Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКАЯ РАБОТА №1

ПО ОП.04 «Основы алгоритмизации и программирования»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-3-22  Кирилов Дмитрий Сергеевич | Проверил преподаватель  \_\_\_\_\_\_\_\_\_Т.Д. Артамонова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2023 года |

Москва 2023

Цель: создать смарт-контракт для управления финансами, который позволяет пользователям отправлять эфиры и получать информацию о балансе.

* Создайте структуру, содержащую поля, такие как address userAddress и uint256 balance (изначально у вас баланс 0, так как никто из пользователей пока что не закинул денежку на смарт-контракт).
* Используйте массив или маппинг (словарь, почти такой же как в питоне. Но так как мы с вами это еще не успели пройти, вы можете использовать массив, но если вам интересно, можете глянуть, что такое маппинг) для хранения информации о пользователях в контракте.
* Реализуйте функцию, которая позволяет пользователям отправлять эфиры на контракт. (количество переведенных эфиров записывается в баланс пользователя. Если вы переводите второй раз, то баланс у вас обновляется, например, отправили вы первый раз 4 эфира, потом вы отправили еще 9, соответственно баланс у вас 13 эфиров)
* Реализуйте функцию getBalance(address user) с использованием модификатора view, которая возвращает баланс конкретного пользователя.
* Добавьте событие, которое регистрирует каждое пополнение баланса.
* Реализуйте функцию для вывода средств из смарт-контракта с проверкой наличия достаточного баланса для вывода.
* С помощью require сделайте проверки, чтобы пользователи могли переводить значения больше 0.

Ход работы:

**1 этап:** создание конструктора и поля с маппингом

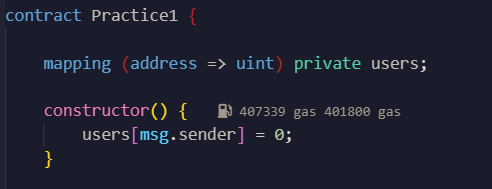


Рисунок 1. создание конструктора и поля с маппингом

**2 этап:** создание функции для пополнения счёта

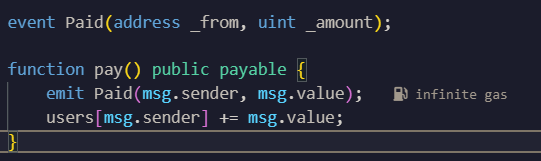


Рисунок 2. создание функции для пополнения счёта

**3 этап:** создание функции для перевода денег на счета

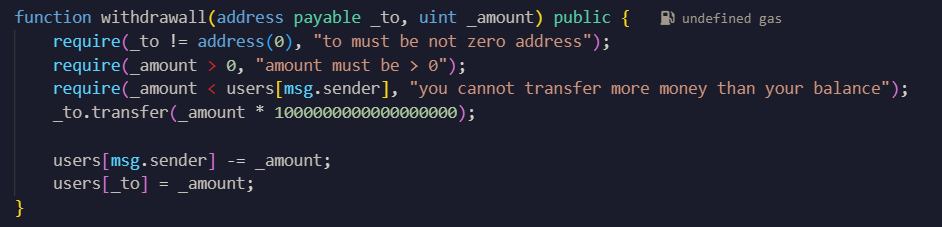


Рисунок 3. создание функции для перевода денег на счета

**4 этап:** создание функции для просмотра баланса счёта, на который были переведены деньги

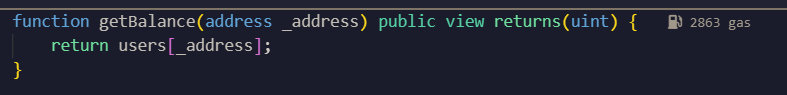


Рисунок 4. создание функции для просмотра баланса счёта, на который были переведены деньги

**5 этап:** в задании было сказано написать простенькую структуру. Но она нужно только для тех случаев, когда вместо маппинга используется массив. В общем я написал её, но нигде не использовал

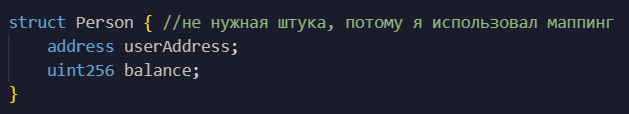


Рисунок 5. Файл с нужными функциями

Вывод: было создан смарт-контракт для управления финансами, который позволяет пользователям отправлять эфиры и получать информацию о балансе.

* Была создана структура, содержащая поля, такие как address userAddress и uint256 balance (изначально у вас баланс 0, так как никто из пользователей пока что не закинул денежку на смарт-контракт).
* Был использован маппинг для хранения информации о пользователях в контракте.
* Была реализована функция, которая позволяет пользователям отправлять эфиры на контракт.
* Была реализована функция getBalance(address user) с использованием модификатора view, которая возвращает баланс конкретного пользователя.
* Было добавлено событие, которое регистрирует каждое пополнение баланса.
* Была реализована функция для вывода средств из смарт-контракта с проверкой наличия достаточного баланса для вывода.
* С помощью require были сделаны проверки, чтобы пользователи могли переводить значения больше 0.