МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Институт информационных технологий, математики и механики

УТВЕРЖДАЮ:

Директор _____ В.П. Гергель

«____» _____ 2018 г.

Рабочая программа дисциплины

Инструментальные средства разработки ПО

Уровень высшего образования
Бакалавриат

Направление подготовки / специальность
02.03.02 Фундаментальная информатика и информационные технологии

Направленность образовательной программы
Инженерия программного обеспечения

Квалификация (степень)
Бакалавр

Форма обучения
Очная

Нижний Новгород

2018

1. Место и цели дисциплины в структуре ОПОП

Место дисциплины в структуре ОПОП

Дисциплина «Инструментальные средства разработки ПО» предназначена для студентов 3-го курса бакалавриата (6 семестр), обучающихся по направлению 02.03.02 «Фундаментальная информатика и информационные технологии», относится к вариативной части ОПОП, является дисциплиной по выбору (Б1.В.ДВ.05.01) Дисциплина опирается на материал курсов «Основы программирования», «Алгоритмы и структуры данных».

Цель освоения дисциплины

Целью освоения дисциплины является овладение инструментальными средствами разработки программного обеспечения.

В курсе рассматриваются инструменты, которые используются программистами на всех стадиях жизненного цикла проектов по разработке ПО: кодирование, тестирование, документирование и так далее. Упор делается на кроссплатформенные инструменты, допускающие автоматизацию рутинных операций.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников)

Формируемые компетенции	**Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций**
Способность к ведению организационно-управленческой деятельности в области фундаментальной информатики и информационных технологий (ПК-2): *- способностью разрабатывать, оценивать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов информационных технологий, реализовывать методы и механизмы оценки и анализа функционирования средств и информационных технологий (ПК-2-1)* *- способность реализовывать процессы управления качеством производственной*	***ЗНАТЬ*** *З1 (ПК-2-1) Основные текстовые форматы.* *З2 (ПК-2-1) Понятие регулярного выражения, основные правила формирования регулярных выражений.* *З3 (ПК-2-1) Основные команды командной строки.* *З4 (ПК-2-2) Возможности скриптовых языков для автоматизации сборки и компиляции программ.* *З5 (ПК-2-2) Понятие интегрированной среды разработки и основных компонент среды разработки.* *З6 (ПК-2-3) Понятие системы контроля версий, ее назначение. Цикл разработки программ с использованием системы контроля версий.* *З7 (ПК-2-2) Процедуру организации отладки и тестирования программ.* *З8 (ПК-2-2) Инструменты инспекции кода.* ***УМЕТЬ*** *У1 (ПК-2-1) Работать с одним из текстовых редакторов, рассмотренном в лекционном материале.* *У2 (ПК-2-1) Работать с командной строкой.* *У3 (ПК-2-2) Разрабатывать скрипты для автоматизации сборки и компиляции программ.*

деятельности, связанной с созданием и использованием информационных технологий, осуществлять мониторинг и оценку качества процессов производственной деятельности (ПК-2-2); *- способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы (ПК-2-3).* *(базовый этап)*	*У4 (ПК-2-2) Работать с интегрированной средой разработки.* *У5 (ПК-2-3) Работать с системой контроля версий.* *У6 (ПК-2-2) Отлаживать и тестировать программы. Разрабатывать автоматические тесты.* *У7 (ПК-2-2) Использовать инструменты инспекции кода.* ***ВЛАДЕТЬ*** *В1 (ПК-2-1) Навыками работы с одним широко известных текстовых редакторов.* *В2 (ПК-2-1) Навыками работы с командной строкой.* *В3 (ПК-2-2) Навыками использования скриптовых языков для автоматизации сборки и компиляции программ.* *В4 (ПК-2-3) Навыками использования системы контроля версий Git.* *В5 (ПК-2-2) Навыками отладки и тестирования программ, а также разработки автоматических тестов на примере GoogleTest.* *В6 (ПК-2-2) Навыками использования инструментов инспекции кода на примере GitHub.*

3. Структура и содержание дисциплины «Инструментальные средства разработки ПО»

Объем дисциплины составляет 3 зачетные единицы, всего 108 часов, из которых 33 часа составляет контактная работа обучающегося с преподавателем:

16 часа занятия лекционного типа,

16 часа практических работ,

1 час мероприятия промежуточной аттестации.

На самостоятельную работу студента отводится 75 часов.

Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины, форма промежуточной аттестации по дисциплине	**Всего (часы)**	в том числе					
		контактная работа (работа во взаимодействии с преподавателем), часы из них					**Самостоятельная работа студента часы**
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	8	1	2			3	6
В т.ч. текущий контроль 2 ч.							
Промежуточная аттестация: Зачет.							

4. Образовательные технологии

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

5. Учебно-методическое обеспечение самостоятельной работы обучающихся

5.1. Виды самостоятельной работы студентов

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.

5.2. 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

6. Фонд оценочных средств для промежуточной аттестации по дисциплине, включающий:

6.1. Перечень компетенций выпускников образовательной программы с указанием результатов обучения (знаний, умений, владений), характеризующих этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Оценка уровня формирования компетенции ПК-2

Индикаторы компетенции	Критерии оценивания (дескрипторы)	Шкала оценивания
ЗНАТЬ *З1 (ПК-2-1) Основные текстовые форматы.* *З2 (ПК-2-1) Понятие регулярного выражения, основные правила формирования регулярных выражений.* *З3 (ПК-2-1) Основные команды командной строки.* *З4 (ПК-2-2) Возможности скриптовых языков для автоматизации сборки и компиляции программ.* *З5 (ПК-2-2) Понятие интегрированной среды разработки и основных компонент среды разработки.* *З6 (ПК-2-3) Понятие системы контроля версий, ее назначение. Цикл разработки программ с использованием системы контроля версий.* *З7 (ПК-2-2) Процедуру организации отладки и тестирования программ.* *З8 (ПК-2-2) Инструменты инспекции кода.* ***УМЕТЬ*** *У1 (ПК-2-1) Работать с одним из текстовых редакторов, рассмотренном в лекционном материале.* *У2 (ПК-2-1) Работать с командной строкой.* *У3 (ПК-2-2) Разрабатывать скрипты для автоматизации сборки и компиляции программ.*	Отсутствие знаний материала, отсутствие способности решения стандартных задач, полное отсутствие навыков, предусмотренных компетенцией.	Плохой уровень формирования компетенции. «Не зачтено»
	Наличие грубых ошибок в основном материале, отсутствие навыков, предусмотренных данной компетенцией	Неудовлетворительный уровень формирования компетенции. «Не зачтено».
	Знать некоторые основные понятия, изучаемые в рамках дисциплины. **Уметь** У1, У4 и У5 с погрешностями. **Владеть** некоторыми основными навыками, демонстрируя их в стандартных ситуациях.	Удовлетворительный уровень формирования компетенции. «Зачтено»
	Знать большинство основных понятий, изучаемых в рамках дисциплины (З1, З2, З3, З4, З5, З6, З7). **Уметь** (У1, У2, У3, У4, У5, У6) с незначительными погрешностями. **Владеть** основными навыками, демонстрируя их в стандартных ситуациях.	Хороший уровень формирования компетенции. «Зачтено»
	Знать основные понятия, изучаемые в рамках дисциплины (З1, З2, З3, З4, З5, З6, З7, З8). **Уметь** (У1, У2, У3, У4, У5, У6, У7) с незначительными погрешностями. **Владеть** всеми основными навыками, демонстрируя их в стандартных ситуациях, в том числе при решении дополнительных задач.	Очень хороший уровень формирования компетенции «Зачтено»
	Знать основной материал, предусмотренный компетенцией (З1, З2, З3, З4, З5, З6, З7, З8), без ошибок и погрешностей. **Уметь** (У1, У2, У3, У4, У5, У6, У7) в	Отличный уровень формирования компетенции «Зачтено»

У4 (ПК-2-2) Работать с интегрированной средой разработки. *У5 (ПК-2-3) Работать с системой контроля версий.* *У6 (ПК-2-2) Отлаживать и тестировать программы. Разрабатывать автоматические тесты.* *У7 (ПК-2-2) Использовать инструменты инспекции кода.* ***ВЛАДЕТЬ*** *В1 (ПК-2-1) Навыками работы с одним широко известных текстовых редакторов.* *В2 (ПК-2-1) Навыками работы с командной строкой.* *В3 (ПК-2-2) Навыками использования скриптовых языков для автоматизации сборки и компиляции программ.* *В4 (ПК-2-3) Навыками использования системы контроля версий Git.* *В5 (ПК-2-2) Навыками отладки и тестирования программ, а также разработки автоматических тестов на примере GoogleTest.* *В6 (ПК-2-2) Навыками использования инструментов инспекции кода на примере GitHub.*	полном объеме. **Владеть** всеми навыками, демонстрируя их не только в стандартных ситуациях, в том числе при решении нестандартных задач.	
	Знать основной материал, предусмотренный компетенцией (З1, З2, З3, З4, З5, З6, З7, З8), без ошибок и погрешностей. **Уметь** (У1, У2, У3, У4, У5, У6, У7) в полном объеме. **Владеть** всеми навыками, демонстрируя их не только в стандартных ситуациях, но и при решении нестандартных задач.	Превосходный уровень формирования компетенции «Зачтено»

6.2. Описание шкал оценивания

Результаты работы студентов оцениваются непрерывно в ходе семестра. При этом оценка сформированности компетенции ПК-2 в части «уметь» и «владеть» оценивается по результатам *выполнения практических работ*. Оценка сформированности компетенции ПК-2 в части «знать» оценивается по результатам письменных опросов (ответы на вопросы) и собеседования по заранее подготовленным вопросам к зачету. В семестре студент должен выполнить все практические работы и отчитаться по всем опросам.

Оценка	Уровень подготовки
Зачтено	Указаны минимальные требования для получения оценки «зачтено». Не хуже, чем минимально достаточный уровень подготовки. Обучаемый в значительной части отвечает на все вопросы, ошибки не носят грубого характера. В случае наличия задолженности по практикуму способен решать стандартные задачи, рассматриваемые в ходе семестра. Освоение материала на уровне не менее 35%.

Не зачтено	Подготовка не достаточна и требует дополнительного изучения материала. Студент дает ошибочные ответы, как на теоретические вопросы, так и на наводящие и дополнительные вопросы. Как правило, имеет задолженности по практикуму. Освоение материала на уровне менее 35%.

6.3. Критерии и процедуры оценивания результатов обучения по дисциплине (модулю), характеризующих этапы формирования компетенций

Для оценивания результатов обучения в виде знаний используются следующие процедуры и технологии:

– письменные или устные опросы в форме собеседования (ответы на вопросы).

Для оценивания результатов обучения в виде умений и владений используются следующие процедуры и технологии:

– практические работы, включающие постановку одной учебной задачи в виде краткой формулировки действий, которые следует выполнить, и описания результата, который нужно получить.

6.4. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенции.

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.

Список контрольных вопросов для проведения опросов с целью оценивания результатов обучения в виде знаний З1, З2, З3, З4, З5, З6, З7, З8 формирования ПК-2.

1. Общее назначение инструментов, примеры.

2. Признаки "хороших" инструментов, с пояснениями.

3. Примеры практик Программной инженерии, их суть.

4. Приведите примеры инструментов, помогающих применять практики.

5. Диаграмма каскадной модели жизненного цикла.

6. Диаграмма работы программиста над задачей.

7. Определение системы контроля версий (СКВ)

8. Основные функции/возможности современных СКВ

9. Преимущества DVCS

10. Centralized Workflow (диаграмма, достоинства и недостатки)

11. Integration Manager Workflow (диаграмма, достоинства и недостатки)

12. Dictator and Lieutenants Workflow (диаграмма, достоинства и недостатки)

13. Модель ветвления GitFlow

14. Рабочий процесс (модель ветвления), используемый в компании GitHub

15. Базовые принципы корректной работы с СКВ

16. Простые истины планирования

17. Практические рекомендации при учете задач (issue tracking)

18. Что нельзя протестировать автоматически?

19. Классификация тестов по назначению.

20. Современная стратегия тестирования (основные 5 утверждений).

21. Основные возможности фреймворков модульного тестирования.

22. Критерии хорошего теста.

23. Возможности Google Test.

24. Порядок использования Google Test.

25. Определение непрерывной интеграции

26. Задачи выделенного сервера

27. Эволюция взглядов на непрерывную интеграцию

28. Travis CI, преимущества и недостатки

29. BuildBot, преимущества и недостатки

30. Определение интегрированной среды разработки (ИСР)

31. Отличия ИСР от редакторов исходного кода

32. Основные функции/возможности современных ИСР

33. История развития билд-систем

34. Плюсы и минусы Makefile

35. Плюсы и минусы CMake

36. Преимущества и недостатки простого текста.

37. Преимущества и недостатки бинарного формата.

38. Примеры ситуаций, когда удобно использовать TXT, XML, YAML, JSON.

39. Легковесные языки разметки. Примеры, назначение, преимущества и недостатки.

40. Синтаксис Markdown (заголовки, стили, списки, ссылки).

41. Примеры использования Markdown, в том числе нестандартные. В чем преимущество использования Markdown в каждой из этих ситуаций.

42. Как будет выглядеть команда в Vim для:

    – Создания 10 копий текущей строки

    – 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`

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)

45. Приведите примеры внутренних документов

46. Приведите примеры внешних документов

47. Распределение ролей при работе над документацией

48. Виды автоматических проверок документации и способы их реализации

49. Популярные форматы внутренних документов, их достоинства и недостатки

50. Содержание/подразбиение README файлов

51. Ключевые принципы при работе с документацией

52. Какие преимущества дает автоматизация

53. Типичные классы задач на автоматизацию (в работе программиста)

54. Философия UNIX

55. Преимущества UNIX при автоматизации

56. Суть деятельности профессионального программиста

6.5. Методические материалы, определяющие процедуры оценивания.
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

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. 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

7.2. 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).

8. Материально-техническое обеспечение дисциплины (модуля)

Имеются в наличии учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа (компьютерный класс с установленным ПО: операционные системы семейства Microsoft Windows, лицензия по подписке Microsoft Imagine, среда разработки семейства Microsoft Visual Studio, лицензия по подписке Microsoft Imagine.), промежуточной аттестации, а также помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет». Учебная и научная литература, учебно-методические материалы, представленные в библиотечном фонде, в электронных библиотеках.

Программа составлена в соответствии с требованиями ОС ННГУ с учетом рекомендаций и ОПОП ВО по направлению 02.03.02 Фундаментальная информатика и информационные технологии, профиль Инженерия программного обеспечения

Автор К.В. Корняков

Заведующий кафедрой_____ Р.Г. Стронгин

Программа одобрена методической комиссией Института информационных технологий, математики и механики ННГУ им. Н.И. Лобачевского

от 20 июня 2018 года, протокол № 10