

6.4总结：连接MySQL、三层架构

time & datetime 库

Python 中处理时间的标准库

提供获取系统时间并格式化输出功能

提供系统级精确计时功能，用于程序性能分析

time库包含三类函数

时间获取：localtime()

时间格式化：strftime() strptime()

程序计时：sleep() perf_counter()

```
import time
# 日期类型与字符串直接互相转化
# 日期转字符串
t1= time.localtime()
print(time.strftime("%Y-%m-%d %H:%M:%S",t1))
# 字符串转日期
timeStr = '2018-01-26 12:55:20'
t2=time.strptime(timeStr, "%Y-%m-%d %H:%M:%S")

#测试程序执行时间差
t_start = time.perf_counter() # 程序从开始到这条代码的运行时间
print("测试")
#开始休眠
time.sleep(5)
print("休眠完毕")
t_end = time.perf_counter()
print( t_end-t_start )

import datetime
t = datetime.datetime.now()
print(datetime.datetime.strftime(t, "%Y-%m-%d %H:%M:%S"))
print(datetime.datetime.strptime("20230211", "%Y%m%d"))
```

连接MySQL操作

pymysql 是在 Python3.x 版本中用于连接 MySQL 服务器的一个库
在 vscode 终端下面直接运行 pip3 install pymysql 安装即可

```
import pymysql
# 连接database
conn = pymysql.connect(
    host="127.0.0.1",
    user="root",
    password="root123456",
    database="test",
    charset="utf8")
print("打开数据库")
#关闭数据库
conn.close()
print("关闭数据库")
```

结构操作

```
# 使用预处理语句创建表
sql1 = """create table if not exists MovieType(
    tid int primary key,
    tname varchar(20),
    tcontent varchar(500),
    tdate datetime )"""

baoma.execute(sql1)
print("操作成功")
```

.execute(sql)的返回值

pymysql中，`cursor.execute(sql)` 的返回值是一个整数，表示：

- 对于 INSERT 、 UPDATE 、 DELETE 语句：返回受影响的行数（即成功插入、更新或删除的行数）。
- 对于 SELECT 语句：返回结果集的行数（即查询返回的总行数）。

数据增删改操作

```
c="insert into MovieType values(1,'喜剧','这是一种搞笑的视频',now())"
result=baoma.execute(c)
conn.commit() #需要提交才到数据库，有错则rollback
print("插入成功")
```

数据查询操作

```
import pymysql sql = "select * from MovieType order by tdate desc"
```

```
#读一行
```

```
baoma.execute(sql)
```

```
data = baoma.fetchone() #执行完毕返回的结果集默认以元组显示
```

```
print(data)
```

```
#读所有
```

```
data = baoma.fetchall() #元组的元组
```

```
print(data)
```

```
for i in data:
```

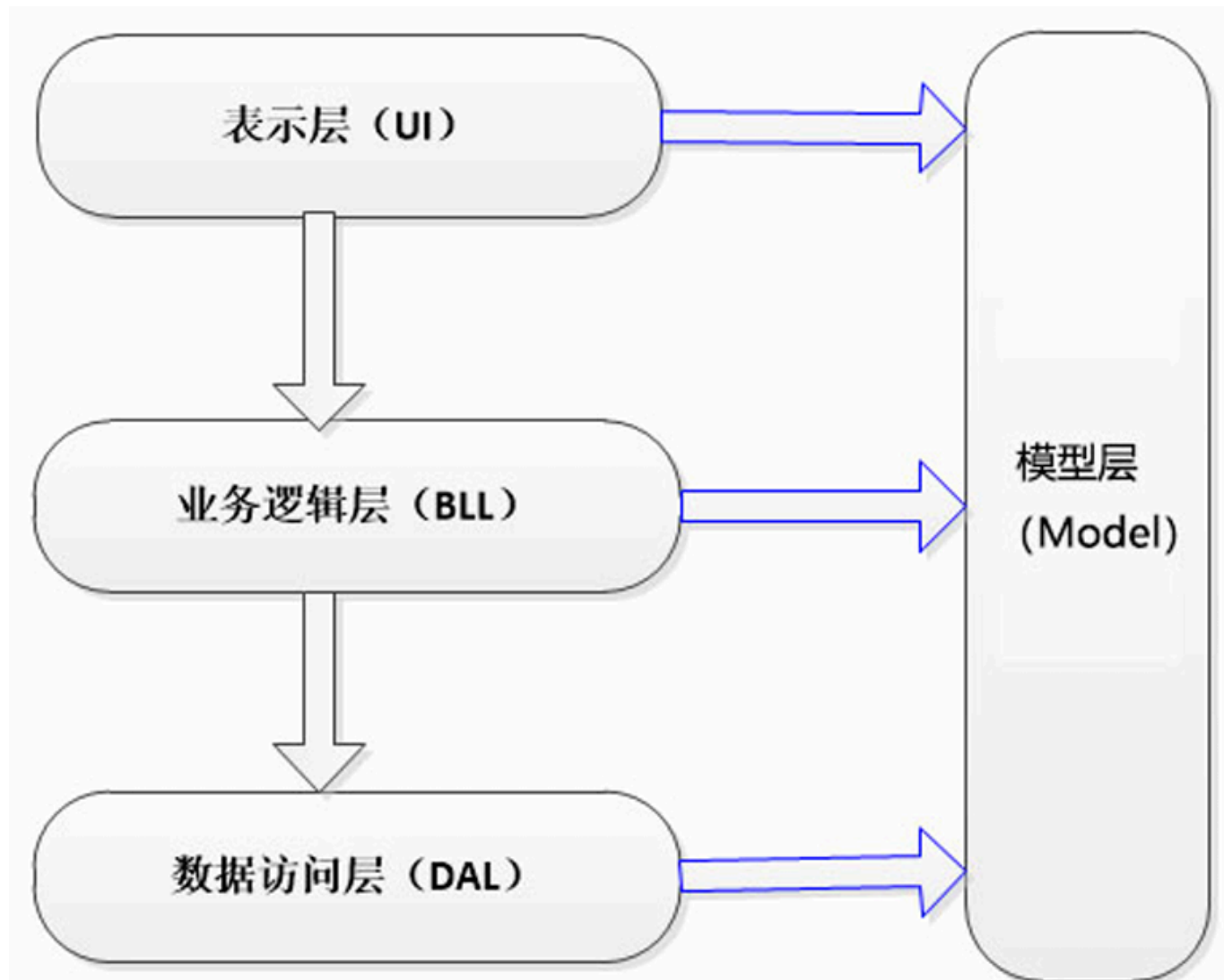
```
    print(i[0],i[1])
```

三层架构

表示层（前端UI）

业务逻辑层（BLL）（Business Logic Layer）

数据访问层（DAL）（Data Access Layer）



作业练习

1. 灵活使用 pymysql 来完成创建表、添加数据、查询数据

```
import pymysql
import datetime
```

```
def dbconnect():
    # 连接database
    conn = pymysql.connect(
        host="127.0.0.1",
        user="root",
        password="root123456",
        database="test",
        port=3306,
        charset="utf8"
    )
    print("连接数据库成功")
    return conn
```

```
def jie gou():
    sql="""create table if not exists MovieType(
    tid int primary key,
    tname varchar(20),
    tcontent varchar(200),
    tdate datetime
    )"""
    # 连接database
    conn = dbconnect()
    jianbiao=conn.cursor()
    jianbiao.execute(sql)
    print("MovieType表创建完成")
    jianbiao.close()
    conn.close()
    print("关闭数据库成功")
```

```
def edit():
    print("示例：1, 喜剧, 这是一个搞笑的电影")
    data=input("请输入要插入的数据（逗号隔开）：").split(",")
    tid=int(data[0])
    tname=data[1]
    tcontent=data[2]
    sql=f"insert into MovieType values({tid},'{tname}','{tcontent}',now())"
    conn = dbconnect()
```

```

inserdata=conn.cursor()
res=inserdata.execute(sql)
if res>0:
    print("添加成功")
else:
    print("添加失败")
conn.commit()
inserdata.close()
conn.close()
print("关闭数据库成功")

```

2. 查询数据时日期处理成如下格式:

1 喜剧 这是一个搞笑的电影 2023 年 08 月 21 日

2 动作 这是一个打斗的电影 2023 年 08 月 21 日

```

def cha():
    sql="select * from MovieType"
    conn=dbconnect()
    show=conn.cursor()
    show.execute(sql)
    res=show.fetchall()
    for i in res:
        print(i[0],i[1],i[2],datetime.datetime.strftime(i[3],"%Y年%m月%d日"))
    show.close()
    conn.close()
    print("关闭数据库成功")

if __name__=="__main__":
    # jieyou()
    # edit()
    cha()

```

上机练习10 三层架构开发豆瓣网

模型层 Model.py

```

import datetime
class MovieType:
    #属性
    tid=0
    tname=""
    tcontent=""
    tdate=datetime.datetime.now()

    #方法
    def __init__(self,tid1,tname1,tcontent1):
        self.tid=tid1

```

```
self.tname=tname1
self.tcontent=tcontent1
```

```
# 数据层 DAL.py
```

```
import pymysql
```

```
class DBHelper:
```

```
    #打开数据库
```

```
    def openDB(self):
```

```
        #把局部变量变成全局变量
```

```
        global conn
```

```
        conn=pymysql.connect(host="127.0.0.1",
                               user="root",
                               password="root123456",
                               database="test",
                               port=3306,charset="utf8")
```

```
        global baoma
```

```
        baoma=conn.cursor()
```

```
#关闭数据库
```

```
def closeDB(self):
```

```
    baoma.close()
```

```
    conn.close()
```

```
def edit(self,sql):
```

```
    self.openDB()
```

```
    result=baoma.execute(sql)
```

```
    conn.commit()
```

```
    self.closeDB()
```

```
    return result
```

```
def chaOne(self,sql):
```

```
    self.openDB()
```

```
    baoma.execute(sql)
```

```
    result=baoma.fetchone()
```

```
    self.closeDB()
```

```
    return result
```

```
def chaAll(self,sql):
```

```
    self.openDB()
```

```
    baoma.execute(sql)
```

```
    result=baoma.fetchall()
```

```
    self.closeDB()
```

```
    return result
```

```
class MovieTypeDAL(DBHelper):
```

```
    # 1.录入电影类型信息
```

```

def insert(self,mt):
    sql=f"insert into MovieType
values({mt.tid},' {mt.tname}',' {mt.tcontent}',now())"
    return self.edit(sql)

# 2.1 显示tid的数据内容
def showformtid(self,tid):
    sql=f"select * from MovieType where tid='{tid}'"
    return self.chaOne(sql)

# 2.2修改电影类型信息
def update(self,mt):
    edtname=0
    edtcontent=0
    if(mt.tname!=""):
        sql1=f"update MovieType set tname='{mt.tname}' where tid={mt.tid}"
        edtname=self.edit(sql1)

    if(mt.tcontent!=""):
        sql2=f"update MovieType set tcontent='{mt.tcontent}' where tid=
{mt.tid}"
        edtcontent=self.edit(sql2)

    return edtname,edtcontent

# 3.删除电影类型信息
def delet(self,tid):
    sql=f"delete from MovieType where tid={tid}"
    return self.edit(sql)

# 4.显示所有电影类型信息
def showall(self):
    sql="select * from MovieType"
    return self.chaAll(sql)

# 业务逻辑层 BLL.py
from Model import MovieType
from DAL import MovieTypeDAL
import datetime

mtdal=MovieTypeDAL()
menu="""\n-----豆瓣网管理系统-----
=====功能菜单=====
1 录入电影类型信息
2 修改电影类型信息

```

- 3 删除电影类型信息
- 4 显示所有电影类型信息
- 0 退出系统

=====

1. 录入电影类型信息

```
def option1():
    tid=input("请输入编号: ")
    res=mtdal.showformtid(tid)
    if not res:
        tname=input("请输入名称: ")
        tcontent=input("请输入内容: ")
        mt=MovieType(tid,tname,tcontent)
        res= mtdal.insert(mt)
        if res>0:
            print("录入成功")
        else:
            print("录入失败")
    else:
        print("编号已存在! ")
    input("\n按下回车键返回主菜单..")
```

2. 修改电影类型信息

```
def option2():
    tid=input("请输入修改编号: ")
    print("\n编号 名称 内容 创建日期")
    res=mtdal.showformtid(tid)
    if not res:
        print("此电影类型不存在! ")
        input("\n按下回车键返回主菜单..")
        return
    print(res[0],res[1],res[2],datetime.datetime.strftime(res[3],"%Y年%m月%d日\n"))
```

```
    tname=input("请输入修改名称（若不修改则直接回车）: ")
    tcontent=input("请输入修改内容（若不修改则直接回车）: ")
    mt=MovieType(tid,tname,tcontent)
    res=mtdal.update(mt)
    print(f"{res[0]}个名称被修改, {res[1]}个内容被修改")
    input("\n按下回车键返回主菜单..")
```

3. 删除电影类型信息

```
def option3():
    tid=input("请输入删除编号: ")
    res1=mtdal.showformtid(tid)
```



```

if not res1:
    print("此电影类型不存在！")
    input("\n按下回车键返回主菜单..")
    return

print("\n编号 名称 内容 创建日期")
print(res1[0],res1[1],res1[2],datetime.datetime.strftime(res1[3],"%Y年%m月%d日\n"))
esc=input("确认要删除吗? y/n:")
if esc.lower() == 'y' or esc.lower() == 'yes':
    res2=mtal.delet(tid)
    if res2>0 :
        print("删除成功")
    else:
        print("删除失败")
    input("\n按下回车键返回主菜单..")

# 4.显示所有电影类型信息
def option4():
    print("\n编号 名称 内容 创建日期")
    res=mtal.showall()
    for i in res:
        print(i[0],i[1],i[2],datetime.datetime.strftime(i[3],"%Y年%m月%d日"))
    input("\n按下回车键返回主菜单..")

# 0.退出系统
def option0():
    esc=input("请问是否退出? y/n:")
    if esc.lower() == 'y' or esc.lower() == 'yes':
        print("已成功退出")
        exit()
    else:
        input("\n按下回车键返回主菜单..")

if __name__=="__main__":
    while True:
        print(menu)
        option=input("请输入操作选项: ")
        if option=="1":
            option1()
        elif option=="2":
            option2()
        elif option=="3":
            option3()
        elif option=="4":
            option4()

```

```
elif option=="0":  
    option0()  
else:  
    print("无效操作，请重新输入！")  
    input("按下回车键继续..")
```