواع داده‌ی اولیه (مقادیر ثابت)

Literal = مقدار ثابت

```
1 public class MyApp {
2
3
4 public static void main(String[] args) {
5     float f = 1.5;
6 }
7
8 }
9
10
```

Type mismatch: cannot convert from double to float

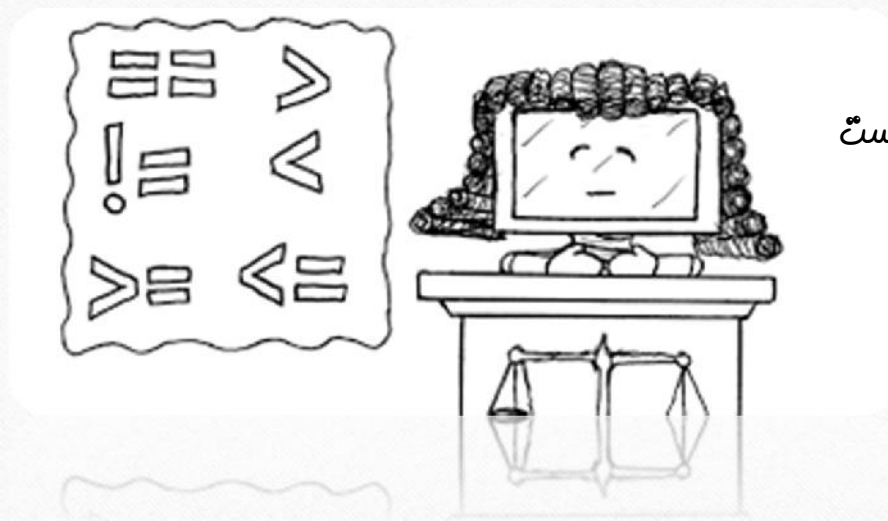
2 quick fixes available:

- [Add cast to 'float'](#)
- [Change type of 'f' to 'double'](#)

- IntegerLiteral: 123, 1231, 12400L
- FloatingPointLiteral: 123.4, 1.2e12, 12f, 12.3d
- BooleanLiteral: true, false
- CharacterLiteral: 'a', '\u123', '\377', ...
- StringLiteral: "a book", "tab\t and newline \n"



عملگرها



- در طراحی زبان جاوا از زبان C و C++ ایده گرفته شده است.
- عملگرها (رابطه‌ای، محاسباتی، منطقی و ...) مشابه همان چیزی است که در زبان سی دیده‌ایم.
- رفتار عملگر تقسیم با اعداد صحیح
- عملگر شرطی سه گانه



مشق شب

با جستجو در فضای مجازی و یا مراجعه به کتاب مرجع، عملگرهای زبان چاوا

(رابطه‌ای، محاسباتی، منطقی، انتساب، پیتی و شرطی)،

اولویت، شرکت‌پذیری و نحوه‌ی استفاده

از آن‌ها را مرور نمایید.



کنترل جریان برنامه

- ساختارهای شرطی

- if-else if-else

- switch

- ساختارهای تکرار (حلقه)

- for

- while

- do-while

- Foreach

- continue و Break



آزمونک

پ.پ.ک دائماً اعداد صحیح را از کاربرد دریافت نماید و هربار مشخص نماید آیا عدد وارد

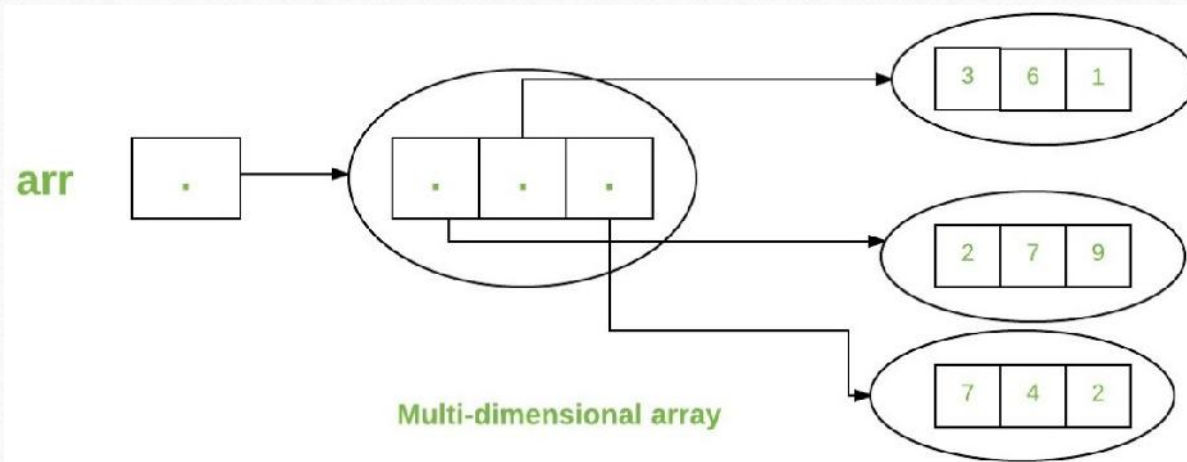
شده اول است یا خیر.

ورود عدد کوچکتر از یک به معنای اتمام برنامه است.



آرایه

- اگرچه در ظاهر شبیه آرایه‌ها در C هستند، اما تفاوت‌های زیادی دارند.



- آرایه‌ها هم شیء هستند!

- عضو `length`

- شروع اندیس از صفر



آرایه (تعریف و مقداردهی اولیه)

```
int[] myIntArray = new int[3];
```

```
int[] myIntArray = {1, 2, 3};
```

```
int[] myIntArray = new int[] {1, 2, 3};
```

```
String[] myStringArray = new String[3];
```

```
String[] myStringArray = {"a", "b", "c"};
```

```
String[] myStringArray = new String[] {"a", "b", "c"};
```

```
String[] myStringArray;
```

```
myStringArray = new String[] {"a", "b", "c"};
```



آرایه‌های چندبعدی

- برخلاف زبان C، اندازه‌ی آرایه‌های یک بعد می‌تواند باهم متفاوت باشد.

```
int[][] num = new int[5][2];  
int num[][] = new int[5][2];  
int[] num[] = new int[5][2];  
num[0][0] = 1;
```

```
int[][] num = { {1,2}, {1, 2, 3}};  
int[][] num = new int[5][];  
num[0] = new int[1];  
num[1] = new int[5];
```



متد (Method)

- مفهوم static
- انواع روش‌های ارسال پارامتر به تابع:
- ارسال با مقدار (Pass By Value)
- ارسال با ارجاع (Pass By Reference)
- متدهای دارای تعداد پارامتر متغیر



آرایه به عنوان ورودی و خروجی متد

- مثال) متدی بنویسید که یک آرایه‌ی دوبعدی (ماتریس) را به عنوان پارامتر دریافت کرده و بلندترین سطر آن را به عنوان خروجی برگرداند.
- آیا با تغییر خروجی، ماتریس اولیه تغییر می‌یابد؟ چرا؟



رشته

- رشته‌ها نیز شی هستند!
- حالت شی رشته شامل آرایه‌ای از نویسه‌ها (کاراکترها) است.
- `String str = new String("ali is a student");`
- `String str = "ali is a student";`
- متدهای رده‌ی رشته
- `charAt`, `concat`, `length`, `contains`, `equals`, `endsWith`, `indexOf`, `lastIndexOf`, `isEmpty`, `split`, `replace`, `replaceAll`
- حالت اشیاء رشته، غیر قابل تغییر (immutable) است.



عبارات منظم (Regular Expressions)

- عبارت منظم = الگوی جستجو در رشته‌ها
- آیا یک رشته، یک رایانامه‌ی معتبر را نشان می‌دهد؟
- آیا یک رشته، یک عدد معتبر را نشان می‌دهد؟



عبارات منظم (نحو)

توصیف	عبارت منظم
هر نویسه‌ای	.
a یا b یا c که پس از آن v یا z آمده باشد	[abc][vz]
هر رقمی، معادل [0-9]	\d
هر غیر رقمی، معادل [^0-9]	\D
هر فاصله‌ی سفیدی	\s
هر غیر فاصله‌ی سفیدی، معادل [^\s]	\S
یک کاراکتر کلمه‌ای، معادل [a-zA-Z_0-9]	\w
یک کاراکتر غیر کلمه‌ای، معادل [^\w]	\W
یک یا چند غیر فاصله‌ی سفید (* معادل صفر یا چند)	\S+
مطابق با X یا Y	X Y
دلخواه (معادل {0, 1})	?



عبارات منظم (مثال)

- اعداد صحیح زوج
- اعداد صحیح به طول زوج
- ایمیل‌های با پسوند دامنه com یا ir
- تعداد فرد کاراکتر نقطه (.)
- اعداد اعشاری
- اعداد صحیح و اعشاری

