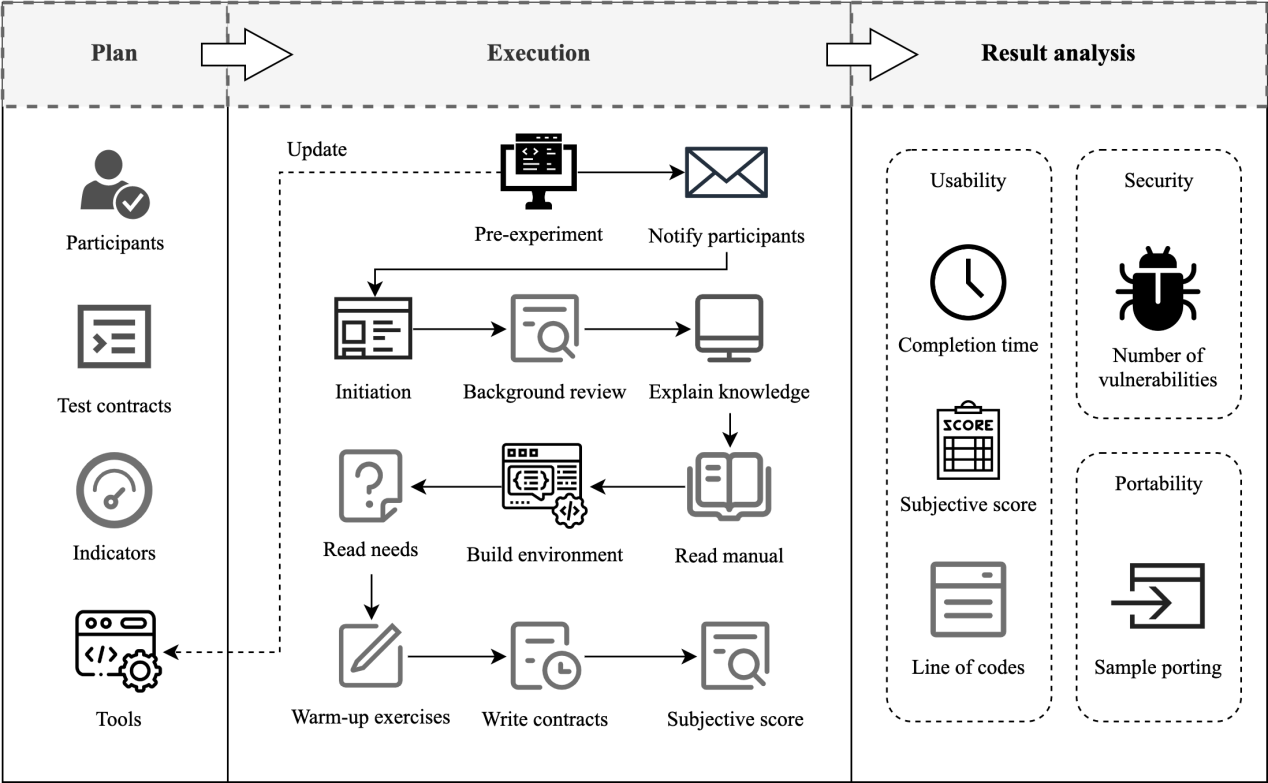
# 实验概览

受控实验是一种实证研究方法，主要通过操纵研究情景中的某一变量或因素来分析其所造成的影响。根据实验对变量的操纵与控制的程度和水平，实验可划分为前实验、准实验和真实验。其中，准实验是指在实验中未按随机原则来选择被试对象，且只对无关变量作尽可能控制的实验。本文从易用性、安全性及可移植性方面，对JSmarter语言展开实验与评估。由于受到环境限制，本文所开展的实验在实验主体选择等环节无法完全满足随机原则等要求，因此属于准实验。

本文开展评估实验的目的是，分析在软件开发能力和智能合约相关知识储备基本相同的情况下，参与者使用本文提出的JSmarter语言与现有主流智能合约语言开发功能相同的智能合约，完成所需时长、开发者的体验和评分、实现合约的代码量和漏洞数量是否存在差异。并对这些差异作出解释，以评估JSmarter在易用性和安全性方面的表现。此外，通过实验中的JSmarter合约样本验证JSmarter的可移植性。

实验执行过程概览如图1所示，分为实验计划、实验执行以及结果收集与分析三个部分。

图1：



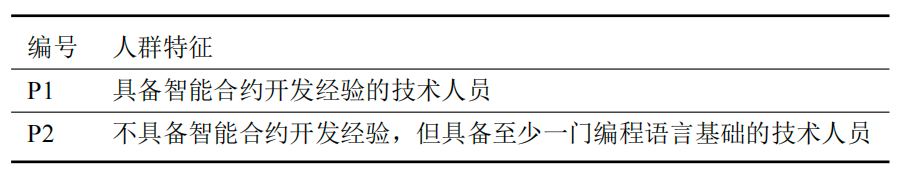
# 实验计划

实验计划已经在论文6.1节阐述。（没时间的话不翻译也行）

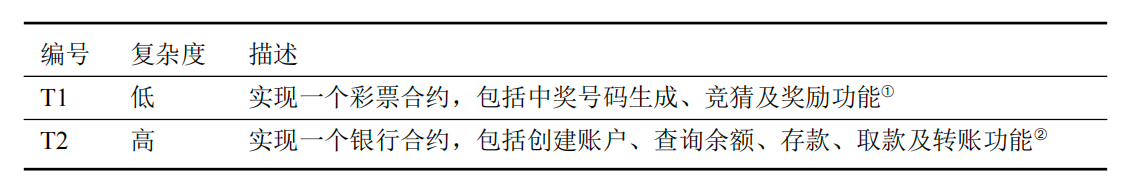
**研究对象。**JSmarter和Java、Go、Solidity、Vyper四种对比语言。本实验对于选取对比语言的考虑分为三个层面。首先，JSmarter是以Java为宿主语言的内部DSL，理应与原生Java进行对照。其次，Go是典型联盟链HF的主流合约开发语言，与其对比可以更好地评估JSmarter应用于联盟链的有效性。最后，Solidity和Vyper是典型公有链以太坊的主流合约开发语言，与其对比可以更好地评估JSmarter应用于公有链的有效性。同时，Java和Go属于通用编程语言，Solidity和Vyper属于DSL，四种语言在类别上也具备代表性。综上，本实验选取Java、Go、Solidity和Vyper作为对比语言，旨在更好地评估JSmarter的有效性。

**实验主体选择。**本实验面向两类人群，如表1所示。依托华为胡杨林基金项目，本实验邀请了48名实验主体（即实验参与者），其中P1类型实验主体16名，P2类型实验主体32名。

表1：



**测试合约选择。**本实验选择来自工业界的智能合约作为测试合约。经过两名具备智能合约开发经验的技术人员评估，实验选择了两个应用场景不同且复杂度递增的测试合约，其功能具有一定代表性，且部分功能容易出现安全漏洞。表2简要描述了上述选择的两个测试合约。



<https://github.com/sonnysangha/lottery-smart-contract>

<https://github.com/sosjk/fabric-chaincode-example-bankmaster>

**分组策略。**实验将48名参与者随机平均地分成四大组ABCD，再将每大组随机平均地分成两小组，每小组内包含2名P1参与者和4名P2参与者。每位参与者都需要使用JSmarter和Solidity、Vyper、Java、Go四种语言之一分别完成两个测试合约的开发。每一大组用于对比JSmarter和另一种语言的表现，每一大组下的两个小组用于排除语言使用顺序的干扰。

**知识背景均衡。**实验在选取参与者时，已经根据其是否具备智能合约开发经验划分成了两类，后续也将分类进行结果分析。且实验选取参与者的来源均一致，分组也满足随机，这些都在一定程度上保证了参与者知识背景的均衡。此外，实验在正式编写合约之前为参与者提供智能合约开发的基础知识演示，以及相关语言的官方文档和使用手册，帮助参与者提高知识储备，使其达到更加均衡的水平。

**合约开发顺序。**实验要求每位参与者均按照先T1后T2的顺序编写测试合约，即从语言的角度，每种语言实现的第一个合约都是T1，第二个合约都是T2。这避免了开发者在两个测试合约编写过程中，对智能合约熟悉程度的递增导致完成时间的递减，从而影响实验结果的有效性。

**预实验。**在正式实验之前，先进行一次小规模的预实验。预实验的目的如下：第一，评估每个测试合约的编写用时，从而在正式实验中给每个测试合约设置最长实现时间，若超出则标记当前测试合约未完成，开始下一个合约的编写任务，便于控制实验的整体时长。第二，评估整个实验的时长，方便在正式实验中通知参与者预留出足够的时间完成实验。第三，收集预实验参与者对于实验指南和语言工具的反馈意见，完善相关文档。第四，验证实验设计的合理性，并根据预实验中可能发生的一些未预料到的情况，优化设计细节。

**评估指标。**

1. 完成时长。使用计时器测量参与者使用各种语言完成每个测试合约的平均时间，包括语言的使用手册阅读时间、开发环境搭建时间，以及每个测试合约的编写时间。时间越短，则说明该语言的易用性越高。
2. 主观评分。在合约编写完成后，要求参与者填写评价问卷，即为JSmarter和另一种语言在开发体验、易学习性等方面进行评分，以此作为易用性的主观衡量标准。平均得分越高，说明该语言的易用性越高。
3. 代码量。统计实验中使用各种语言编写合约的代码量，代码行数越少代表程序越简单。从语言特性的角度，代码量可以作为一种侧面辅助印证该语言易用性的指标。
4. 漏洞数量。通过智能合约测试工具和人工校验结合的方式，检测实验中使用各语言编写合约中的漏洞类型和数量。检测出的漏洞数量越少，说明该语言的安全性越高。

**实验工具。**

1. 实验指南。实验指南分为两部分。第一部分是对于整体实验的说明讲义，这是为了让参与者大致了解实验的背景和目的。第二部分是对于智能合约编程基础知识讲解的幻灯片，其中包括使用五种语言编写智能合约的简单示例，以便参与者尽快熟悉智能合约开发的基本逻辑，这是考虑到参与者群体可能此前对于智能合约领域的相关知识了解较少。
2. 调研问卷。实验涉及两次问卷调研。背景问卷用于在实验前调研参与者的编程技术背景和智能合约相关的知识储备情况。评价问卷用于在实验后了解参与者对于各语言在开发体验、易学习性、语法设计的简洁易用性、使用手册的易理解性和环境搭建的便捷性等方面的评价，以及对于JSmarter的建议，便于其持续优化。
3. 语言工具。实验会提供五种语言开发智能合约的使用手册或官方文档，包括本文作者撰写的JSmarter使用手册。实验还提供了对于各语言建议的开发环境以及详细的搭建步骤，包括Solidity和Vyper的Remix在线环境，Golang的Lightly在线环境和Goland本地环境，以及Java和JSmarter的IntelliJ IDEA本地环境。
4. 测试合约描述。实验会给出对于两个测试合约的功能需求描述，这由三名具备智能合约开发经验的技术人员仔细检查过其完整性和准确性。

# 实验执行

1. 实验启动。首先向参与者说明实验内容，然后为每位参与者分配一个实验包。
2. 背景调研。要求参与者填写实验包中的背景问卷，以收集参与者的知识背景和编程经验。
3. 知识讲解和问答。为参与者展示并讲解区块链和智能合约的基础知识和各语言编程示例，并进行第一轮问答环节，以解答参与者对于智能合约及其编程语言的任何疑问。
4. 手册阅读与环境搭建。要求参与者阅读所分配到的两门编程语言的使用手册以及搭建相应的开发环境。为每个参与者启动计时器，记录其阅读手册和搭建环境所需的时间。
5. 需求阅读和问答。要求参与者阅读两个测试合约的功能需求描述，并进行第二轮问答环节，以解答参与者对于待实现合约的所有疑问。
6. 热身练习。要求参与者使用各语言完成一个简单的Hello world编程的热身练习，以确保参与者可以正常使用所有语言进行编程。
7. 合约编写。参与者有105min的时间来实现测试合约。参与者必须按照实验包中指定的语言和测试合约之间的对应关系来完成合约编写。为每个参与者启动计时器，记录每个测试合约的完成时间。合约完成的标志是在相应语言的开发环境中成功编译。若超过最长时间仍未完成，则结束当前测试合约的编写，标记为未完成，并进入下一个合约的编写任务，若当前测试合约已经是T2，则结束合约编写环节。编写过程中允许参考实验包中提供的各语言使用手册，但不允许参与者之间交流，以避免给实验结果带来偏差。
8. 主观评分。在合约编写完成后，要求参与者填写实验包中的评价问卷，收集参与者对于各语言的评价以及对于JSmarter的建议。至此，实验结束。

# 结果收集与分析

在结果分析之前，本文首先进行无效数据的去除。根据完成情况，实验中的合约样本可以分为三类，一是未在最长时限内完成开发，二是在时限内完成但经人工检查合约的功能实现存在错误，三是在时限内完成且经人工检查合约的功能实现正确。其中，前两类样本视为无效数据。