**Framework**

Какие цели преследует Framework?

- Повышение скорости разработки

- Повышение стабильности кода

**- Стандартизация написания кода**

**Особенности реализации Framework’а**

В абсолютном большинстве framework’ов весь код приложения разделяется на максимально обособленные не связанные между собой модули(компоненты). Особенность в том, что он использует только эти компоненты. Компоненты могут быть вложены друг в друга и даже более того само приложение является одной большой компонентой.

Компонента отвечает:

- Формирование разметки части страницы

- Обработку событий элементов на этой части страницы

- Сохранность данных

- Обработка части страницы

Приложение состоит из компонентов. Центральная компонента, называется app и лежит в папке js. Бизнес-логика отдельно (файл modules). Компонента должны быть “тупой”.

**Template.js**

В функцию Template записываются все шаблоны компонентов, шаблоны страниц записываются в прототип функции Template.

function Template(){

}

**Component.js**

class Component {

constructor({id, parent, template, templateParams=null, callbacks={}, className}){

this.id = id;

this.parent = parent;

this.callbacks = callbacks;

this.render(template(templateParams), className);

this.addEventListeners();

}

show(){

document.getElementById(this.id).classList.remove(‘hide’);

}

hide(){

document.getElementById(this.id).classList.add(‘hide’);

}

style.css

.hide{display:none;}

render(template, className){

const elem = document.createElement(‘div’);

elem.setAttribute(‘id’, this.id);

if (className) {

elem.classList.add(className);

}

elem.innerHTML = template;

if (this.parent){

document.getElementById(this.parent).appendChild(elem);

}

else

document.querySelector(‘body’).appendChild(elem);

}

}

addEventListeners(){}

Методы show, hide и render нельзя переопределять. Метод adddEventListeners переопределять нужно.

index.js

const template = new Template;

window.onload = function(){

new App({id:’app’,template:()=>’’});

}

App.js

class App extends Component {

constructor(props){

super(props);

this.header = new Header({id:’header’, parent: this.id, template: template.headerTemplate, callbacks:{ showPage:(name)=>this.showPage(name)}

});

}

showPage(name){

if(this[name]?.show){

this[name].show();

}

}

}

headerTemplate.js

Template.prototype.headerTemplate=()=>`

<button id=’show2D’>2D</button>

<button id=’showCalc’>Calc</button>

…

`;

Header.js

class Header extends Component{

addEventListeners(){

document.getElementById(‘show2D’).addEventListener(‘click’,()=>this.callbacks.showPage(‘graph2D’));

document.getElementById(‘showCalc’).addEventListener(‘click’,()=>this.callbacks.showPage(‘calc’));

}

}

В app необходимо создать 2 пустые компоненты graph2D и calc.