


КОНКУРСНОЕ ЗАДАНИЕ
КОМПЕТЕНЦИИ
«РАЗРАБОТКА РЕШЕНИЙ С
ИСПОЛЬЗОВАНИЕМ БЛОКЧЕЙН
ТЕХНОЛОГИЙ»

2023 г.

Сотисово,
директор филиала генерального
директора по конкурсам
и выставкам ОАи ПК
А.А. Сотисово



Конкурсное задание разработано экспертным сообществом и утверждено Менеджером компетенции, в котором установлены нижеследующие правила и необходимые требования владения профессиональными навыками для участия в соревнованиях по профессиональному мастерству.

Конкурсное задание включает в себя следующие разделы:

1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ.....	2
1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ	2
1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «РАЗРАБОТКА РЕШЕНИЙ С ИСПОЛЬЗОВАНИЕМ БЛОКЧЕЙН ТЕХНОЛОГИЙ»	2
1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ	5
1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ.....	5
1.5.2. Структура модулей конкурсного задания (инвариант/вариатив)	6
2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ.....	21
2.1. Личный инструмент конкурсанта.....	21
3. Приложения	22

1. ОСНОВНЫЕ ТРЕБОВАНИЯ КОМПЕТЕНЦИИ

1.1. ОБЩИЕ СВЕДЕНИЯ О ТРЕБОВАНИЯХ КОМПЕТЕНЦИИ

Требования компетенции «Разработка решений с использованием блокчейн технологий» определяют знания, умения, навыки и трудовые функции, которые лежат в основе наиболее актуальных требований работодателей отрасли.

Целью соревнований по компетенции является демонстрация лучших практик и высокого уровня выполнения работы по соответствующей рабочей специальности или профессии.

Требования компетенции являются руководством для подготовки конкурентоспособных, высококвалифицированных специалистов / рабочих и участия их в конкурсах профессионального мастерства.

В соревнованиях по компетенции проверка знаний, умений, навыков и трудовых функций осуществляется посредством оценки выполнения практической работы.

Требования компетенции разделены на четкие разделы с номерами и заголовками, каждому разделу назначен процент относительной важности, сумма которых составляет 100.

1.2. ПЕРЕЧЕНЬ ПРОФЕССИОНАЛЬНЫХ ЗАДАЧ СПЕЦИАЛИСТА ПО КОМПЕТЕНЦИИ «РАЗРАБОТКА РЕШЕНИЙ С ИСПОЛЬЗОВАНИЕМ БЛОКЧЕЙН ТЕХНОЛОГИЙ»

Перечень видов профессиональной деятельности, умений и знаний и профессиональных трудовых функций специалиста (из ФГОС/ПС/ЕТКС..) и базируется на требованиях современного рынка труда к данному специалисту

Таблица №1

Перечень профессиональных задач специалиста

№ п/п	Раздел	Важность в %
1	Организация работы	5
	Специалист должен знать и понимать: - Документацию и правила по охране труда и технике безопасности; - Основные принципы безопасной работы с ПК; - Значимость планирования всего рабочего процесса, как выстраивать эффективную работу и распределять рабочее время.	
	Специалист должен уметь: - Выполнять требования по охране труда и технике безопасности; - Выполнять требования техники безопасности при работе с ПК;	

	<ul style="list-style-type: none"> - Идентифицировать и использовать средства индивидуальной защиты; - Организовывать рабочее место для максимально эффективной работы; - Эффективно использовать рабочее время; - Работать эффективно, постоянно отслеживая результаты работы; - Внедрять и постоянно использовать высокие стандарты качества работ и технологий. 	
2	<p>Коммуникативные и межличностные навыки общения</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> - Важность поддержания знаний на высоком уровне; - Основные требования к смежным профессиям; - Значение построения продуктивных рабочих отношений; - Важность умения решать конфликтные ситуации и недопонимания. 	5
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> - Формулировать задание по требованиям к продукту; - Консультировать и рекомендовать продукцию или решения по новым технологиям; - Опрашивать заказчика точно и детально для понимания требований; - Давать ясные инструкции по эксплуатации; - Подготовить письменные отчеты о проделанной работе; - Производить оценку стоимости и необходимого времени для выполнения задания; - Адаптироваться к изменениям в смежных профессиях. 	
3	<p>Программирование узла блокчейн</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> - Сферы и назначение блокчейн-технологии; - Принципы работы блокчейн-технологии; - Формирование транзакций; - Формирование блоков. Механизмы консенсусов; - Передача блоков; - Возникновение ответвлений (форков) и механизмы определения основной цепочки; - Симметричные и асимметричные алгоритмы шифрования; - Как работают сетевые протоколы (tcp/ip); - Создание и работа peer-to-peer сети; - Существующие решения на рынке: EOS, Ethereum, Bitcoin, HyperLedger, Solana, TON, Polkadot, Waves. 	20
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> - Правильно выбирать и применять технологию; - Правильно выбирать и применять шаблоны и алгоритмы при разработке; - Программировать на одном из языков высокого уровня, такие как Python, Go, C++, Java, JavaScript, C# и т.п.; - Работать со средами разработки (IDE); - Использовать принципы Объектно-Ориентированного Программирования (ООП) или Функционального Программирования (ФП); - Читать и понимать существующий код; 	

	<ul style="list-style-type: none"> - Покрытие кода программы тестами; - Тестировать, отлаживать и оптимизировать код программы. 	
4	<p>Программирование умных контрактов</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> - Принципы работы умных контрактов в виртуальной среде; - Принципы создания умного контракта; - Принципы тестирования контракта; - Механизмы доставки контракта в среду выполнения; - Ограничения умных контрактов; - Взаимодействие умных контрактов; - Частые ошибки и уязвимости при создании умных контрактов. 	45
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> - Создавать умные контракты для определенной виртуальной среды; - Понимание ограничений на программный код, которые накладывает VM; - Тестирование умных контрактов; - Деплой умных контрактов; - Знание одного из языков программирования для написания контрактов. 	
5	<p>Программирование интерфейса</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> - Принципы создания удобного и функционального интерфейса; - Принципы взаимодействия со сторонним ПО. 	10
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> - Применять навыки системной аналитики для создания внешнего вида интерфейса; - Применять навыки программирования для создания интерфейса; - Настроить взаимодействие смарт-контракта с внешней средой. 	
6	<p>Информационная безопасность</p> <p>Специалист должен знать и понимать:</p> <ul style="list-style-type: none"> - Современные протоколы аутентификации с использованием асимметричной криптографии; - Протоколы распределения асимметричных ключей в открытых системах; - Специфику передаваемых данных в открытой информационной системе; - Типовые уязвимости смарт контрактов 	15
	<p>Специалист должен уметь:</p> <ul style="list-style-type: none"> - Применять навыки информационной безопасности для создания надежной системы аутентификации; - Учитывать характер данных, передаваемых в транзакциях. 	

1.3. ТРЕБОВАНИЯ К СХЕМЕ ОЦЕНКИ

Сумма баллов, присуждаемых по каждому аспекту, должна попадать в диапазон баллов, определенных для каждого раздела компетенции, обозначенных в требованиях и указанных в таблице №2.

Таблица №2

Матрица пересчета требований компетенции в критерии оценки

Критерий/Модуль						Итого баллов за раздел ТРЕБОВАНИЙ КОМПЕТЕНЦИИ
Разделы ТРЕБОВАНИЙ КОМПЕТЕНЦИИ И		А	Б	В	Г	
	1	1	1	1	2	5
	2	0	0	0	5	5
	3	7	7	5	1	20
	4	20	20	4	1	45
	5	4	4	–	2	10
6	5	5	4	1	15	
Итого баллов за критерий/модуль		37	37	14	12	100

1.4. СПЕЦИФИКАЦИЯ ОЦЕНКИ КОМПЕТЕНЦИИ

Оценка Конкурсного задания будет основываться на критериях, указанных в таблице №3:

Таблица №3

Оценка конкурсного задания

Критерий	Методика проверки навыков в критерии
А Решение на базе блокчейн-платформы №1	Решение проверяется путем просмотра демонстрации видеоматериалов, подготовленных Конкурсантами в соответствии с конкурсным заданием; проверки отчета о проделанных действиях, составленного Конкурсантами в соответствии с конкурсным заданием; тестированием функционала разработанного решения. Для более объективной оценки могут быть получены комментарии ТАП.
Б Решение на базе блокчейн-платформы №2	Решение проверяется путем просмотра демонстрации видеоматериалов, подготовленных Конкурсантами в соответствии с конкурсным заданием; проверки отчета о проделанных действиях, составленного Конкурсантами в соответствии с конкурсным заданием; тестированием функционала разработанного решения. Для более объективной оценки могут быть получены комментарии ТАП.

В	Аналитика	Проверка производится путем сверки предоставленных решений с подготовленными разработчиками задания ключами-ответами.
Г	Презентация	Осуществляется оценка только представленной в докладе информации. Разрешены уточняющие вопросы, позволяющие оценить корректность и понимание Конкурсантами представленной информации. Ответы на вопросы оцениваются отдельно.

1.5. КОНКУРСНОЕ ЗАДАНИЕ

Общая продолжительность Конкурсного задания¹: 18 ч.

Количество конкурсных дней: 3 дня

Вне зависимости от количества модулей, КЗ должно включать оценку по каждому из разделов требований компетенции.

Оценка знаний участника должна проводиться через практическое выполнение Конкурсного задания. В дополнение могут учитываться требования работодателей для проверки теоретических знаний / оценки квалификации.

1.5.1. Разработка/выбор конкурсного задания

Конкурсное задание состоит из 4 модулей, включает обязательную к выполнению часть (инвариант) – 2 модуля, и вариативную часть – 2 модуля. Общее количество баллов конкурсного задания составляет 100.

Обязательная к выполнению часть (инвариант) выполняется всеми регионами без исключения на всех уровнях чемпионатов.

Количество модулей из вариативной части, выбирается регионом самостоятельно в зависимости от материальных возможностей площадки соревнований и потребностей работодателей региона в соответствующих специалистах. В случае если ни один из модулей вариативной части не подходит под запрос работодателя конкретного региона, то вариативный (е) модуль (и) формируется регионом самостоятельно под запрос работодателя. При этом, время на выполнение модуля (ей) и количество баллов в критериях оценки по аспектам не меняются.

Матрица конкурсного задания доступна по ссылке <https://disk.yandex.ru/i/y8wsqrfdz8NUJw>

1.5.2. Структура модулей конкурсного задания (инвариант/вариатив)

¹ Указывается суммарное время на выполнение всех модулей КЗ одним конкурсантом.

Модуль А. Решение на базе блокчейн-платформы №1 (инвариант)

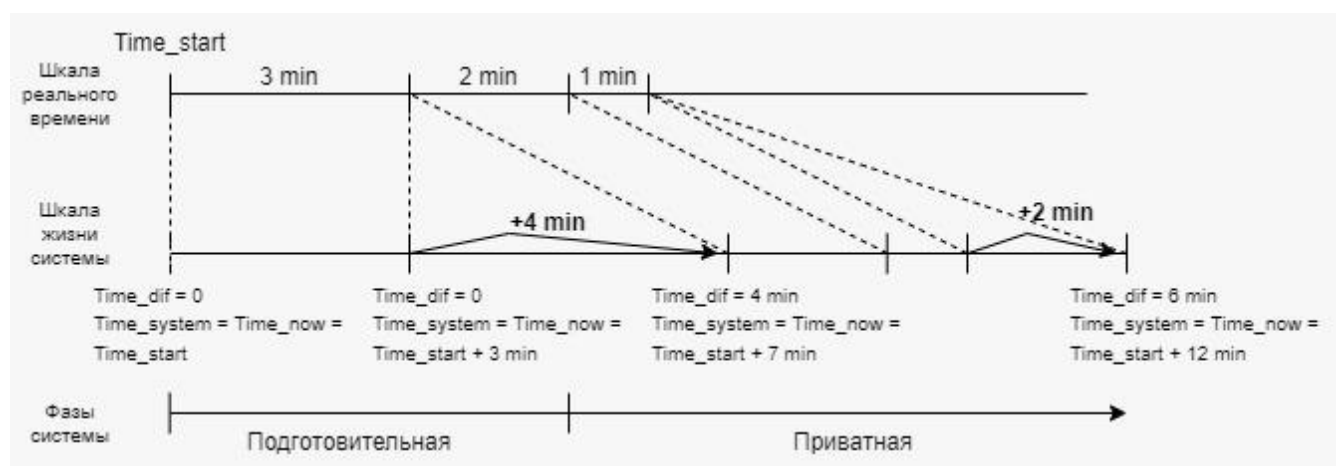
Время на выполнение модуля 7 часов

Задания:

Конкурсантам необходимо реализовать сервис для работы с токеном стандарта ERC20 на платформе Ethereum. Помимо основного функционала, приведенного в шаблоне стандарта, необходимо реализовать дополнительную бизнес-логику по привлечению инвесторов к проекту в соответствии с техническим заданием.

С момента публикации контракта начинается отсчет времени жизни системы (глобальная переменная контракта). Всем пользователям системы должна быть доступна функция управления временем (этой глобальной переменной) В системе должна быть предусмотрена кнопка, нажатие на которую каждый раз увеличивает время жизни системы на 1 минуту.

Жизненный цикл системы состоит из 3 фаз: Подготовительная – 5 минут, Приватная (Private) – 10 минут, Публичная (Public)– все оставшееся время.



Пояснения:

В момент старта системы фиксируется время (определяется по системному времени) в переменную Time_start. Также вводятся три переменные Time_dif = 0 (разница времени), Time_now (время в текущий момент, определяется по системному времени), Time_system (время жизни системы, определяется как Time_now + Time_dif). Определение текущей фазы системы происходит в результате сравнения переменных Time_start и Time_system.

В момент старта системы $Time_start = Time_now = Time_system$.

Например, система была запущена 20 апреля 2022 года в 14:00:00 по Московскому времени. Тогда:

1. При старте системы

Time_start = 1650452400
Time_now = 1650452400
Time_system = 1650452400
Time_dif = 0

2. Через 3 минуты

Time_start = 1650452400
Time_now = 1650452580
Time_dif = 0

Time_system = Time_now + Time_dif = 1650452580

$(\text{Time_system} - \text{Time_start})/60 = (1650452580 - 1650452400)/60 = 180/60 = 3$ –

Время жизни системы 3 минуты, значит мы в подготовительной фазе

3. Перемотали время на 4 минуты вперед

Time_start = 1650452400
Time_now = 1650452580
Time_dif = Time_dif + 4 * 60 = 240

Time_system = Time_now + Time_dif = 1650452580 + 240 = 1650452820

$(\text{Time_system} - \text{Time_start})/60 = (1650452820 - 1650452400)/60 = 420/60 = 7$ –

Время жизни системы 7 минут, значит мы в приватной фазе

4. Прошло еще 2 минуты

Time_start = 1650452400
Time_now = 1650452700
Time_dif = 240

Time_system = Time_now + Time_dif = 1650452700 + 240 = 1650452940

$(\text{Time_system} - \text{Time_start})/60 = (1650452940 - 1650452400)/60 = 540/60 = 9$ –

Время жизни системы 9 минут, значит мы в приватной фазе

5. Прошла еще 1 минута и перемотали еще на 2 минуты вперед

Time_start = 1650452400
Time_now = 1650452760

Time_dif = Time_dif + 2 * 60 = 240 + 120 = 360

Time_system = Time_now + Time_dif = 1650452760 + 360 = 1650453120

$(\text{Time_system} - \text{Time_start})/60 = (1650453120 - 1650452400)/60 = 720/60 = 12$ –

Время жизни системы 12 минут.

Общие сведения

- Наименование: CryptoMonster
- Символ: CMON

- Общая капитализация: 10 000 000
- Количество десятичных знаков: 12
- Стандарт: ERC20

Распределение монет

Все токены системы изначально принадлежат владельцу системы (Owner).

В системе предусмотрено несколько раундов распределения токенов, которое выглядит следующим образом:

Seed – 10%

Private – 30%

Public – 60%

Управление каждой группой токенов возлагается владельцем токенов на соответствующего провайдера (Таблица A1):

Таблица A1

Роль	Адрес в сети Ethereum
Владелец системы (Owner)	
Private провайдер	
Public провайдер	

В системе при старте должны существовать пользователи из таблицы A1. В системе должны быть реализованы функция авторизации и функция регистрации новых пользователей.

Распределение Seed токенов

Seed раунд предназначен для выдачи токенов начальным инвесторам, которые уже сделали вклад в проект. Токены должны быть переведены на счет соответствующих инвесторов (Таблица A2) сразу после запуска системы.

Таблица A2

Обозначение пользователя	Адрес в сети Ethereum	Количество токенов, CMON
Investor1		300 000
Investor2		400 000
Best friend		200 000

В системе при старте должны существовать пользователи из таблицы A2.

Фаза распределения Private токенов

Во время подготовительной фазы потенциальные пользователи системы подают заявки Private провайдеру о том, что готовы инвестировать в будущий проект.

Поля в заявке:

- имя;
- адрес в сети Ethereum.

Заявки модерируются Private провайдером:

- принимаются – адрес добавляется в whitelist фазы;
- отклоняются – заявка удаляется из списка.

Стадия покупки Private токенов начинается через 5 минут после старта системы, стадия длится в течение 10 минут. По завершении стадии приватной продажи сразу запускается стадия свободной продажи.

Покупка Private токена должна быть ограничена объемом в 100 000 CMON за одну транзакцию.

Стоимость одного токена CMON в Private фазу составляет 0.00075 ETH.

Если в этот срок токен пытается купить пользователь, не находящийся в whitelist, транзакция отклоняется с ошибкой «*Free sale not started*».

Покупка Public токенов

Сразу после завершения стадии приватной продажи все пользователи получают возможность свободной покупки токена.

Покупка Public токена должна быть ограничена объемом в 5 000 CMON за одну транзакцию.

Стоимость одного токена CMON в Public фазу составляет 0.001 ETH. Необходимо предусмотреть возможность изменения стоимости токена Public провайдером.

Помимо свободной продажи Public токен также предназначен для вознаграждения партнеров проекта, рекламодателей и т.д. Необходимо предусмотреть функционал для безвозмездного перевода указанного количества

Public токенов на указанный адрес (комиссию за перевод оплачивает Public провайдер).

Прочие требования

Один и тот же пользователь может обладать токенами из любой группы. Работа с токенами каждой группы должна осуществляться в соответствии с вышеописанной логикой.

Владелец системы (Owner) должен иметь возможность распоряжаться непроданными и только непроданными токенами раунда **только после окончания фазы**.

Графический интерфейс сервиса

Графический интерфейс должен включать следующий необходимый функционал пользователя:

Панель Владельца системы:

- просмотр полной информации об активах конкретного пользователя с учетом всех его токенов в разных группах;

Панель Private провайдера:

- обработка заявок в whitelist (добавление и отказ) до начала фазы продажи Private токенов, видны Имя пользователя, адрес в сети;
- публикация whitelist в контракт;
- просмотр информации о токенах Private группы конкретного пользователя;

Панель Public провайдера:

- выдача вознаграждения партнерам проекта с указанием адреса пользователя и суммы вознаграждения;
- просмотр информации о токенах Public группы конкретного пользователя;

Любой провайдер и владелец токенов может рассматриваться и как обычный пользователь, и иметь возможность пользоваться функционалом для работы с токеном:

- передавать его другим пользователям;
- предоставлять право распоряжаться своими токенами другому пользователю;
- оставлять заявку на добавление в whitelist;
- участвовать в приватной продаже токенов в случае его добавления в whitelist;
- покупать токены в фазу свободной продажи.

В графическом интерфейсе каждого пользователя должны отображаться:

- **таймер:** время жизни системы;
- **таймер:** общая длительность Private фазы;
- **таймер:** время с момента начала свободной продажи;
- роль пользователя в системе;
- адрес пользователя в сети;
- баланс ETH в сети;
- баланс CMON в сети.

Демонстрация

Засчитываются при полном выполнении шага. Демонстрация может выполняться в графическом интерфейсе или путем непосредственного вызова функций контракта. Заполните приведенные в конкурсном задании таблицы с используемыми адресами в сети Ethereum:

1. Выполнить запуск системы.
2. Опубликовать контракт в сети, показать время старта системы
3. Подать 3 разные заявки с разными адресами на добавление адреса в whitelist
4. Принять первую и третью заявку, отклонить вторую заявку
5. Перевести 3 000 CMON со счета Investor2 на счет Bestfriend с помощью функционала системы (смарт-контракт или GUI)
6. Вызвать функцию покупки токена пользователем из whitelist, не дожидаясь начала Private фазы, при этом показать изменение баланса ETH.
7. Ускорить систему как минимум на одну минуту (или больше для достижения начала Private фазы), показать время работы системы
8. Купить 150 000 CMON первым пользователем, включенным в whitelist, при этом показать изменение баланса ETH.
9. Купить 60 000 CMON любым пользователем, не включенным в whitelist, при этом показать изменение баланса ETH.
10. Купить 85 000 CMON вторым пользователем, включенным в whitelist, при этом показать изменение баланса ETH.
11. Ускорить систему как минимум на одну минуту (или больше для достижения начала Public фазы), показать время работы системы
12. Купить 6 500 CMON любым пользователем, при этом показать изменение баланса ETH.
13. Купить 4 500 CMON любым пользователем, при этом показать изменение баланса ETH.
14. Увеличить стоимость Public токена до 0.002.
15. Купить 4 500 CMON любым пользователем, при этом показать изменение баланса ETH.
16. Передать любому пользователю 10 000 CMON Public токенов в качестве вознаграждения.
17. Перевести доступные Владельцу системы (Owner) 250 000 CMON, оставшиеся после завершения разных фаз пользователю Bestfriend

ВАЖНО!!! Для быстрого разворачивания и запуска решения должен быть создан bat-файл с именем «StartX.bat», где X – номер рабочего места. Необходимо привести инструкции по установке, запуску и эксплуатации готового решения. Код разработанного решения должен быть задокументирован.

ВАЖНО!!! Необходимо привести достаточные пояснения к работе всех функций смарт-контракта, описание входных параметров и возвращаемых значений.

ВАЖНО!!! При выполнении Модуля А у участников ограничен доступ к сети Интернет

ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «2023_АХ», где X – номер рабочего места. Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.

Модуль Б. Решение на базе блокчейн-платформы №2 (вариатив)

Время на выполнение модуля 7 часов

Задания:

В рамках Модуля Б необходимо:

1. Выполнить инициализацию тестовой сети платформы Hyper Ledger Fabric.
2. Разработать смарт-контракт на одном из поддерживаемых платформой языков.
3. Создать в сети организации с названиями Shops, Users
4. Создать в сети канал blockchain2023 для размещения в нем chaincode разрабатываемого смарт-контракта.
5. Реализовать функции в соответствии с заданной логикой.
6. Продемонстрировать работу решения в соответствии с заданным сценарием.

Пояснения: Ролевая модель.

В системе предусмотрены следующие роли: магазины, покупатели, поставщик.

Магазин в рамках задания – это аккаунт в блокчейн-сети, который принадлежит организации Shops, аккаунт создается в системе при ее запуске либо вводится в систему в ходе ее использования путем регистрации новых аккаунтов. Всего в систему при старте должно быть заведено 4 магазина согласно табл. Б1.

Поставщик в рамках задания – это аккаунт в блокчейн сети, который создается при запуске сети и принадлежит организации Shops.

Таблица Б1 – Список магазинов

№ магазина	Город	Баланс, JekkyCoin
1	Дмитров	1000
2	Калуга	900
3	Москва	1050
4	Рязань	700

При старте в системе должны быть зарегистрированы:

- ✓ 4 адреса для магазинов с заданным балансом (табл. Б1).
- ✓ Поставщик Золотая рыбка (логин goldfish, на счету 100 JekkyCoin).

Принадлежат организации Shops.

✓ Покупатель Романов Роман Романович (логин roman, на счету 80 JekkyCoin). Принадлежат организации Users.

✓ Покупатель Николаев Николай Николаевич (логин nikola, на счету 90 JekkyCoin). Принадлежат организации Users.

Товары поставщика

Поставщик поставяет товары в магазин. Товар является объектом контракта, которому присущи следующие атрибуты: название, производитель, дата изготовления, срок годности, температура хранения, единица измерения (или кг или шт), базовая цена товара. Поставщик может создавать любые объекты, которые он готов поставить в магазины. При этом Магазины (и только магазины!) должны получить возможность увидеть товары, заведенные в систему.

Поставка товаров от поставщика в магазин

Поставка товаров в магазин осуществляется по следующей схеме. Заказ продуктов осуществляется из аккаунта Магазина. Для каждого выбранного товара указывается объем закупаемого товара (в кг или шт). После этого производится расчет конечной цены товара на основе базовой цены товара и объема закупаемого товара по следующему принципу.

$$ЦТ = БЦ * К * КЛ,$$

где

ЦТ – цена товара

БЦ – базовая цена

КЛ – количество товара (в кг или шт).

К – коэффициент ($K = 1$, если $KЛ \leq 100$; $K = 0,95$, если $100 < KЛ \leq 1000$; $K = 0,9$, если $KЛ > 1000$)

Например, закупка 200 кг креветок по базовой цене 0,03 JekkyCoin за килограмм составит:

$$ЦТ = 0,03 * 0,95 * 200 = 5,7 \text{ JekkyCoin}$$

Магазин должен подтвердить свою готовность купить товар по рассчитанной стоимости. После подтверждения товар отгружается со склада. Товар должен доставляться в условиях определенного температурного режима (характерного для конкретного заказанного товара). Считаем, что функция генератора случайных чисел будет являться датчиком температуры. За время доставки товара должно быть сгенерировано 6 транзакций в блокчейн-системе с показаниями температуры (случайные числа в диапазоне от -50 до +50). Если температурный режим был нарушен (выходит за рамки температурного режима, установленного для товара), то цена поставки снижается на 10% за каждое нарушение. Если температурный режим был нарушен, то Магазину выводится конечная цена продукта (с условием снижения цены в случае нарушения температурного режима) и предлагается сделать выбор: принять товар или отказаться от товара. Если Магазин выбрал принять товар, то поставщик получает конечную сумму стоимости товара, а Магазину возвращается остаток (10% за каждое нарушение). Если Магазин выбирает отказаться от товара, то Магазин получает полный возврат стоимости. Если условия доставки не были нарушены или если Магазин принял товар, то данный товар становится доступен для покупки обычными пользователями по цене, увеличенной на 50% по отношению к закупочной.

Например.

Магазин №1 готов купить 200 кг креветок по цене 5,7 JekkyCoin. Температурный режим хранения товара от -20 до -5. При доставке датчик показал следующие значения: -10, -5, 0, -7, 1, -15. Температурный режим был нарушен дважды. Поэтому итоговая цена товара будет равна:

$$5,7 * 0,8 = 4,56 \text{ JekkyCoin} \text{ – закупочная цена за 200 кг}$$

$$4,15528 / 200 = 0,0228 \text{ JekkyCoin} \text{ – закупочная цена за 1 кг}$$

$$0,0228 * 1,5 = 0,0342 \text{ JekkyCoin} \text{ – розничная цена в продажу.}$$

Покупка товаров в магазине

После того как товар поступил в магазин, покупателю становится доступен список товаров, которые он может купить, а также розничная цена для покупки.

В случае обнаружения брака, покупатель может инициировать процедуру возврата денег. Деньги должны вернуться на счет покупателя, а товар в Магазин только после того, как Магазин подтвердит возврат.

Для каждого пользователя системы должен быть доступен реализованный функционал посредством графического интерфейсов.

Через интерфейс должен быть реализован следующий функционал:

1. Создание товара (поставщик).
2. Заказ товаров (Магазины).
3. Подтверждение получения товара (Магазины).
4. Покупка товаров в магазине (покупатель).
5. Просмотр информации о товарах в магазине (все).
6. Возврат товара в случае брака (покупатель).
7. Подтверждение возврата товара (Магазины)
8. Просмотр своего баланса (все).

В интерфейсе зарегистрированных пользователей системы (поставщик, Магазины, покупатель) должен быть «личный кабинет», в котором отображаются: все личные данные пользователя, его текущий баланс, его статус (роль), история всех совершенных им действий. В интерфейсе для аккаунта Магазин должны отображаться: текущий баланс, номер магазина, город, список товаров.

Должна быть дана инструкция по установке, запуску и использования решения.

ВАЖНО!!! Для быстрого разворачивания и запуска решения должен быть создан исполняемый файл с именем «StartX.***», где X – номер рабочего места участника. Реализованное решение должно запускаться на ПК участника. Необходимо привести инструкции по установке, запуску и эксплуатации готового решения. Код разработанного решения должен быть задокументирован.

Демонстрация

Необходимо продемонстрировать работоспособность разработанной системы. Демонстрация работы приложения осуществляется путем видеозахвата экрана и комментариев разработчиков. При демонстрации необходимо показывать изменения состояния блокчейн-системы. Для демонстрации решения необходимо выполнить действия:

1. Зайти в аккаунт Магазина №1. Показать информацию о магазине.
2. Зайти в аккаунт Поставщика. Создать товар «Кальмар», производство ОкеанТорг, дата изготовления 05.04.2022, срок хранения 30 дней, температура хранения от -10 до 0, базовая цена 0,002 JekkyCoin за 1 кг. Создать товар «Лосось. Стейк», производство Северное Море, дата изготовления

01.04.2022, срок хранения 30 дней, температура хранения от -15 до -5, базовая цена 0,004 JekkyCoin за 1 кг.

3. Зайти в аккаунт Магазин №2. Купить 100 кг товара «Кальмар». Показать температурные режимы. В любом случае получить товар.

4. Зайти в аккаунт Магазин №3. Купить 400 кг товара «Лосось. Стейк». Показать температурные режимы. Если было нарушение – отказаться.

5. Зайти в аккаунт Магазин №1. Купить 400 кг товара «Лосось. Стейк». Показать температурные режимы. В любом случае получить товар.

6. Зайти в аккаунт Романов Роман Романович. Показать баланс. Посмотреть товары в Магазине №2 и их цену. Купить 5 килограмм товара «Кальмар».

7. Зайти в аккаунт Николаев Николай Николаевич. Показать баланс. Посмотреть товары в Магазине №1 и их цену. Купить 10 килограмм товара «Лосось. Стейк».

8. Зайти в аккаунт Николаев Николай Николаевич. Вернуть товар «Лосось. Стейк» в связи с браком.

9. Зайти в аккаунт Магазина №1. Подтвердить возврат. Показать Баланс.

10. Зайти в аккаунт Николаев Николай Николаевич. Показать баланс.

ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «2023_БХ», где Х – номер рабочего места. Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.

Модуль В. Аналитика (вариатив)

Время на выполнение модуля 2 часа

Задание:

Смарт-контракт был написан с использованием компилятора языка программирования Solidity версии 0.8.0. Не разрешается изменять описание и определение приведенных в контракте структур данных. Код смарт-контракта может быть размещен в открытом доступе.

Конкурсантам необходимо сформировать:

- отчет по синтаксису – указать строки, в которых допущены синтаксические ошибки и их исправление;

- отчет по функционалу – описать функционал каждого контракта, в приведенном файле;

- отчет по логике – указать на ошибки в логике работы контракта, внести предложения по их устранению;

- отчет по проверкам – предложить текст сообщений об ошибке в существующих в контракте проверках;

- отчет по информационной безопасности – указать на существующие недостатки контракта (ошибки, уязвимости и т.д.) с точки зрения информационной безопасности.

- оптимизация смарт-контракта – указать на недостатки разработанного смарт-контракта, предложить варианты его оптимизации;

- тестирование смарт-контракта – реализовать тесты для автоматизированной проверки работы смарт-контракта для следующих функций: *create_user()*, *participate()*, *finish()*.

```
1 pragma solidity >0.5.0;
2 contract Contract{
3
4     struct User{
5         uint balance;
6         string login;
7     }
8     mapping(string >> address) public logins;
9     mapping(address >> User) public users;
10    address root = msg.sender;
11
12    function create_user(string memory login, string memory FIO) public{
13        require(logins[login] ==
140x0000000000000000000000000000000000000000000000000000000000000000, "_____");
15        login[login] = msg.sender;
16        users[msg.sender] = User(FIO, msg.sender.balance, login);
17    }
18    function get_balance(address user_address) public view returns(uint){
19        return(users[user_address].balance);
20    }
21
22    function send_money(address payable adr_to) public payable {
23        adr_to.transferring(msg.sender);
24    }
25
26    struct Donation{
27        uint donate_id;
28        string name;
29        address user;
30        uint amount;
```

```

31     uint deadline;
32     address[] sender;
33     uint[] value;
34     bool status;
35     string info;
36 }
37
38     Donation donation;
39
40     function ask_to_donate(string memory name, uint amount, uint deadline, string
memory info) public {
41         address[] memory sender;
42         uint value;
43         donation.push(Donation(donation.length, name, msg.sender, amount,
deadline, sender, value, false, info));
44     }
45
46     function participate(uint donation_id) public payable{
47         require(donation[donation_id].status == false, "_____");
48         require(msg.value > 0, "_____");
49         donation[donation_id].sender.push(msg.sender);
50         donation[donation_id].value.push(msg.value);
51     }
52
53     function get_donation(uint donation_id) public view returns(uint, string
memory, address, uint, uint, bool){
54         return(donation_id, donation[donation_id].name, donation[donation_id].user,
donation[donation_id].amount, donation[donation_id].deadline,
donation[donation_id].status);
55     }
56     function get_donation_2(uint donation_id) public view returns(address[]
memory, uint[] memory, string memory) {
57         return(donation[donation_id].sender, donation[donation_id].value,
donation[donation_id].data);
58     }
59
60     function get_donation_number() public view return(uint) {
61         return donation.len();
62     }
63
64     function get_total(uint donation_id) public view returns(uint){
65         uint total = 0;
66         for (uint i = 0; i < donation[donation_id].value.length; i++){
67             total = donation[donation_id].value[i];
68         }

```

```

69     return total;
70 }
71
72 function finish(uint donation_id) public{
73     require(msg.sender == donation[donation_id].user, "_____");
74     require(donation[donation_id].status == false, "_____");
75     uint total = get_total(donation_id);
76     if (total ** 2 >= donation[donation_id].amount){
77         payable(donation[donation_id].user).transfer(total);
78     }
79     else{
80         for (uint i = 0; i < donation[donation_id].value.length; i++) {
81             payable(donation[donation_id].sender[i]).transfer(donation[donation_id].value[i]);
82         }
83     }
84     donation[donation_id].status = false;
85 }
86 }

```

ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «2023_B». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.

Модуль Г. Презентация (инвариант)

Время на выполнение модуля 2 часа

Задания:

На работу Модуля Г отводится 2 часа. Первый час времени используется для подготовки презентации блокчейн-решения (на выбор участников), которое было разработано в рамках модуля А или модуля Б. Второй час используется для заслушивания докладов участников.

В ходе презентации необходимо отобразить следующие моменты:

- ✓ техническое задание проекта;
- ✓ планирование всех задач проекта;
- ✓ архитектуру проекта: основные функциональные модули системы и их взаимосвязь;
- ✓ подробную схему работы модуля авторизации;
- ✓ обоснование использования технологии блокчейн;
- ✓ стек технологий для разработки;
- ✓ графический интерфейс решения;
- ✓ возможные уязвимости и недостатки проекта;
- ✓ план дальнейшего развития проекта;
- ✓ план коммерциализации проекта.

Регламент устного выступления: 7 минут на доклад + 3 минуты ответы на вопросы экспертов.

ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «2023_ГХ», где Х – номер рабочего места. Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку (презентация проекта).

2. СПЕЦИАЛЬНЫЕ ПРАВИЛА КОМПЕТЕНЦИИ²

2.1. Личный инструмент конкурсанта

Список материалов, оборудования и инструментов, которые конкурсант может или должен привезти с собой на соревнование.

Нулевой - нельзя ничего привозить.

2.2. Материалы, оборудование и инструменты, запрещенные на площадке

Список материалов, оборудования и инструментов, которые запрещены на соревнованиях по различным причинам.

Конкурсантам запрещено использовать мобильные устройства связи, персональные компьютеры и иную личную технику на конкурсной площадке в дни с С-1 до С3 во время нахождения на конкурсной площадке. В случае нарушения данного требования Конкурсант в первый раз получает предупреждение, в следующий раз отстраняется от выполнения текущего Модуля.

2.3. Дополнительно

Вопрос	Специальные правила
Сохранение решения конкурсантами	<ul style="list-style-type: none">Решение должно быть сохранено в указанную в Конкурсном задании директорию, в случае нарушения инструкций решение не проверяется.
Временной регламент	<ul style="list-style-type: none">В случае выполнения участником модуля Конкурсного задания в непредназначенное для этого время, в том числе во время выполнения другого модуля, такое решение не будет оценено.За не прекращение выполнения Конкурсного задания по команде «СТОП» Главного эксперта или ответственных за хронометраж Экспертов в соответствии с временем выполнения Модуля или в иных ситуациях к Конкурсантам применяются штрафные санкции в виде снижения общей суммы баллов на 1 балл.
Технические средства – физические носители данных	<ul style="list-style-type: none">Конкурсантам разрешается использовать только физические носители, предоставляемые организатором чемпионата. Запрещается использовать любые другие носители в устройствах Конкурсантов.

² Указываются особенности компетенции, которые относятся ко всем возрастным категориям и чемпионатным линейкам без исключения.

	<ul style="list-style-type: none"> • Нельзя выносить за пределы рабочей площадки физические носители или любые другие портативные устройства хранения данных. • Физические носители данных должны предъявляться главному эксперту в конце каждого дня для безопасного хранения, их нельзя выносить за пределы рабочей площадки. • При выявленном нарушении работа модуля не оценивается.
Технические средства — персональные устройства для фото- и видеосъемки	<ul style="list-style-type: none"> • Экспертам и переводчикам разрешается использовать на рабочей площадке персональные устройства для фото- и видеосъемки, при этом запрещена фиксация информации о Конкурсном задании, критериях и ведомостях оценки, деталях выполнения задания Конкурсантами.
Программное обеспечение	<ul style="list-style-type: none"> • Конкурсанты могут создавать программные продукты, оформлять инструкции или делать заметки, находясь на рабочей площадке, однако их никогда нельзя забирать с рабочей площадки. • За использование материалов, файлов, подготовленных вне конкурсного времени и за пределами конкурсной площадки, в том числе шпаргалок, материалов полученных в сети Интернет (если иное не указано в Конкурсном задании), выполняемый модуль Конкурсного задания не оценивается.
Отказ оборудования	<ul style="list-style-type: none"> • Если имеется явное доказательство того, что конкурсанты сами причинили ущерб оборудованию, им не будет предоставляться замена и дополнительное время.
Контроль за конкурсантами	<ul style="list-style-type: none"> • Конкурсантов необходимо постоянно контролировать во время их работы. Эксперты, в чьи обязанности входит контроль, должны принять меры для того, чтобы их заменил другой эксперт, если им необходимо уйти. • Экспертам не разрешается контролировать своего конкурсанта-компатриота.
Поведение конкурсантов	<ul style="list-style-type: none"> • За использование ненормативной лексики устно во время выполнения Конкурсного задания или во время защиты своих работ, а также письменно в представленных к проверке файлах к Конкурсанту применяются штрафные санкции в виде снижения общей суммы баллов на 5 баллов. • Общение участников во время выполнения Конкурсного задания запрещено, в случае неоднократного нарушения запрета участники дисквалифицируются на оставшееся время конкурсного дня.

3. Приложения

Приложение №1 [Инструкция по заполнению матрицы конкурсного задания](#)

Приложение №2 [Матрица конкурсного задания](#)

Приложение №3 Критерии оценки

Приложение №4 [Инструкция по охране труда и технике безопасности по компетенции «Разработка решений с использованием блокчейн технологий».](#)