# 第一阶段：

## 第一部分*python语言基础*

*1. 程序运行过程:*

*源代码--编译* *-->字节码--解释-->机器码*

*|----1次-----|*

*|-----------运行时------------|*

*2. pycharm常用快捷键(百度搜索)*

*3. python内存管理*

*引用计数:*

*定义:每个对象存储被引用的次数,如果数量为零,则"销毁".*

*缺点:存在循环引用的现象.*

*标记清除:*

*定义:在内存容量不够时,从栈帧中开始扫描内存,标记正在使用的对象.*

*(不被使用的对象,即为可回收)*

*缺点:扫描内存耗时长.*

*分代回收:*

*定义:根据回收频次将内存分为多个区域("青年代","中年代","老年代")*

*避免每次扫描时范围过大.*

*优化:*

*尽少产生垃圾,对象池(小整数对象池/字符串池..),手动回收(慎重)*

*4. 容器*

*种类:字符串(只存字符编码值)/列表(预留空间)/元组(按需分配)/字典(单个元素读写速度最快)/集合(不重复/数学运算)*

*内存图:......*

*通用操作:+ \* 比较* *in 索引/切片*

*5. 函数*

*设计原则:单一职责*

*参数:*

*实参* *-- 调用时*

*位置实参* *(1,2,3)*

*序列实参* *(\*[1,2,3]) 备注:拆*

*关键字实参* *(a = 1,b = 2)*

*字典关键字实参(\*\*{"a":1,"b":2})备注:拆*

*形参* *-- 定义时*

*默认形参(a = 1,b = "")*

*位置形参(a,b)*

*星号元组形参(\*args) 备注:合*

*命名关键字形参(\*,b) (\*args,b)*

*双星号字典形参(\*\*kwargs) 备注:合*

*万能参数(\*args,\*\*kwargs)*

*# 3.内存管理*

*# 循环拼接字符串*

*# str\_result = ""*

*# for i in range(100):*

*# # 两个字符串拼接后产生新字符串,替换str\_result存储的引用后,产生垃圾.*

*# str\_result =str\_result + str(i)*

list\_result = []

**for** i **in** range(100):

list\_result.append(str(i))

str\_result = **""**.join(list\_result)

print(str\_result)

*# 4.容器*

list01 = [3,4,54,6,7]

*# 切片获取数据时创建新列表(拷贝原始数据)*

list02 = list01[1:4]

list02[0] = **"qtx"**

print(list01)

*# 切片修改数据时不创建新列表*

list01[1:4] = [**"qtx"**]

## 第二部分*面向对象* *11:10*

*OOA/OOD/OOP*

*字面意思:考虑问题从对象的角度出发.*

*三大特征:*

*封装:分*

*继承:隔*

*多态:做*

*设计原则:*

*开闭*

*单一职责*

*依赖倒置(抽象)*

*组合复用*

*里氏替换*

*迪米特(低耦合)*

*案例:老王开车去东北/图形(员工)管理器/技能系统..*

# 第三部分*python 核心*

*1. 程序结构*

*导包的路径从项目根目录开始计算.*

*导包是否成功的原则: sys.path + 自定义路径 可以正确找到模块.*

*2. 异常处理*

*异常现象:报错后返回给调用者,不再向后执行.*

*处理目的:让异常流程(向上)改为正常(向下)流程.*

*try:*

*可能出错的代码*

*except:*

*处理*

*3. 迭代*

*字面意思:在上一次基础上的重复/反复.*

*可迭代对象:\_\_iter\_\_()*

*迭代器:\_\_next\_\_()*

*4. 生成器*

*生成器函数:*

*def 函数名():*

*yield 数据*

*生成器表达式:*

*(对变量的处理* *for 变量* *in 可迭代对象)*

*class 生成器伪代码:*

*def \_\_iter\_\_():*

*return self*

*def \_\_next\_\_():*

*yield 以前的代码*

*return yield后面的数据*

*价值:惰性/延迟操作(省内存)*

*面试题:请简述迭代器与生成器.*

*5. 函数式编程:*

*函数作为参数: 将逻辑传入函数*

*函数作为返回值:逻辑连续(压岁钱)*

*闭包* *--> 装饰器(拦截)*