实验 02：数据类型

一、实验目的

1. 掌握 Java 的基本数据类型和内置数据类型的使用。
2. 熟悉字符串的处理，包括 `String` 类和 `StringBuffer` 类的使用。
3. 理解基本数据类型的包装类及其应用场景。
4. 掌握 `BigInteger` 类的使用，处理大整数运算。
5. 熟悉 Java 中的日期和时间处理类，包括 `Date`、`SimpleDateFormat`、`Calendar` 以及 Java 8 新增的日期和时间类。

二、实验学时

2 学时

三、实验类型

验证性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

IntelliJ IDEA，以及 Java 运行所需要的相关基础环境。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

1. 编写程序，演示基本数据类型的使用。
2. 使用 `String` 类和 `StringBuffer` 类进行字符串操作。
3. 使用基本数据类型的包装类进行数据转换和操作。
4. 使用 `BigInteger` 类进行大整数运算。
5. 使用 `Date`、`SimpleDateFormat`、`Calendar` 以及 Java 8 新增的日期和时间类进行日期和时间的处理。

六、实验内容及步骤

1. 基本数据类型的使用

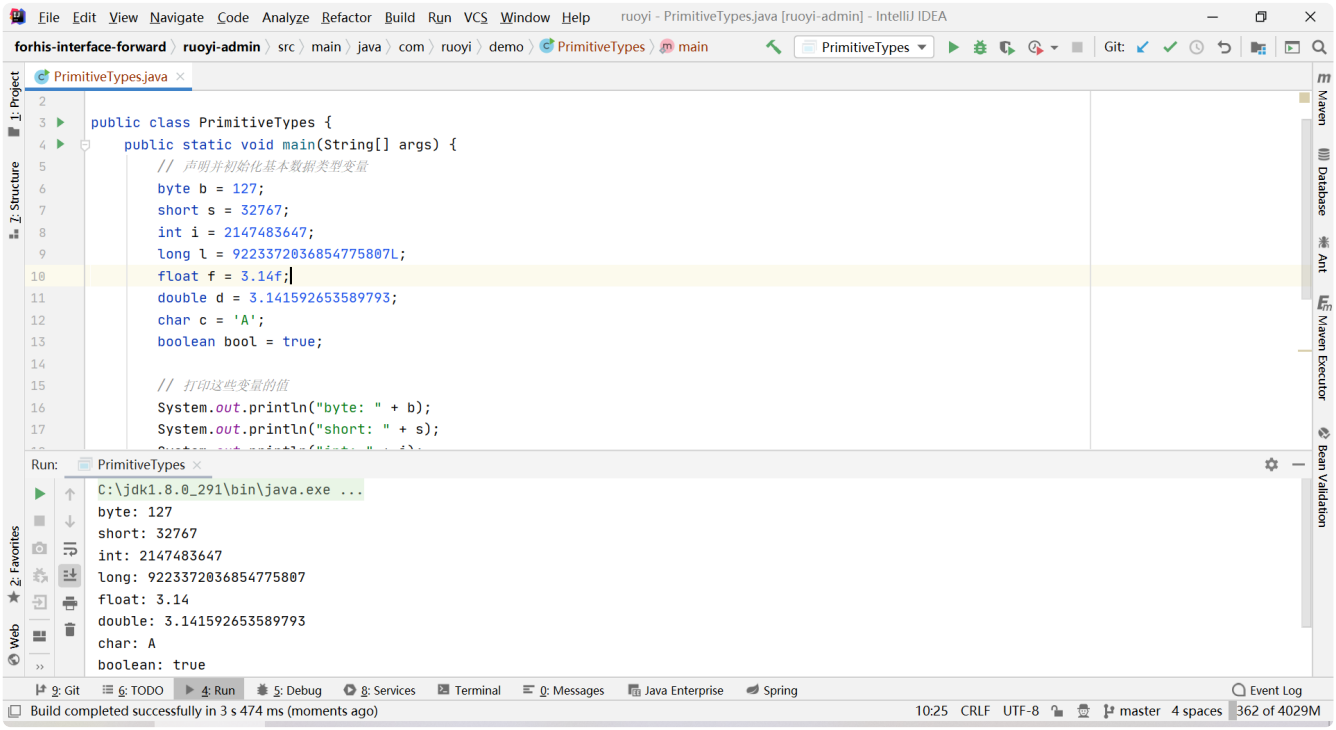
步骤：

- 声明并初始化各种基本数据类型的变量。
- 打印这些变量的值。

示例代码：

Java

```java
public class PrimitiveTypes {
    public static void main(String[] args) {
        // 声明并初始化基本数据类型变量
        byte b = 127;
        short s = 32767;
        int i = 2147483647;
        long l = 9223372036854775807L;
        float f = 3.14f;
        double d = 3.141592653589793;
        char c = 'A';
        boolean bool = true;

        // 打印这些变量的值
        System.out.println("byte: " + b);
        System.out.println("short: " + s);
        System.out.println("int: " + i);
        System.out.println("long: " + l);
        System.out.println("float: " + f);
        System.out.println("double: " + d);
        System.out.println("char: " + c);
        System.out.println("boolean: " + bool);
    }
}
```



2. 字符串处理

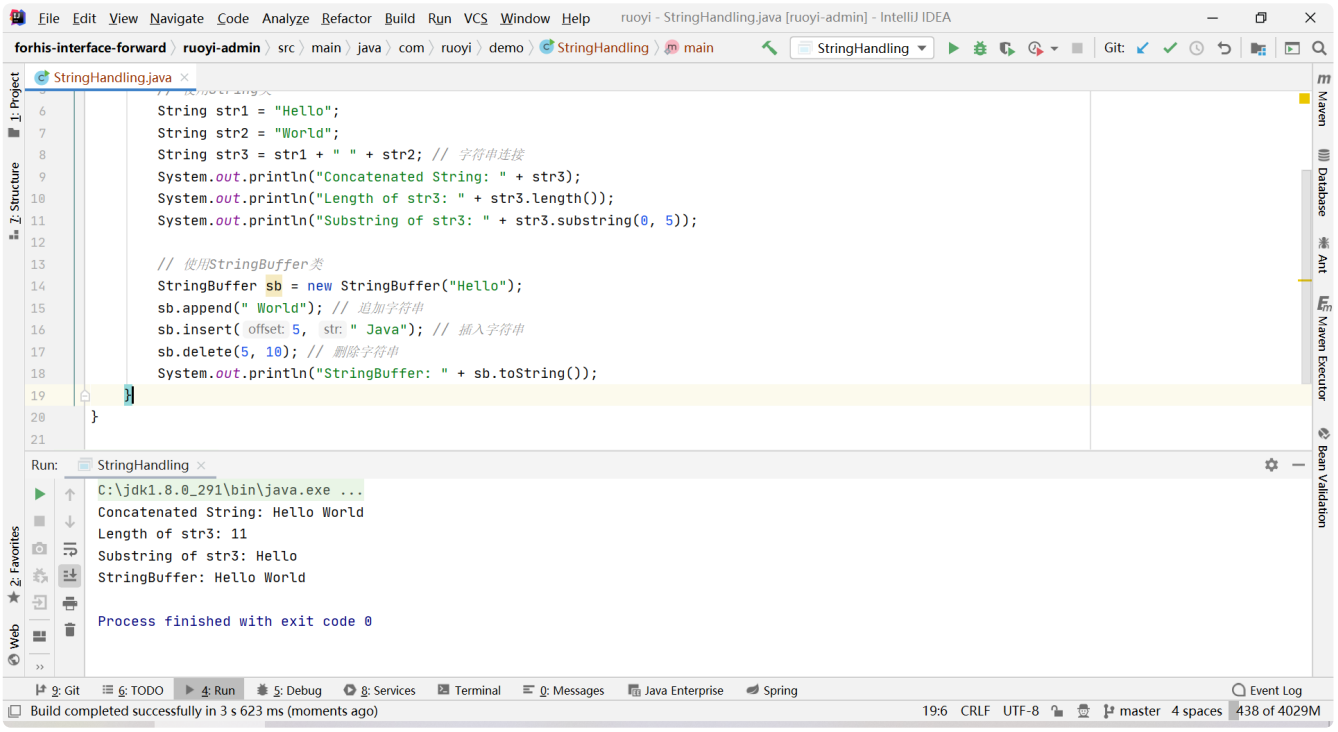
步骤：

- 使用 `String` 类进行字符串的创建、连接、比较等操作。
- 使用 `StringBuffer` 类进行字符串的追加、插入、删除等操作。

示例代码：

Java

```java
public class StringHandling {
    public static void main(String[] args) {
        // 使用String类
        String str1 = "Hello";
        String str2 = "World";
        String str3 = str1 + " " + str2; // 字符串连接
        System.out.println("Concatenated String: " + str3);
        System.out.println("Length of str3: " + str3.length());
        System.out.println("Substring of str3: " + str3.substring(0, 5));

        // 使用StringBuffer类
        StringBuffer sb = new StringBuffer("Hello");
        sb.append(" World"); // 追加字符串
        sb.insert(5, " Java"); // 插入字符串
        sb.delete(5, 10); // 删除字符串
        System.out.println("StringBuffer: " + sb.toString());
    }
}
```



3. 基本数据类型的包装类

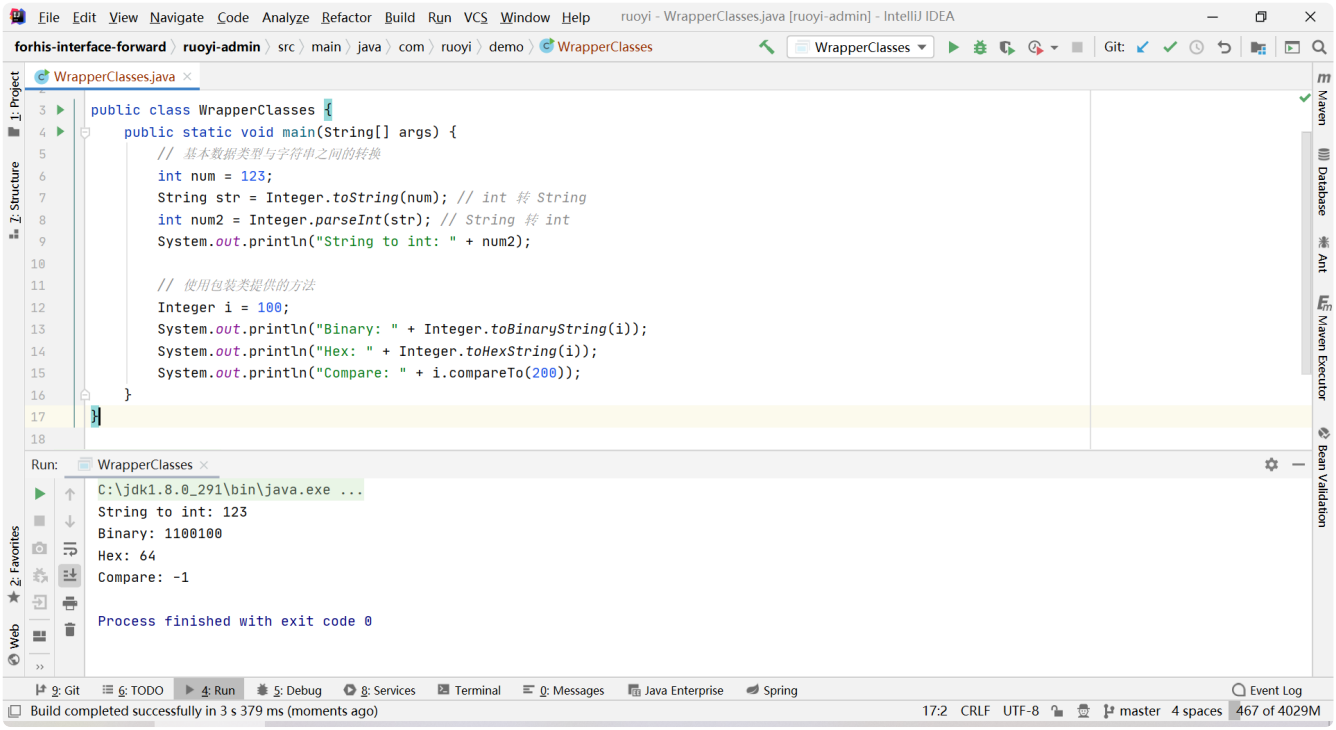
步骤：

- 使用包装类进行基本数据类型与字符串之间的转换。
- 使用包装类提供的方法进行数据操作。

示例代码：

Java

```java
public class WrapperClasses {
    public static void main(String[] args) {
        // 基本数据类型与字符串之间的转换
        int num = 123;
        String str = Integer.toString(num); // int 转 String
        int num2 = Integer.parseInt(str); // String 转 int
        System.out.println("String to int: " + num2);

        // 使用包装类提供的方法
        Integer i = 100;
        System.out.println("Binary: " + Integer.toBinaryString(i));
        System.out.println("Hex: " + Integer.toHexString(i));
        System.out.println("Compare: " + i.compareTo(200));
    }
}
```



4. BigInteger 类的使用

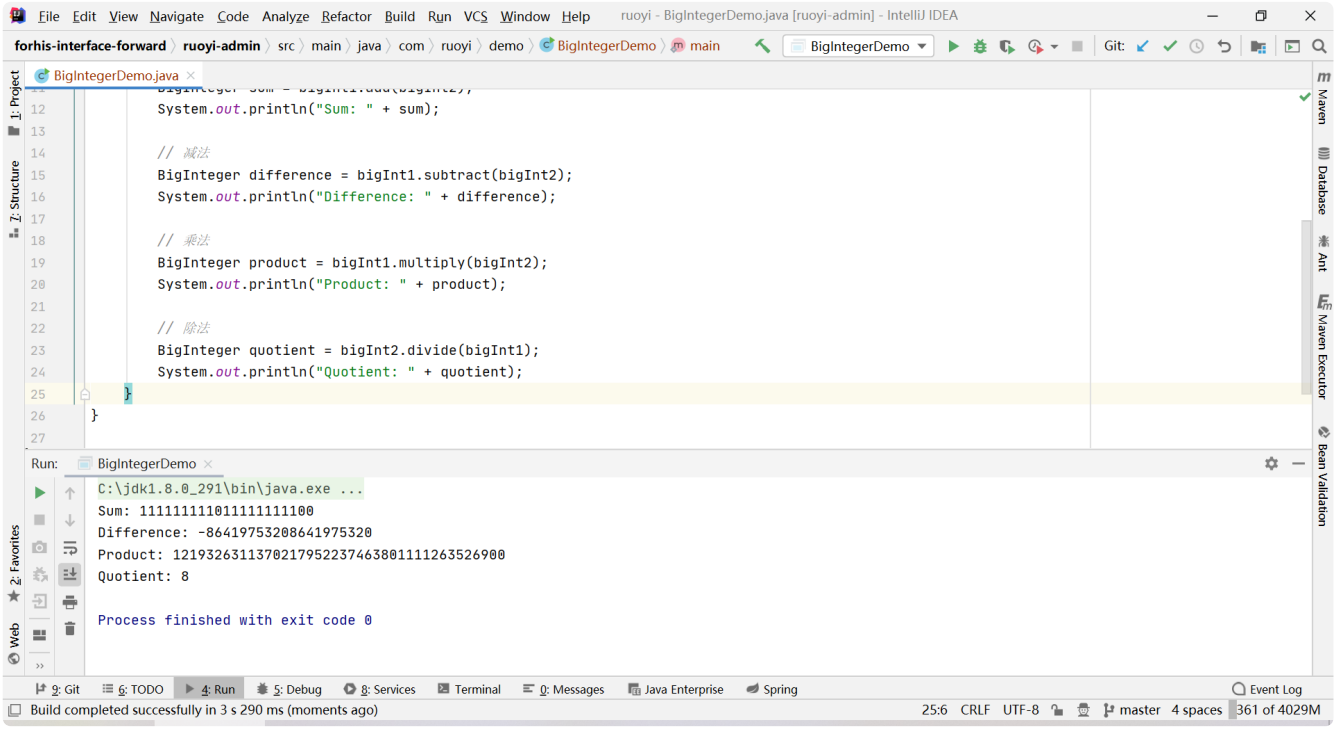
步骤：

- 使用 `BigInteger` 类进行大整数的加减乘除运算。

示例代码：

Java

```java
import java.math.BigInteger;

public class BigIntegerDemo {
    public static void main(String[] args) {
        BigInteger bigInt1 = new BigInteger("12345678901234567890");
        BigInteger bigInt2 = new BigInteger("98765432109876543210");

        // 加法
        BigInteger sum = bigInt1.add(bigInt2);
        System.out.println("Sum: " + sum);

        // 减法
        BigInteger difference = bigInt1.subtract(bigInt2);
        System.out.println("Difference: " + difference);

        // 乘法
        BigInteger product = bigInt1.multiply(bigInt2);
        System.out.println("Product: " + product);

        // 除法
        BigInteger quotient = bigInt2.divide(bigInt1);
        System.out.println("Quotient: " + quotient);
    }
}
```



5. 日期和时间类的使用

步骤：

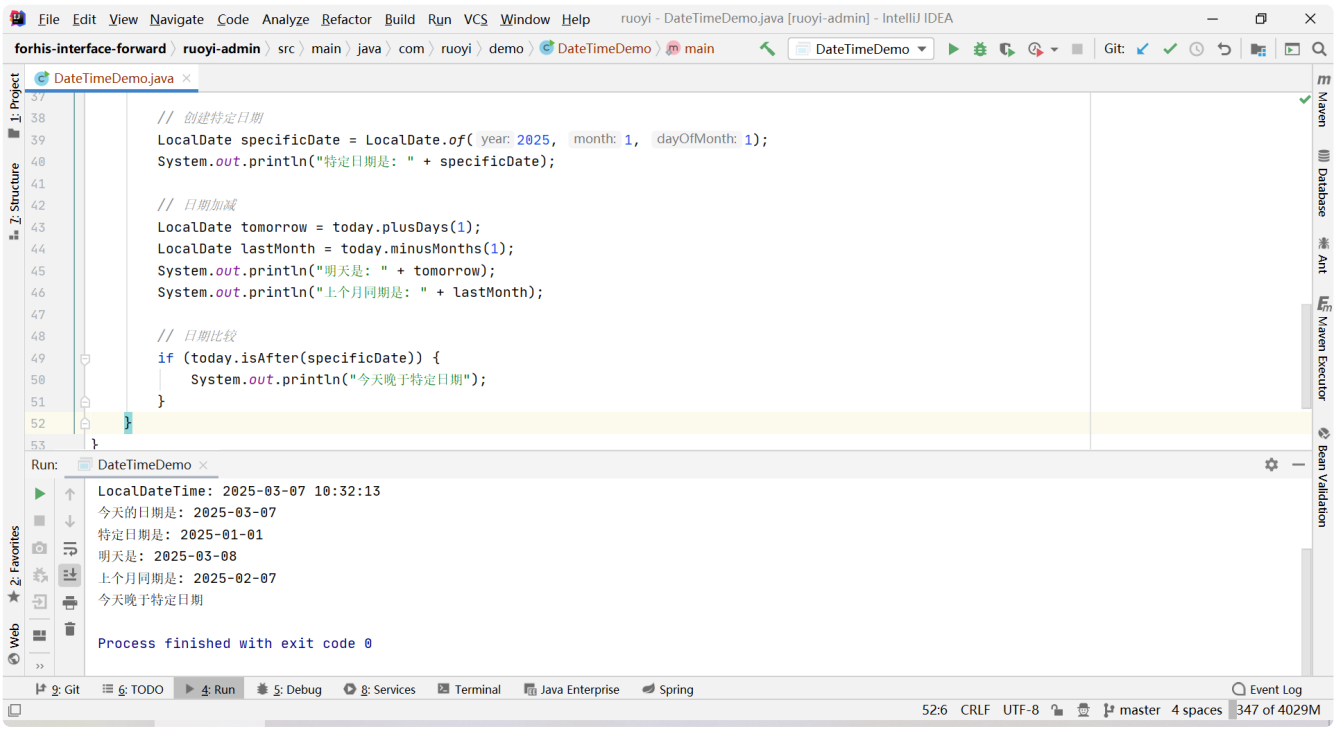
- 使用 `Date` 类和 `SimpleDateFormat` 类进行日期和时间的格式化。
- 使用 `Calendar` 类进行日期和时间的操作。
- 使用 Java 8 新增的日期和时间类（如 `LocalDate`、`LocalTime`、`LocalDateTime`）进行日期和时间的处理。

示例代码：

Java

```java
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.time.LocalDate;
import java.time.LocalTime;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;

public class DateTimeDemo {
    public static void main(String[] args) {
        // 使用Date类和SimpleDateFormat类
        Date date = new Date();
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        System.out.println("Formatted Date: " + sdf.format(date));

        // 使用Calendar类
        Calendar calendar = Calendar.getInstance();
        calendar.setTime(date);
        calendar.add(Calendar.DAY_OF_MONTH, 7); // 增加7天
        System.out.println("Date after 7 days: " + sdf.format(calendar.getTime()));

        // 使用Java 8新增的日期和时间类
        LocalDate localDate = LocalDate.now();
        LocalTime localTime = LocalTime.now();
        LocalDateTime localDateTime = LocalDateTime.now();
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");

        System.out.println("LocalDate: " + localDate);
        System.out.println("LocalTime: " + localTime);
        System.out.println("LocalDateTime: " + dtf.format(localDateTime));

        // 获取当前日期
        LocalDate today = LocalDate.now();
        System.out.println("今天的日期是: " + today);

        // 创建特定日期
        LocalDate specificDate = LocalDate.of(2025, 1, 1);
        System.out.println("特定日期是: " + specificDate);
```

```
39
40          // 日期加减
41          LocalDate tomorrow = today.plusDays(1);
42          LocalDate lastMonth = today.minusMonths(1);
43          System.out.println("明天是: " + tomorrow);
44          System.out.println("上个月同期是: " + lastMonth);
45
46          // 日期比较
47          if (today.isAfter(specificDate)) {
48              System.out.println("今天晚于特定日期");
49          }
50      }
51  }
```



七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

（1）学生演示汇报实验内容的完成情况，实验指导老师现场打分。

（2）指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。

（3）实验考核结束后，进行公布成绩。