

《区块链基础》第 4 次作业

2025 年 06 月 04 日

基于蚂蚁链 CloudIDE 区块链开发环境，编写并调试以下程序：

1、Solidity 智能合约结构

1. 1 使用 CloudIDE 编译和部署应用程序小例子，基本功能应包括，对区块链状态变量进行 Get 和 Set 操作，即读写操作；

1. 2 解释 Solidity 智能合约的基本结构；

2、整数

2. 1 编写合约，计算两个正整数的和、差、积、商并输出，包括：

正常 Case：输入、输出全部在整型范围内；

异常 Case：变量不在其值域范围内，同时解析输出结果的机理；

2. 2 编写合约，计算两个正整数的取余结果，并输出；

2. 3 练习左移运算符和右移运算符；

3、布尔类型与枚举类型

3. 1 进行两个整型之间的比较运算，包括大于、小于、等于，并获得输出结果；

3. 2 练习枚举变量和整形数据类型的默认转换；

4、字符串和定长字节数组

4. 1 创建多个具有不同长度的定长字节数组，并对这个数组进行赋值和取值操作（注意字节数组的长度）；

4. 2 将一个字符串转换为定长字节数组；

5、地址类型

5. 1 编写 Solidity 智能合约，获取链账户地址，打印输出，并与蚂蚁联盟链托管的链账户地址进行比较；

5. 2 使用区块链浏览器查询交易详情；

6、数据位置

6. 1 使用数据位置关键字对状态变量，局部变量进行强制指定；

6. 2 尝试分析 Storage 和 Memory 的区别；

7、数组

7. 1 创建定长整型数组，并获取该定长整型数组元素；

7. 2 创建动态字符串数组，并向该数组中插入两个字符串：“Hello”，“World”；

7. 3 练习变长字节数组和字符串之间的转换；

8、映射

8. 1 创建一个 mapping，并往该映射中填充数据；

8. 2 练习使用 identity 作为 key 创建一个 mapping，并往该映射中填充数据；

9、结构体

9. 1 练习创建一个结构体来描述“车”；

9. 2 练习创建一个“车”的数组；

9. 3 练习创建一个“车”的映射；

9. 4 以实际的应用场景（比如停车场管理系统、员工人事管理系统、学生花名册管理系统）为例，对应用场景的需求进行描述，并实现包含数组、结构体、映射 3 种类型变量的嵌套定义、变量的读、写操作；

10、时间单位

10. 1 练习在 Solidity 合约中获取当前时间戳；

10. 2 练习对普通整型变量和时间单位进行运算；

11、Solidity 函数

11.1 创建接收两个无符号整型参数的函数，并返回这两个无符号整型参数的乘积；

11.2 编写一个函数，使用命名函数的方式来调用函数并返回值；

11.3 编写 internal 函数、external 函数和 public 函数并体验他们的区别；

12、Solidity 条件语句

12.1 创建一个打分函数：

如果分数大于等于 90，输出“优秀”；

如果分数小于 90 且大于等于 60，输出“良好”；

如果分数小于 60 且大于等于 30，输出“一般”；

否则，输出“差”；

13、Solidity 循环语句

13.1 分别使用三种循环语句计算 $1*2*3*\dots*100$ 的值，并输出；

13.2 练习使用 break 和 continue 跳出循环；

14、蚂蚁链常用平台接口函数

14.1 使用区块接口函数，获取系统中的 gas 最大值并输出；

14.2 使用加密接口函数对数据进行加密，并输出加密后的结果；

15、CloudIDE 使用调试技巧

15.1 将合约部署到本地模拟器中，并调用合约查看输出；

15.2 对上一步的合约进行单步调试，掌握单步调试的流程；

15.3 创建一个合约检测任务，并查看检测结果。

相关要求：

1、养成良好的代码编写习惯，包括但不限于智能合约、函数、变量等的命名规范、代码缩进、代码注释等；

2、养成良好的软件程序文档编写规范及习惯，比如代码的设计目的，预期的运行结果，调试过程（包括失败的过程及原因）等；

3、所有源代码，连同对应的调试过程及运行结果（截图的形式）整合到一个 WORD（或 PDF）文件里进行提交；

三、提交要求

1、为锻炼独立学习和应用能力，作业**务必独立完成！发现雷同作业，关联的各位同学均没有成绩！**

2、将所有作业整合成一份 WORD（或者 PDF）文件，统一提交至邮箱 23648094@qq.com

3、提交截止时间：6月22日24:00前；

4、邮件主题格式：04_2025 区块链_学号_姓名_提交日期（8位数格式，比如 20250622）

邮件附件格式：对 Word（或 PPT）进行打包，文件命名规则同上