



Утвърдил:
/ /

Утвърден от Факултетен съвет
с протокол № /

СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”

Факултет по Математика и Информатика

Специалност: Информатика, Информационни системи, Компютърни науки,
Математика и информатика, Софтуерно инженерство

--	--	--	--	--	--	--

Курс: 2+
Учебна година: 2017/2018
Семестър: IV

УЧЕБНА ПРОГРАМА

Дисциплина: Мултимедийни технологии /
Multimedia Technologies

Ж	6	1	0
---	---	---	---

 Избираема дисциплина

Преподавател: ас. Траян Илиев

Асистент: ас. Траян Илиев

Учебна заетост	Форма	Хорариум
Аудиторна заетост	Лекции	30
	Семинарни упражнения	
	Практически упражнения (хоспетиране)	30
Обща аудиторна заетост		60
Извънаудиторна заетост	Курсов учебен проект	50
	Доклад/Презентация	10
	Подготовка на домашни работи	10
	Подготовка за финален тестови изпит	10
	Самостоятелна работа с ресурси	10
Обща извънаудиторна заетост		90
ОБЩА ЗАЕТОСТ		150
Кредити аудиторна заетост		2
Кредити извънаудиторна заетост		3
ОБЩО ЕКСТ		5

№	Формиране на оценката по дисциплината	% от оценката
1.	Финален курсов проект	45%
2.	Финален тестови изпит	45%
3.	Домашни работи през семестъра (текущ контрол)	10%
Анотация на учебната дисциплина:		
<p>Целта на курса е да запознае с актуалните тенденции при разработка на мултимедийни/ хипермедийни/ 3D VR уеб приложения и услуги с Angular 5, TypeScript, Redux, REST, GraphQL, Three.js, Whitestorm.js и HTML 5 JS APIs.</p> <p>Участниците ще придобият практически опит при използване на стандартни за индустрията JavaScript библиотеки и платформи – Node.js, Express, Angular, RxJS за разработка на уеб приложения и услуги от край до край (<i>fullstack</i>).</p> <p>Курсът запознава с актуални стандарти и спецификации като <i>Web Components (Custom Tags + HTML Templates + Shadow DOM + Component Linking)</i>, <i>Google Material Design (MD)</i> и други.</p> <p>Включена е практическа разработка на <i>single page application (SPA) TypeScript</i> клиенти с разпространени библиотеки като Angular 5, RxJS, NGRX, Apolo и уеб услуги (<i>REST/ JSON APIs</i>) с Node.js + Express + MongoDB, както под формата на примерни проекти, така и като част от курсовите проекти реализирани екипно през семестъра.</p> <p>Участниците ще имат възможност да приложат на практика някои от наложилите се методи за проектиране на <i>изживяването на потребителя (User Experience Design– UXD)</i>.</p> <p>Преобладаващата част от използваните материали са достъпни в електронна форма на английски език в Интернет. Кодът ще бъде достъпен в <i>GitHub</i> под отворен лиценз.</p>		
Предварителни изисквания:		
<p>Очаква се участниците да могат да боравят свободно с технически английски език. Необходимо е добро познаване на стандартните уеб дизайн технологии – HTML, CSS и JavaScript, както и известен практически опит при разработка на уеб приложения с някои от разпространените JS библиотеки.</p>		

Очаквани резултати:

По време на курса студентите ще придобият знания и умения за:

- *обектно-ориентиран JavaScript* и шаблони (*JS design patterns*) и уеб интерактивност;
- *новости в ECMAScript 6, 7 и 8* – класове, конструктори, *let* и *var*, ламбда функции, ES6 модули, *destructuring*, *Rest/Spread*, *Promises*, *async/await*, *import()* и др.;
- *TypeScript* – функции, интерфейси, класове, функционални интерфейси, ламбди, *enums*, модули, *generics*, *tsconfig.json*, *typings*, опции на компилатора;
- конфигуриране и използване на билд инструменти – *npm*, *webpack*;
- цялостна методология (*Lean UX*) за проектиране на изживяването на потребителя (*User Experience Design – UXD*);
- *Single Page Applications (SPA)* с *Angular 5* и *TypeScript - MVVM*, *Angular Command Line Interface (CLI)*, свързване (*binding*), интерполация и вградени директиви, реактивни форми и валидация, услуги, рутиране и гардове, асинхронни операции и реактивно програмиране (*Angular 5 HttpClient* и *RxJS*), *Test Driven Development (TDD)*, анимации;
- *REST (REpresentational State Transfer)* и *HATEOAS (Hypermedia as the Engine of Application State)*;
- разработка на уеб услуги – *REST/JSON APIs* с *Node.js + Express*;
- работа с *NoSQL* база от данни *MongoDB*;
- разработка на уеб графика и анимация с *CSS 3* и *Canvas*;
- вграждане на *аудио* и *видео* в уеб приложения, създаване на собствен видео плейър с *HTML 5* и *JS*;
- разработка на интерактивни мултимедийни приложения с *HTML 5 APIs – Audio, Video, Canvas, Camera & Microphone, Geolocation & Maps, Local Storage, History, File API, WebSocket*;
- разработка на цялостни (*fullstack*) разпределени хипермедийни приложения (*mashups*) – *Angular 5* клиент + *REST/JSON APIs* + *MongoDB*;
- 3D графика, виртуална реалност и уеб-базирани мултиплейър компютърни игри с *Angular 5, Three.js, Whitestorm.js* и *Physi.js*.

Студентите ще изградят умения за целенасочено търсене, анализиране и използване на информация, както и умения за работа в екип по избрана тема.

Учебно съдържание

№	Тема:	Хорариум
1.	Представяне на курса, въведение в областта. Видове медии. Основни термини и понятия. Хипертекст и хипермедия. Софтуерен архитектурен стил <i>REST (REpresentational State Transfer)</i> и <i>HATEOAS (Hypermedia As The Engine Of Application State)</i> . Разработка от край до край на цялостни, разпределени хипермедийни приложения (<i>Web 2.0 Mashups</i>) – с богат JavaScript клиент, собствени и готови уеб услуги в стил <i>REST</i> и <i>БД</i> .	2
2.	Мултимедия и Интернет – специфика на проектирането на мултимедийни уеб приложения. Проектиране и тестване на изживяването на потребителите – <i>User Experience (UX)</i> . Стартиране на мултимедийен уеб проект – дефиниране на целите, анализ на групите потребители и техните нужди. Дефиниране на сценарии за употреба. Подход на прототипиране. <i>Lean UX</i> методология за разработка на мултимедийни уеб приложения.	2
3.	Обектно-ориентиран <i>JavaScript</i> . Примитивни типове и обекти. Достъп до обектите по референция. Свойства, функции и методи. Използване на <i>this</i> . Методи <i>call</i> , <i>apply</i> и <i>bind</i> . Прототипно наследяване (<i>prototypal inheritance</i>), полиморфизъм и предефиниране на методи (<i>method overriding</i>), класове и конструктори. Класово наследяване и използване на <i>instanceof</i> . Шаблони при реализация на обектно-ориентирани JS приложения (<i>JS design patterns</i>) - <i>Prototype, Constructor, Module, Singleton, Observer, Factory, Mixin, Decorator</i> . Използване на <i>Visual Studio Code</i> .	3
4.	Разработка на <i>растерна, векторна графика</i> и <i>анимация</i> – специфика, техники, технологии, файлови формати, инструментални средства, <i>Web</i> графика и анимация, анимация с <i>CSS 3</i> . Използване на <i><canvas></i> таг и JS програмни интерфейси (<i>API</i>) за визуализация на графика с <i>HTML 5</i> . Събитийно-ориентирано програмиране с <i>JavaScript</i> . Разработка на интерактивни мултимедийни приложения с <i>HTML 5</i> и <i>JavaScript</i> – слушатели на събития, <i>drag-and-drop</i> . Разработка на <i>видео</i> за уеб –	5

	<p>специфика, технологии, файлови формати, инструменти. Вграждане на видео с HTML 5 таг <code><video></code>, създаване на собствен видео плейър с HTML 5 и JS. Разработка на аудио материали за уеб – специфика, технологии, файлови формати, инструменти. Вграждане на звук в мултимедийни уеб приложения с HTML 5 таг <code><audio></code>. Създаване на прости уеб игри чрез комбиниране на анимация, интерактивност и звук с <i>HTML 5</i> и <i>JS</i>.</p>	
5.	<p>Новости в <i>ECMAScript</i> 6, 7 и 8 - класове, конструктори, <i>let</i> и <i>var</i>, ламбда функции (<code>=></code>), <i>destructuring</i>, <i>Rest/Spread</i>, <i>Promises</i>, <i>async/await</i>, <i>import()</i> и др. JavaScript модули и модулни системи - <i>CommonJS</i> и <i>ES 6</i>. Инструменти за изглаждане на уеб приложения (<i>build toolchain</i>) - <i>npm</i>, <i>babel</i>, <i>lite-server</i>, <i>webpack</i>. Конфигуриране на <i>ES6</i> среда за разработка с <i>webpack</i>, <i>webpack-dev-server</i>, <i>hot module replacement</i>, <i>development/production/test</i> конфигурации.</p>	4
6.	<p>Въведение в TypeScript – функции, интерфейси, <i>duck typing</i>, класове и конструктори, <i>public/private</i> свойства, <i>get</i> and <i>set</i> достъп до свойства, <i>static</i> и <i>instance</i> страни, функционални типове и интерфейси, ламбда изрази и използване на <i>this</i>, <i>optional/default/rest</i> параметри, разширяване на интерфейси, <i>enums</i>. Модули в <i>TypeScript</i> – <i>namespaces/modules</i>, използване на <i>import</i>, <i>export</i> и <i>require</i>. <i>Generic</i> параметри на типа – <i>generic</i> функции и класове, <i>generic</i> конструктори, <i>bounded generics</i>. Извеждане на типа (<i>type inference</i>) и съвместимост на типове (<i>type compatibility</i>). Декоратори. Конфигуриране и разгръщане на <i>TypeScript</i> проект – <i>tsconfig.json</i>, <i>typings</i>, опции на компилатора. Инструменти – <i>tsc</i>, <i>npm</i>, <i>webpack</i>.</p>	4
7.	<p><i>Single Page Applications (SPA)</i> с Angular 5 и <i>TypeScript</i> - <i>Angular Command Line Interface (CLI)</i>, уеб компоненти, модули, свързване (<i>binding</i>), интерполация и вградени директиви, <i>pipes</i>, <i>Angular</i> услуги, инжектиране на зависимостите (<i>dependency injection - DI</i>), темплейт-базирани и реактивни форми. Създаване на собствени валидации и мулти-провайдъри. Асинхронни операции и</p>	12

	<p>реактивно програмиране (<i>Angular HttpClient</i> и <i>RxJS</i>). Рутиране – <i>HTML 5 History API</i>, предаване на данни (<i>data</i>), асинхронно резолвиране (<i>resolves</i>), условия за преход (<i>guards</i>), „мързеливо“ зареждане (<i>lazy loading</i>). Анимации. Зони. Детекция на промените (<i>change detection</i>). <i>Angular Material</i>. Разработка на примерно приложение - <i>Test Manager</i>.</p>	
8.	<p><i>Redux</i> архитектура. Предсказуемо реактивно управление на състоянието на приложението – <i>NGRX/store</i>. Изчисляване на производни данни в стил <i>Reselect</i> с <i>RxJS</i>. <i>NGRX/@Effects</i>. <i>NGRX Router</i> интеграция. <i>IndexedDB</i> интеграция. Рефакторинг на примерното приложение – <i>Test Manager NGRX</i>.</p>	4
9.	<p>Типизиран достъп до данните и управление на кеша с <i>GraphQL</i>. <i>GraphQL Schema Definition Language (SDL)</i>. Заявки (<i>queries</i>), мутации (<i>mutations</i>) и обновяване на данните в реално време (<i>subscriptions</i>). Използване на <i>GraphQL</i> в <i>Angular</i> приложения с <i>Apollo</i> клиент. Рефакторинг на приложение – <i>Test Manager GraphQL</i>.</p>	4
10.	<p><i>HTTP</i> протокол и мултимедийни типове (<i>MIME</i> типове). Разработка на уеб услуги (<i>APIs</i>) – <i>JavaScript Object Notation (JSON) APIs</i> на <i>Node.js</i> + <i>Express.js</i> сървърна платформа. Съхраняване и извличане на данните в/от <i>NoSQL</i> база от данни <i>MongoDB</i>. Софтуерен архитектурен стил <i>REST (REpresentational State Transfer)</i> и <i>HATEOAS (Hypermedia as the Engine of Application State)</i>. Шаблони и добри практики при проектиране и реализация на <i>REST/JSON APIs</i>. Практическа реализация на <i>TestManager JSON API</i> с <i>Express.js</i>.</p>	8
11.	<p>Разработка на интерактивни мултимедийни приложения с <i>HTML 5</i> и <i>JavaScript APIs: Audio, Video, Canvas, Camera & Microphone, Geolocation & Maps, Local Storage, History, File API, WebSocket</i>. Разработка на разпределени (<i>fullstack</i>) хипермедийни приложения (<i>Web 2.0 Mashups</i>) – <i>SPA Angular 5</i> клиент + <i>REST / JSON APIs</i> + <i>MongoDB</i>.</p>	4

12.	3D графика и виртуална реалност с <i>Three.js</i> . Интеграция на 3D графика с <i>Three.js</i> в <i>Angular</i> приложения с <i>Whitestorm.js</i> (опростен код, пост ефекти) и <i>Physi.js</i> (физическа симулация). Разработка на уеб-базирани мултиплейър компютърни игри с <i>Angular</i> , <i>Whitestorm.js</i> и <i>Physi.js</i> .	8
-----	---	---

Конспект за изпит

№	Въпрос
1.	Представяне на курса, въведение в областта. Видове медии. Основни термини и понятия. Хипертекст и хипермедия. Софтуерен архитектурен стил <i>REST</i> (<i>REpresentational State Transfer</i>) и <i>HATEOAS</i> (<i>Hypermedia As The Engine Of Application State</i>). Разработка от край до край на цялостни, разпределени хипермедийни приложения (<i>Web 2.0 Mashups</i>) – с богат JavaScript клиент, собствени и готови уеб услуги в стил <i>REST</i> и <i>БД</i> .
2.	Мултимедия и Интернет – специфика на проектирането на мултимедийни уеб приложения. Проектиране и тестване на изживяването на потребителите – <i>User Experience (UX)</i> . Стартиране на мултимедийен уеб проект – дефиниране на целите, анализ на групите потребители и техните нужди. Дефиниране на сценарии за употреба. Подход на прототипиране. <i>Lean UX</i> методология за разработка на мултимедийни уеб приложения.
3.	Обектно-ориентиран <i>JavaScript</i> . Примитивни типове и обекти. Достъп до обектите по референция. Свойства функции и методи. Използване на <i>this</i> . Методи <i>call</i> , <i>apply</i> и <i>bind</i> . Прототипно наследяване (<i>prototypal inheritance</i>), полиморфизъм и предефиниране на методи (<i>method overriding</i>), класове и конструктори. Класово наследяване и използване на <i>instanceof</i> . Шаблони при реализация на обектно-ориентирани JS приложения (<i>JS design patterns</i>) - <i>Prototype</i> , <i>Constructor</i> , <i>Module</i> , <i>Singleton</i> , <i>Observer</i> , <i>Factory</i> , <i>Mixin</i> , <i>Decorator</i> . Използване на <i>Visual Studio Code</i> .
4.	Разработка на <i>растерна</i> , <i>векторна графика</i> и <i>анимация</i> – специфика, техники, технологии, файлови формати, инструментални средства, <i>Web</i> графика и анимация, анимация с <i>CSS 3</i> . Използване на <i><canvas></i> таг и JS програмни интерфейси (<i>API</i>) за визуализация на графика с <i>HTML 5</i> . Събитийно-ориентирано програмиране с <i>JavaScript</i> . Разработка на интерактивни мултимедийни приложения с <i>HTML 5</i> и <i>JavaScript</i> – слушатели на събития, <i>drag-and-drop</i> .

5.	Разработка на <i>video</i> за уеб – специфика, технологии, файлови формати, инструменти. Вграждане на видео с HTML 5 таг <i><video></i> , създаване на собствен видео плейър с HTML 5 и JS. Разработка на <i>audio</i> материали за уеб – специфика, технологии, файлови формати, инструменти. Вграждане на звук в мултимедийни уеб приложения с HTML 5 таг <i><audio></i> . Създаване на уеб игри чрез комбиниране на анимация, интерактивност и звук с <i>HTML 5</i> и <i>JS</i> .
6.	Новости в <i>ECMAScript 6, 7 и 8</i> - класове, конструктори, <i>let</i> и <i>var</i> , ламбда функции (<i>=></i>), <i>destructuring</i> , <i>Rest/Spread</i> , <i>Promises</i> , <i>async/await</i> , <i>import()</i> и др. JavaScript модули и модулни системи - <i>CommonJS</i> и <i>ES 6</i> . Инструменти за изглаждане на уеб приложения (<i>build toolchain</i>) - <i>npm</i> , <i>babel</i> , <i>lite-server</i> , <i>webpack</i> . Конфигуриране на <i>ES6</i> среда за разработка с <i>webpack</i> , <i>webpack-dev-server</i> , <i>hot module replacement</i> , <i>development/production/test</i> конфигурации.
7.	Въведение в TypeScript – функции, интерфейси, <i>duck typing</i> , класове и конструктори, <i>public/private</i> свойства, <i>get</i> and <i>set</i> достъп до свойства, <i>static</i> и <i>instance</i> страни, функционални типове и интерфейси, ламбда изрази и използване на <i>this</i> , <i>optional/default/rest</i> параметри, разширяване на интерфейси, <i>enums</i> . Модули в <i>TypeScript</i> – <i>namespaces/modules</i> , използване на <i>import</i> , <i>export</i> и <i>require</i> . <i>Generic</i> параметри на типа – <i>generic</i> функции и класове, <i>generic</i> конструктори, <i>bounded generics</i> . Извеждане на типа (<i>type inference</i>) и съвместимост на типове (<i>type compatibility</i>). Декоратори. Конфигуриране и разгръщане на <i>TypeScript</i> проект – <i>tsconfig.json</i> , <i>typings</i> , опции на компилатора. Инструменти – <i>tsc</i> , <i>npm</i> , <i>webpack</i> .
8.	<i>Single Page Applications (SPA)</i> с <i>Angular 5</i> и <i>TypeScript</i> - <i>Angular Command Line Interface (CLI)</i> , уеб компоненти, модули, свързване (<i>binding</i>), интерполация и вградени директиви, <i>pipes</i> , <i>Angular</i> услуги, инжектиране на зависимостите (<i>dependency injection - DI</i>), темплейт-базирани и реактивни форми. Създаване на собствени валидации и мулти-провайдъри.
9.	Асинхронни операции и реактивно програмиране (<i>Angular HttpClient</i> и <i>RxJS</i>), анимации. Рутиране – <i>HTML 5 History API</i> , предаване на данни (<i>data</i>), асинхронно резолвиране (<i>resolves</i>), условия за преход (<i>guards</i>), „мързеливо“ зареждане (<i>lazy loading</i>). Анимации. Зони. Детекция на промените (<i>change detection</i>). <i>Angular Material</i> .
10.	<i>Redux</i> архитектура. Предсказуемо реактивно управление на състоянието на приложението – <i>NGRX/store</i> . Изчисляване на производни данни в стил

	<i>Reselect</i> с <i>RxJS</i> . <i>NGRX/@Effects</i> . <i>NGRX Router</i> интеграция. <i>IndexedDB</i> интеграция.
11.	Използване на <i>WebSockets</i> . Реактивна <i>WebSocket</i> имплементация с <i>RxJS</i> . <i>Test Driven Development (TDD)</i> – <i>unit</i> и <i>end-to-end</i> тестване на <i>Angular</i> компоненти и приложения с използване на <i>Karma</i> и <i>Protractor</i> .
12.	Типизиран достъп до данните и управление на кеша с <i>GraphQL</i> . <i>GraphQL Schema Definition Language (SDL)</i> . Заявки (<i>queries</i>), мутации (<i>mutations</i>) и обновяване на данните в реално време (<i>subscriptions</i>). Използване на <i>GraphQL</i> в <i>Angular</i> приложения с <i>Apollo</i> клиент.
13.	<i>HTTP</i> протокол и мултимедийни типове (<i>MIME</i> типове). Разработка на уеб услуги (<i>APIs</i>) – <i>JavaScript Object Notation (JSON) APIs</i> на <i>Node.js</i> + <i>Express.js</i> сървърна платформа. Съхраняване и извличане на данните в/от <i>NoSQL</i> база от данни <i>MongoDB</i> . Софтуерен архитектурен стил <i>REST (REpresentational State Transfer)</i> и <i>HATEOAS (Hypermedia as the Engine of Application State)</i> . Шаблони и добри практики при проектиране и реализация на <i>REST/JSON APIs</i> . Практическа реализация на <i>JSON API</i> с <i>Express.js</i> .
14.	Разработка на интерактивни мултимедийни приложения с <i>HTML 5</i> и <i>JavaScript APIs</i> – <i>Audio, Video, Canvas, Camera & Microphone, Geolocation & Maps, Local Storage, History, File API, WebSocket</i> . Разработка на разпределени (<i>fullstack</i>) хипермедийни приложения (<i>Web 2.0 Mashups</i>) – <i>SPA Angular 5</i> клиент + <i>REST / JSON APIs</i> + <i>MongoDB</i> .
15.	3D графика и виртуална реалност с <i>Three.js</i> . Интеграция на 3D графика с <i>Three.js</i> в <i>Angular</i> приложения с <i>Whitestorm.js</i> (опростен код, пост ефекти) и <i>Physi.js</i> (физическа симулация). Разработка на уеб-базирани мултиплейър компютърни игри с <i>Angular, Whitestorm.js</i> и <i>Physi.js</i> .

Библиография

Основна:

1. Официален уеб сайт на библиотеката Angular – <https://angular.io>
2. Официален уеб сайт на TypeScript – <http://www.typescriptlang.org/>
3. Сайт на Mozilla Developer Network за съвременни уеб технологии – <https://developer.mozilla.org/>
4. Gothelf, J., Lean UX: Applying Lean Principles to Improve User Experience, O'Reilly, 2013
5. Osmani, A., Learning JavaScript Design Patterns, O'Reilly - Creative Commons Attribution-Noncommercial-No Derivative Works 3.0 – <http://addyosmani.com/resources/essentialjsdesignpatterns/book/>
6. Mardan, A., Pro Express.js: Master Express.js: The Node.js Framework For Your Web Development, 1st ed., APress, 2014, ISBN-10: 1484200381

Дата:

5 декември 2017 г

Съставил:

ас. Траян Илиев