КОНКУРСНОЕ ЗАДАНИЕ
категория «Школьники»

Компетенция: blokcheyn-tehnologii-r

Выдержка из официального конкурсного задания, опубликованного на abilympics-russia.ru (раздел «Компетенции 2026»).
Страницы источника, относящиеся к категории, приведены далее в исходном виде.

Собрано автоматически: Региональный центр Абилимпикс республики Башкортостан.

2. Конкурсное задание

2.1. Краткое описание задания

2.1.1. Категория участников «ШКОЛЬНИКИ»: в ходе выполнения конкурсного задания участникам необходимо создать смарт-контракт для системы «Безопасное дорожное движение», который позволяет регистрировать автомобили, а также своевременно уплачивать штрафы при нарушении правил дорожного движения (ПДД), а также опубликовать контракт в сети. Также необходимо выполнить анализ контракта на наличие ошибок.

2.1.2. Категория участников «СТУДЕНТЫ»: в ходе выполнения конкурсного задания участникам необходимо создать смарт-контракт для системы «Безопасное дорожное движение», который позволяет регистрировать автомобили, своевременно уплачивать штрафы при нарушении правил дорожного движения (ПДД), а также продлевать срок действия водительского удостоверения. Создать приватный блокчейн Ethereum с помощью geth и задеплоить контракт. Также необходимо выполнить анализ контракта на наличие ошибок.

2.1.3. Категория участников «СПЕЦИАЛИСТЫ»: в ходе выполнения конкурсного задания участникам необходимо создать смарт-контракт для системы «Безопасное дорожное движение», который позволяет регистрировать автомобили, своевременно уплачивать штрафы при нарушении правил дорожного движения (ПДД), а также продлевать срок действия водительского удостоверения. Создать приватный блокчейн Ethereum с помощью geth и задеплоить контракт. Код разработанного контракта необходимо задокументировать и покрыть тестами. Также необходимо выполнить анализ контракта на наличие ошибок.

2.2. Структура и подробное описание конкурсного задания

Категория участников	Наименование и описание модуля	Время	Результат
Школьники	**Модуль А.** Аналитика	1 час	Составленные отчеты по исправлению ошибок в контракте
	Модуль Б. Реализация смарт-контракта	1 час 30 мин	Разработанный и опубликованный сети контракт
	Модуль В. Презентация	30 мин	Видео с демонстрацией работоспособности разработанной системы
Время выполнение всех модулей: **3 часа**			
Студенты	**Модуль А.** Аналитика	1 час	Составленные отчеты по исправлению ошибок в контракте
	Модуль Б. Реализация смарт-контракта	1 час 30 мин	Разработанный и опубликованный сети контракт
	Модуль В. Презентация	30 мин	Видео с демонстрацией работоспособности разработанной системы
Время выполнение всех модулей: **3часа**			
Специалисты	**Модуль А.** Аналитика	1 час	Составленные отчеты по исправлению ошибок в контракте
	Модуль Б. Реализация смарт-контракта	1 час 30 мин	Разработанный и опубликованный сети контракт
	Модуль В. Презентация	30 мин	Видео с демонстрацией работоспособности разработанной системы

Время выполнение всех модулей: **3 часа**

2.3 Последовательность выполнения задания.
2.3.1. Категория участников «ШКОЛЬНИКИ»:
Модуль А. Аналитика

В приведенном ниже коде смарт-контракта:

1. Исправьте синтаксические ошибки.
2. Составьте отчет с указанием, в какой строке и какое именно изменение вы внесли.
3. Опубликуйте контракт в сети.
4. В ходе тестирования исправьте логические ошибки в контракте.
5. Предположите и опишите сферу использования данного смарт-контракта.

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] == 0x0000000000000000000000000000000000000000,
        "_____");
14          login[login] = msg.sender;
15          users[msg.sender] = User(FIO, msg.sender.balance, login);
16      }
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.transfering(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  }
```

Модуль Б. Реализация смарт-контракта

В рамках конкурсного задания необходимо создать смарт-контракт для системы «Безопасное дорожное движение», который позволяет регистрировать автомобили, а также своевременно уплачивать штрафы при нарушении правил дорожного движения (ПДД).

В систему Безопасного дорожного движения вовлечены следующие ***субъекты***: водитель, сотрудник дорожно-патрульной службы (ДПС), государственный банк.

Объектом системы является транспортное средство.

Характеристики транспортного средства:

- Категория (A, B, C);
- Рыночная стоимость (Pc, eth);
- Срок эксплуатации (Сэ, лет);

Характеристики водителя:

- ФИО;
- Водительское удостоверение со всеми параметрами (номер, срок действия, категория транспортного средства);
- Год начала водительского стажа (для определения стажа; например, если стаж начался в 2001 году, то на 2025 год стаж будет равен 24 года);
- Количество ДТП;
- Количество неоплаченных штрафов;
- Текущий баланс.

Банк в рамках задания – это аккаунт в блокчейн-цепи, который задается в генезис-блоке. При старте системы на счету данного аккаунта должно быть 1000 eth. На счет данного аккаунта поступают все штрафы, оплачиваемые водителями.

В системе есть две основные роли пользователей: водитель и сотрудник ДПС. У каждого из них есть свой личный кабинет. Сотрудник ДПС может также являться водителем.

Функционал для каждой из ролей в системе:

Водитель:

- добавляет в своем личном кабинете данные водительского удостоверения: номер, срок действия, категория транспортного средства (все водительские удостоверения в рамках текущего конкурсного задания перечислены в табл. 1);
- отправляет запрос на регистрацию транспортного средства;
- создает запрос на продление срока действия водительского удостоверения;
- оплата штрафа.

Сотрудник ДПС:

- подтверждает данные водительского удостоверения (функция может быть реализована автоматически);
- регистрирует транспортное средство (функция может быть реализована автоматически);
- продлевает срок прав (функция может быть реализована автоматически);
- выписывает штраф;

Таблица 1 - Водительские удостоверения в рамках текущего конкурсного задания

Номер	Срок действия	Категория
000	11.01.2024	A
111	12.05.2028	B
222	09.09.2025	C
333	13.02.2030	A
444	10.05.2025	B
555	24.06.2032	C
666	31.03.2023	A

Пояснение функционала.

При старте в системе должны быть зарегистрированы:

✓ аккаунт Банка (1000 eth на счету);

✓ 1 сотрудник ДПС он же Водитель 1 Иванов Иван Иванович (без указания водительского удостоверения и транспортного средства, водительский стаж 2 года, 0 ДТП, 0 неоплаченных штрафов, на счету 50 eth).

✓ Водитель 2 Семенов Семен Семенович (без указания водительского удостоверения и транспортного средства, водительский стаж 5 лет, 0 ДТП, 0 неоплаченных штрафов, на счету 50 eth).

✓ Водитель 3 Петров Петр Петрович (без указания водительского удостоверения и транспортного средства, водительский стаж 10 лет, 3 ДТП, 0 неоплаченных штрафов, на счету 50 eth).

Остальные пользователи системы должны иметь возможность регистрироваться в системе самостоятельно.

Добавление водительского удостоверения

Когда водитель добавляет данные водительского удостоверения, запрос на рассмотрение отправляется в службу ДПС. В службе ДПС производится проверка, что данное водительское удостоверение есть в базе и все его поля заполнены правильно. После этого происходит подтверждение водительского удостоверения. Если при проверке возникает ошибка, то водителю приходит соответствующее сообщение об отказе. Предпочтительно выполнение автоматической проверки.

Регистрация транспортного средства

При запросе на регистрацию транспортного средства, происходит проверка соответствия категории регистрируемого транспортного средства категории транспортного средства в водительском удостоверении. Регистрация производится, если категории совпадают. Если при проверке возникает ошибка, то водителю приходит соответствующее сообщение об отказе. Предпочтительно выполнение автоматической проверки.

Продление срока действия водительского удостоверения.

При запросе на продление срока действия водительского удостоверения, происходит проверка двух параметров:

- ✓ текущая дата должна быть не больше, чем за месяц до окончания срока действия удостоверения;
- ✓ у водителя должны быть оплачены все штрафы.

Продление производится, если оба условия выполнены. Если при проверке возникает ошибка, то водителю приходит соответствующее сообщение об отказе. Предпочтительно выполнение автоматической проверки.

Оплата штрафа

Водитель имеет возможность оплатить выписанные ему штрафы. Стоимость одного штрафа 10 eth. Если оплата штрафа производится в первые 5 дней, то стоимость штрафа снижается в 2 раза и составляет 5 eth. **ВАЖНО!!!!** 1 день в задании = 5 секунд реального времени.

Оформление штрафа

Сотрудник ДПС может выписать штраф водителю. Штраф выписывается по номеру водительского удостоверения.

Код разработанного решения должен быть задокументирован.

Модуль В. Презентация (вариативный)

В рамках данного модуля необходимо продемонстрировать работоспособность разработанного контракта. Демонстрация работы контракта осуществляется путем видеозахвата экрана и комментариев разработчиков. При демонстрации необходимо показывать изменения состояния блокчейн-системы (это можно сделать любым доступным способом, например, с использованием Remix, Ganache, Geth).

Для демонстрации решения необходимо выполнить действия из табл. 2.

Таблица 2 – Порядок действий при демонстрации решения

Действия	Действия
1. Зайти в аккаунт Банк, проверить баланс.	2. Зайти в аккаунт Иванова Ивана Ивановича, проверить баланс и данные водителя.
3. Зарегистрировать нового водителя Романов Роман Романович (без указания водительского удостоверения и транспортного средства, водительский стаж 6 лет, 1 ДТП, 0 неоплаченных штрафов, на счету 1000 eth).	4. Зарегистрировать нового водителя Дмитриев Дмитрий Дмитриевич (без указания водительского удостоверения и транспортного средства, водительский стаж 3 года, 1 ДТП, 0 неоплаченных штрафов, на счету 1000 eth).
5. Зайти в аккаунт Дмитриева Дмитрия Дмитриевича. Добавить ему удостоверение 111. Показать изменение информации.	6. Зайти в аккаунт Семенова Семена Семеновича. Добавить ему удостоверение 222. Показать изменение информации.
7. Зайти в аккаунт Петрова Петра Петровича. Добавить ему удостоверение 222. Показать изменение информации. Показать сообщения в личном кабинете.	8. Зайти в аккаунт Романова Романа Романовича. Добавить ему удостоверение 111. Показать изменение информации. Показать сообщения в личном кабинете.

9. Зайти в аккаунт Романова Романа Романовича. Добавить ему удостоверение 333. Показать изменение информации.	10. Зайти в аккаунт Петрова Петра Петровича. Добавить ему удостоверение 444. Показать изменение информации.
11. Зайти в аккаунт Петрова Петра Петровича. Добавить ему транспортное средство категории B. Показать изменение информации.	12. Зайти в аккаунт Дмитриева Дмитрия Дмитриевича. Добавить ему транспортное средство категории B. Показать изменение информации.
13. Зайти в аккаунт Семенова Семена Семеновича. Добавить ему категорию A. Показать изменение информации. Показать сообщения в личном кабинете.	14. Зайти в аккаунт Романова Романа Романовича. Добавить ему категорию C. Показать изменение информации. Показать сообщения в личном кабинете.
15. Зайти в аккаунт ДПС. Выписать штраф по номеру водительского удостоверения 111.	16. Зайти в аккаунт ДПС. Выписать штраф по номеру водительского удостоверения 444.
17. Зайти в аккаунт Петрова Петра Петровича. Оплатить штраф. Показать изменение информации в личном кабинете.	18. Зайти в аккаунт Банк, проверить баланс

По желанию можно продемонстрировать дополнительный функционал.

По итогу выполнения демонстрации заполнить табл. 3 соответствия адресов и пользователей системы

Таблица 3 – Соответствие аккаунтов и адресов в сети Ethereum

Пользователь	**Адрес в сети Ethereum**
Банк	
Иванов Иван Иванович	
Семенов Семен Семенович	
Петров Петр Петрович	
Романов Роман Романович	
Дмитриев Дмитрий Дмитриевич	

Должна быть дана инструкция по запуску и использованию решения.

2.3.2. Категория участников «СТУДЕНТЫ»:

Модуль А. Аналитика

В приведенном ниже коде смарт-контракта:

1. Исправьте синтаксические ошибки.
2. Составьте отчет с указанием, в какой строке и какое именно изменение вы внесли.
3. Опубликуйте контракт в сети.
4. В ходе тестирования исправьте логические ошибки в контракте.
5. Предположите и опишите сферу использования данного смарт-контракта.

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: