Рабочая программа общеуниверситетского факультатива «Введение в блокчейн и разработка на Solidity»

(разработана BANKEX Foundation, Москва 2018 год)

Учебный год: 2018/2019
Статус: факультатив
Преподаватель: Сенотов Дмитрий Игоревич (читает лекции, ведет семинары и принимает экзамены/зачеты)
Кто читает: BANKEX Foundation
Где читается: Факультет компьютерных наук
Язык: русский
Уровень: бакалавриат
Когда читается: 3, 4 модули
Кредитов: 2

1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает требования к образовательным результатам и результатам обучения студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину «***Введение в блокчейн и разработка на Solidity***», учебных ассистентов и студентов всех направлений, изучающих данную дисциплину.

2. Место дисциплины в структуре образовательной программы

Настоящая дисциплина является факультативной.

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

- опыт программирования (на одном из языков, например, JavaScript, Python, C++);

- владение английским языком на базовом уровне.

Программа курса предусматривает лекции (12 часов), семинары (10 часов) и практические занятия (10 часов).

3. Цели освоения дисциплины

Цель курса – обучить студентов основам создания децентрализованных приложений на блокчейне Эфириума с помощью языка программирования Solidity.

В результате изучения дисциплины «*Введение в блокчейн и разработка на Solidity*» студенты должны:

- знать основы теоретической составляющей технологии блокчейн;
- понимать фундаментальные идеи и отличия блокчейнов Биткоина и Эфириума;
- уметь разрабатывать смарт-контракты на языке Solidity и тестировать их.

4. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Выполнять самостоятельно теоретические задания, направленные на понимание различных составляющих технологии блокчейн, таких как цифровые подписи и криптографические хеш-функции;
- Выполнять сасомостоятельно практические задания по программной реализации смарт контрактов с помощью языка программирования Solidity.

5. Тематический план учебной дисциплины

N	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа (дз)
			Лекции	Семинары	Практи-ческие занятия	
1.	Базовая теория блокчейна		4	2		6
2.	Платформа Ethereum		4	2		10
3.	Язык Solidity в среде Remix		2	3	5	14
4.	Библиотека web3.js		2	3	5	14
		76	12	10	10	44

6. Формы контроля и структура итоговой оценки

- Текущий контроль: письменная аудиторная контрольная работа на теорию (40 минут), практическая аудиторная контрольная работа по написанию кода (60 минут) и 8 домашних заданий;
- Итоговый контроль – письменный экзамен.

Оценки по всем формам контроля выставляются по 10-ти балльной шкале. Способ округления накопленной оценки текущего контроля и экзамена: в пользу студента.

Формирование оценки: текущая оценка работы студентов учитывает активность на семинарских и практических занятиях (Оаудиторная), правильность решения домашних работ (Од/з), результаты контрольных работ (Ок/р). В диплом выставляется оценка за итоговый контроль (Оитоговая), которая является результирующей оценкой по учебной дисциплине.

Результирующая оценка за итоговый контроль в форме экзамена выставляется по следующей формуле, где О(экзамен) – оценка за работу, выполненную на письменном экзамене:

$$О(итоговая) = 0,5 \cdot О(экзамен) + 0,2 \cdot О(к/р) + 0,2 \cdot О(д/з) + 0,1 \cdot О(аудиторная)$$

Накопленная оценка по 10-ти балльной шкале, состоящая из О(аудиторная) + О(к/р) + О(д/з), определяется перед итоговым контролем и озвучивается студентам не позднее, чем за 2 дня до экзамена.

Примеры тем домашних заданий:

- Описать одну из проблем блокчейн-индустрии и ее возможные решения;
- Написать смарт-контракт для хранения информации с возможностью разрешать ее просмотр другому пользователю;
- Описать задачу консенсуса и разные подходы к ее решению.

Примеры экзаменационных вопросов:

- Написать смарт-контракт простой лотереи (люди платят за участие, создатель контракта может вызвать функцию рандомного выбора победителя) и тесты к нему;
- Найти уязвимость в смарт-контракте токена ERC-20 с дополнительным функционалом.

7. Содержание дисциплины

Тема 1. Базовая теория блокчейна

Введение: Биткоин. Централизованный реестр. Цифровые подписи. Временные отметки. Система utxo. Децентрализованный реестр. P2P-сети. Как достичь консенсуса. Хеш-функции. Proof of work. Проблема двойных трат. Блоки и цепочки блоков. Дерево Меркла. Сложность майнинга. Награда за создание блока. Комиссии за транзакции.

Тема 2. Платформа Ethereum

Основные различия Эфириума и Биткоина. Отличие системы utxo от балансов.

Базовая теория Эфириума. Виды узлов. Транзакции. Газ.

Пользовательский аккаунт. Metamask. Основная сеть, тестовые сети. Faucet.

Теория смарт-контрактов. Аккаунт смарт-контракта. Газ в смарт-контрактах. Создание контракта. Языки для написания смарт-контрактов (Solidity). Oracles. Bytecode, OPcode, ABI.

Виртуальная машина Эфириума (EVM). Различные способы хранения данных. Stack-machine.

Тема 3. Solidity in Remix

Remix - онлайн среда разработки для Solidity.

Основы Solidity. Version pragma, import, комментарии. Переменные состояния. Основные типы. Конструкторы. Функции, типы функций. Настройки Remix.

Выпуск смарт-контрактов в Remix. Вызов функций. Повторный запуск контракта. Разные виды вызова выполнения функций (вызов, отправка транзакций).

Подробности Solidity. Типы (struct, enum, mapping). Модификаторы view и pure. Видимость функций. Модификатор payable, fallback функции.

Продвинутые смарт-контракты. Свойства блока и транзакций. Обработка ошибок (assert, require, revert и exceptions). Модификаторы функций. Наследование, интерфейсы. События. Библиотеки. Calls, delegated calls.

Разбор существующих имплементаций.

Тема 4. Библиотека web3.js

Среды разработки смарт-контрактов. Настройка среды. Структура проекта. Truffle framework. Тестовые фреймворки. Генерация ключей. Подпись/отправка транзакций. Библиотека web3. Компиляция. Тестирование. Mocha. Запуск контрактов с web3. Запуск контрактов с infura. Пример приложения.

8. Учебно-методическое и информационное обеспечение дисциплины

Основная литература:

1. *Прасти Н.,* Блокчейн. Разработка приложений. Разработка децентрализованных приложений в реальном времени на платформе Ethereum. СПб.: «БХВ-Петербург», 2018.
2. *Antonopoulos A. M.* Mastering Bitcoin: Programming the open blockchain. – "O'Reilly Media, Inc.", 2017.
3. *Antonopoulos A. M., Wood G.* Mastering Ethereum – 2018. URL: https://github.com/ethereumbook/ethereumbook (accessed 29 October 2018)
4. *Buterin V. et al.* A next-generation smart contract and decentralized application platform //white paper. – 2014. URL: https://github.com/ethereum/wiki/wiki/White-Paper (accessed 29 October 2018)
5. *Nakamoto S.* Bitcoin: A Peer-to-Peer Electronic Cash System. – 2008. URL: https://bitcoin.org/bitcoin.pdf (accessed 29 October 2018)
6. *Narayanan A.* et al. Bitcoin and cryptocurrency technologies: a comprehensive introduction. – Princeton University Press, 2016.
7. *Wood G.* Ethereum: A secure decentralised generalised transaction ledger //Ethereum project yellow paper. – 2014. – Т. 151. – С. 1-32. URL: https://ethereum.github.io/yellowpaper/paper.pdf (accessed 29 October 2018)

9. Программные средства

Освоение дисциплины требует ноутбук с любой ОС, где установлен браузер (желательно Chrome / Chromium), Node.JS, NPM, Truffle, ganache, а также текстовый редактор или IDE (рекомендуем Visual Studio Code или WebStorm).

10. Материально-техническое обеспечение дисциплины

Аудитории для лекционных и практических занятий должны быть оснащены беспроводным доступом в интернет и оборудованы индивидуальными рабочими местами для студентов и преподавателя, иметь экран и проектор для показа презентаций, маркерную доску и маркеры.