1. **各个正则表达式的解释。**

2.邮箱字符串正则表达式的理解：

^[A-Za-zd]+([-\_.][A-Za-zd]+)\*@([A-Za-zd]+[-.])+[A-Za-zd]{2,4}$

以字符@为分隔串将正则表达式分为两部分。，^表示以此开头，$表示以这几个单词结尾。 前面一部分代表着用户名，可以是一个或多个单词字符或者是带有连接线-下划线\_或点.的；后面一部分表示邮箱域至少要有一个“.”和两个单词，2指的是最少要带两个单词，4指的是最大可带的单词设为4

3. **正则表达式解释 /\b([a-z]+) \1\b/ig：**

* **捕获的表达式，正如 [a-z]+ 指定的，包括一个或多个字母。**
* **正则表达式的第二部分是对以前捕获的子匹配项的引用，即，单词的第二个匹配项正好由括号表达式匹配。\1 指定第一个子匹配项。**
* **单词边界元字符 \b 确保只检测整个单词。否则，诸如 "is issued" 或 "this is" 之类的词组将不能正确地被此表达式识别。**
* **正则表达式后面的全局标记 g 指定将该表达式应用到输入字符串中能够查找到的尽可能多的匹配。**
* **表达式的结尾处的不区分大小写 i 标记指定不区分大小写。**

**②继承不同方式的理解。**

*·原型链：*

核心：将父类的实例作为子类的原型

优点：父类方法可以复用  
缺点：

·父类的引用属性会被所有子类实例共享

·子类构建实例时不能向父类传递参数

·*构造函数：*

核心：将父类构造函数的内容复制给了子类的构造函数。这是所有继承中唯一一个不涉及到prototype的继承

优点：和原型链继承完全反过来。

* 父类的引用属性不会被共享
* 子类构建实例时可以向父类传递参数

缺点：父类的方法不能复用，子类实例的方法每次都是单独创建的。

*·Object.creat()：*

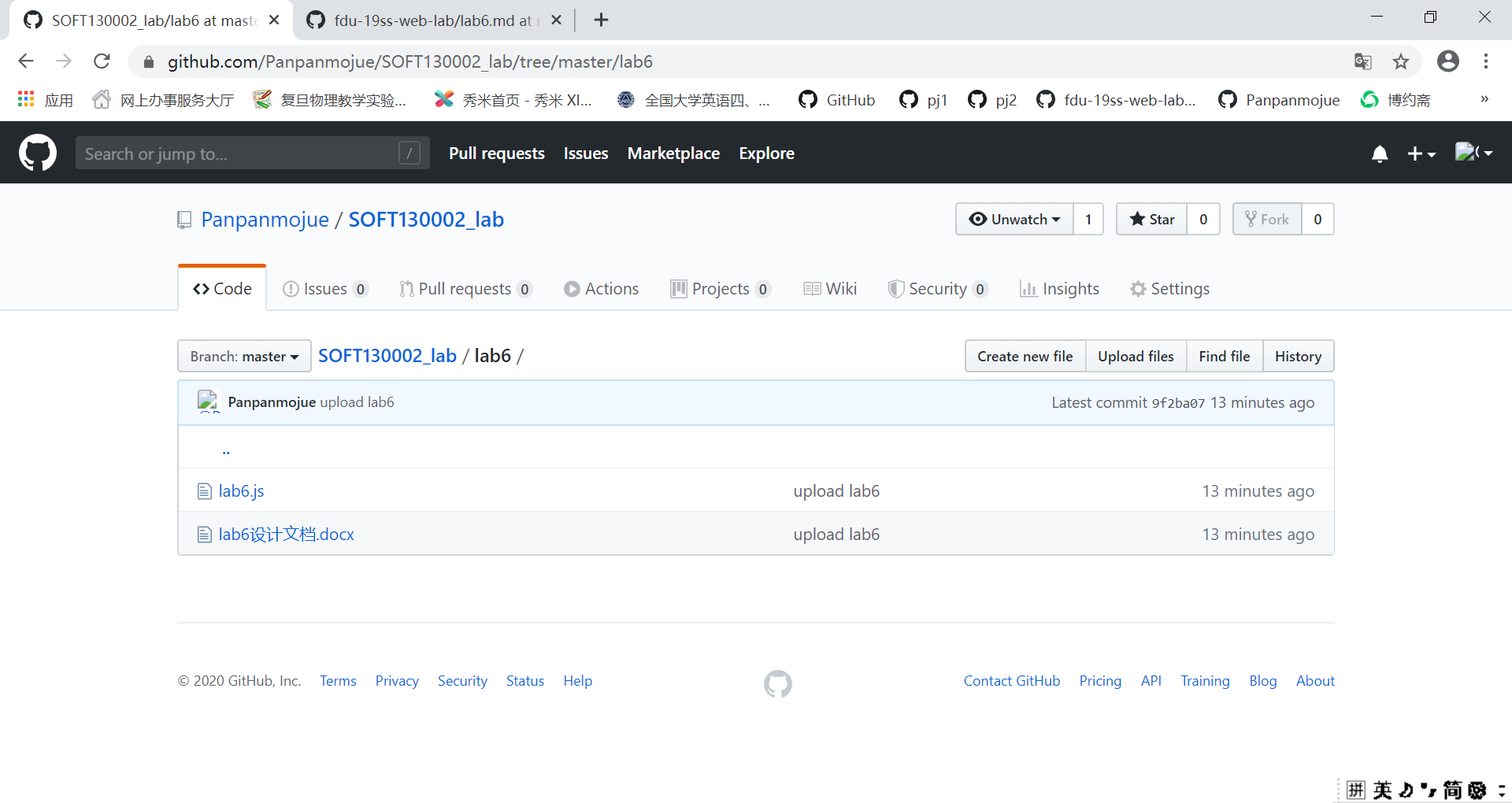
前两者构造出来的都是类，而这个创建出来的是继承父类的一个子类实例。可以给这个子类直接添加自己的属性和方法

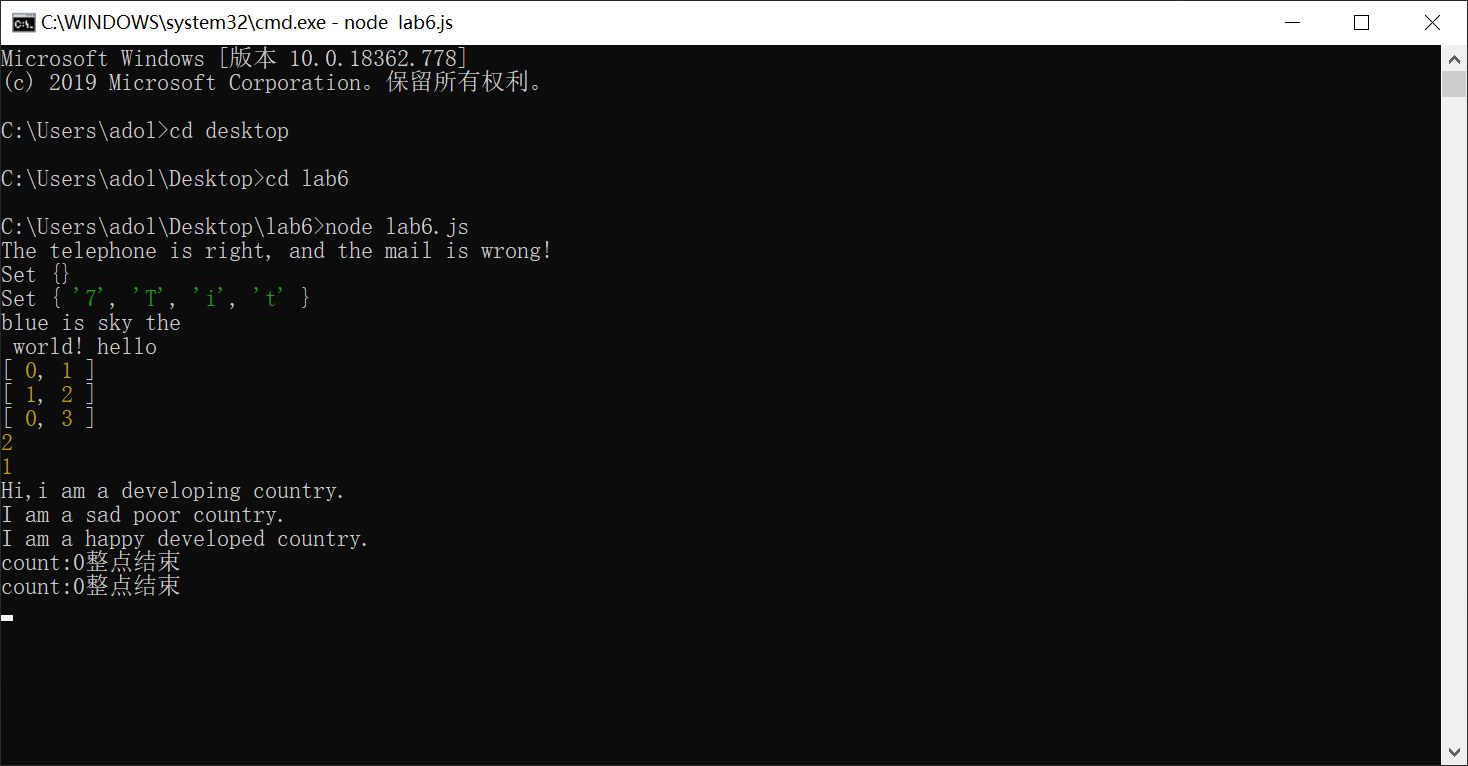
​ **③Map、Set、Array之间的区别和使用**

**·Array:** 数组对象，是使用单独的变量名来存储一系列的值。

**·Set:** 类似于数组，但是成员的值都是唯一的，没有重复的值。Set本身是一个构造函数，用来生成 Set 数据结构。

**·Map:**中的value常用于储存编号，key用来储存我们需要寻找的值，这样便于寻找。Map类似于对象，也是键值对的集合，但是“键”的范围不限于字符串，各种类型的值（包括对象）都可以当作键。Map 结构提供了“值—值”的对应。





.邮箱字符串正则表达式的理解：

^[A-Za-zd]+([-\_.][A-Za-zd]+)\*@([A-Za-zd]+[-.])+[A-Za-zd]{2,4}$

以字符@为分隔串将正则表达式分为两部分。，^表示以此开头，$表示以这几个单词结尾。

前面一部分代表着用户名，可以是一个或多个单词字符或者是带有连接线-下划线\_或点.的；后面一部分表示邮箱域至少要有一个“.”和两个单词，2指的是最少要带两个单词，4指的是最大可带的单词设为4

先给出手机号和邮箱的正则表达式，判断用户输入的手机号码和邮箱是否符合对应的正则表达式，从而完成相应的输出

3. **正则表达式解释 /\b([a-z]+) \1\b/ig：**

* **捕获的表达式，正如 [a-z]+ 指定的，包括一个或多个字母。**
* **正则表达式的第二部分是对以前捕获的子匹配项的引用，即，单词的第二个匹配项正好由括号表达式匹配。\1 指定第一个子匹配项。**
* **单词边界元字符 \b 确保只检测整个单词。否则，诸如 "is issued" 或 "this is" 之类的词组将不能正确地被此表达式识别。**
* **正则表达式后面的全局标记 g 指定将该表达式应用到输入字符串中能够查找到的尽可能多的匹配。**
* **表达式的结尾处的不区分大小写 i 标记指定不区分大小写。**

.首先以空格为分隔符将原字符串拆成不同的部分，并用一个名为array的数组来记录各部分

用一个名为check的数组来记录相邻两个单词，利用正则表达式进行验证，如果相同，则将其作为一个元素计入result数组中，并记一次数。

若最终计数不超过十个，即result数组中的元素个数不超过10则直接输出；如果超过十个，就对result中的元素按照首字母进行分类，按首字母的顺序取前十个于集合。

4.（Set可以自动删去重复的内容）

把wantInput里有actualInput 里的elem删掉 剩下的就是坏了的键

5.首先以空格为分隔符将原字符串拆成不同的部分，并用一个名为array的数组来记录各部分

# 再借助数组的reverse方法翻转数组顺序

然后进行一个循环，依次检查array中的元素是否为空格，如果不是空格，则将其赋值给result数组 最后把数组输出转化为字符串

6.首先将数组中的元素的大小、元素所在的位置存于Map中 然后一个个地判断，如果map里面有target-num[i]的元素 则说明存在符合标准的数组 将其打印出来 利用循环 找出所有满足要求的数组

7.首先用一个for循环对str中的字符元素一个一个地进行遍历 map最初为空 若出现了之前没有出现过的元素 即map.has(str[i])为false 就记一次数 并创立一个key为str[i]的map 这样下次出现相同的元素时 map.has(str[i])为true，就不计数 最后输出计数个数即可