* 导入一些需要的模块，如pymysql, pyhive, pandas, numpy等，这些模块是用于连接数据库，处理数据，进行数学计算的。
* 定义一些全局变量，如HIVE\_HOST, MYSQL\_HOST, MYSQL\_USER等，这些变量是用于配置数据库的连接信息的。
* 定义一个类Struct\_data，这个类是用于对文本数据进行后结构化处理的，即将文本数据中的信息提取出来，转换为结构化的数据。这个类的主要方法有：
  + \_\_init\_\_方法，这个方法是用于初始化类的实例的，传入一些参数，如data, data\_process\_dict, extract\_rule\_dict, computer\_rule\_dict等，这些参数是用于对文本进行预处理，提取，计算的规则字典。
  + data\_process方法，这个方法是用于对文本数据进行预处理的，即将文本数据中的一些特殊字符，空格，换行等去除或替换，使文本数据更规范，更易于处理。
  + extract\_data方法，这个方法是用于对文本数据进行提取的，即根据提取规则字典中的正则表达式，从文本数据中匹配出一些信息，如阻断方式，阻断时间，病理类型，肿瘤大小等，将这些信息存储在一个字典中，作为提取的结果。
  + computer\_data方法，这个方法是用于对提取出来的数据进行计算的，即根据计算规则字典中的函数，对提取出来的数据进行一些数学运算，如求和，平均，最大值，最小值等，将这些运算的结果存储在一个字典中，作为计算的结果。
  + main方法，这个方法是用于调用其他的方法，得到最终的后结构化结果的，即先对文本数据进行预处理，然后进行提取，然后进行计算，最后将提取和计算的结果合并成一个字典，作为最终的结果。
* 定义一个函数struct\_data，这个函数是用于对手术描述和检查结论字段进行后结构化处理的，即将这两个字段中的文本数据提取出来，转换为结构化的数据，方便后续的分析和处理。这个函数的主要逻辑有：
  + 首先，将res\_string列表转换为列表的列表，方便修改元素。
  + 然后，定义两个空列表，分别用于存储手术描述和检查结论字段的数据。
  + 接着，遍历res\_string中的每个元素，即每条手术病理信息的记录。
  + 对于每个元素，分别提取出手术描述和检查结论字段的值，如果为空，就用空字符串代替，然后分别添加到对应的列表中。
  + 然后，调用Struct\_data这个类，传入手术描述和检查结论的列表，以及一些其他的参数，如data\_process\_dict, extract\_rule\_dict\_operate, computer\_rule\_dict等，这些参数是用于对文本进行预处理，提取，计算的规则字典。
  + 接着，调用Struct\_data类的main方法，得到两个结果列表，分别表示手术描述和检查结论的后结构化结果，每个元素是一个字典，包含result和error两个键，result的值是一个字典，表示提取出来的信息，error的值是一个列表，表示提取过程中的错误信息。
  + 然后，遍历两个结果列表，将提取出来的信息插入到原来的记录中，形成新的字段，如阻断方式，阻断时间，病理类型，肿瘤大小等，如果没有提取出来的信息，就用None代替。
  + 在插入值的过程中，还需要对一些特殊的情况进行处理，如：
    - 如果阻断时间的值包含时和分，就需要将其转换为分钟数，如1小时30分钟，就转换为90分钟。
    - 如果阻断时间的值包含分和秒，就需要将其转换为分钟数，并保留两位小数，如2分钟30秒，就转换为2.5分钟。
    - 如果阻断时间的值是一个数字，就直接使用它，如3.5，就表示3.5分钟。
    - 如果阻断时间的值不是以上任何一种情况，就不做任何处理，如未知，就保留为未知。
    - 如果肿瘤大小的值是一个范围，就需要取出最大值，如1-2，就取2。
    - 如果肿瘤大小的值是一个数字，就直接使用它，如3，就表示3厘米。
    - 如果肿瘤大小的值不是以上任何一种情况，就不做任何处理，如未见，就保留为未见。
  + 最后，将res\_string列表转换回列表的元组，作为返回值。
* 定义一个函数get\_data\_from\_hive，这个函数是用于从Hive数据库中读取手术病理信息的数据的，即根据指定的SQL语句，从Hive数据库中查询出需要的数据，然后将数据转换为列表的元组的格式，作为返回值。这个函数的主要逻辑有：
  + 首先，使用pyhive模块中的connect方法，根据HIVE\_HOST, HIVE\_PORT, HIVE\_USER等全局变量，创建一个连接Hive数据库的对象。
  + 然后，使用连接对象的cursor方法，创建一个执行SQL语句的游标对象。
  + 接着，使用游标对象的execute方法，执行传入的SQL语句，从Hive数据库中查询出需要的数据。
  + 然后，使用游标对象的fetchall方法，将查询出的数据全部取出，存储在一个列表中，每个元素是一个元组，表示一条记录。
  + 最后，返回这个列表，作为返回值。
* 定义一个函数write\_data\_to\_mysql，这个函数是用于将数据写入MySQL数据库中的，即根据指定的表名，字段名，数据，将数据批量插入到MySQL数据库中的对应的表中。这个函数的主要逻辑有：
  + 首先，使用pymysql模块中的connect方法，根据MYSQL\_HOST, MYSQL\_USER, MYSQL\_PASSWORD, MYSQL\_DB等全局变量，创建一个连接MySQL数据库的对象。
  + 然后，使用连接对象的cursor方法，创建一个执行SQL语句的游标对象。
  + 接着，根据传入的表名，字段名，数据，构造一个插入数据的SQL语句，使用游标对象的executemany方法，批量执行这个SQL语句，将数据插入到MySQL数据库中的对应的表中。
  + 然后，使用连接对象的commit方法，提交事务，使数据的插入生效。
  + 最后，使用游标对象的close方法，关闭游标，使用连接对象的close方法，关闭连接。
* 定义一个主函数main，这个函数是用于调用其他的函数，完成整个程序的流程的
* 首先，导入了一些需要的模块，如pymysql, pyhive, pandas等，用于连接数据库，处理数据，打印日志等。
* 然后，定义了一些全局变量，如Hive的连接信息，MySQL的连接信息，表名，字段名，日志文件名等，用于配置的代码的参数。
* 接着，定义了一个类，叫做Struct\_data，用于对手术描述和检查结论字段进行后结构化处理，即将这两个字段中的文本信息提取出来，转换为结构化的数据，方便后续的分析和处理。这个类的主要方法如下：
  + init: 构造方法，用于初始化类的属性，如data, data\_process\_dict, extract\_rule\_dict, computer\_rule\_dict等，这些属性是用于对文本进行预处理，提取，计算的规则字典。
  + data\_process: 数据预处理方法，用于对文本进行一些清洗和规范化，如去除空格，换行，标点符号等，返回处理后的文本。
  + extract: 数据提取方法，用于对文本进行正则匹配，提取出感兴趣的信息，如阻断方式，阻断时间，病理类型，肿瘤大小等，返回提取出的信息和错误信息。
  + computer: 数据计算方法，用于对提取出的信息进行一些计算，如将阻断时间转换为分钟数，将肿瘤大小转换为厘米数等，返回计算后的信息和错误信息。
  + main: 主方法，用于调用上述的方法，对每条文本进行后结构化处理，返回处理后的结果列表，每个元素是一个字典，包含result和error两个键，result的值是一个字典，表示提取出来的信息，error的值是一个列表，表示提取过程中的错误信息。
* 然后，定义了一个函数，叫做struct\_data，用于调用Struct\_data这个类，对手术描述和检查结论字段进行后结构化处理，然后将处理后的结果插入到原来的记录中，形成新的字段，如阻断方式，阻断时间，病理类型，肿瘤大小等，如果没有提取出来的信息，就用None代替。这个函数的参数和返回值如下：
  + res\_string: 一个列表，每个元素是一个元组，表示一条手术病理信息的记录，包含多个字段，如患者编号，手术日期，手术描述，检查结论等。
  + return: 一个列表，每个元素是一个元组，表示经过处理后的手术病理信息的记录，其中手术描述和检查结论字段被拆分为多个字段，分别表示不同的信息，如阻断方式，阻断时间，病理类型，肿瘤大小等。
* 然后，定义了一个函数，叫做hive\_2\_mysql，用于从Hive数据库中读取手术病理信息的数据，调用struct\_data函数进行后结构化处理，然后写入到MySQL数据库中。这个函数的参数和返回值如下：
  + table\_name: 一个字符串，表示要读取的Hive表的名称，如form\_operate\_pathology\_info。
  + return: 一个整数，表示写入到MySQL数据库中的记录的数量，如果发生异常，就返回-1。
* 最后，定义了一个主函数，用于调用hive\_2\_mysql函数，传入表名，打印日志，显示结果。