**Учебно-тематический план дополнительной профессиональной программы**

Автономная некоммерческая организация дополнительного профессионального образования центр повышения квалификации «АИС»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| «Разработка высокопроизводительной системы» | | | | | | |
| Цель обучения: | | | узнать принципы и подходы в построении высокопроизводительной системы;  освоить паттерны, научится выбирать оптимальную архитектуру системы под профильные задачи, и тем самым оптимизировать работоспособность высокопроизводительных систем. | | | |
| Категория слушателей: | | | разработчики, архитекторы, руководители проектов, аналитики | | | |
|  | | |  | | | |
| Срок обучения: | | | 24 академических часов | | | |
| Режим занятий: | | | 5 академических часов в день | | | |
| Форма обучения: | | | очно (с применением дистанционных образовательных технологий и электронного обучения) | | | |
| № | Наименование | Всего, | | В том числе | | Форма |
|  | разделов и тем | часов | | лекции | практические занятия | контроля |
| **1.** | **Принципы построения высокопроизводительной системы**Отказоустойчивость, высокопроизводительность, горизонтальная масштабируемость, высокодоступность, disaster-recoveryОценка стоимости исправления ошибок на разных стадиях жизненного цикла разработки ПОВыбор оптимальной архитектуры системы для конкретной задачи: монолитная, микросервисная, событийно-ориентированная, etc | **6** | | **3** | **3** |  |
| **2.** | **Использование паттернов в построении высокопроизводительной системы**GRASPApplication Integration Patterns (Messaging, Transactional, Service, Remote Procedure Invocation)Architecture Patterns (Layered Architecture, Event-Driven Architecture, etc)Использование паттернов микросервисной архитектуры: "API Gateway", "Event Sourcing", "CQRS", "Saga", etc | **8** | | **4** | **4** |  |
| **3.** | **Практики построения высокопроизводительной системы**Мониторинг производительности приложенияПостроение CI/CD процессовРеализация интеграционных и end-to-end с помощью TestcontainersТехники увеличение производительности автоматизированных тестов: распараллеливание тестов, реализация независимых stateless-тестов, etcИзмерение покрытия тестов, pact-тестирование, etcАвтоматизация анализа кода: выявление уязвимостей, маскировка чувствительных/клиентский данных, внедрение статических анализаторовРеализация горизонтального масштабирования системРеализация disaster-recovery -Реализация zero-downtimeПартиционирование, шардирование и денормализацияПримеры и кейсы успешной реализации архитектур высокопроизводительных систем | **8** | | **4** | **4** |  |
|  | **Всего:** | **22** | | **11** | **11** |  |
|  | **Итоговая аттестация** | **2** | |  | **2** | **зачет** |
|  | **Итого:** | **24** | | **11** | **13** |  |