## Hunger Games



Every year, in the ruins of what was once North America, the **Capitol** of the nation of **Panem** - a technologically advanced, utopian city where the nation's most wealthy and powerful citizens live, forces each of its 12 districts to send a teenage boy and a girl, between the ages of 12 and 18, to compete in the **Hunger Games**: a nationally televised event in which 'tributes' fight each other within an arena, until one survivor remains.

This time of the year has come and it’s time for the 100th hunger game where you should send the new pair of teenage boy and girl.

Create a **Capitol** contract which:

* **adds** person by **age** and **gender** (hint: use **struct** for storing the person)
* **chooses** one girl and one boy:
* you are not allowed to choose the two people from one gender
* they should be between **12** and **18** years old
* they should be chosen by **random** function (you can use **block.timestamp** but it is not safe or **oraclize** -> learn more about it from oraclize documentation)
* you can **check how many girls and boys** are added -> returns a **positive number**
* after choosing the pair (boy and girl) set the **start date** of the hunger games and the **end date** (the hunger games should **last 5** minutes)
* after the **end** of the hunger game, **check** if the boy and girl are alive (use **random** 0 - dead, 1 - alive, use **modifier** for checking if the hunger game ended)

Use **modifiers** where it is appropriate.

Add appropriate **Events** for the functions

Test and play around with the contract!

## Pet Adoption



In our adoption sanctuary, we have 5 kinds of animals

Fish  
Cat   
Dog  
Rabbit  
Parrot

The sanctuary **will store** information for all the people who have adopted a pet.

After creating the contract, we should add the owner of the adoption sanctuary.

* Create a contract called **PetSanctuary**:
  + function **add** (animalKind, howManyPieces) – **only** the owner can give shelter to animals in the sanctuary
  + function **buy** (personAge, personGender, animalKind)
* save when the animal is bought
* you can only adopt **one** animal for lifetime
* **Men** can only buy dog and fish
* **Women** can buy from every kind, but if they are under 40, they are not allowed to buy a cat
* function **giveBackAnimal**(animalKind)
* you can give the animal back in the first 5 minutes after adoption

Think about how to store people’s information in the contract.

Use **modifiers** where it is appropriate.

Use **events** to notify everyone.

## Vodka Crowdsale



There are less than 4 months until the **Football World Cup 2018** starts. It will take place in **Russia** and you know what Russians drink?! Yeah, exactly - **Vodkaaa**!

But there is a problem - they do not have money to buy vodka. So, they have heard about **ICOs** (Crowdsales) and want to make one to raise money for Vodkaaa. The problem is no one knows how to write Smart Contracts, so they gave it to you to write it.

* Create a **VodkaToken** contract which will be an ERC20Basic Token Standard:

|  |
| --- |
| contract **ERC20Basic** {  function **totalSupply()** public view returns (uint256);  function **balanceOf(address who)** public view returns (uint256);  function **transfer(address to, uint256 value)** public returns (bool);  event **Transfer(address indexed from, address indexed to, uint256 value)**; } |

* name = "**VodkaToken**";
* symbol = "**VODKA**";
* decimals = **18**
* initialSupply = **1 000 000** tokens with 18 decimals
* Create a **VodkaCrowdsale** contract which will:
* takes the **startTime**, **endTime**, **rateOfTokensToGivePerEth**, **walletToStoreTheEthers** and **vodkaToken** as a reward
* store how much **weiAmout** is contributed as a persistent variable
* function **buyTokens()** -> **payable**
  + check if the purchase is valid **within the period**
  + you are not allowed to buy zero tokens
  + it should calculate the tokens buy the **rateOfTokensToGivePerEth** and **wei** **contributed** and **transfer** them to the balance of the contributor
  + finally **forward the weiAmount to the wallet**
* function **hasEnded()**
  + to check if the crowdsale has ended -> returns **bool**
* Use **modifiers** where it is appropriate
* Add appropriate **Events** for the functions
* Test and play around with the contract