知识导入基础编程题

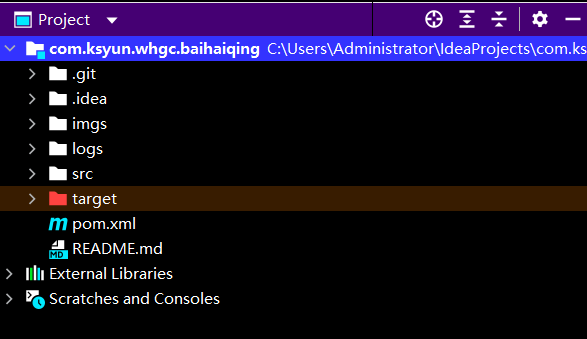
**工程形式：**创建maven工程，源码顶级包路径为com.ksyun.whgc.姓名汉语全拼小写，如：张三同学，源码顶级包路径应为com.ksyun.whgc.zhangsan

**提交形式：**创建git本地代码库，将本地代码库压缩打包提交（包含.git目录）；

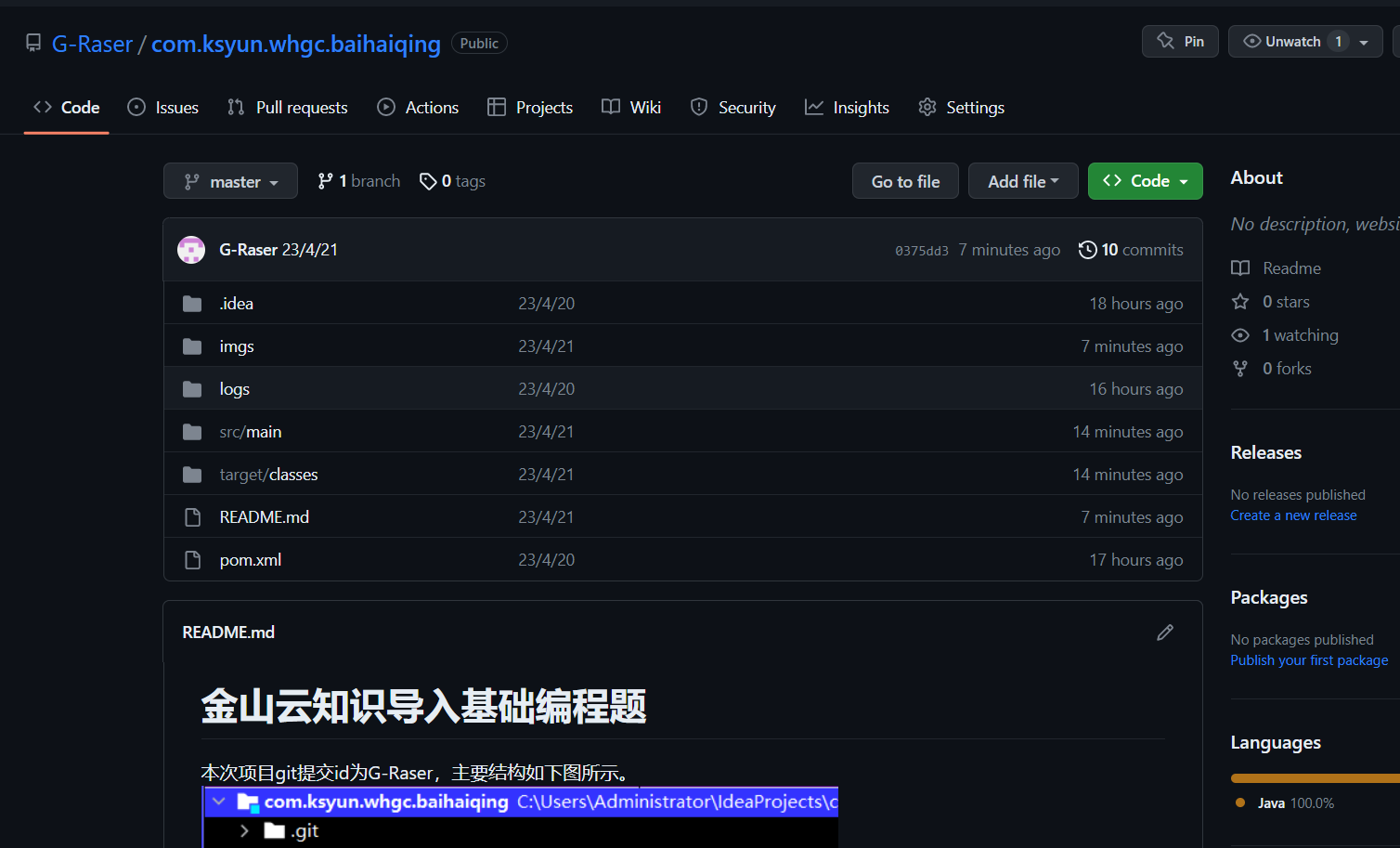
[**README.md**](https://gitee.com/zhenrong-wang/hpc-now/blob/master/README.md)**:**工程说明文件，置于工程根目录，详细说明每题的实现路径、测试方法路径、git提交commit id等；

**评分细则**：每个功能点（方法得分项）需有至少一次git提交记录，缺少一次提交记录扣2分；功能实现错误不得分；无测试方法不得分；编码规范问题一处扣2分；

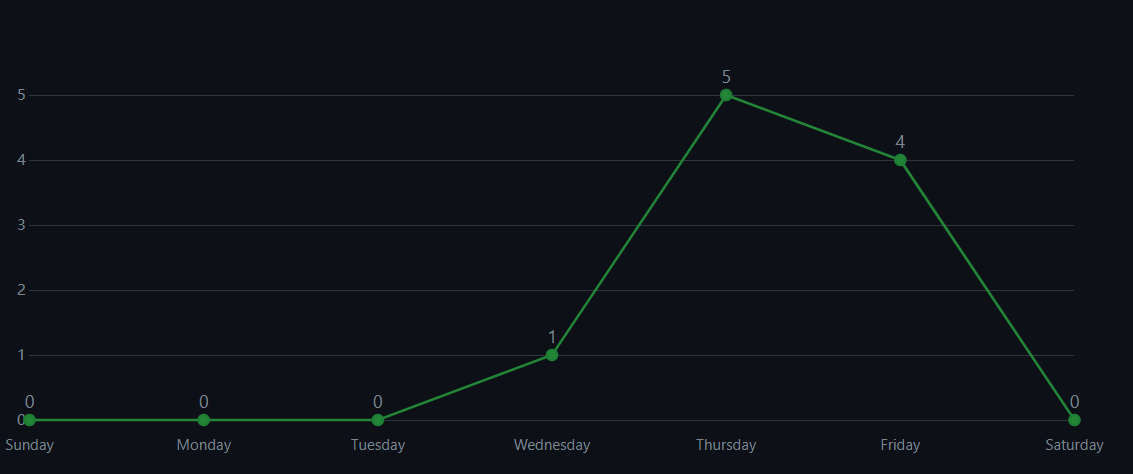
本项目总结构如下图所示。



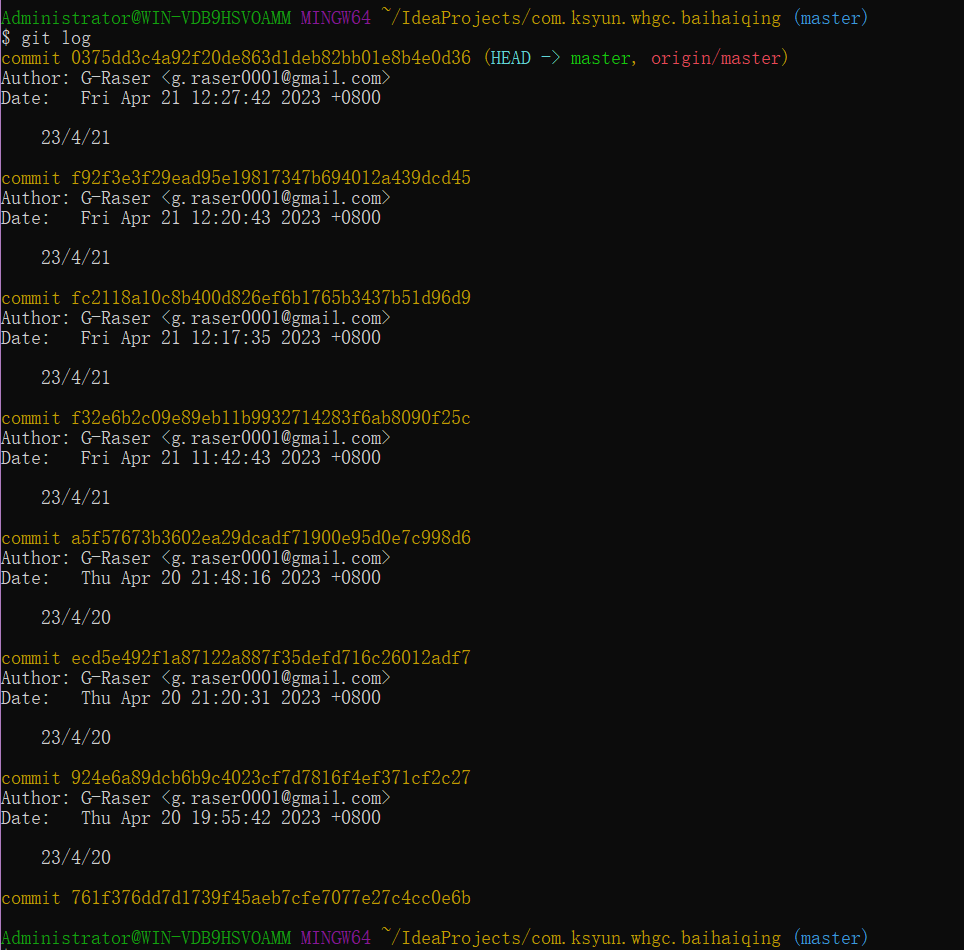
Github项目地址为：https://github.com/G-Raser/com.ksyun.whgc.baihaiqing，页面如下所示。



提交记录如下图所示。



git提交commit id如下所示。



1.编写字符串工具类实现下如下四个方法，方法接收字符串参数并实现如下目标：  
 1.1反转字符串参数，并作为方法结果返回(10分)  
 1.2统计字符串中每个字母出现的次数，以Map结构返回（10分）  
 1.3打印出现次数最多的字母及其次数（10分）  
 1.4 返回不含有重复字符的最长子串（20分）

MyStringUtils.java

package Question1**;**import java.util.ArrayList**;**import java.util.HashMap**;**import java.util.List**;**import java.util.Map**;***/\*\*  
 \** ***@author*** *Haiqing Bai  
 \** ***@date*** *2023/4/19  
 \*/*public class MyStringUtils {  
 */\*\*  
 \* 首字母变小写  
 \** ***@param*** *str  
 \** ***@return*** *\*/* public static String reverseStr(String str) {  
 char[] chars = str.toCharArray()**;** String reverse = ""**;** for (int i = chars.length - **1;** i >= **0;** i--) {  
 reverse += chars[i]**;** }  
 return reverse**;** }  
 public static HashMap<Character**,** Integer> countNum(String str){  
 HashMap<Character**,** Integer> map = new HashMap<>()**;** for (char c : str.toCharArray()) {  
 //将此字符串转换为一个新的字符数组。  
 //4.使用获取到的字符，去Map集合判断key是否存在  
 if (map.containsKey(c)) {  
 //key存在  
 Integer value = map.get(c)**;** value++**;** map.put(c**,** value)**;** } else {  
 //key不存在  
 map.put(c**, 1**)**;** }  
 }  
 return map**;** }  
 public static void printMost(String str){  
 HashMap<Character**,** Integer> myMap=*countNum*(str)**;** Integer maxValue = **0;** Character markedChar = '?'**;** for (Character key : myMap.keySet()) {  
 if(myMap.get(key)>maxValue){  
 maxValue=myMap.get(key)**;** markedChar=key**;** }  
 }  
 System.*out*.println(markedChar+' '+maxValue)**;** }  
  
 public static String getLonger(String target) {  
 String longerStr = ""**;** int maxLonger = **0;** if(target == null || target.length() == **0**) {  
  
 return"illegal"**;** }  
 // 结束递归  
 boolean out = false**;** StringBuffer sb = new StringBuffer()**;** for(int i=**0;** i<target.length()**;**i++) {  
 String chat = target.charAt(i) + ""**;** if(!sb.toString().contains(chat)) {  
 sb.append(chat)**;** if(i == target.length() -**1**) {  
 //整个字符串是否完全遍历  
 longerStr = sb.toString()**;** maxLonger = sb.length()**;** out = true**;** }  
 } else {  
 if(maxLonger < sb.length() ) {  
 longerStr = sb.toString()**;** maxLonger = sb.length()**;** }  
 // 每次递归去掉首字母  
 target = target.substring(**1**)**;** // 不可能获得更长的子字符串了  
 if(maxLonger == target.length()) {  
 out = true**;** }  
 break**;** }  
 }  
 if(!out) {  
 *getLonger*(target)**;** }  
 return longerStr**;** }  
}

StringUtilsDemo.java

package Question1**;**import java.util.Map**;**import java.util.Scanner**;**import static Question1.MyStringUtils.\***;***/\*\*  
 \** ***@author*** *Haiqing Bai  
 \** ***@date*** *2023/4/20  
 \*/*public class StringUtilsDemo {  
 public static void main(String[] args) {  
// Scanner sc = new Scanner(System.in);  
// System.out.println("请输入一个字符串：");  
// String str = sc.next();  
 String str= "abcdafgc"**;** String str1=*reverseStr*(str)**;** System.*out*.println("原字符串："+str)**;** System.*out*.println("反转测试结果："+str1)**;** Map<Character**,**Integer> mapResult=*countNum*(str)**;** //展示结果map  
 System.*out*.println("字母出现次数统计map展示：")**;** for (Character key : mapResult.keySet()) {  
 Integer value = mapResult.get(key)**;** System.*out*.println(key + "=" + value)**;** }  
 System.*out*.println("出现次数最多字母展示：")**;** *printMost*(str)**;** System.*out*.println("最长非重复字母子串展示：")**;** System.*out*.println(*getLonger*(str))**;** }  
}

结果：



1. 创建学生对象，包括姓名、年龄、住址、出生年月等属性；模拟生成学生集合，基于字段注解、打印输出该集合，要求输出的第一行数据为字段名称、后续行数据为集合数据；不限输出格式，可以是excel、htm、txt等；  
    2.1.定义注解类：用于标注集合对象属性；（10分）  
    2.2.定义工具类输出集合对象:集合输出时根据属性的标注输出属性字段名;(20分)

PrintInfoDemo.java

package Question2**;**import java.util.HashSet**;**import java.util.Set**;**import java.util.Iterator**;**import static Question2.Set2Excel.*toExcel***;***/\*\*  
 \** ***@author*** *Haiqing Bai  
 \** ***@date*** *2023/4/19  
 \*/*public class PrintInfoDemo {  
 public static void main(String[] args) throws Exception {  
 Set<Object> set = new HashSet<Object>()**;** set.add(new Student("123"**,**"China"**,**"张三"**,**"女"**,**"2006-03-09"))**;** set.add(new Student("111"**,**"USA"**,**"罗斯"**,**"男"**,**"2003-01-19"))**;**// Iterator<Object> it = set.iterator();  
// while (it.hasNext()) {  
// Student myStudent = (Student) it.next();  
// System.out.println(myStudent.getName());  
// }  
 String filePath="C:\\Users\\Administrator\\Desktop\\Test1.xlsx"**;** *toExcel*(set**,**filePath)**;** }  
}

Set2Excel.java

package Question2**;**import com.alibaba.excel.EasyExcel**;**import com.alibaba.excel.annotation.ExcelProperty**;**import java.lang.reflect.Field**;**import java.util.ArrayList**;**import java.util.List**;**import java.util.Set**;**import java.lang.reflect.InvocationHandler**;**import java.lang.reflect.Proxy**;**import java.util.Map**;***/\*\*  
 \** ***@author*** *Haiqing Bai  
 \** ***@date*** *2023/4/19  
 \*/*class Set2Excel {  
 public static void toExcel(Set<Object> mySet**,**String filePath){  
 List<Object> list1 = new ArrayList<Object>(mySet)**;** System.*out*.println("文件输出路径：")**;** System.*out*.println(filePath)**;** System.*out*.println("当前集合对应列表：")**;** System.*out*.println(list1)**;** Class studentClass=Student.class**;** Field[] fields = studentClass.getDeclaredFields()**;**//// List<String> heads=new ArrayList<>();  
// for(Field field : fields){  
// System.out.println(field.getName());  
//// heads.add(field.getName());  
// }  
//// list1.add(0,heads);  
 EasyExcel.*write*(filePath**,** studentClass).sheet("学生表").doWrite(list1)**;** System.*out*.println("集合数据输出Excel文件完毕！")**;** }  
}

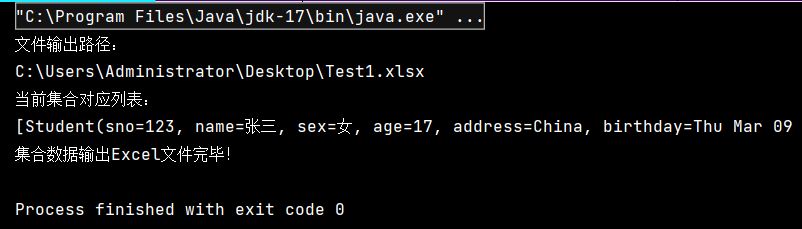
Student.java

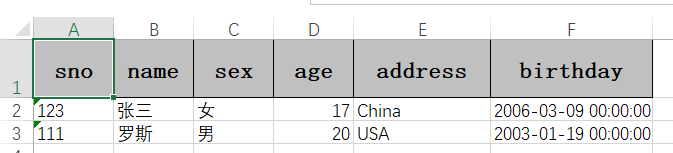
package Question2**;**import com.alibaba.excel.EasyExcel**;**import com.alibaba.excel.ExcelWriter**;**import com.alibaba.excel.annotation.ExcelIgnore**;**import com.alibaba.excel.annotation.ExcelProperty**;**import com.alibaba.excel.write.metadata.WriteSheet**;**import lombok.Data**;**import java.util.Calendar**;**import java.text.SimpleDateFormat**;**import java.util.Date**;**import java.text.ParseException**;***/\*\*  
 \** ***@author*** *Haiqing Bai  
 \** ***@Date*** *2023/4/19  
 \*/*////学生类  
@Data  
class Student  
{  
 @StudentAnno(sno="000"**,**name = "Anomy"**,**age = **0**)  
// @ExcelProperty(value = "学号",index = 0)  
 @ExcelProperty(index = **0**)  
 private String sno**;**// @ExcelProperty(value = "{姓名",index = 1)  
 @ExcelProperty(index = **1**)  
 private String name**;**// @ExcelProperty(value = "性别",index = 2)  
 @ExcelProperty(index = **2**)  
 private String sex**;**// @ExcelProperty(value = "年龄",index = 3)  
 @ExcelProperty(index = **3**)  
 private int age**;**// @ExcelProperty(value = "地址",index = 4)  
 @ExcelProperty(index = **4**)  
 private String address**;**// @ExcelProperty(value = "生日",index = 5)  
 @ExcelProperty(index = **5**)  
 private Date birthday**;** public String getSno() {  
 return sno**;** }  
  
 public String getName() {  
 return name**;** }  
  
 public String getSex() {  
 return sex**;** }  
  
 public int getAge() {  
 return age**;** }  
  
 public String getAddress() {  
 return address**;** }  
  
 public Date getBirthday() {  
 return birthday**;** }  
 public void setSno(String sno) {  
 this.sno = sno**;** }  
  
 public void setName(String name) {  
 this.name = name**;** }  
  
 public void setSex(String sex) {  
 this.sex = sex**;** }  
  
 public void setAddress(String address) {  
 this.address = address**;** }  
  
 public void setBirthday(Date birthday) {  
 this.birthday = birthday**;** }  
 public Student()  
 {  
 }  
 public Student(String sno**,**String address**,**String name**,**String sex**,** String bd) {  
 Calendar now = Calendar.*getInstance*()**;** try {  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd")**;** Date bdd = null**;** bdd = sdf.parse(bd)**;** this.birthday=bdd**;** } catch (ParseException e) {  
 e.printStackTrace()**;** }  
 this.age = now.get(Calendar.*YEAR*) - this.birthday.getYear()-**1900;** this.sno = sno**;** this.address = address**;** this.name = name**;** this.sex = sex**;** }  
}

StudentAnno.java

package Question2**;**import java.lang.annotation.ElementType**;**import java.lang.annotation.Retention**;**import java.lang.annotation.RetentionPolicy**;**import java.lang.annotation.Target**;***/\*\*  
 \** ***@author*** *Haiqing Bai  
 \** ***@date*** *2023/4/19  
 \*/*@Target({ElementType.*FIELD***,**ElementType.*METHOD*})//作用目标，字段,方法  
@Retention(RetentionPolicy.*RUNTIME*)//作用范围，整个运行期间  
public @interface StudentAnno {  
 */\*\*  
 \* Returns an array of the kinds of elements an annotation type  
 \* can be applied to.  
 \** ***@return*** *an array of the kinds of elements an annotation type  
 \* can be applied to  
 \*/* String sno() default "000"**;** String address() default "earth"**;** String name() default "Anomy"**;** int age() default **0;** String sex() default "?"**;**}

结果：





1. 编写一个程序，启动三个线程，三个线程的ID分别是A，B，C；，每个线程将自己的ID值在屏幕上打印5遍，打印顺序是ABCABC… （20分）

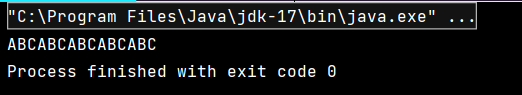
MyThread.java

package Question3**;**public class MyThread {  
 int count = **1;** public synchronized void printA() { //A线程  
 while(count<=**0**) {  
 try {  
 this.wait()**;** } catch (InterruptedException e) {  
 e.printStackTrace()**;** }  
 }  
 System.*out*.print("A")**;** count--**;** this.notifyAll()**;** }  
 public synchronized void printB() { //B线程  
 while(count!=**0**) {  
 try {  
 this.wait()**;** } catch (InterruptedException e) {  
 e.printStackTrace()**;** }  
 }  
 System.*out*.print("B")**;** count--**;** this.notifyAll()**;** }  
 public synchronized void printC() { //C线程  
 while(count>=**0**) {  
 try {  
 this.wait()**;** } catch (InterruptedException e) {  
 e.printStackTrace()**;** }  
 }  
 System.*out*.print("C")**;** count=-count**;** this.notifyAll()**;** }  
}

ThreadDemo.java

package Question3**;***/\*\*  
 \** ***@author*** *Administrator  
 \*/*public class ThreadDemo  
{  
 public static final int *PRINTSIZE* = **5;** public static void main(String[] args) {  
 MyThread thread = new MyThread()**;** new Thread(new Runnable() {  
 @Override  
 public void run() {  
 for(int i=**0;** i<PRINTSIZE**;** i++) {  
 thread.printA()**;** }  
 }  
 }).start()**;** new Thread() {  
 @Override  
 public void run() {  
 for(int i=**0;** i<PRINTSIZE**;** i++) {  
 thread.printB()**;** }  
 }  
 }.start()**;** new Thread(new Runnable() {  
 @Override  
 public void run() {  
 for(int i=**0;** i<PRINTSIZE**;** i++) {  
 thread.printC()**;** }  
 }  
 }).start()**;** }  
}

结果：



备注：git commit id查看

