

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

И.о. декана факультета информационных
технологий и компьютерной безопасности



_____/ А.В. Бредихин /

19.03.2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Технологии совместной разработки»

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Технологии искусственного интеллекта

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2024

Автор программы

_____/ Н.А. Рындин

И.о. заведующего кафедрой
систем
автоматизированного
проектирования и
информационных систем

_____/ П.Ю. Гусев

Руководитель ОПОП

_____/ Д.В. Иванов

Воронеж 2024

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Формирование у студентов теоретических знаний в области технологий совместной разработки программного обеспечения. Развитие практических навыков по организации и управлению командной работой над проектами в сфере информационных систем и технологий. Подготовка студентов к эффективному применению современных инструментов и методологий для совместной разработки, развертывания и эксплуатации информационных систем.

1.2. Задачи освоения дисциплины

- изучение основных концепций и принципов технологий совместной разработки программного обеспечения;
- освоение методов организации командной работы, распределения ролей и обязанностей в команде;
- изучение инструментов для совместной работы над проектами, таких как системы контроля версий, средства управления проектами, средства автоматизации сборки и развертывания;
- ознакомление с методологиями гибкой разработки и их применение в командной работе;
- изучение подходов к документированию, коммуникации и обеспечению безопасности в процессе совместной разработки;
- приобретение навыков тестирования, отладки и эксплуатации программных систем, разработанных в команде;
- формирование умений по оценке и улучшению процесса совместной разработки на основе анализа метрик и показателей эффективности.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Технологии совместной разработки» относится к дисциплинам части, формируемой участниками образовательных отношений, блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Технологии совместной разработки» направлен на формирование следующих компетенций:

ПК-1 – способен анализировать предметную область, определять современные подходы и стандарты автоматизации в процессе проектирования и разработки информационных систем;

ПК-6 – способен выполнять работы по созданию (модификации), развертыванию и сопровождению информационных систем и ресурсов для различных прикладных областей.

| Компетенция | Результаты обучения, характеризующие сформированность компетенции |
|--------------------|--|
| ПК-1 | знать основные концепции, принципы и инструменты технологий совместной разработки программного обеспечения |
| | уметь применять методы организации командной работы, распределения ролей и обязанностей в команде |
| | владеть навыками использования систем контроля версий, средств управления проектами и автоматизации сборки/развертывания |
| ПК-6 | знать методы тестирования, отладки и эксплуатации программных систем, разработанных в команде |
| | уметь организовывать эффективную коммуникацию и документирование в процессе совместной разработки |
| | владеть навыками обеспечения безопасности при совместной разработке программного обеспечения |

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Технологии совместной разработки» составляет 3 з. е.

Распределение трудоемкости дисциплины по видам занятий:

Очная форма обучения

| Виды учебной работы | Всего часов | Семестр |
|---------------------------------------|-------------|---------|
| | | 6 |
| Аудиторные занятия (всего) | 54 | 54 |
| В том числе: | | |
| лекции | 18 | 18 |
| лабораторные работы (ЛР) | 36 | 36 |
| Самостоятельная работа | 54 | 54 |
| Курсовой проект | - | - |
| Часы на контроль | 0 | 0 |
| Виды промежуточной аттестации – зачет | + | + |
| Общая трудоемкость: | | |
| академические часы | 108 | 108 |
| зачетные единицы | 3 | 3 |

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

5.1. Содержание разделов дисциплины и распределение трудоемкости по видам занятий

Очная форма обучения

| № п/п | Наименование темы | Содержание раздела | Лекц. | Лаб. зан. | СРС | Всего, час. |
|-------|---|--|-------|-----------|-----|-------------|
| 1 | Введение в технологии совместной разработки | Понятие и основные принципы технологий совместной разработки программного обеспечения. Преимущества и проблемы, возникающие при совместной разработке. Обзор инструментов и платформ для совместной работы над проектами | 2 | 4 | 6 | 12 |
| 2 | Организация командной работы | Методы формирования и управления командой разработчиков. Распределение ролей и обязанностей в команде. Планирование и координация работы команды. | 2 | 4 | 6 | 12 |
| 3 | Инструменты для совместной разработки | Системы контроля версий (Git, SVN и др.). Средства управления проектами (Trello, Jira и др.). Интеграция инструментов разработки (IDE, CI/CD, тестирование). | 2 | 4 | 6 | 12 |
| 4 | Методологии гибкой разработки | Основные принципы и ценности гибкой разработки. Методология Scrum: роли, артефакты, ритуалы. Методология Kanban: визуализация, ограничение WIP, непрерывный поток. | 2 | 4 | 6 | 12 |
| 5 | Документирование и коммуникация | Использование Markdown для документирования проекта. Организация эффективной коммуникации в команде разработчиков. Инструменты для совместной работы над документацией. | 2 | 4 | 6 | 12 |
| 6 | Развертывание и эксплуатация | Автоматизация процессов сборки и развертывания (CI/CD). Мониторинг и поддержка работоспособности системы. Управление конфигурацией и релизами. | 2 | 4 | 6 | 12 |
| 7 | Безопасность и конфиденциальность | Управление доступом и правами пользователей. Обеспечение безопасности при совместной разработке. Защита конфиденциальной информации. | 2 | 4 | 6 | 12 |
| 8 | Тестирование и отладка | Инструменты для автоматизированного тестирования. Методы отладки и устранения ошибок в совместной разработке. Интеграция тестирования в процесс разработки. | 2 | 4 | 6 | 12 |
| 9 | Оценка и улучшение процесса разработки | Анализ метрик и показателей эффективности команды. Выявление и устранение узких мест в процессе | 2 | 4 | 6 | 12 |

| | | | | | | |
|--------|--|--|----|----|----|-----|
| | | разработки. Внедрение улучшений в процесс совместной разработки. | | | | |
| Итого: | | | 18 | 36 | 54 | 108 |

5.2. Перечень лабораторных работ

1. Знакомство с системами контроля версий (Git, SVN и другие).
2. Основы работы с Git (создание репозитория, ветвление, слияние).
3. Использование Trello или Jira для управления проектом.
4. Распределение ролей и обязанностей в команде.
5. Использование GitHub/Gitlab/Gitverse для совместной работы над проектом.
6. Интеграция инструментов разработки (IDE, CI/CD, тестирование).
7. Применение Scrum в командной работе.
8. Использование Kanban для управления процессом разработки.
9. Использование Markdown для документирования проекта.
10. Организация эффективной коммуникации в команде.
11. Автоматизация развертывания приложения (CI/CD).
12. Мониторинг и поддержка работоспособности системы.
13. Управление доступом и правами пользователей.
14. Обеспечение безопасности при совместной разработке.
15. Использование инструментов для автоматизированного тестирования.
16. Отладка и устранение ошибок в совместной разработке.
17. Анализ метрик и показателей эффективности команды.
18. Внедрение улучшений в процесс совместной разработки

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

Не предусмотрено учебным планом.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1. Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

- «аттестован»;
- «не аттестован».

| Компетенция | Результаты обучения, характеризующие сформированность компетенции | Критерии оценивания | Аттестован | Не аттестован |
|-------------|---|--------------------------------------|---|---|
| ПК-1 | знать основные концепции, принципы и инструменты технологий совместной разработки, владеть навыками использования систем контроля версий, средств управления проектами и автоматизации сборки/развертывания, методологии гибкой разработки и особенности их применения в командной работе | Лабораторная работа 1, 2, 3, 4, 5 | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |
| | владеть навыками использования систем контроля версий, средств управления проектами и автоматизации сборки/развертывания, обеспечения безопасности при совместной разработке программного обеспечения | Лабораторная работа 1, 2, 3, 4, 5 | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |
| ПК-6 | знать методы тестирования, отладки и эксплуатации программных систем, разработанных в команде, уметь анализировать метрики и показатели эффективности командной работы для улучшения процесса совместной разработки | Лабораторная работа 1, 2, 3, 4, 5 | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |
| | владеть навыками внедрения улучшений в процесс совместной разработки программного обеспечения | Лабораторная работа 1, 2, 3, 4, 5 | Выполнение работ в срок, предусмотренный в рабочих программах | Невыполнение работ в срок, предусмотренный в рабочих программах |

7.1.2. Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5 семестре для очной формы обучения по двухбалльной системе:

- «зачтено»;
- «не зачтено».

| Компетенция | Результаты обучения, характеризующие сформированность компетенции | Критерии оценивания | Отлично | Хорошо | Удовл. | Неудовл. |
|-------------|---|--|--|---|--|--------------------------------------|
| ПК-1 | знать основные концепции, принципы и инструменты технологий совместной разработки, владеть навыками использования систем контроля версий, средств управления проектами и автоматизации сборки/развертывания, методологии гибкой разработки и особенности их применения в командной работе | Тест | Выполнение теста на 90-100% | Выполнение теста на 80-90% | Выполнение теста на 70-80% | В тесте менее 70% правильных ответов |
| | уметь использовать системы контроля версий, средств управления проектами и автоматизации сборки/развертывания, обеспечения безопасности при совместной разработке программного обеспечения | Решение стандартных практических задач | Задачи решены в полном объеме и получены верные ответы | Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах | Продемонстрирован верный ход решения в большинстве задач | Задачи не решены |
| | владеть навыками работы с системой контроля версий Git, владеть навыками обеспечения безопасности при совместной разработке программного обеспечения | Решение прикладных задач в конкретной предметной области | Задачи решены в полном объеме и получены верные ответы | Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах | Продемонстрирован верный ход решения в большинстве задач | Задачи не решены |
| ПК-6 | знать методы тестирования, отладки и эксплуатации | Тест | Выполнение теста на 90-100% | Выполнение теста на 80-90% | Выполнение теста на 70-80% | В тесте менее 70% |

| | | | | | | |
|--|---|--|--|---|--|--------------------|
| | программных систем, разработанных в команде, уметь анализировать метрики и показатели эффективности командной работы для улучшения процесса совместной разработки | | | | | правильных ответов |
| | уметь анализировать метрики и показатели эффективности командной работы для улучшения процесса совместной разработки | Решение стандартных практических задач | Задачи решены в полном объеме и получены верные ответы | Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах | Продемонстрирован верный ход решения в большинстве задач | Задачи не решены |
| | владеть навыками внедрения улучшений в процесс совместной разработки программного обеспечения | Решение прикладных задач в конкретной предметной области | Задачи решены в полном объеме и получены верные ответы | Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах | Продемонстрирован верный ход решения в большинстве задач | Задачи не решены |

7.2. Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1. Примерный перечень заданий для подготовки к тестированию

1. Какие виды систем контроля версий вам известны:

- Локальная система контроля версий;
- Централизованная система контроля версий;
- Распределенная система контроля версий;
- **Все вышеперечисленные.**

2. Какие из перечисленных системы относятся к средствам управления проектами:

- **Trello;**
- Git;
- Github;
- SVN;
- **Jira.**

3. В каких системах можно настроить CI/CD:

- Jira;
- **Gitlab;**
- SVN;
- **Github.**

4. Каких элементов нет в Markdown:

- Таблиц;
- Картинок;
- Списков;
- **Надстрочных символов.**

5. Какая из методологий не относится к семейству гибких (Agile):

- Scrum;
- Kanban;
- **Водопад (она же каскадная);**
- Scrumban.

6. Какой тип тестирования программного обеспечения нельзя автоматизировать:

- Модульное тестирование;
- Интеграционное тестирование;
- Системное тестирование;
- **Приемочное тестирование.**

7. Какая из указанных систем не относится к системам мониторинга и поддержки систем:

- Zabbix;
- Kafka;
- ELK;
- **Gitverse.**

8. В каком текстовом редакторе нет отладки исходного кода:

- **Блокнот;**
- Notepad++;
- Sublime Text;
- VScode.

9. С помощью какой программы можно организовать автоматизированное тестирование:

- VSCode;
- **Selenium;**
- Pycharm;
- Nginx.

10. С помощью чего нельзя организовать эффективную коммуникацию в команде:

- **Одноклассники;**
- Slack;
- Jira;
- Telegram.

7.2.2. Примерный перечень заданий для решения стандартных задач

1. С помощью какой команды создается проект в Git:

- **init;**
- config;
- rebase;
- commit.

2. С помощью какой команды скачиваются изменения проекта проекта в Git:

- amend;
- push;
- **pull;**
- rebase.

3. С помощью какой команды создается новая ветка в Git:

- **branch;**
- diff;
- log;
- checkout.

4. С помощью какой команды посмотреть список измененных файлов в текущей ветке Git:

- clone;
- **status;**
- commit;
- push.

5. Что делает команда move в SVN:

- Копирует файл;
- Обновляет содержимое локальной копии;
- **Перемещает файл или папку.**
- Удаляет файл или папку.

6. Что делает команда update в SVN:

- Отправляет все изменения локальной копии в репозиторий;
- **Обновляет содержимое локальной копии до самой последней версии из репозитория;**

- Удаляет файл или папку из локальной копии проекта;
- Перемещает файл или папку в другое место.

7. Какая альтернатива у команды update из SVN в Git:

- push;
- **pull**;
- add;
- commit.

8. С помощью какой команды в SVN можно добавить новые файлы на сервере:

- add;
- push;
- **import**;
- merge.

9. Как посмотреть записи каталога SVN:

- all;
- **list**;
- commit;
- checkout.

10. Какая команда одинаково работает и в SVN, и в Git:

- update;
- push;
- **status**;
- pull.

7.2.3. Примерный перечень заданий для решения прикладных задач

1. С помощью какой команды объединяются ветки в Git:

- checkout;
- **merge**;
- status;
- branch.

2. С помощью какой команды можно изменить последний коммит в Git:

- edit;
- change;
- **amend**;
- branch.

3. С помощью какой команды можно преобразовать ветку путем перебазирования в Git:

- amend;

- push;
- **rebase**;
- commit.

4. Как посмотреть изменения между коммитами и в Git, и в Svn:

- list;
- **diff**;
- status;
- log.

5. Как частично мигрировать изменения из одной ветки в другую в Git:

- merge;
- **cherry-pick**;
- checkout;
- branch.

6. Как отменить последние изменения в SVN:

- ls;
- status;
- **revert**;
- up.

7. Как отменить последние изменения в Git:

- push;
- **revert**;
- log;
- status.

8. Как переименовать файл в SVN:

- **mv**;
- del;
- lock;
- add.

9. Как удалить ветку в SVN:

- lock;
- **delete**;
- commit;
- merge.

10. Как узнать кто и какие изменения вносил в файл в SVN:

- diff;
- **blame**;
- log;
- up.

7.2.4. Примерный перечень вопросов для подготовки к зачету

1. Какие основные понятия и принципы используются в технологиях совместной разработки программного обеспечения?
2. Какие преимущества и проблемы могут возникнуть при совместной разработке?
3. Какие инструменты и платформы существуют для совместной работы над проектами?
4. Какие методы формирования и управления командой разработчиков известны?
5. Каким образом можно распределить роли и обязанности в команде?
6. Как можно планировать и координировать работу команды?
7. Какие системы контроля версий вам известны?
8. Что такое роли, артефакты и ритуалы в методологии Scrum?
9. Что такое визуализация, ограничение WIP и непрерывный поток в методологии Kanban?
10. Что такое Markdown и как его использовать для документирования проекта?
11. Как организовать эффективную коммуникацию в команде разработчиков?
12. Какие существуют инструменты для совместной работы над документацией?
13. Как автоматизировать процесс сборки и развертывания приложения?
14. Какие средства мониторинга и поддержания работоспособности системы вам известны?
15. Какие методы управления конфигурацией и релизами вам известны?
16. Как управлять доступом и правами пользователей?
17. Какие средства обеспечения безопасности при совместной разработке вам известны?
18. Какие компоненты защиты конфиденциальной информации вам известны?
19. Какие инструменты автоматизированного тестирования вам известны?
20. Какие методы отладки и устранения ошибок в совместной работе чаще всего используются?
21. Как интегрировать тестирование в процесс разработки?
22. Какие методы анализа метрик и показателей эффективности команды существуют?

23. Как выявить и устранить узкие места в процессе разработки?

24. Как грамотно внедрять улучшения в процесс совместной разработки?

7.2.5. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 20 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 30.

Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 16 баллов.

Оценка «Удовлетворительно» ставится в случае, если студент набрал от 16 до 20 баллов

Оценка «Хорошо» ставится в случае, если студент набрал от 21 до 25 баллов.

Оценка «Отлично» ставится, если студент набрал от 26 до 30 баллов.

7.2.6. Паспорт оценочных материалов

| № п/п | Контролируемые разделы (темы) дисциплины | Код контролируемой компетенции | Наименование оценочного средства |
|-------|---|--------------------------------|---|
| 1 | Введение в технологии совместной разработки | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 2 | Организация командной работы | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 3 | Инструменты для совместной разработки | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 4 | Методологии гибкой разработки | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 5 | Документирование и коммуникация | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 6 | Развертывание и эксплуатация | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 7 | Безопасность и конфиденциальность | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
| 8 | Тестирование и отладка | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |

| | | | |
|---|--|------------|---|
| 9 | Оценка и улучшение процесса разработки | ПК-1, ПК-6 | Тест, защита лабораторных работ, требования к курсовому проекту |
|---|--|------------|---|

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методике выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1. Перечень учебной литературы, необходимой для освоения дисциплины

1. Проектные методологии управления. Agile и Scrum : учебное пособие / Ю. Д. Агеев, Ю. А. Кавин, И. С. Павловский [и др.]. — Москва : Аспект Пресс, 2018. — 160 с. — ISBN 978-5-7567-0982-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/86125.html> (дата обращения: 15.05.2024). — Режим доступа: для авторизир. Пользователей
2. Аппело, Ю. Agile-менеджмент: лидерство и управление командами / Ю. Аппело ; перевод А. Олейник ; под редакцией А. Обуховой. — Москва : Альпина Паблишер, 2024. — 534 с. — ISBN 978-5-9614-

6361-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/137839.html> (дата обращения: 15.05.2024). — Режим доступа: для авторизир. пользователей

3. Майк, Кон Agile: оценка и планирование проектов / Кон Майк ; перевод В. Ионов. — Москва : Альпина Паблишер, 2018. — 424 с. — ISBN 978-5-9614-6947-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/82576.html> (дата обращения: 15.05.2024). — Режим доступа: для авторизир. пользователей

8.2. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем

Перечень ПО, включая перечень лицензионного программного обеспечения:

- ОС Windows 7 Pro;
- Microsoft Office 64-bit;
- Mozilla Firefox;
- Google Chrome;

Ресурсы информационно-телекоммуникационной сети «Интернет»:

- <http://window.edu.ru>
- <http://www.edu.ru>
- Образовательный портал ВГТУ

Современные профессиональные базы данных и информационно-справочные системы:

- <https://proglib.io>
- <https://habr.com>

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения обучения по дисциплине используется компьютерный класс. Аудитория:

Компьютерный класс

Учебная аудитория для проведения учебных занятий

Комплект учебной мебели:

- рабочее место преподавателя (стол, стул);
- рабочие места обучающихся (столы, стулья);
- персональные компьютеры с установленным ПО, подключенные к сети Интернет (12 шт.);
- принтер;
- доска магнитно-маркерная поворотная.

Помещение для самостоятельной работы: читальный зал с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

По дисциплине «Технологии совместной разработки» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

| Вид учебных занятий | Деятельность студента |
|------------------------|--|
| Лекция | Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии. |
| Лабораторная работа | Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания. |
| Самостоятельная работа | Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none">– работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;– выполнение домашних заданий и расчетов; |

| | |
|---------------------------------------|---|
| | <ul style="list-style-type: none"> – работа над темами для самостоятельного изучения; – участие в работе студенческих научных конференций, олимпиад; – подготовка к промежуточной аттестации. |
| Подготовка к промежуточной аттестации | <p>Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.</p> |

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

| № п/п | Перечень вносимых изменений | Дата внесения изменений | Подпись заведующего кафедрой, ответственной за реализацию ОПОП |
|----------|-----------------------------|----------------------------|--|
|----------|-----------------------------|----------------------------|--|