chp2 Java基础知识

- **1**：**Java**语言的词法
- **2**：数据类型
- **3**：变量
- **4**：操作符
- **5**：流程控制语句
- **6**：数组
- **7**：字符串
- **7**：练习

1：Java 词法

- 空格 分号
- 注释
- 标识符
- 关键字

1.1 空格和标识符—Java 词法

Java是一种自由格式的语言，可以用任意个空格、制表符、换行符隔开每个词。如：

```
System.
out.
println
( "Counted " + count + " chars." );
```

分号作为语句的结束符，每个语句必须以分号作为结束符。

1.2 注释—Java 词法

三种注释格式：

- **//** 实现单行注释

- **/*** 这是
  ***** 一段注释，
  ***** 它跨越了多个行
  ***/**

- **/** JDK的javadoc工具用这种注释信息能　　自动从程序中抽出类的公共接口形成文档。
  …
  ***/**

1.3 标识符—Java 词法

- 定义：标识符用来作为类、方法和变量的名字
- 语法规则：以字母、下划线(_)、美元符号
- ($)开始，后跟这三种符号或数字。
  例如：identifier　　_sys_ var1
  $change
  　　userName　　user_ name
- 必须要区分大小写
- 没有长度限制

1.4 关键字—Java 词法

abstract	boolean	break	byte	case	catch
char	class	const*	continue	default	do
double	else	extends	final	finally	float
for	goto*	if	implements	import	instanceof
int	interface	long	native	new	null
package	private	protected	public	return	short
static	super	switch	synchronized	this	throw
throws	transient	try	void	volatile	while

关键字是由系统定义的一些词，它们在程序里代表特定的含义。
定义表示符时要避免使用这些词

2：数据类型

1：分类

- 基本数据类型(primitive type)
- 引用数据类型(reference type)

2：存储在什么地方

2.1 primitive type—数据类型

基本类型	大小	最小值	最大值	包装器类型
boolean	-	-	-	Boolean
char	16-bits	Unicode 0	Unicode 2^16-1	Character
byte	8-bits	-128	127	Byte
short	16-bits	-2^15	+2^15-1	Short
int	32-bits	-2^31	+2^31-1	Integer
long	64-bit	-2^63	+2^63-1	Long
float	32-bits	IEEE754	IEEE754	Float
double	64-bits	IEEE754	IEEE754	Double
void	-	-	-	Void

基本类型 取值范围 及其包装类

2.1 reference type-Java 数据类型

- 数组Array ，类 classe，接口interface属于 *reference* 类型。

reference 类型的变量是所引用的对象的内存地址。

```
int[] studentAges = new int[10];
java.util.Date today = new java.util.Date();
List students = new ArrayList();
```

2.2 存储、内存分配－Java 数据类型

	特点	缺点	存储内容
堆[heap]	可以动态地分配内存大小，生存期也不必事先告诉编译器	速度相对栈慢	由new创建的对象
栈[stack]	1：存取速度比堆要快，仅次于寄存器；2：栈数据可以共享	存在栈中的数据大小与生存期必须是确定的，缺乏灵活性	基本数据类型的值；对象引用

3：变量

- 变量的定义
- 变量的作用域
- 变量的声明
- 变量的默认值

3.1 变量的定义—变量

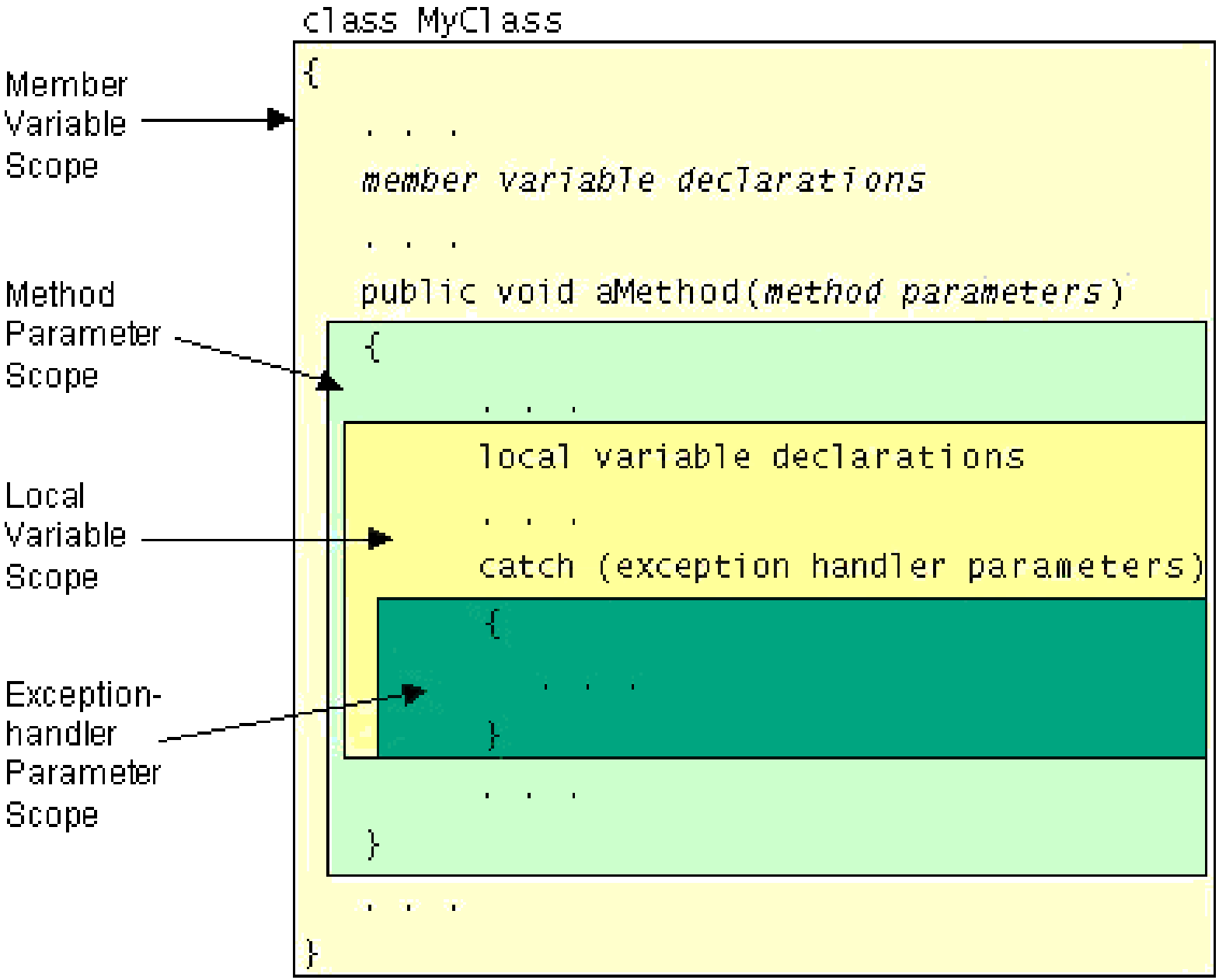
变量名是一个合法的标识符

- 它是字母、数字、下划线或美元符“$”的序列
- 变量名不能以数字开头
- 不能为保留字
- 变量名区分大小写
- 变量名应具有一定的含义，以增加程序的可读性

3.2 变量作用域—变量

变量的作用域指明可访问该变量的一段代码
按作用域来分，变量可以有下面几种：

- 局部变量 **Local variable**
- 类变量 **Member variable**
- 方法参数 **Method parameter**
- 例外处理参数 **Exception-handler parameter**


class MyClass
{
. . .
member variable declarations
. . .
public void aMethod(method parameters)
{
. . .
local variable declarations
. . .
catch (exception handler parameters)
{
. . .
}
. . .
}
. . .
}
Member Variable Scope
Method Parameter Scope
Local Variable Scope
Exception-handler Parameter Scope

3.3 变量的声明—变量

格式：**type identifier[=value][,identifier[=value]…]**;

例如：**int a, b, c;**

double d1, d2=0.0;

方法参数和例外处理参数的变量值是由调用者给出。

【规范写法】每行定义一个变量

```
int a;
int b;
double d2;
```

3.3 变量默认值—变量

Variable	Value
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
boolean	false
All reference types	null

【注】只有当变量作为类的成员使用时，Java才保证给定其默认值

4：操作符

- 操作符分类
- 操作符优先级

4.1 操作符分类

按功能分类：

- 算术操作符(+，-，*，/，%，++，--)
- 关系操作符(>，<，>=，<=，= =，!=)
- 布尔逻辑操作符(!，&&，||, |, &)
- 位操作符(>>，<<，>>>，&，|，^，~)
- 赋值操作符(=，及其扩展赋值运算符如+=)
- 条件操作符(?：)
- 其它 (包括分量操作符 ·，下标操作符 []，实例操作符 instanceof，内存分配操作符new，强制类型转换操作符 (类型)，方法调用操作符 () 等)

4.2 操作符优先级—操作符

Ulcer Addicts Really Like C A lot.
Ulcer--Unary-一元操作符
Addicts—Arithmetic(and shift)—算术、移位
Really—Relational—关系操作符
Like— Logic(and bitwise)—逻辑、位操作符
C- Conditional(ternary)-条件、三元操作符
A lot -- Assignment--赋值操作符
——引自Java编程思想
【提示】在不确定的地方加小括号，防止出错又便于理解。

5：流程控制语句

- if-else
- while
- do-while
- for
- switch

[注]Java没有goto

6：数组

- 声明数组
- 创建数组
- 初始化数组
- 二维数组
- 改变数组

6.1 声明数组—数组

两种方式：

```
int studentAges[];
int[] studentAges;  //推荐使用
```

6.2 创建数组—数组

- 1）创建数组时，必须指定数组长度
- 2）两种创建数组的方式

```
int[] studentAges = new int[3];
int[] studentAges = {20, 21, 22};
```

遍历数组，数组下标从0开始

```
//...
for (int i = 0; i < studentAges.length; i ++ ) {
        System.out.println(studentAges[i]);
}
//...
```

6.3 初始化数组—数组

两种方式：

1）先定义，后分别为数组中的每个元素赋值

2）定义同时初时化，这个限制稍微大一些，
因为有些数组是在运行时才能决定长度的

参考：net.xxt.chp2.ArrayInit

6.4 多维数组—数组

Java中多维数组被看作数组的数组。例如二维数组为一个特殊的一维数组，其每个元素又是一个一维数组。

```
int[][][] points = new int[2][][];
//int[][][] points2 = new int[][2][]; // ERROR，多维数组定义时必须至少指定第一维的长度
//int[][][] points3 = new int[2][][2];  // ERROR，在没有明确前一维的长度时，不能定义下一维的长度
int[][][] points4 = new int[2][2][3]; // 指定所有维度的大小

points[0] = new int[2][3]; // 可以未每一维再指定不同的长度
points[1] = new int[1][2];
```

参考：net.xxt.chp2.MutiDimArray

6.5 改变数组

- 不能改变一个已定义的数组的大小
- 可以改变引用变量所指向的数组

7：字符串

- 字符串初始化
- 字符串比较
- 字符串常见操作

7.1 字符串初始化—字符串

```
String name = null;

// 用一个字符串常量直接给字符串赋值
name = "yangyuwei";

// 使用new String()初始化字符串
name = new String("yangyuwei");

String nickname = name;
```

????比较三者的区别，从内存分配方面考虑

7.2 字符串比较—字符串

```
String first = "yangyuwei";
String second = new String("yangyuwei");
String third = "yangyuwei";

System.out.println(first == second);
System.out.println(first == third);
```

【思考输出？？】

7.3 字符串常见操作—字符串

length()

trim()

replace(String, String)

split(String)

startsWith(String)

substring(int)

substring(int, int)

8：练习

1）heap stack的区别是什么，使用情况？

2）基本数据类型中int long short byte的取值范围是多少，用十进制表示。

3）思考7.2例子中的原因，详细分析

4）查找资料：String StringBuffer这两个类的关系及常见使用注意事项

5）查找资料：java注释、javadoc，养成写注释、写好注释的习惯