МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»

Институт информационных технологий, математики и механики
(факультет / институт / филиал)

УТВЕРЖДАЮ:

Декан факультета/директор института

_____ В.П. Гергель
«____»_____2019

Рабочая программа дисциплины

Инструментальные средства
разработки программного
обеспечения
(наименование дисциплины (модуля))

Уровень высшего образования
Бакалавриат
(бакалавриат / магистратура / специалитет)

Направление подготовки / специальность
02.03.02 Фундаментальная информатика и информационные технологии
(указывается код и наименование направления подготовки / специальности)

Направленность образовательной программы
Инженерия программного обеспечения
(указывается профиль / магистерская программа / специализация)

Форма обучения
очная
(очная / очно-заочная / заочная)

Нижний Новгород

2019 год

Лист актуализации

Визирование РПД для исполнения в очередном учебном году

Председатель МК
__ _____ 2019 г.

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2019-2020 учебном году на заседании кафедры
_____

Протокол от __ _____ 20___ г. № __
Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель МК
__ _____ 20__ г.

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2020-2021 учебном году на заседании кафедры
_____

Протокол от __ _____ 20__ г. № __
Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель МК
__ _____ 20___ г.

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2021-2022 учебном году на заседании кафедры
_____

Протокол от __ _____ 20___ г. № __
Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель МК
__ _____ 20___ г.

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2022-2023 учебном году на заседании кафедры
_____

Протокол от __ _____ 20___ г. № __
Зав. кафедрой _____

1. Место дисциплины в структуре ОПОП

Дисциплина относится к части, формируемой участниками образовательных отношений.
Б1.В.ДВ.03.01 Инструментальные средства разработки программного обеспечения

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	**Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции**		**Наименование оценочного средства**
	Индикатор достижения компетенции* (код, содержание индикатора)	**Результаты обучения по дисциплине****	
ПК-2.: Способен проектировать программное обеспечение	**ПК-2.1**: Знает типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения **ПК-2.2**: Знает методы и средства проектирования программного обеспечения **ПК-2.3.**: Знает методы и средства проектирования баз данных	*Знать:* *Основные текстовые форматы.* *Понятие регулярного выражения, основные правила формирования регулярных выражений.* *Основные команды командной строки.* *Возможности скриптовых языков для автоматизации сборки и компиляции программ.* *Понятие интегрированной среды разработки и основных компонент среды разработки.* *Понятие системы контроля версий, ее назначение. Цикл разработки программ с использованием системы контроля версий.* *Процедуру организации отладки и тестирования программ.* *Инструменты инспекции кода.*	*Собеседование,*
	ПК-2.4.: Умеет использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Уметь: *Работать с одним из текстовых редакторов, рассмотренном в лекционном материале.* *Работать с командной строкой.* *Разрабатывать скрипты для*	*Задача*

	ПК-2.5.: Умеет применять методы и средства проектирования программного обеспечения, структур данных, баз данных	*автоматизации сборки и компиляции программ.* *Работать с интегрированной средой разработки.* *Работать с системой контроля версий.* *Отлаживать и тестировать программы. Разрабатывать автоматические тесты.* *Использовать инструменты инспекции кода.* *Пользоваться навыками работы с одним широко известных текстовых редакторов.* *Пользоваться навыками работы с командной строкой.* *Пользоваться навыками использования скриптовых языков для автоматизации сборки и компиляции программ.* *Пользоваться навыками использования системы контроля версий Git.* *Пользоваться навыками отладки и тестирования программ, а также разработки автоматических тестов на примере GoogleTest.* *Пользоваться навыками использования инструментов инспекции кода на примере GitHub..*	

3. Структура и содержание дисциплины

3.1. Трудоемкость дисциплины

	Очная форма обучения
Общая трудоемкость	**3 ЗЕТ**
Часов по учебному плану	**108**
в том числе	
аудиторные занятия (контактная работа): **- занятия лекционного типа** **- занятия семинарского типа** **- текущий контроль (КСР)**	**33** **16** **16** **1**
самостоятельная работа	**75**
Промежуточная аттестация –зачет	

3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине (модулю)	**Всего (часы)**	в том числе				
		контактная работа (работа во взаимодействии с преподавателем), часы из них				**Самостоятельная работа обучающегося, часы**
		Занятия лекционного типа	**Занятия семинарского типа**	**Занятия лабораторного типа**	**Всего контактных часов**	
Рабочее место программиста	5	1	0		1	4
Текстовые форматы	6	1	1		2	4
Обработка текста и регулярные выражения	6	1	1		2	4
Текстовые редакторы	6	1	1		2	4
Автоматизация: командная строка	6	1	1		2	4
Автоматизация: скриптовые языки	6	1	1		2	4
Системы контроля версий	6	1	1		2	4
Интегрированные среды разработки	6	1	1		2	4
Описание и построение проектов	6	1	1		2	4
Анализ бинарных модулей	6	1	1		2	4
Контроль качества кода	7	1	1		2	5
Отладка	8	1	1		2	6
Тестирование	8	1	1		2	6
Непрерывная интеграция	8	1	1		2	6
Профилирование и оптимизация производительности	8	1	1		2	6
Командная разработка. Формирование сообщества	9	1	2		3	6
Текущий контроль (КСР)	1				1	
Промежуточная аттестация –зачет						
Итого	108	16	16	0	33	75

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа
Промежуточная аттестация проходит в традиционной форме (зачет).

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Используются образовательные технологии в форме лекций и практических занятий. При чтении лекций используются электронных презентаций.

Выполнение практических работ на следующие темы:

- Настройка окружения разработки.
- Проектирование программного интерфейса (API), создание пользовательской документации.
- Разработка функциональности.
- Интеграция кода в общий проект.
- Создание модульных тестов.

Образовательные материалы для самостоятельной работы студентов

- Страуструп Б. Курс «Язык программирования C++ для профессионалов». http://www.intuit.ru/studies/courses/98/98/info
- Сафонов А. Возможности Visual Studio 2013 и их использование для облачных вычислений. Лекция 1: Концепция современной интегрированной среды разработки приложений. http://www.intuit.ru/studies/courses/13996/1223/lecture/23386
- Котляров В. Основы тестирования программного обеспечения. http://www.intuit.ru/studies/courses/48/48/info

Положение о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся в ННГУ от 13.02.2014. http://www.unn.ru/site/images/docs/obrazov-org/Formi_stroki_kontrolya_13.02.2014.pdf

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий:

5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикатора достижения компетенций)	**Шкала оценивания сформированности компетенций**						
	плохо	**неудовлетворительно**	**удовлетворительно**	**хорошо**	**очень хорошо**	**отлично**	**превосходно**
	Не зачтено		Зачтено				
Знания	Отсутствие знаний теоретическо	Уровень знаний ниже минималь-ных	Минимально допустимый уровень знаний.	Уровень знаний в объеме, соответствующ	Уровень знаний в объеме, соответствующ	Уровень знаний в объеме, соответствую	Уровень знаний в объеме, превышающе

	го материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	требований. Имели место грубые ошибки.	Допущено много негрубых ошибки.	ем программе подготовки. Допущено несколько негрубых ошибок	ем программе подготовки. Допущено несколько несущественных ошибок	щем программе подготовки, без ошибок.	м программу подготовки.
Умения	Отсутствие минималь-ных умений. Невозмож-ность оценить наличие умений вследствие отказа обучающего-ся от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продемонстрированы все основные умения, решены все основные задачи с отдельными несущест-венным недочетами, выполнены все задания в полном объеме.	Продемонстрированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
Навыки	Отсутствие владения материалом. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки.	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами.	Продемонст-рированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов.	Продемонстрирован творческий подход к решению нестандартных задач.

Шкала оценки при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	Превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
	Удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже

		«удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	Неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

5.2.1 Контрольные вопросы

Вопрос	Код компетенции *(согласно РПД)*
1. Общее назначение инструментов, примеры.	ПК-2
2. Признаки "хороших" инструментов, с пояснениями.	ПК-2
3. Примеры практик Программной инженерии, их суть.	ПК-2
4. Приведите примеры инструментов, помогающих применять практики.	ПК-2
5. Диаграмма каскадной модели жизненного цикла.	ПК-2
6. Диаграмма работы программиста над задачей.	ПК-2
7. Определение системы контроля версий (СКВ)	ПК-2
8. Основные функции/возможности современных СКВ	ПК-2
9. Преимущества DVCS	ПК-2
10. Centralized Workflow (диаграмма, достоинства и недостатки)	ПК-2
11. Integration Manager Workflow (диаграмма, достоинства и недостатки)	ПК-2
12. Dictator and Lieutenants Workflow (диаграмма, достоинства и недостатки)	ПК-2
13. Модель ветвления GitFlow	ПК-2
14. Рабочий процесс (модель ветвления), используемый в компании GitHub	ПК-2
15. Базовые принципы корректной работы с СКВ	ПК-2
16. Простые истины планирования	ПК-2
17. Практические рекомендации при учете задач (issue tracking)	ПК-2
18. Что нельзя протестировать автоматически?	ПК-2

19. Классификация тестов по назначению.	ПК-2
20. Современная стратегия тестирования (основные 5 утверждений).	ПК-2
21. Основные возможности фреймворков модульного тестирования.	ПК-2
22. Критерии хорошего теста.	ПК-2
23. Возможности Google Test.	ПК-2
24. Порядок использования Google Test.	ПК-2
25. Определение непрерывной интеграции	ПК-2
26. Задачи выделенного сервера	ПК-2
27. Эволюция взглядов на непрерывную интеграцию	ПК-2
28. Travis CI, преимущества и недостатки	ПК-2
29. BuildBot, преимущества и недостатки	ПК-2
30. Определение интегрированной среды разработки (ИСР)	ПК-2
31. Отличия ИСР от редакторов исходного кода	ПК-2
32. Основные функции/возможности современных ИСР	ПК-2
33. История развития билд-систем	ПК-2
34. Плюсы и минусы Makefile	ПК-2
35. Плюсы и минусы CMake	ПК-2
36. Преимущества и недостатки простого текста.	ПК-2
37. Преимущества и недостатки бинарного формата.	ПК-2
38. Примеры ситуаций, когда удобно использовать TXT, XML, YAML, JSON.	ПК-2
39. Легковесные языки разметки. Примеры, назначение, преимущества и недостатки.	ПК-2
40. Синтаксис Markdown (заголовки, стили, списки, ссылки).	ПК-2
41. Примеры использования Markdown, в том числе нестандартные. В чем преимущество использования Markdown в каждой из этих ситуаций.	ПК-2
42. Как будет выглядеть команда в Vim для: – Создания 10 копий текущей строки	ПК-2

– Перевода всей строки в верхний регистр (капитализация)	
43. Предложите регулярное выражение для поиска: – Имен всех классов в вашем C++ проекте – Поиска дат в формате `2013-09-18` или `14-01-01` – IP-адресов, номеров банковских карт, HEX-представления чисел типа `int`	ПК-2
44. Предложите командную строку для: – Поиска всех вызовов виртуального метода в директории с исходниками – Печати всех заголовков первого и второго уровня в файле Markdown (#-нотация)	ПК-2
45. Приведите примеры внутренних документов	ПК-2
46. Приведите примеры внешних документов	ПК-2
47. Распределение ролей при работе над документацией	ПК-2
48. Виды автоматических проверок документации и способы их реализации	ПК-2
49. Популярные форматы внутренних документов, их достоинства и недостатки	ПК-2
50. Содержание/подразбиение README файлов	ПК-2
51. Ключевые принципы при работе с документацией	ПК-2
52. Какие преимущества дает автоматизация	ПК-2
53. Типичные классы задач на автоматизацию (в работе программиста)	ПК-2
54. Философия UNIX	ПК-2
55. Преимущества UNIX при автоматизации	ПК-2
56. Суть деятельности профессионального программиста	ПК-2

5.2.2. Типовые задания для оценки сформированности компетенции ПК-2

Темы практических работ

1. Настройка окружения разработки.

2. Проектирование программного интерфейса (API), создание пользовательской документации.
3. Разработка функциональности.
4. Интеграция кода в общий проект.
5. Создание модульных тестов.

6. Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Страуструп Б. Курс «Язык программирования C++ для профессионалов». http://www.intuit.ru/studies/courses/98/98/info
2. Сафонов А. Возможности Visual Studio 2013 и их использование для облачных вычислений. Лекция 1: Концепция современной интегрированной среды разработки приложений. http://www.intuit.ru/studies/courses/13996/1223/lecture/23386
3. Котляров В. Основы тестирования программного обеспечения. http://www.intuit.ru/studies/courses/48/48/info

б) Программное обеспечение и Интернет-ресурсы

1. Интегрированная среда разработки: Microsoft Visual Studio (лицензионное ПО), Eclipse (свободное ПО), Qt Creator (свободное ПО), Xcode (свободное ПО). Одна из перечисленных на выбор студента.
2. Клиент системы контроля версий Git: git-scm [https://git-scm.com] (свободное ПО), GitHub Desktop [https://desktop.github.com] (свободное ПО). Один из перечисленных на выбор студента.
3. GitHub [http://github.com] (свободный ресурс).
4. Утилита для сборки проектов CMake [http://cmake.org] (свободное ПО).
5. Google Test [https://github.com/google/googletest] (свободное ПО).
6. Travis CI [https://travis-ci.org].
7. Redmine [http://www.redmine.org] (свободное ПО).

7.Материально-техническое обеспечение дисциплины

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: операционная система Windows (лицензия), Microsoft Visual Studio (лицензия).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду

Программа составлена в соответствии с требованиями ОС ННГУ 02.03.02 Фундаментальная информатика и информационные технологии.

Автор К.В. Корняков

Рецензент (ы) _____

Заведующий кафедрой_____ Р.Г. Стронгин