

Java程序设计编程作业-2

作业说明

请按照本文档的题目要求和代码框架完成编程作业。代码框架的示例结构如下所示，在 `src` 文件夹目录下，每一个题目对应一个文件夹(package)。在package内部，通常题目要求实现若干个类(class)，分别对应一个java源代码文件（未完整实现）。此外，每个题目有一个包含main函数的公有类 `Test.java`，功能是对题目中待实现的类进行测试。最后，在 `src` 文件夹目录下有一个 `testutil` 包，是测试时所用到的工具，同学无需阅读或更改。

```
1  └─hw1
2      └─README.pdf
3      |
4      └─src
5          └─problem1
6              └─Class1.java (待实现)
7              └─Class2.java (待实现)
8              └─Test.java
9          └─problem2
10             └─Class1.java (待实现)
11             └─Test.java
12             ...
13             |
14             └─testutil
```

同学们完成作业的建议步骤如下：

1. 按照文档，在代码框架中完成题目所要求的功能类的完整实现。
2. 运行每个题目的 `Test` 类，根据测试结果进行调试。
3. **严格按照以下文件结构和命名提交源代码文件，并打包成 zip 格式上传，文件命名为 学号_姓名.zip。**

```
1  └─2020123456_张三 (学号_姓名)
2      └─report.pdf/txt/md (实验报告，可选)
3      |
4      └─src
5          └─problem1 (抽象名，根据实际题名决定，小写)
6              └─Class1.java (抽象名，根据实际类名决定，首字母大写)
7              └─Class2.java
8              └─... (其他java源代码)
9          |
10         └─problem2
11             └─Class1.java
12             └─... (其他java源代码)
13             ...
```

请注意：

- 编程作业为个人作业，请独立完成。如果提交的代码参考了资料或个人，请在report中标注来源，否则视为抄

袭。

- 编程作业采用自动化评判，因此不按照规定格式提交作业可能导致错误。
- 请严格按照代码框架和文档要求对成员变量、类、函数、文件命名，函数参数类型、名称、顺序和函数返回值类型请严格按照文档实现，此外类、成员变量和函数访问权限请按照文档要求指定。**Java严格区分大小写，不符合要求的命名会导致错误。**
- 请不要更改代码框架的文件结构和命名，请不要删除代码框架中java源文件开头的包声明 `package ...;`。
- 除了题目要求外，可以根据需要实现其他类、函数和成员变量。可以提交其他新增的源代码文件，在新增的源代码文件开头必须进行包声明 `package ...;`。
- `Test` 类对应的源代码文件以及 `testutil` 包可以不提交，同时同学们也可以不改写。无论它们是否被提交或改写，评判过程中会对这两类文件重新覆盖。
- `Test` 类包含了测试程序的一些基本测例。无特殊说明的情况下，**作业框架中提供的测例是评判时测例的子集**。最终的评判分数为同学所提交代码在全体测例中通过的比例。同学们可以在 `Test` 类中根据需要补充测例，进行测试。但是本次作业提供的测例是评判时的全部测例。
- 实验报告是可选的，内容为除了源代码外额外需要说明的内容，例如参考的资料，复杂题目的实现思路等。**实验报告不影响本次评判。**
- 文档和实验框架的错误、歧义和bug可联系助教wujy22@mails.tsinghua.edu.cn。
- Java API官方在线文档：<https://docs.oracle.com/en/java/javase/19/docs/api/>

Problem 1: Date (25")

在包 `date` 中创建功能类 `Date`，表示日期。该类所要求的接口见以下伪代码。

1. 定义三个重载的构造函数，在创建 `Date` 对象时，可用以下三种格式来进行初始化：

```
1  /**
2   * 参数为一个模式化的字符串，使用月份的缩写，格式为"Month.Day,Year"，形如："Feb.7,2005"。
3   * 月份的缩写分别是：Jan. Feb. Mar. Apr. May. Jun. Jul. Aug. Sept. Oct. Nov. Dec.
4   * 日期前的0省略，例如"Feb.07,2005"不是合法参数，不会出现这种情况。
5   * @param s 字符串。
6   */
7  Date(String s);
8
9  /**
10 * 参数为一个字符串和两个整数。保证参数是一个合法的日期，例如年份为-10的情况不会出现。
11 * @param m 月份缩写，见上，例如"Feb."。
12 * @param d 日，整数类型。
13 * @param y 年，整数类型。
14 */
15 Date(String m, int d, int y);
16
17 /**
18 * 参数为三个整数 year、month、day，分别表示年、月、日。保证参数是一个合法的日期。
19 * @param d 日，整数类型。
20 * @param m 月，整数类型。
21 * @param y 年，整数类型。
22 */
23 Date(int d, int m, int y);
```

2. 日期的字符串输出，为公有函数。

```
1  /**
2  * 该日期的格式化字符串输出，格式为："Year/Month/Day"，例如"2018/10/11"，注意月份和日期前不要输出额外的0，例如"2018/07/03"不是正确的。
3  * @return 该日期对应的格式化字符串。
4  */
5  String toString();
```

测例数据范围：

- 年份满足 $1000 \leq Year \leq 9999$ 。
- 月份满足 $1 \leq Month \leq 12$ 。
- 日期满足 $1 \leq Day \leq 31$ 。
- 所有出现的日期都是合法的，例如2022/2/31的情况不需要考虑。

Problem 2: HugeInteger (50")

`long` 数据类型最多能表示19位十进制整数，那么对于更大的整数应该如何描述呢？在包 `hugeinteger` 中创建功能类 `HugeInteger`，该类用来存放和操作一个不超过100位的十进制大整数。该类所要求的接口见以下伪代码。

1. 2个重载的构造函数，支持通过 `long` 和 `String` 类型初始化一个 `HugeInteger` 对象。

```
1  /**
2  * 构造函数，用来对大整数进行初始化。参数为一个long类型。
3  * @param i
4  */
5  HugeInteger(long i);
6
7  /**
8  * 构造函数，用来对大整数进行初始化。参数为一个字符串。
9  * @param s 例如"0","1234","2354246324142356"。
10 */
11 HugeInteger(String s);
```

2. 大整数的十进制字符串表示，公有函数。

```
1  /**
2  * 大整数的十进制字符串表示。
3  * @return 例如"0","1234","2354246324142356"。
4  */
5  String toString();
```

3. 实现大整数的加法，公有函数。

```

1  /**
2  * 定义 add 成员函数，实现两个大整数的加法。参数为一个HugeInteger对象，无返回值，例如：
3  * HugeInteger A = new HugeInteger("12345");
4  * HugeInteger B = new HugeInteger("1234");
5  * A.add(B);
6  * 此时，A为13579，B为1234。
7  * @param b 另一个HugeInteger对象。
8  */
9  void add(HugeInteger b);

```

4. 定义2个重载的大整数关系运算。

```

1  /**
2  * 定义大整数关系运算的成员函数。
3  * a.compareTo(b)的输出：
4  * a<b: -1
5  * a==b: 0
6  * a>b: 1
7  * @param o 另一个HugeInteger对象。
8  * @return -1,0,1。
9  */
10 int compareTo(HugeInteger o);
11
12 /**
13 * 定义大整数关系运算的成员函数。
14 * a.compareTo(b)的输出：
15 * a<b: -1
16 * a==b: 0
17 * a>b: 1
18 * @param o 另一个整数，用long类型表示。
19 * @return -1,0,1。
20 */
21 int compareTo(long o);

```

测例数据范围：

- 题目中出现的所有大整数参数和运算结果满足： $0 \leq X < 10^{100}$ 。

限制：

- 本题禁用java.math.BigInteger，否则本题计0分。

Problem 3: SavingsAccount (25")

在包savingsaccount中创建功能类SavingsAccount，表示银行账户。该类所要求的接口见以下伪代码。

1. 银行的存款年利率是固定的，为一个范围在[0,1]中的实数，是所有客户共享的，与特定的账户无关。年利率初始值为1%。
2. 定义一个公有成员函数用来将年利率设定为一个新的值。利率的调整会影响到银行中的所有客户。

```
1  /**
2   * 将年利率设定为一个新的值。
3   * @param interestRate 新的年利率。
4   */
5  void modifyInterestRate(double interestRate);
```

3. 定义一个构造函数，参数为一个实数类型表示存入的余额。

```
1  /**
2   * 构造函数，存入一定余额。
3   * @param savings 存入的余额，非负。
4   */
5  SavingsAccount(double savings);
```

4. 定义一个公有成员函数用来计算客户每个月的利息收入。

```
1  /**
2   * 计算客户每个月的利息收入，计算公式：当前存款 *  $[(1 + \text{年利率})^{(1/12)} - 1]$ 。该利息随后要加入到余额中，
   * 即使用复利。
3   * @return savings 该月的利息。
4   */
5  double calculateMonthlyInterest();
```

测例数据范围：

- $0 \leq \text{savings} \leq 10^6$.
- $0 \leq \text{interestRate} \leq 1$.
- 每个客户的当前存款总额不超过 10^8 。