

СПРАВКА

**ЗА ОТРАЗЯВАНЕ НА ПОСТЪПИЛИТЕ ПРЕДЛОЖЕНИЯ ОТ ОБЩЕСТВЕНИТЕ КОНСУЛТАЦИИ НА ПРОЕКТА НА НАРЕДБА
ЗА ПРИДОБИВАНЕ НА КВАЛИФИКАЦИЯ ПО ПРОФЕСИЯТА КОД 061303 „РАЗРАБОТКА НА СОФТУЕР“**

Период на обществените консултации: 8 октомври – 11 ноември 2025 г.

№	Организация/потребител (вкл. начина на получаване на предложението)	Бележки и предложения	Приети/ неприети	Мотиви
1.	Мартин Ветов и екипът на ПГ по телекомуникации Получено по е-поща	1. Формулировка на ЕРУ: ЕРУ 4.6: „Създава уебстраници Аргументи: Концепцията за „адаптивен дизайн“, или често срещана като „мобилна версия“, е предшественика на концепцията за „респонсив дизайн“, при която чрез процентно преоразмеряване се преминава към гъвкава основа. Двете концепции се различават в своята реализация и продължават да се развиват. Предложение за промяна и актуализация: Учениците да усвоят адаптивен и респонсив дизайн като две отделни концепции за промяна на изгледа на уебстраници, спрямо размера на използваното устройство. 2. Формулировка на ЕРУ: ЕРУ 6.1: „Създава интерфейс с помощта на базови HTML/CSS шаблони “. Аргументи: В съвременната разработка на уеб приложения разграничаването на стилове и свойства в CSS за визуално оформление трябва да съпътства изучаването на CSS библиотеки Bootstrap или Tailwind, тъй като те предоставят множество креативни и оптимизирани решения за създаване на интуитивен потребителски интерфейс. Предложение за промяна и актуализация: Учениците да усвоят работата с CSS библиотеки Bootstrap или Tailwind. 3. Формулировка на ЕРУ: ЕРУ 6.3: „Конфигурира интерфейс според потребителските нужди “. Аргументи: Анализирането на влиянието на структурата и визуалните елементи върху потребителското поведение бива съпътствано от анализиране влиянието на анимациите и микроинтеракциите, тъй като динамичните елементи подобряват ориентацията и обратната връзка на потребителя.	Частично приети	Част от бележките са приети с допълнени текстове в РУ 4.6, РУ 6.1, РУ 6.3, РУ 11.2. Предложения относно текстове за Реализирането на алгоритми с използване на графи и дървета за представяне и търсене на данни. Добавено е ново РУ 7.2 Прилага алгоритмичен дизайн при разработка на основни програмни модули и сървърна логика Не се приемат следните предложения: Предложение за промяна и актуализация: Учениците да усвоят работата с CSS библиотеки Bootstrap или Tailwind. Мотив: Това предложение вече е отразено, но без да се посочва изрично наименование на библиотеки, поради факта, че те биват често актуализирани и посочването на наименование на софтуерен продукт, би довело до бързото морално остаряване на ДОС. В проект на ДОС CSS библиотеките посочени в становището са рефирани като UI библиотеки, например в „РУ 6.1. Познава различни UI библиотеки и компоненти за ускоряване на разработката“. Предложение за промяна и актуализация на РУ 91.: Учениците да усвоят принципите за разпределение и управление на ресурсите в екипа – време, човешки и физически ресурси, спрямо целите и етапите на проекта. Мотив: тази материя е част от изучаването на цикъла на софтуерна разработка и различните методологии за разработка и в този смисъл вече е засегнато от РУ 9.1.

№	Организация/потребител (вкл. начина на получаване на предложението)	Бележки и предложения	Приети/ неприети	Мотиви
		Предложение за промяна и актуализация: Учениците да усвоят анализиране влиянието на анимациите и микроинтеракциите на потребителя с интерфейса. 4. Формулировка на ЕРУ: ЕРУ 9.1: „Изпълнява професионални дейности, свързани с жизнения цикъл на софтуерната разработка Аргументи: При познаване фазите на жизнения цикъл на софтуера, учениците трябва да могат да разпределят и управляват ресурсите в екипа – време, човешки и физически ресурси, спрямо целите и етапите на проекта, като част от етапа „Анализ“. Предложение за промяна и актуализация: Учениците да усвоят принципите за разпределение и управление на ресурсите в екипа – време, човешки и физически ресурси, спрямо целите и етапите на проекта. 5. Формулировка на ЕРУ: ЕРУ 12.2: „Реализира проект, базиран на архитектура от микроуслуги, съгласно зададено изискване“. Аргументи: Микроуслугите са прекалено сложна архитектура за учениците. Изискват се познания за мрежова комуникация, DevOps и контейнери, което ги прави неприложими в училищна среда. Предложение за промяна и актуализация: Учениците да усвоят внедряването и наблюдаването на работата на примерен софтуерен продукт (напр. уеб приложение) в избрана облачна среда, без използване на микроуслуги. 6. Формулировка на ЕРУ: ЕРУ 12.4: „Прилага CI/CD pipelines за автоматизиране на процесите по изграждане, тестване и внедряване“. Аргументи: Тази тема изисква познания и инфраструктура, които надхвърлят възможностите на училищата. Инструменти като Jenkins, GitLab CI и GitHub Actions предполагат DevOps умения и реални среди. Целта на обучението трябва да бъде ориентирана към проектно и компетентно базирано обучение, което да доведе до усвояване на основни програмни концепции, и изграждане на логическо мислене в учениците. Предложение за промяна и актуализация: Учениците да се научат да описват основните етапи на процеса по изграждане,		Предложение за промяна и актуализация на РУ 12.2: Учениците да усвоят внедряването и наблюдаването на работата на примерен софтуерен продукт (напр. уеб приложение) в избрана облачна среда, без използване на микроуслуги. Мотив: микроуслугите като концепция, се търси от бизнеса, в т.ч. и от начинаещи (джуниър) кадри. Нещо повече, през последните години в регламента на Националната олимпиада по ИТ, микроуслугите са добавени като очакван компонент в проекти в старшата категория. Съществуват достатъчно възможности концепцията да се разгледа по подходящ за учениците начин. Също, това ЕРУ е очаквано само за по-високата – трета степен и е важен компонент от по-високата степен на професията. Предложение за промяна и актуализация на РУ 12.4: Учениците да се научат да описват основните етапи на процеса по изграждане, тестване и внедряване на софтуер и да могат да демонстрират ръчно изпълнение на тези стъпки в учебна среда. Мотив: внедряването на софтуер е важна част от жизнения цикъл на софтуерната разработка и премахването му би довело до сериозен дефицит в подготовката на учениците. Това е търсено умение от страна на бизнеса, а също така съществуват редица инструменти, включително и безплатни, с които може да се реализира в опростен вариант, подходящ за ученици. Колкото до предложената формулировка, това е нещо, което вече се покрива от други предложени в проекта на ДОС. Предложения за допълване с конкретни технологии и протоколи (т.7) Мотив: Споменаването им крие риск от моралното остаряване при скорошна тяхна промяна/замяна от подобрени технологии. Въпреки това би са добавени конкретни знания и умения в РУ 11.2 относно автентикация и оторизация

№	Организация/потребител (вкл. начина на получаване на предложението)	Бележки и предложения	Приети/ неприети	Мотиви
		тестване и внедряване на софтуер и да могат да демонстрират ръчно изпълнение на тези стъпки в учебна среда. 7. Предложения за актуализация на проекта на ДОС: А. Да се споменат конкретно съвременни уеб протоколи за автентикация и управление на достъп – JWT, OAuth. Те са ключови при изграждане на реални приложения и същевременно могат да бъдат използвани и в учебна среда. Б. Да се добави ЕРУ, което в смисъла на обучението в училищна среда да има за цел да доразвие и доизгради напреднали умения за работа с алгоритми и структури от данни. То от своя страна ще развие логическото мислене и уменията за решаване на широк набор от задачи (problem solving skills), нужни в днешния свят на Индустрия 4.0 и Индустрия 5.0. Новата ЕРУ да включва: 1. Прилагане на алчни алгоритми и рекурсия (и алгоритми с пълно изчерпване) за решаване на оптимизационни задачи. Ученикът да обяснява принципа на работа и да анализира коректността на решенията. 2. Реализирането на алгоритми с използване на графи и дървета за представяне и търсене на данни. Структурите от данни граф и дърво са основополагащи за реалните софтуерни системи – файлови системи, навигационни карти, социални мрежи. Ще развият в учениците пространствено мислене и да могат да моделират зависимости и йерархии. 3. В проекта на ДОС липсва и използване на библиотеки за измерване и сравнение на ефективността на angrorHTMH(benchmarking). Това е важен практически елемент, който насърчава аналитичното мислене – защо дадено решение е по-бързо и по-ефективно, както и развива самостоятелния експериментален подход. Може да се реализира с достъпни инструменти, които са вградени в езиките за програмиране, както и с библиотеки, даващи сумирани резултати (например BenchmarkDotNet, rich-bench и др.) В заключение, предлаганите промени целят адаптиране на стандарта към реалните възможности на учениците. Темите за DevOps и контейнеризация следва да се разглеждат на концептуално ниво, докато фокусът в професионалното образование трябва да бъде върху алгоритмичното мислене, структуриране		Предложения за включване на текстове за използване на библиотеки за измерване и сравнение на ефективността на angrorHTMH(benchmarking). Мотив: Относно сравнението на производителността (benchmarking), това е похват, който по-скоро следва да се демонстрира от учители при усвояване на конкретни теми от теорията и практиката и не е нужно да се посочва изрично в текста на ДОС.

№	Организация/потребител (вкл. начина на получаване на предложението)	Бележки и предложения	Приети/ неприети	Мотиви
		на кода и основите на проектирането на приложения. Предложението за нова ЕРУ възстановява баланса между теория и практика и осигурява стабилна основа за бъдещо надграждане в по-високи степени на обучение, когато в учениците е развито аналитично и логическо мислене, и е изграден навика за самостоятелност.		
2.	Иван Илиев – учител МГ „Акад. Кирил Попов“, Пловдив Получено по е-поща	Учител съм от МГ „Акад. Кирил Попов“ Пловдив, преподаващ 7 години по професията „Системен програмист“, като препоръката ми е да се включат следните компоненти към ДОС на бъдещата професия „Разработка на софтуер“: ЕРУ 14: Създаване на софтуер за вградени системи ЕРУ 15: Създаване на игри Описание на единиците, които предлагам: За ЕРУ 14 са същите, каквито в момента при „Системен програмист“ са при модул Вградени системи. Ако се приеме и ЕРУ 15, ще опиша подробно следващите точки: – изучаване на data oriented design – изучаване на Entity-Component-Systems (горните две парадигми се различават от ООП, като допринасят за бързината на приложенията и дават нови възможности при проектирането на софтуер) – развитие на абстрактното мислене и работа в екип, защото направата на игра включва познания по операционни системи, аудио и видео обработка, асинхронен и паралелен код, познания по OSI модела (за направата на мултиплеър), алгоритмично мислене, както и познания по графичен дизайн и рендирането на самия екран. Минимум часове – 72 часа теория и 72 часа практика. Който желае, може да увеличи часовете чрез РПП. Допълнителни мотиви от мен: Учениците ще бъдат подготвени в изграждането на почти всякакъв тип софтуер (без операционни системи или изкуствен интелект*), като по този начин няма да бъдат засегнати от бъдещите тенденции на големите компании, а и ще имат шанс да бъдат не просто техни работници, а конкурентни, изграждащи собствен бизнес.	Частично приети	Създава се ново ЕРУ 13 „Програмиране на вградени микрокомпютърни системи“ Мотив: Областта е релевантна, има устойчива традиция в ПОО (професия „Системен програмист“). Секторният съвет не възразява и темата е ключова за хардуерно-софтуерна интеграция. Добавя се като ново ЕРУ за III СПК. Предшното ЕРУ 13 се преномерираща на ЕРУ 14. ЕРУ Създаване на игри – не се приема Мотив: Разработването на игри изисква комплекс от знания (графика, звук, OS, мрежи), който надхвърля рамката и целите на професията. Основните принципи (работа с графики, асинхронност и др.) вече се покриват от други ЕРУ. Добавянето би разширило необосновано обхвата на професия „Разработка на софтуер“.

№	Организация/потребител (вкл. начина на получаване на предложението)	Бележки и предложения	Приети/ неприети	Мотиви
		* Познанията на ниско ниво (C и C++), които се изискват при направата на предложените две допълнителни разновидности софтуер (вградени системи и игри), освен че ще ги направят разбиращи от хардуер и софтуер, биха им отворили врати и за писането на драйвери или допринасянето за развитието на операционните системи, защото ще имат основата. Изкуственият интелект си има своя отделна професия, изискваща задълбочаване (и предварителни познания), така че липсата му тук е желателна и няма да разводни и без това важната основа, която се преподава. Разбира се, че учителят ще има свободата да избира езика за програмиране – такъв, който е подходящ за модулите и владее най-добре. При вградените системи може да се използва C, C++ или Rust. При създаването на игри подходящи езици са C/C++/C#/GDScript, както и комбинация от библиотеки, писани на C/C++ или готови енджини като Godot/Unity/Unreal.		