**每日作业卷**

就业班JavaSE--day06【Date、DateFormat、Calendar、Math、System、包装类、正则表达式】

# 关卡1

## 训练案例1

### 训练考核知识点

Date类

### 训练描述

请说出你对Date类的理解

### 操作步骤描述

1. 请问Date类用于表示什么 ? Date类是操作最简单的日期类，代表一个瞬间。
2. 请问如何使用Date 类获取当前的日期? public Date()
3. 请问如何把当前的日期转换为毫秒值? public long getTime()
4. 请问如何把毫秒值转换为日期? public Date(long date)

## 训练案例2

### 训练考核知识点

日期格式化

### 训练描述

请阐述如何实现日期对象与指定格式的字符串之间的转换

### 操作步骤描述

1. 请说出日期格式化的核心类是什么? DateFormat
2. 请说出日期指定格式的常见的规则是什么 ? yyyy年MM月dd日HH:mm:ss
3. 请说出如何把日期转换为指定格式的字符串 ? public final String format(Date date)
4. 请说出如何把指定格式的字符串转换为日期 ?

Date dTime = format.parse(sTime);

## 训练案例3

### 训练考核知识点

Calendar类

### 训练描述

请阐述Calendar的基本使用

### 操作步骤描述

1. 请说出如何获取Calendar对象 public static Calendar getInstance()
2. 请说出Calendar的常见方法有哪些,作用是什么?

public int get(int field) //获取时间字段值 int field对应的字段

public final void set(intyear,intmonth,int date) //设置年月日，可查看对应重载

public final Date getTime() //获取该日历对象转成的日期对象

1. 请说出Calendar注意事项有哪些?

在获取月份时，Calendar.MONTH + 1

在获取星期几 Calendar.DAY\_OF\_WEEK – 1

获取日期时 Calendar.DAY\_OF\_MONTH 不需要特殊的操作，他直接返回一个月中的第几天

## 训练案例4

### 训练考核知识点

System类

### 训练描述

请阐述System类的理解

### 操作步骤描述

1. 请说System类的特点是什么?

System类：系统类，主要用于获取系统的属性数据，没有构造方法

1. 请说System类的常见方法有哪些,作用是什么?

public static void exit(int status)：让操作系统终止该java程序 0代表正常终止

public static void gc()：暗示java虚拟机回收不使用的对象(垃圾)

public static String getProperty(String key) ：查找制定的系统属性 , 系统属性参照API

## 训练案例5

### 训练考核知识点

Math类

### 训练描述

请阐述你对Math类的理解

### 操作步骤描述

1. 请阐述Math类的概念?

Math类包含用于执行基本数学运算的方法，如初等指数、对数、平方根和三角函数。

1. 请说出Math类的常见方法有哪些,作用是什么?

public static int abs(int a)：返回a的绝对值  
public static double ceil(double a) ：求大于等于a的最小整数,返回该整数的小数形式

public static double floor(double a) ：求小于等于a的最小整数

public static long round(double a) ：对a进行四舍五入,返回四舍五入后的值

public static double pow(double a, double b) ：求a的b次幂,返回double形式

## 训练案例6

### 训练考核知识点

包装类型

### 训练描述

请阐述你对基本数据类型包装类的理解

### 操作步骤描述

1. 请说出基本数据类型包装类的主要作用是什么?

可以提供更多的操作基本数值的功能

1. 请说出基本数据类型对应的八个包装类分别是什么?

int对应的是Integer，char对应的Character，其他6个都是基本类型首字母大写

1. 请说出如何把字符串转换为对应的基本数据类型,有什么主意点?

调用包装类的 parseXxx 静态方法。调用包装类的 valueOf() 方法转换为基本类型的包装类，会自动拆箱。

## 训练案例7

### 训练考核知识点

自动装箱

自动拆箱

### 训练描述

请阐述你对自动装箱和自动拆箱的理解

### 操作步骤描述

1. 请说出你对自动装箱的理解

将基本类型的数值转成基本类型对应的包装类对象

1. 请说出你对自动拆箱的理解

将基本类型包装类的对象转成该基本类型的数值

## 训练案例8

### 训练考核知识点

正则表达式

### 训练描述

请阐述你对正则表达式的理解

### 操作步骤描述

1. 请说出什么是正则表达式

正则表达式是专门解决字符串规则匹配的工具。

1. 请说出正则表达式的作用是什么?

正则表达式也是一个字符串，用来定义匹配规则。

1. 请说出字符串中关于正则表达式的常见方法有哪些?

public boolean matches(String regex) //被匹配的字符串调用方法,接收正则表达式字符串,如果匹配返回true,否则返回false

public String[] split(String regex) //被匹配的字符串调用方法,将字符串中的字符与规则匹配,如果符合就匹配为分隔符,将一个完整的字符串通过这些分隔符切割为不同的子字符串

1. 请说出正则表达式的匹配规则是什么?

. 匹配除 "\n" 之外的任何单个字符。要匹配包括 '\n' 在内的任何字符，请使用象 '[.\n]' 的模式。

\d 匹配一个数字字符。等价于 [0-9]。

\D 匹配一个非数字字符。等价于 [^0-9]。

\s 匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [ \f\n\r\t\v]。

\S 匹配任何非空白字符。等价于 [^ \f\n\r\t\v]。

\w 匹配包括下划线的任何单词字符。等价于'[A-Za-z0-9\_]'。

\W 匹配任何非单词字符。等价于 '[^A-Za-z0-9\_]'。

## 训练案例9

### 训练考核知识点

Date类

SimpleDateFormat类

### 训练描述

请用代码实现:

获取当前的日期,并把这个日期转换为指定格式的字符串,如2088-08-08 08:08:08

### 操作步骤描述

1. 创建包com.itheima.level1
2. 定义类Test09
3. 提供main方法,在main方法中
   1. 获取当前日期对象 now;
   2. 创建SimpleDateFormat对象 df,并制定日期格式
   3. 调用df的format(Date date) 方法,传入now; 接收返回的字符串
   4. 打印这个字符串

package com.java.test01;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Test01 {

public static void main(String[] args) {

Date now = new Date();

System.out.println(now);

String rule = "yyyy年MM月dd日HH:mm:ss";

DateFormat df = new SimpleDateFormat(rule);

String s = df.format(now);

System.out.println(s);

}

}



## 训练案例10

### 训练考核知识点

Date类

SimpleDateFormat类

异常处理

### 训练描述

请用代码实现如下需求:

把你的生日字符串表示转换为对应的日期对象.

### 操作步骤描述

1. 在com.itheima.level1包下创建Test10类
2. 在Test10类中,提供main方法,在main方法中
   1. 创建SimpleDateFormat对象 df,并制定日期格式
   2. 调用df的parse(String str)方法,传入你的生日字符串;结束返回的日期对象
   3. 打印这个日期对象
   4. 使用try...catch处理解析异常

package com.java.test02;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class Test02 {

public static void main(String[] args) {

String rule = "yyyy年MM月dd日";

DateFormat df = new SimpleDateFormat(rule);

try {

Date date = df.parse("1997年07月31日");

System.out.println(date);

} catch (ParseException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}



## 训练案例11

### 训练考核知识点

Calendar类

### 训练描述

使用代码实现

500天后是几几年几月几日.

### 操作步骤描述

1. 创建包com.itheima.level1中创建Test11
2. 在Test11类中,提供main方法,在main方法中
   1. 获取当前日历对象
   2. 调用日期对象的add()方法,让当前日历向后移动500天
   3. 获取日历中的年,月,日
   4. 输出几几年几月几日.

**package** com.java.test03;

**import** java.util.Calendar;

**import** java.util.Date;

**public** **class** Test03 {

**public** **static** **void** main(String[] args) {

Calendar rightNow = Calendar.*getInstance*();

rightNow.add(Calendar.*DAY\_OF\_MONTH*, +500);

**int** year = rightNow.get(Calendar.*YEAR*);

**int** month = rightNow.get(Calendar.*MONTH*)+1;

**int** day = rightNow.get(Calendar.*DAY\_OF\_MONTH*);

System.*out*.println(year + "年" + month + "月" + day + "日");

}

}



## 训练案例12

### 训练考核知识点

正则表达式

### 训练描述

编写一个校验用户名的程序,检测键盘录入的用户名是否合法

要求:必须以英文开头,只能包含英文,数字和`\_`;最少6位,做多12位

### 操作步骤描述

1. 在com.itheima.level1中创建Test12类
2. 在Test12类中提供main方法,在main方法中
   1. 创建键盘录入对象
   2. 接收用户录入的用户名字符串
   3. 调用字符串的matches方法,判断用户名是否合法
   4. 如果合法就打印: 用户名合法
   5. 如果不合法就打印: 用户名不合法

**package** com.java.test04;

**import** java.util.Scanner;

**public** **class** Test04 {

**public** **static** **void** main(String[] args) {

System.*out*.println("请输入用户名：");

Scanner scanner = **new** Scanner(System.*in*);

String s = scanner.nextLine();

String regex = "[a-zA-Z][a-zA-Z\_0-9]{5,11}";

**boolean** result = s.matches(regex);

**if** (result) {

System.*out*.println("用户名合法");

} **else** {

System.*out*.println("用户名不合法");

}

}

}







# 关卡2

## 训练案例1

### 训练考核知识点

Date类

### 训练描述

请使用代码实现

计算你活了多少天

### 操作步骤描述

1. 创建包com.itheima.level2
2. 在包中创建测试类Test13
3. 在Test13类中,提供main方法,在main方法中
   1. 把你生日的字符串,转换为日期对象
   2. 把日期对象转换为对应的毫秒值
   3. 获取当前系统的毫秒值
   4. 当前系统毫秒值-出生时间对于的毫秒值计算你活的毫秒值
   5. 把你活毫秒值转换为天
   6. 输出天数
   7. 使用try...catch处理异常

**package** com.java.test05;

**import** java.text.DateFormat;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **class** Test05 {

**public** **static** **void** main(String[] args) {

String rule = "yyyy年MM月dd日HH:mm:ss";

DateFormat format = **new** SimpleDateFormat(rule);

String birth = "1997年07月31日00:00:00";

Date date = **null**;

**try** {

date = format.parse(birth);

} **catch** (ParseException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**long** time1 = date.getTime();

date = **new** Date();

**long** time2 = date.getTime();

**long** time = time2 - time1;

**long** day = time/(3600\*24\*1000);

System.*out*.println(time);

System.*out*.println(day);

}

}



## 训练案例2

### 训练考核知识点

System类

Date类

SimpleDateFormat类

### 训练描述

写代码验证使用SimpleDateFormat把日期转换为字符串的时候;

创建SimpleDateFormat对象,比较消耗程序的执行性能

### 操作步骤描述

1. 在com.itheima.level2包下创建Test14类
2. 在Test14类中提供main方法,在main方法中
   1. 使用System的currentTimeMillis()方法获取当前操作系统的毫秒值,作用程序执行的开始时间,使用start变量接收
   2. 需要测试执行性能的代码
   3. 使用System的currentTimeMillis()方法获取当前操作系统的毫秒值,作用程序执行的结束时间,使用end变量接收
   4. 计算代码执行花费的时间 end - start,输出代码执行消耗的时间
3. 在Test14类中提供一个静态方法testSimpleDataFormatInLoop(),在方法中
   1. 创建日期(Date)对象 now
   2. 写一个循环100000次的for循环
   3. 在循环中
      1. 创建SimpleDateFormat对象 df,并制定日期格式为yyyy-MM-dd
      2. 调用df的format(Date date),传入now
   4. 在main方法中,需要测试执行的性能的代码处调用testSimpleDataFormatInLoop()方法
   5. 记录程序执行的效率
4. 在Test14类,完整复制testSimpleDataFormatInLoop()方法
   1. 改名为testSimpleDataFormatOutLoop
   2. 把创建SimpleDateFormat的代码移动到循环外部
   3. 在main方法中,注释调用testSimpleDataFormatInLoop();
   4. 在语句下一行调用testSimpleDataFormatOutLoop方法
   5. 运行程序,查看输出结果,并记录
5. 得出结论: 创建日期格式对象,确实比较消耗程序性能,所以在实际开发一般会写一个工具类专门处理,日期的格式化与解析.

**package** com.java.test06;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **class** Test06 {

**public** **static** **void** testSimpleDataFormatInLoop() {

Date now = **new** Date();

**for** (**int** i = 0; i < 1000; i++) {

DateFormat df = **new** SimpleDateFormat("yyyy-MM-dd");

String sNow = df.format(now);

}

}

**public** **static** **void** testSimpleDataFormatOutLoop() {

Date now = **new** Date();

DateFormat df = **new** SimpleDateFormat("yyyy-MM-dd");

**for** (**int** i = 0; i < 1000; i++) {

String sNow = df.format(now);

}

}

**public** **static** **void** main(String[] args) {

**long** start = System.*currentTimeMillis*();

//testSimpleDataFormatInLoop();

*testSimpleDataFormatOutLoop*();

**long** end = System.*currentTimeMillis*();

**long** time = end - start;

System.*out*.println(time);

}

}

## 训练案例3

### 训练考核知识点

正则表达式

字符串的split(String regex)方法

### 训练描述

使用正则表达式切割字符串

\* 192.168.105.27 按照 . 号切割字符串

\* 18 22 40 65 按照空格切割字符串

提示: 由于 . 号 在正则表达式中用于匹配除换行符以外的任意字符,会把字符串切的连渣都不剩,此处使用 . 号需要转义; 使用\\.作为切割使用正则表达式

### 操作步骤描述

1. 在com.itheima.level2包下创建Test15
2. 在Test15类中提供main方法,在main方法中
   1. 定义字符串变量str1,赋值为192.168.105.27
   2. 调用str1的split的方法,传入 “\\.”,使用字符春数组strs1接收
   3. 遍历数组strs1,输出每一个元素
   4. 定义字符串变量str2,赋值为 18 22 40 65
   5. 调用str1的split的方法,传入“\\s+”,使用字符春数组strs2接收
   6. 遍历数组strs2,输出每一个元素

**package** com.java.test07;

**public** **class** Test07 {

**public** **static** **void** main(String[] args) {

String str1 = "192.168.105.27";

String[] strs1 = str1.split("\\.");

**for** (String s : strs1) {

System.*out*.print(s + " ");

}

System.*out*.println("-----------");

String str2 = "18 22 40 65";

String[] strs2 = str2.split("\\s+");

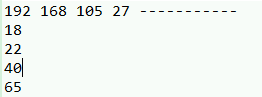
**for** (String s : strs2) {

System.*out*.println(s);

}

}

}



## 训练案例4

### 训练考核知识点

正则表达式

字符串类的split方法

字符串类的matches方法

### 训练描述

请使用代码描述:

有一段文本 " anglebaby 女 18 515151511 "

分别描述一个人的姓名 性别 年龄(int) 和QQ号码信息,完成如下需求:

1. 创建一个Person类,包含性别,性别,年龄和QQ号码属性,使用解析出来信息创建一个Person对象,并打印这个人信息

2. 并判断QQ号是否合法,匹配规则,全部都是数字,第一位非0,长度为5到12位.

### 操作步骤描述

1. 在com.itheima.level2包中
2. 创建Student类
   1. 属性: 姓名(name),性别(sex),年龄(int age),QQ号码(qq)
   2. 提供空参,有参构造方法以及setters与getters方法
   3. 重写toString()方法,返回属性值
3. 创建Test16类,在Test16类中:
   1. 提供main方法,在main方法中
      1. 定义字符串变量str,赋值为:" anglebaby 女 18 515151511 "
      2. 取出字符str两端的空格,避免切割的时候,产生空字符串
      3. 使用”\\s+”正则表达式切割字符串,因为中间的空格可能包含制表符,多个空格
      4. 切割后,字符串数组0索引->姓名,1索引->性别,2索引->年龄,3索引->QQ号
      5. 把年龄字符串转换int类型,赋值给变量age
      6. 使用满参构造创建Student对象s,传入对应的数据
      7. 打印学生s信息
      8. 判断该学生的QQ号码是否合法,如果合法输出QQ号码合法,否则输出QQ号码不合法

**package** com.java.test08;

**public** **class** Test08 {

**public** **static** **void** main(String[] args) {

String str = " anglebaby 女 18 515151511 ";

String str1 = str.trim();

String[] strings = str1.split("\\s+");

**int** age = Integer.*parseInt*(strings[2]);

Student student = **new** Student(strings[0], strings[1], age, strings[3]);

System.*out*.println(student.toString());

String s = strings[3];

String regex = "[1-9][0-9]{4,9}";

**boolean** result = s.matches(regex);

**if** (result) {

System.*out*.println("QQ号码合法");

} **else** {

System.*out*.println("QQ号码不合法");

}

}

}



## 训练案例5

### 训练考核知识点

正则表达式

字符串类的replaceAll(String regex,string str)方法

### 训练描述

通过代码实现:

把字符串 "Hello12345World6789012" 中所有数字替换为 "#"

提示: 字符串类的replaceAll(String regex,string str)方法,使用str替换掉当前字符串中所有与正则表达式匹配的字符

### 操作步骤描述

1. 在com.itheima.level2包中
2. 创建Test17类,在Test17类中:
   1. 提供main方法,在main方法中
   2. 定义字符串变量str,赋值为:"Hello12345World6789012"
   3. 调用str的replaceAll(String regex,String str),第一个参数传入”\\d+”,第二个参数传入”#”,定义字符串变量newStr接收替换后的字符串
   4. 输出: newStr

**package** com.java.test09;

**public** **class** Test09 {

**public** **static** **void** main(String[] args) {

String str = "Hello12345World6789012";

String newStr = str.replaceAll("\\d+", "#");

System.*out*.println(newStr);

}

}



# 关卡3

## 训练案例1

### 训练考核知识点

SimpleDateFormat

### 训练描述

使用SimpleDateFormat类,把2016-12-18转换为2016年12月18日.

提示: SimpleDateFormat对象的日期格式字符串可以通过[**applyPattern**](mk:@MSITStore:C:\Users\58334\Desktop\JDK_API_1_6_zh_CN.CHM::/java/text/../../java/text/SimpleDateFormat.html#applyPattern(java.lang.String))([String](mk:@MSITStore:C:\Users\58334\Desktop\JDK_API_1_6_zh_CN.CHM::/java/text/../../java/lang/String.html) pattern)进行修改

要求:异常处理使用throws的方式

**package** com.java.test10;

**import** java.text.DateFormat;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **class** Test10 {

**public** **static** **void** main(String[] args) **throws** Exception {

String rule1 = "yyyy-MM-dd";

DateFormat format1 = **new** SimpleDateFormat(rule1);

String sTime = "2016-12-18";

Date dTime = **null**;

dTime = format1.parse(sTime);

String rule2 = "yyyy年MM月dd日";

DateFormat format2 = **new** SimpleDateFormat(rule2);

String sNow = format2.format(dTime);

System.*out*.println(sNow);

}

}



## 训练案例2

### 训练考核知识点

使用System类获取当前系统的毫秒值

统计代码的执行时间

String拼接

StringBuilder拼接

### 训练描述

请用代码实现

分别使用String的+= 和StringBuilder的append方法对字符串做100000次拼接,

计算String拼接100000次花费时间与StringBuilder拼接100000次所花费时间并打印

**package** com.java.test11;

**public** **class** Test11 {

**public** **static** **void** testString() {

String s = "a";

String result = "";

**for** (**int** i = 0; i < 100000; i++) {

result += s;

}

}

**public** **static** **void** testStringBuilder() {

String s = "a";

StringBuilder stringBuilder = **new** StringBuilder();

**for** (**int** i = 0; i < 100000; i++) {

stringBuilder.append(s);

}

}

**public** **static** **void** main(String[] args) {

**long** start = System.*currentTimeMillis*();

//testString();

*testStringBuilder*();

**long** end = System.*currentTimeMillis*();

**long** time = end - start;

System.*out*.println(time);

}

}





## 训练案例3

### 训练考核知识点

正则表达式

### 训练描述

模拟验证手机号的过程，按照以下要求实现相关代码（友情提示：不一定要用正则）

a. 提示用户在控制台输入手机号，用一个字符串对其进行保存

b. 判断该手机号位数是否是11位

c. 判断该手机号，是否都是数字

d. 判断改手机号最后3为数字如果是相同数字则认为是靓号

**package** com.java.test12;

**import** java.util.Scanner;

**public** **class** Test12 {

**public** **static** **void** main(String[] args) {

System.*out*.println("请输入手机号：");

Scanner scanner = **new** Scanner(System.*in*);

String number = scanner.nextLine();

String regex1 = ".{11}";

String regex2 = "[0-9]{1,11}";

**int** len = number.length();

**char** s1 = number.charAt(len - 1);

**char** s2 = number.charAt(len - 2);

**char** s3 = number.charAt(len - 3);

Boolean result1 = number.matches(regex1);

Boolean result2 = number.matches(regex2);

Boolean result3 = (s1 == s2 && s1 == s3);

**if** (result1) {

System.*out*.println("该手机号位数是11位");

} **else** {

System.*out*.println("该手机号位数不是11位");

}

**if** (result2) {

System.*out*.println("该手机号都是数字");

} **else** {

System.*out*.println("该手机号不都是数字");

}

**if** (result3) {

System.*out*.println("该手机号是靓号");

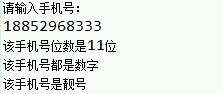
} **else** {

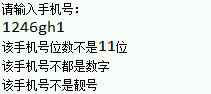
System.*out*.println("该手机号不是靓号");

}

}

}





## 训练案例3

### 训练考核知识点

IO流

集合

正则表达式

字符串的split方法

数组

### 训练描述

请使用实现:

已知项目的根目录下有一个student.txt,内容如下:

1 郝婷婷 女 30

2 赵鹏 男 35

3 周强 男 38

4 李艳欣 女 37

5 朱可可 女 41

6 张少荣 女 30

7 王小军 男 28

该文件中,每一行表示一个学生信息,从前往后依次是学号,姓名,姓名,年龄

要求:

1. 读取文件内容,把每一个学生信息封装为一个学生对象,存储到ArrayList集合中
2. 打印集合中的学生信息
3. 计算平均年龄
4. 异常处理使用throws的方式

**package** com.java.test13;

**import** java.io.BufferedReader;

**import** java.io.FileReader;

**import** java.util.ArrayList;

**public** **class** Test13 {

**public** **static** **void** main(String[] args) **throws** Exception {

BufferedReader bufferedReader = **new** BufferedReader(**new** FileReader("student.txt"));

String line;

ArrayList<Student> arrayList = **new** ArrayList<Student>();

**while** ((line = bufferedReader.readLine()) != **null**) {

Student student = **new** Student();

String[] strings = line.split("\\s+");

student.setId(strings[0]);

student.setName(strings[1]);

student.setSex(strings[2]);

student.setAge(strings[3]);

arrayList.add(student);

}

**for** (**int** i = 0; i < arrayList.size(); i++) {

System.*out*.println(arrayList.get(i).toString());

}

**int** allAge = 0;

**for** (**int** i = 0; i < arrayList.size(); i++) {

allAge += Integer.*parseInt*(arrayList.get(i).getAge());

}

**int** avgAge = allAge / arrayList.size();

System.*out*.println("平均年龄：" + avgAge);

}

}

