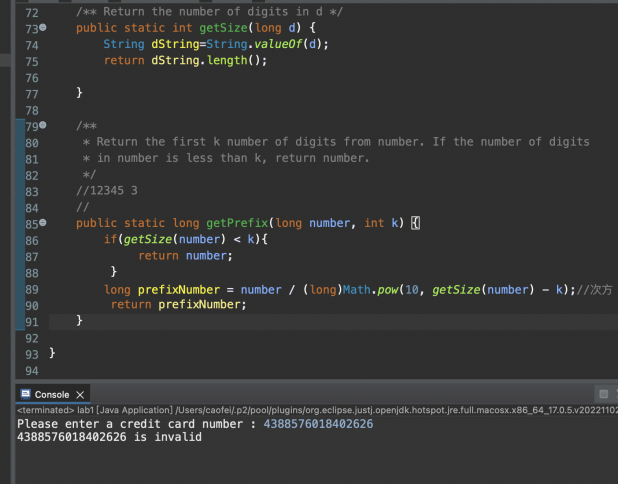
Lab1

Problem description:

Lab1 is to do a method to analyze whether the bank card number is correct or not. The steps are as follows: the first step is to multiply the even-digit number by 2, and if the multiplied by 2 is a two-digit number, add the tens digit and the ones digit, and add Add all the resulting numbers. The second step is to add all the odd digit numbers. The third step is the sum of the data obtained in the first and second steps. The fourth step is to divide the number obtained in the third part by 10. If it is divisible, the bank card number is valid, and if it is not divisible, the bank card number is invalid

Analysis:

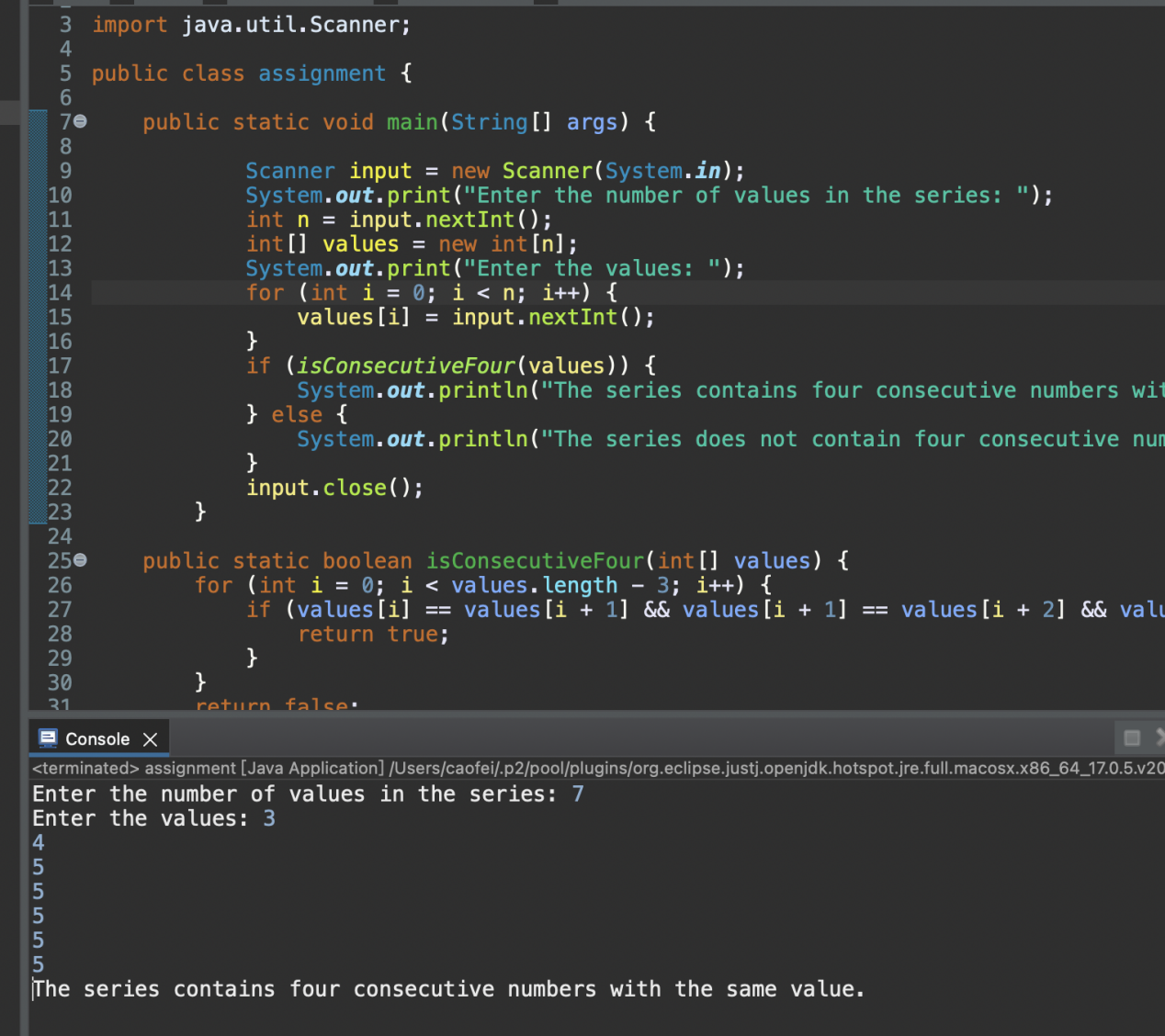
In order to achieve the above steps, we need to make several methods to achieve. The first step is to get the sum of each odd number of bits. The method first converts the long type into a character type, and obtains each odd digit, and uses charAt to locate the specific character position and uses the type to convert it into a number, and accumulates each odd digit. The second step is similar to the first step, but there is one more step to add the tens and ones digit of the two-digit number. So we make a method called getDigit() to determine whether it is greater than 10 after multiplying by 2. Then you need to verify that the card number starts with（• 4 for Visa cards ,• 5 for Master cards ,• 37 for American Express cards ,• 6 for Discover cards ) Therefore, it is necessary to verify whether the first few digits are the required numbers, and it is necessary to use a method to compare the input number with the card number, but because it needs to be compared with the first few digits of the card number, it is necessary to create a new method to obtain the first few digits of the bank card number. The specific number of digits should be the same as the number of digits to be compared, and at the same time, in order to obtain this number, it is necessary to make a method to obtain the length of the input number. So a getsize() method is made to achieve it. Because it needs to be converted to a string type to get the length, type conversion is used. The last method is used to confirm whether the card number is reasonable, and the above methods are summarized and assembled in the method of judging the card number is reasonable. The last is verification, call the method in the main function and introduce the scanner to input the card number, and finally judge whether the card number is reasonable.



Lab2

Problem description and Analysis:

Lab2 needs to input a row of numbers and judge whether there are four consecutive identical numbers in the sequence. Therefore, in main, the length of the input sequence should be judged first, and then the specific values in the input sequence should be traversed, and a method should be constructed to judge whether have four consecutive identical numbers



中文分析：

lab1是要做一个分析银行卡号码是否正确的方法，

1. 偶数位的数字乘2，若乘2后为两位数，将十位数与个位数相加，并将所得到所有的数字相加。
2. 将所有奇数位的数字相加。
3. 将第一部和第二步所得到的数据总和。
4. 将第三部所得到的数字除以10，如果整除则这个银行卡号合法，如果不整除则银行卡号不合法。

为了做到如上的步骤，需要做几个方法来实现第二个方法是获得每个奇数位的总和。首先先将long型转换成字符型，并获得每个奇数位，并用charAt定位到具体的字符位置并用类型转换成数字，将每个奇数位的做累加。第二步和第一步相似，但是多一个将有两位数的数字十位数与个位相加。所以我们重新做一个方法叫做getDigit（）用于判断是否在乘2后大于10。之后则需要验证是否开头是以

• 4 for Visa cards   
• 5 for Master cards   
• 37 for American Express cards   
• 6 for Discover cards

开头的卡号，因此需要验证前几位是否为所需要的数字，因此需要用到方法比较输入的数字与卡号数字对比，但是因为需要与卡号前几位数字做对比，因此新建一个方法来获得银行卡数字的前几位，具体位数应当与对比的数字位数相同，与此同数为了获得这个数字需要引入方法获得所输入的数字长度。因此做了一个getsize方法来实现。因为需要转换为字符串型才可以得到长度所以用到了类型转换。最后一个方法则是用来确认是否卡号合理，将以上的方法汇总并集合在判断卡号合理的方法中。最后是验证，在main函数中调用方法并引入scanner来输入卡号，最终判断是否卡号合理