ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

КАФЕДРА КИБЕРНЕТИКИ

ОДОБРЕНО  
  
протокол № 18 / 03   
  
от « 31 » мая 2018 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

МЕТОДОЛОГИЯ ПРОГРАММНОЙ ИНЖЕНЕРИИ (ТЕХНОЛОГИЯ РАЗРАБОТКИ, ВЕРИФИКАЦИЯ И СЕРТИФИКАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ)

|  |  |
| --- | --- |
| Направление подготовки (специальность) | 09.04.04 Программная инженерия |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Семестр** | **Интерактив** | **Трудоемкость, кред.** | **Общий объем курса, час.** | **Лекции, час.** | **Практич. занятия, час.** | **Лаборат. работы, час.** | **СРС, час.** | **КСР, час.** | **Форма(ы) контроля, экз./зач./КР/КП** |
| 3 |  | 4 | 144 | 16 | 16 | 16 | 96 | 0 | З |
| ИТОГО | 0 | 4 | 144 | 16 | 16 | 16 | 96 | 0 |  |

Группа: М20-504

АННОТАЦИЯ

В рамках данной дисциплины студенты знакомятся с базовыми подходами в области разработки программного обеспечения, а также в области управления процессами разработки программного обеспечения. В рамках курса обучаемые студенты будут изучать базовые теоретические и практические понятия, используемые в классических и современных методах разработки программного обеспечения. Также студенты будут изучать специфические методы разработки программного обеспечения, применяемые в различных отраслевых областях промышленности. Кроме прочего студенты знакомятся с понятиями процессов сертификации программного обеспечения, необходимыми условиями и процедурами сертификации. Изучают особенности процессов сертификации программного обеспечения, используемого в различных отраслях промышленности. Студенты изучают особенности процессов разработки сертифицируемого и несертифицируемого программного обеспечения.

1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью курса является ознакомление студентов с современными технологиями верификации программного обеспечения, приобретение ими навыков использования современных инструментальных средств для верификации программных систем. Курс знакомит студентов с методиками тестирования промышленных программных систем и охватывает вопросы построения тестового окружения, планирования системы тестов, анализа и улучшения покрытия тестируемой системы, интеграционного и нагрузочного тестирования, тестирования пользовательских интерфейсов. Также в программу курса входит рассмотрение процессов верификации проектной документации, а также процессов формальной инспекции кода и документации. Детально рассматриваются различные типы документации, сопровождающей процесс верификации, и процессы разработки, в которых эти документы участвуют. В завершение курса дается обзор методов разработки устойчивого программного обеспечения, позволяющих уменьшить продолжительность фазы тестирования без снижения его качества. Основным инструментальным средством, используемым при обучении, является Microsoft Visual Studio Team Edition for Software Testers.

По завершении курса студент овладевает всеми основными навыками, необходимыми для участия в верификации сложных промышленных систем.

В ходе изучения данной дисциплины предусматривается:

- проведение лекционных занятий в соответствии с утвержденной сеткой часов;

- самостоятельное дистанционное обучение с использованием материалов курса, размещенных на сайте Интернет-университета информационных технологий ИНТУИТ

- самостоятельное изучение литературы, поиск и анализ информации, размещаемой на сайтах, посвященных тестированию программного обеспечения, освоение теоретического материала, подготовка к семинарским занятиям;

- проведение лабораторных работ с использованием материалов курса, размешенных на сайте Интернет-университета информационных технологий ИНТУИТ

- сдача зачета по изучаемым темам.

В результате изучения дисциплины студент должен:

- обладать знаниями обо всех основных методах тестирования программного кода;

- иметь представление о комплексном тестировании программных систем;

- иметь представление о процессах, связанных с промышленной разработкой программных систем;

- знать методы создания надежных программных систем;

- иметь представление о стандартах качества в области разработки программного обеспечения;

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Технология разрабтоки, верификация и сертификация программного обеспечения» относится к вариативной части общенаучного цикла и является обязательной дисциплиной для студента.

Дисциплина не требует специальной подготовки, кроме полученной слушателями в процессе изучения дисциплин профессионального цикла, предусмотренных федеральным государственным образовательным стандартом высшего профессионального образования по направлению подготовки «Программная инженерия» (квалификация (степень) «магистр»).

Дисциплина способствует освоению навыков и знаний, необходимых для участия в верификации и сертификации сложных промышленных систем.

3. КОМПЕТЕНЦИИ СТУДЕНТА, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ/ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ И КОМПЕТЕНЦИИ СТУДЕНТА ПО ЗАВЕРШЕНИИ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОПК-8 – Способен осуществлять эффективное управление разработкой программных средств и проектов

ПК-11 – Способен проектировать основные компоненты операционных систем

ПК-12 – Способен проектировать вспомогательные и специализированные языки программирования и языки представления данных

ПК-18 – Владеет навыками создания компонент операционных систем и систем реального времени

ПК-20 – Владеет навыками организации промышленного тестирования создаваемого программного обеспечения

ПК-25 – Способен к планированию и организации работ в проектах разработки программного продукта в соответствии с действующими нормами и требованиями заказчика

ПК-26 – Способен к участию в составлении технической документации и установленной отчетности по утвержденным формам

ПК-6 – Способен к пониманию существующих подходов к верификации моделей программного обеспечения

УК-3 – Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной цели

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ п.п** | **Наименование раздела учебной дисциплины** | **Недели** | **Лекции, час.** | **Практ. занятия / семинары, час.** | **Лабораторные работы, час.** | **Обязат. текущий контроль (форма\*, неделя)** | **Аттестация раздела (форма\*, неделя)** | **Максимальный балл за раздел\*\*** |
|  | *3 Семестр* |  |  |  |  |  |  |  |
| 1 | Методы тестирования и верификации программного обеспечения | 1-8 | 8 | 8 | 8 | БДЗ-8 | КИ-8 | 30 |
| 2 | Процессы верификации и сертификации программного обеспечения | 9-16 | 8 | 8 | 8 | БДЗ-17 | КИ-18 | 30 |
|  | *Итого за 3 Семестр* |  | 16 | 16 | 16 |  |  | 60 |
|  | **Контрольные мероприятия за 3 Семестр** |  |  |  |  |  | З | 40 |

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

|  |  |
| --- | --- |
| **Обозначение** | **Полное наименование** |
| БДЗ | Большое домашнее задание |
| КИ | Контроль по итогам |
| З | Зачет |

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Недели** | **Темы занятий / Содержание** | **Лек., час.** | **Пр./сем., час.** | **Лаб., час.** |
|  | *3 Семестр* | 16 | 16 | 16 |
| **1-8** | **Методы тестирования и верификации программного обеспечения** | 8 | 8 | 8 |
| 1 - 2 | **Виды и методы тестирования программного обеспечения. Место тестирования в жизненном цикле разработки.** 1. Жизненный цикл разработки программного обеспечения  2. Модели жизненного цикла  o Водопадная  o Спиральная  o С прототипированием  o С короткими итерациями  3. Современные технологии разработки программного обеспечения:  o Microsoft Solutions Framework  o Rational Unified Process  o eXtreme Programming  4. Ролевой состав коллектива разработчиков, взаимодействие между ролями в различных технологических процессах  5. Документация, создаваемая на различных этапах жизненного цикла  6. Задачи и цели процессов тестирования и верификации  7. Тестирование, верификация и валидация – различия в понятиях  8. Типы процессов тестирования и верификации и их место в различных моделях жизненного цикла  o Модульное тестирование  o Интеграционное тестирование  o Системное тестирование  o Нагрузочное тестирование  o Формальные инспекции  9. Верификация сертифицируемого программного обеспечения | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 | 2 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 3 - 4 | **Тестирование программного кода. Повторяемость тестирования** 1. Задачи и цели тестирования программного кода  2. Методы тестирования  o Черный ящик  o Стеклянный ящик  o Тестирование моделей  o Анализ программного кода (инспекции)  3. Тестовое окружение  o Драйверы и заглушки (структурный код)  o Тестирующие классы (объектно-ориентированный код)  o Генераторы сигналов (событийно-управляемый код)  4. Тестовые примеры  o Тест-требования как основной источник информации для тестирования  o Определение ожидаемых выходных значений  o Тестовые примеры как конкретная последовательность проверки тест-требований  o Тест-план  o Трассировка тестовых примеров на тест-требования  o Оценка качества тест-плана  o Оценка качества тестируемого кода – статистика выполнения тестов  5. Покрытие программного кода  o Понятие покрытия  o Уровни покрытия  - По строкам программного кода  - По веткам условных операторов  - По компонентам логических условий  o Классы эквивалентности  o Граничные условия  o Проверка робастности (выхода за границы диапазона)  o Метод MC/DC для уменьшения количества тестовых примеров при 3-м уровне покрытия кода  o Методы проверки системы тестов на полноту покрытия на каждом из трех уровней покрытия  6. Задачи и цели обеспечения повторяемости тестирования при промышленной разработке программного обеспечения  7. Предусловия для выполнения теста, настройка тестового окружения  8. Настройки по умолчанию для тестовых примеров и их последовательностей  9. Зависимость между тестовыми примерами и способы ее избежать | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 | 2 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 5 - 6 | **Документация, сопровождающая процесс тестирования.** 1. Технологические процессы и роли в проекте, документация, создаваемая в ходе жизненного цикла проекта, ее назначение.  2. Тест-требования  o Технологические цепочки и роли участников проекта, использующих тест-требования. Связь тест-требований с другими типами проектной документации.  o Полнота тест-требований  o Непротиворечивость тест-требований  o Тестируемость тест-требований  3. Тест-планы  o Технологические цепочки и роли участников проекта, использующих тест-планы. Связь тест-планов с другими типами проектной документации.  o Возможные формы подготовки тест-планов  - Сценарии  - Таблицы  - Конечные автоматы  o Использование тест-планов для ручного и автоматического тестирования  o Генераторы тестов  4. Отчеты о прохождении тестов  o Технологические цепочки и роли участников проекта, использующих отчеты о прохождении тестов. Связь отчетов о прохождении тестов с другими типами проектной документации.  o Возможные формы представления информации об успешности прохождения тестовых примеров  o Группировка тестовых примеров  o Автоматический анализ результатов тестирования  5. Отчеты о покрытии программного кода  o Технологические цепочки и роли участников проекта, использующих отчеты о покрытии. Связь отчетов о покрытии с другими типами проектной документации.  o Возможные формы отчетов о покрытии  o Покрытие на уровне исходных текстов и на уровне машинных кодов  6. Отчеты о проблемах  o Технологические цепочки и роли участников проекта, использующих отчеты о проблемах. Связь отчетов о проблемах с другими типами проектной документации.  o Возможные формы отчетов о проблемах  o Создание отчета о проблеме и его трассировка на программный код и документацию  7. Трассировочные таблицы  o Технологические цепочки и роли участников проекта, использующих трассировочные таблицы. Связь трассировочных таблиц с другими типами проектной документации.  o Возможные формы трассировочных таблиц  o Обеспечение целостности трассировки при повторении тестирования | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 | 2 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 7 - 8 | **Формальные инспекции.** 1. Задачи и цели проведения формальных инспекций  2. Формальные инспекции программного кода  o Этапы формальной инспекции и роли ее участников  o Особенности этапа просмотра инспектируемого кода: выделение ключевых точек и построение или использование таблиц трассировки; проверка стиля кодирования; проверка надежности кода  o Особенности этапа проведения собрания: распределение ролей; управление собранием; документирование собрания  o Особенности этапа завершения: обеспечение прослеживаемости документации, контроль за внесением изменений    3. Формальные инспекции проектной документации  o Этапы формальной инспекции и роли ее участников  o Особенности этапа просмотра документации: несогласованность и противоречивость проектной документации и их типы; методы выявления несогласованности  o Особенности этапа завершения: влияние несогласованности документации на процесс разработки; трассировка изменений на программный код («волны изменений») | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 | 2 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| **9-16** | **Процессы верификации и сертификации программного обеспечения** | 8 | 8 | 8 |
| 9 | **Модульное тестирование** 1. Задачи и цели модульного тестирования  2. Определение границ модуля  3. Проектирование тестового окружения  o «Один метод – несколько тестов»  o Заглушки – имитация внешней среды  4. Подготовка данных для модульного тестирования  o Табличное представление  o Потоковая загрузка  o Отражения и заполнения  5. Использование автоматизированных средств модульного тестирования | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 1 | 1 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 10 | **Интеграционное тестирование.**  1. Задачи и цели интеграционного тестирования  2. Подходы к организации интеграционного тестирования  o Ранняя интеграция  o Поздняя интеграция  o Послойная интеграция  o Иерархическая интеграция  3. Тестирование межмодульных интерфейсов  4. Определение границ применимости требований | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 1 | 1 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 11 | **Тестирование пользовательского интерфейса** 1. Задачи и цели тестирования пользовательского интерфейса  2. Подходы к тестированию интерфейсов  o Ручное тестирование  o Сценарии  o Эмуляция событий  3. Методы автоматизации тестирования пользовательского интерфейса  4. Полнота тестирования пользовательского интерфейса | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 1 | 1 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 12 | **Системное тестирование** 1. Задачи и цели системного тестирования  2. Планирование системного тестирования  3. Подготовка испытательных стендов  4. Нагрузочное тестирование  5. Особенности системного тестирования и приемо-сдаточных испытаний при разработке сертифицируемого программного обеспечения | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 1 | 1 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 13 | **Методы разработки устойчивого кода** 1. Классификация проблем, возникающих при работе программных систем  o Сбои  o Отказы  o Аварии  2. Методы разработки устойчивого кода  o Критические точки  o Предусловия (assertions)  o Обработка исключений  o Автоматический анализ программного кода на устойчивость  o Сбор и обработка информации о сбоях и отказах | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 1 | 1 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 14 | **Процесс управления качеством при промышленной разработке программного обеспечения**  1. Задачи и цели управления качеством  2. Система менеджмента качества по ISO 9000 и CMMi  3. Аудит процессов разработки и тестирования  4. Корректирующие действия и коррекция процессов  5. Оптимизация процессов | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 1 | 1 | 1 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |
| 15 - 16 | **роцесс управления конфигурациями при промышленной разработке программного обеспечения. Разработка сертифицируемого программного обеспечения** 1. Задачи и цели управления конфигурациями  2. Идентификация объектов разработки  3. Управление версиями  4. Жизненные циклы объектов разработки  5. Управление связями между объектами разработки  6. Базовые версии  7. Целостность конфигурации  8. Управление качеством и управление конфигурациями при разработке сертифицируемого программного обеспечения. | Всего аудиторных часов | Всего аудиторных часов | Всего аудиторных часов |
| 2 | 2 | 2 |
| Онлайн | Онлайн | Онлайн |
|  |  |  |

Сокращенные наименования онлайн опций:

|  |  |
| --- | --- |
| **Обозначение** | **Полное наименование** |
| ЭК | Электронный курс |
| ПМ | Полнотекстовый материал |
| ПЛ | Полнотекстовые лекции |
| ВМ | Видео-материалы |
| АМ | Аудио-материалы |
| Прз | Презентации |
| Т | Тесты |
| ЭСМ | Электронные справочные материалы |
| ИС | Интерактивный сайт |

ТЕМЫ ЛАБОРАТОРНЫХ РАБОТ

|  |  |
| --- | --- |
| **Недели** | **Темы занятий / Содержание** |
|  | *3 Семестр* |
|  | 1. Обзор общих подходов к тестированию, изучение архитектуры программного комплекса «Калькулятор». Постановка и обсуждение задачи, используемой в качестве сквозного примера в ходе семинарских занятий. Обсуждение общей архитектуры системы, разбиение на модули. |
|  | 2. Тестовые примеры. Обсуждение подходов к тестированию системы. Классы эквивалентности, граничные условия, робастность (тестирование за границей диапазонов). |
|  | 3. Возможности MVSTE по ручному тестированию и описанию тестовых примеров (Manual Testing). Демонстрация простейшего подхода к тестированию ПО и его реализация в MVSTE. |
|  | 4. Тестовое окружение. Проектирование тестового окружения. Тестовые классы, тестовые проекты. Демонстрация написания тестового драйвера для функций калькулятора. |
|  | 5. Покрытие программного кода. Методы проверки покрытия, методы улучшения покрытия. Покрытие по MC/DC. Отчеты о покрытии. Формы отчетов о покрытии, анализ неполноты покрытия, влияние зависимостей тестовых примеров на полноту покрытия. |
|  | 6. Покрытие программного кода. Методы проверки покрытия, методы улучшения покрытия. Покрытие по MC/DC. Отчеты о покрытии. Формы отчетов о покрытии, анализ неполноты покрытия, влияние зависимостей тестовых примеров на полноту покрытия. |
|  | 7. Зависимости тестовых примеров. Инициализация тестового окружения перед выполнением. Выполнение последовательностей тестов. Зависимость по общим данным, зависимость по состоянию системы/модуля. |
|  | 8. Формальная инспекция программного кода. Демонстрация процедуры формальных инспекций на примере проведения формальной инспекции программного кода. |
|  | 1. Модульное тестирование: задачи и цели. Понятие модуля и его границ. Тестирование классов. Проектирование тестового окружения. Демонстрация написания тестового драйвера и создание тестового окружение для функций калькулятора. |
|  | 2. Конфигурации тестов и упорядоченные тесты (ordered tests) в MVSTE. Демонстрация возможности зависимости тестовых примеров от их порядка и от тестового окружения, а также средств создания упорядоченных тестов в MVSTE. |
|  | 3. Интеграционное тестирование. Тестирование межмодульных интерфейсов, определение границ тестируемой области. |
|  | 4. Тестирование информационного обмена между модулями. |
|  | 5. Демонстрация возможности MVSTE по интеграционному тестированию (тестирование с постоянной интеграцией). |
|  | 6. Критерии начала и окончания системного тестирования, |
|  | 7. Границы системного тестирования |
|  | 8. Тестирование пользовательского интерфейса |
|  | 9. Управление конфигурациями при верификации программного обеспечения |
|  | 10. Управление качеством при верификации программного обеспечения |

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

1. Лекционные занятия:

a. комплект электронных презентаций/слайдов,

b. аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук)

2. Практические занятия:

a. компьютерный класс,

b. презентационная техника (проектор, экран, компьютер/ноутбук),

c. стандартный пакет программ Microsoft Office.

d. пакет Microsoft Visual Studio for Software Testers 2005 и выше

6. ТРЕБОВАНИЯ К ФОНДУ ОЦЕНОЧНЫХ СРЕДСТВ В РАМКАХ РЕАЛИЗУЕМОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В качестве оценочного средства используется 100 бальная семестровая система, учитывающая посещаемость занятий, активность (выполнение домашних занятий), выполнение тематических домашних заданий по каждому разделу, контрольно-тестовая работа по каждому разделу. Каждый раздел проходит аттестацию.

ПЗ – посещаемость лабораторных занятий (еженедельно) не менее 80% +2 балла

не менее 50% +1 балл

менее 50% 0 баллов

ТДЗ – выполнения тематического ДЗ (по каждому разделу)

Выполнено не менее 90% +28 баллов

Выполнено от 80-до 89% +20 балла

Выполнено от 70-до 79% +15 балла

Выполнено от 60-до 69% +10 балла

Выполнено от 40-до 59% +5 балл

Менее 39% 0 баллов

КИ – аттестация раздела (контроль по итогам раздела) Раздел аттестуется, если набрано не менее 50%

Самостоятельная работа студента включает: Повторение теоретического материала –

Выполнение ДЗ

Выполнение ТДЗ

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА:

1. ЭИ С 77 Основы тестирования и верификации программного обеспечения : учебное пособие, Санкт-Петербург: Лань, 2020

2. ЭИ З-91 Технология разработки программного обеспечения : учебное пособие, Санкт-Петербург: Лань, 2019

3. 004 С38 Верификация программного обеспечения : учебное пособие, С. В. Синицын, Н. Ю. Налютин, Москва: Интернет-Университет информационных технологий; Бином. Лаборатория знаний, 2008

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. Информационные материалы, размещенные на сайте http://cyber.mephi.ru (http://cyber.mephi.ru)

2. Синицын С.В., Налютин Н.Ю. Верификация программного обеспечения. http://www.intuit.ru/department (http://www.intuit.ru/department/se/verify/)

3. Microsoft Solutions Framework. Методология создания программных решений (http://www.microsoft.com/Rus/Msdn/msf/Default.mspx)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Специальное материально-техническое обеспечение не требуется

Рабочая программа дисциплины составлена в соответствии с ОС НИЯУ МИФИ (ФГОС) и учебным планом основной образовательной программы (программ).

Автор(ы):

|  |  |  |
| --- | --- | --- |
|  | Иващенко Максим Георгиевич |  |