

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Физический факультет



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разработка надежного ПО на языке Rust

направление подготовки: **03.04.01 Прикладные математика и физика**
Профиль: **Прикладные математика и физика. Информационные процессы и системы**

Форма обучения: очная

Семестр	Общий объем	Виды учебных занятий (в часах)				Промежуточная аттестация (в часах)				
		Контактная работа обучающихся с преподавателем			Самостоятельная работа, не включая период сессии	Самостоятельная подготовка к промежуточной аттестации	Контактная работа обучающихся с преподавателем			
		Лекции	Практические занятия	Лабораторные занятия			Консультации	Зачет	Дифференцированный зачет	Экзамен
1	2	3	4	5	6	7	8	9	10	11
	1	108	32		48	26			2	

Всего 108 часов / 3 зачётные единицы, из них:
- контактная работа 82 часа

Компетенции ПК-2

Руководитель программы
д.ф.-м.н.

И. Б. Логашенко

Новосибирск, 2024

Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы	3
2. Место дисциплины в структуре образовательной программы	3
3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося	3
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий.....	4
5. Перечень учебной литературы	6
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся..	6
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	6
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине	7
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	7
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.....	8
Приложение 1 Аннотация по дисциплине	
Приложение 2 Оценочные средства по дисциплине	

1. Перечень планируемых результатов обучения по дисциплине, соотнесённых с установленными в программе индикаторами достижения компетенций

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
ПК-2. Способен осуществлять научно-технические разработки при исследовании самостоятельных тем.	ПК-2.1. Проводит работы по обработке и анализу научно-технической информации и результатов исследований. ПК-2.2. Разрабатывает планы и методические программы проведения исследований и разработок. ПК-2.3. Оформляет результаты опытно-конструкторских работ	Знать: состав и порядок принятия мер по обеспечению надежности функционирования АСУ ТП и предотвращению техногенных катастроф Уметь: создавать программы контроля и управления технологическими объектами, используя среды для разработки технологического программного обеспечения Владеть: технологиями создания, внедрения, эксплуатации и сопровождения АСУ ТП, языками технологического программирования (промышленный стандарт IEC-61131).

2. Место дисциплины в структуре образовательной программы

Дисциплина «Разработка надежного ПО на языке Rust» является одной из дисциплин по выбору по направлению подготовки 03.04.01 Прикладные математика и физика. Курс «Промышленные контроллеры» развивает знания, умения и навыки, сформированные у обучающихся по результатам общей базовой подготовки в рамках программ бакалавриата. Дисциплина дает магистранту необходимые знания, навыки и предоставляет инструменты для выполнения проектов на стыке областей физики, математики и информационных технологий.

3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Трудоемкость дисциплины – 3 з.е. (108 часов)

Форма промежуточной аттестации: 1 семестр – дифференцированный зачет

№	Вид деятельности	Семестр
		1
1	Лекции, час	32
2	Практические занятия, час	-
3	Лабораторные занятия, час	48
4	Занятия в контактной форме, час, из них	82
5	из них аудиторных занятий, час	80
6	в электронной форме, час	-
7	консультаций, час	-
8	промежуточная аттестация, час	2
9	Самостоятельная работа, час	26
10	Всего, час	108

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Лекции (32 часа)

Наименование темы и их содержание	Объем, час
Лекция 1. Общая классификация языков по назначению и модели исполнения. Общие свойства динамических языков (динамическая типизация, модель трансляции и исполнения)	2
Лекция 2. Язык Ruby: основные конструкции языка, коллекции. Функциональный стиль программирования в Ruby: блоки и замыкания. Итераторы. Реорганизующее присваивание.	6
Лекция 3. Особенности объектной модели Ruby: унифицированность объектного представления, модули и примеси, инкапсуляция. Динамическое изменение классов, элементы Meta-Object Protocol (MOP) в Ruby. JRuby и взаимодействие с Java, Java Scripting API	2
Лекция 4. Регулярные выражения. Классы символов. Жадные и нежадные выражения. Основные операции с регулярными выражениями.	2
Лекция 5. Классификация языков по парадигмам программирования. Функциональное программирование (ФП). Неподвижное состояние объекта как ключевое отличие ФП от ООП. Чистые функции, функции высших порядков. Функции, как объекты первого класса. Лексические контексты, анонимные функции, замыкания. Основные семейства функциональных языков. Историческая связь динамических и функциональных языков.	2
Лекция 6. Общие характеристики семейства языков Lisp: единое представление кода и данных, S-выражения, модель трансляции и исполнения, REPL. Язык Clojure, как современный представитель семейства Lisp: основные структуры языка. Компонентное тестирование в Clojure.	2
Лекция 7. Функциональные возможности Clojure: коллекции, реорганизующее присваивание, мемоизация, отложенные вычисления, бесконечные структуры данных. Абстрагирование данных с помощью функциональных примитивов (пары, числа Черча). Моделирование времени с помощью потоков. Символьные вычисления. Преимущества и недостатки ФП в сравнении с ООП.	2
Лекция 8. Императивные возможности Clojure. Software Transactional Memory. Многопоточность. Ссылки, атомы, агенты, переменные, виды транзакций. Взаимодействие с Java.	4

Лекция 9. Управляемая кодогенерация. Макросы в Lisp (на примере Clojure). Модель исполнения макросов. Макросы, как способ расширения языка.	2
Лекция 10. Понятие о проблемно-специфичных языках (DSL) и языках сценариев. Методы построения и генерации DSL.	2
Лекция 11. Динамические объектные модели. CLOS: обобщенный динамический полиморфизм, обобщенные функции и мультииметоды, вспомогательные методы. Реализация элементов CLOS в Clojure. Интроспекция, введение в МОР.	2
Лекция 12. Сквозная функциональность (cross-cutting concerns), проблема модульности. Традиционные методы обеспечения модульности в условиях сквозной функциональности	2
Лекция 13. Аспектно-ориентированное программирование (АОП). Динамические лексические контексты, их реализация в Clojure. Элементы АОП в CLOS.	2
Итого:	32

Лабораторные занятия (48 часов)

Содержание лабораторного занятия	Объем, час
Разбор теоретической части, решение заданий по темам: 1.1. Общая классификация языков по назначению и модели исполнения. Общие свойства динамических языков (динамическая типизация, модель трансляции и исполнения) 1.2. Язык Ruby: основные конструкции языка, коллекции. Функциональный стиль программирования в Ruby: блоки и замыкания. Итераторы. Реорганизующее присваивание. 1.3. Особенности объектной модели Ruby: унифицированность объектного представления, модули и примеси, инкапсуляция. Динамическое изменение классов, элементы Meta-Object Protocol (MOP) в Ruby. JRuby и взаимодействие с Java, Java Scripting API. 1.4. Регулярные выражения. Классы символов. Жадные и нежадные выражения. Основные операции с регулярными выражениями.	8
Разбор теоретической части, решение заданий по темам 2.1. Классификация языков по парадигмам программирования. Функциональное программирование (ФП). Неподвижное состояние объекта как ключевое отличие ФП от ООП. Чистые функции, функции высших порядков. Функции, как объекты первого класса. Лексические контексты, анонимные функции, замыкания. Основные семейства функциональных языков. Историческая связь динамических и функциональных языков. 2.2. Общие характеристики семейства языков Lisp: единое представление кода и данных, S-выражения, модель трансляции и исполнения, REPL. Язык Clojure, как современный представитель семейства Lisp: основные структуры языка. Компонентное тестирование в Clojure. 2.3. Функциональные возможности Clojure: коллекции, реорганизующее присваивание, мемоизация, отложенные вычисления, бесконечные структуры данных. Абстрагирование данных с помощью функциональных примитивов. Моделирование времени с помощью потоков. Символьные вычисления. Преимущества и недостатки ФП в сравнении с ООП.	16

2.4. Императивные возможности Clojure. Software Transactional Memory. Многопоточность. Ссылки, атомы, агенты, переменные, виды транзакций. Взаимодействие с Java.	
Разбор теоретической части, решение заданий по темам 3.1. Управляемая кодогенерация. Макросы в Lisp (на примере Clojure). Модель исполнения макросов. Макросы, как способ расширения языка. 3.2. Понятие о проблемно-специфичных языках (DSL) и языках сценариев. Методы построения и генерации DSL. 3.3. Динамические объектные модели. CLOS: обобщенный динамический полиморфизм, обобщенные функции и мультиметоды, вспомогательные методы. Реализация элементов CLOS в Clojure. Интроспекция, введение в МОР. 3.4. Сквозная функциональность (cross-cutting concerns), проблема модульности. Традиционные методы обеспечения модульности в условиях сквозной функциональности. 3.5. Аспектно-ориентированное программирование (АОП). Динамические лексические контексты, их реализация в Clojure. Элементы АОП в CLOS, 3.6. Применение АОП в проектирование. Преимущества и недостатки по сравнению с традиционными методами проектирования. Примеры задач, эффективно решаемых с помощью АОП.	24

Проведение семинарских занятий осуществляется в форме практической подготовки, предусматривающей участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью в области информационных технологий, связанных с проведением научных и практических работ.

Самостоятельная работа студентов (26 ч)

Перечень занятий на СРС	Объем, час
Подготовка к лабораторным занятиям.	16
Изучение теоретического материала, не освещаемого на лекциях	10

5. Перечень учебной литературы

1. Абельсон, Х. Структура и интерпретация компьютерных программ: [пер. с англ.] / Москва: Добросвет : КДУ, 2011. – 608с., ISBN 978-5-98227-829-6 (10 экз.)
2. Фултон, Х. Программирование на языке Ruby: руководство / Москва : ДМК Пресс, 2007. – 684с., ISBN 5-94074-357-9 (1 экз.)

6. Перечень учебно-методических материалов по самостоятельной работе обучающихся.

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет

7.1 Ресурсы сети Интернет

Для освоения дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет;
- «Российская национальная платформа открытого образования» (<http://openedu.ru/>), edX (www.edx.org);
- Веб-страницы ведущих международных центров СИ.

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС, электронную почту.

7.2 Современные профессиональные базы данных:

- Реферативно-поисковая база данных Reaxys (Elsevier)
- Реферативно-библиографическая база данных Scopus (Elsevier)
- Реферативно-библиографическая база данных Scifinder (Chemical Abstracts Service)
- Библиометрическая база данных Web of Science Core Collection (Thomson Reuters Scientific LLC.)
- База данных полнотекстовых научных журналов JSTOR.
- Электронная библиотека диссертаций Российской государственной библиотеки (ЭБД РГБ)
- Электронные ресурсы российской научной библиотеки eLibrary.ru
- Электронные ресурсы издательства American Chemical Society (ACS)
- Электронные ресурсы издательства Annual Reviews
- Электронные ресурсы Freedom Collection издательства Elsevier
- Электронные ресурсы издательства The Royal Society of Chemistry (RSC)
- Электронные ресурсы издательства Wiley

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

8.1 Перечень программного обеспечения

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий приложения для работы с документами и презентациями.

8.2 Информационные справочные системы

Не используются.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для реализации дисциплины используются специальные помещения:

- Учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации;
- Помещения для самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду НГУ.

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине

Перечень результатов обучения по дисциплине и индикаторов их достижения представлен в разделе 1.

10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль успеваемости:

Текущий контроль осуществляется в ходе семестра путем опроса в начале каждой лекции по материалам предыдущей лекции, а также проведения опроса студентов в начале каждого лабораторного занятия по теме, рассмотренной на предыдущем занятии.

Промежуточная аттестация:

Освоение компетенций оценивается согласно шкале оценки уровня сформированности компетенции. Положительная оценка по дисциплине выставляется в том случае, если заявленная компетенция ПК-2 сформирована не ниже порогового уровня в части, относящейся к формированию способности использовать специализированные знания в области технологии разработки программного обеспечения в профессиональной деятельности.

Окончательная оценка работы студента в течение семестра происходит на дифференцированном зачете. Дифференцированный зачет в конце семестра в устной форме. Вопросы подбираются таким образом, чтобы проверить уровень сформированности компетенции ПК-2.

Вывод об уровне сформированности компетенций принимается преподавателем. Каждый вопрос билета оценивается от 0 до 5 баллов. Положительная оценка ставится, когда

все компетенции освоены не ниже порогового уровня. Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине «Разработка надежного ПО на языке Rust»

Таблица 10.1

Код компетенции	Индикатор	Результат обучения по дисциплине	Оценочное средство
ПК-2	ПК-2.1. Проводит работы по обработке и анализу научно-технической информации и результатов исследований.	Знать: состав и порядок принятия мер по обеспечению надежности функционирования АСУ ТП и предотвращению техногенных катастроф	Дифференцированный зачет.
	ПК-2.2. Разрабатывает планы и методические программы проведения исследований и разработок.	Уметь: создавать программы контроля и управления технологическими объектами, используя среды для разработки технологического программного обеспечения	Работа на семинарских занятиях при обсуждении типовых ошибок, затруднений, а также идей по решению задач рамках дисциплины.
	ПК-2.3. Оформляет результаты опытно-конструкторских работ	Владеть: технологиями создания, внедрения, эксплуатации и сопровождения АСУ ТП, языками технологического программирования (промышленный стандарт IEC-61131).	Работа на семинарских занятиях при решении задач в рамках дисциплины.

Таблица 10.2

Критерии оценивания результатов обучения	Шкала оценивания
<p><u>Дифференцированный зачет:</u></p> <ul style="list-style-type: none"> – самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, отсутствие затруднений в объяснении процессов и явлений, а также при формулировке собственных суждений, – точность и корректность применения терминов и понятий, – наличие исчерпывающих ответов на дополнительные вопросы. 	<i>Отлично</i>

<p>При изложении ответа на вопрос(ы) преподавателя обучающийся мог допустить непринципиальные неточности.</p>	
<p><u>Дифференцированный зачет:</u></p> <ul style="list-style-type: none"> – самостоятельность, осмысленность, структурированность, логичность и аргументированность изложения материала, наличие затруднений в объяснении отдельных процессов и явления, а также при формулировке собственных суждений, – точность и корректность применения терминов и понятий при наличии незначительных ошибок, – наличие полных ответов на дополнительные вопросы с возможным присутствием ошибок. 	<i>Хорошо</i>
<p><u>Дифференцированный зачет:</u></p> <ul style="list-style-type: none"> – теоретический и фактический материал в слабой степени подкреплен ссылками на научную литературу и источники, – частичное понимание и неполное изложение причинно-следственных связей, – самостоятельность и осмысленность в изложении материала, наличие ошибок в логике и аргументации, в объяснении процессов и явлений, а также затруднений при формулировке собственных суждений, – корректность применения терминов и понятий, при наличии незначительных ошибок, – наличие неполных и/или содержащих существенные ошибки ответов на дополнительные вопросы. 	<i>Удовлетворительно</i>
<p><u>Дифференцированный зачет:</u></p> <ul style="list-style-type: none"> – фрагментарное и недостаточное представление теоретического и фактического материала, не подкрепленное ссылками на научную литературу и источники, – непонимание причинно-следственных связей, – отсутствие осмысленности, структурированности, логичности и аргументированности в изложении материала, – грубые ошибки в применении терминов и понятий, – отсутствие ответов на дополнительные вопросы. 	<i>Неудовлетворительно</i>

10.3 Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения

Пример задания для решения

1. Задан набор символов и число **n**. Опишите функцию, которая возвращает список всех строк длины **n**, состоящих из этих символов и не содержащих двух одинаковых символов, идущих подряд. Например, для символов 'a', 'b', 'c' и n=2 результат должен быть ("ab" "ac" "ba" "bc" "ca" "cb") с точностью до перестановки. Не допускается использовать циклы или рекурсию.

Примеры вопросов для изучения

- Интерпретация и компиляция. АОТ-, ЛИТ- компиляция. Компиляция в динамических языках.
- Виды типизации. Преимущества и недостатки различных видов типизации. Типизация в динамических языках.

- Семантика регулярных выражений. Основные операции с регулярными выражениями.

Вопросы для дифференцированного зачета

1. Основные характеристики и области применения динамических языков.
2. Интерпретация и компиляция. АОТ-, JIT- компиляция. Компиляция в динамических языках.
3. Виды типизации. Преимущества и недостатки различных видов типизации. Типизация в динамических языках.
4. Язык Ruby, классификация, основные реализации. Встроенные структуры данных.
5. Семантика регулярных выражений. Основные операции с регулярными выражениями.
6. Объектная модель Ruby.
7. Динамическое изменение объектной модели в Ruby: элементы Meta-Object Protocol.
8. Основные положения функциональной парадигмы программирования. Неподвижное состояние, преимущества и недостатки.
9. Понятие лексического контекста. Замыкания.
10. Побочные эффекты. Чистые функции. Преимущества и недостатки.
11. Функции как объекты первого класса. Функции высших порядков: функционал, оператор. Примеры. Операторы композиции и частичного применения (каррирования).
12. Основные функции преобразования коллекций: map, reduce, filter. Отличия от прямой итерации по коллекции. Примеры использования (на Ruby или Clojure).
13. Язык Clojure. Концепция LISP: код как данные, связь с АСД. Модель компиляции/исполнения. REPL.
14. Встроенные типы и структуры данных Clojure. Основные операции.
15. Основные управляющие структуры Clojure: вызов функции, ветвления, цикл. Связь рекурсии и цикла. Хвостовая рекурсия в Clojure. Императивные управляющие структуры: последовательное выполнение, doseq.
16. Генератор for в Clojure, связь с map/reduce/filter.
17. Мемоизация. Область и примеры применения.
18. Реорганизующее присваивание (destructuring) в Clojure. Использование при объявлении/вызове функции, в let.
19. Отложенные вычисления на примере Clojure. Ленивые последовательности, delay. Основные операции над ленивыми последовательностями.
20. Потоки данных. Моделирование состояния с помощью потоков. Примеры использования. Бесконечные потоки.
21. Квотирование (quote). Виды квотирования в Clojure. Обратные операции: unquote, eval.
22. Специальные формы. Макросы. Модель исполнения. Применение макросов.
23. Особенности разрешения символов в eval и макросах. Внутренние переменные в макросах.
24. Разделение ответственостей. Принцип KISS. Связь с модульностью и абстракцией.
25. Ответственности 2-го класса (cross-cutting concerns). Примеры. Способы разделения ответственостей.
26. Инверсия управления (принцип Голливуда). Примеры применения.
27. Внедрение зависимостей (Dependency Injection, DI). Элементарное DI. DI с использованием контейнера. Связь с порождающими шаблонами проектирования.
28. Формы управления параллелизмом без блокировок.

29. Atomic-типы. Atomic-ссылка в Clojure, основные операции. Агенты. Future, promise.
30. Транзакционная память. Multi-Version Concurrency Control. Реализация в Clojure: алгоритм выполнения транзакции, изоляция транзакций.
31. Понятие распределенной транзакции. CAP-теорема. Транзакции типа Copy-Modify-Merge.
32. Формы полиморфизма. Полиморфизм в динамических языках. Принцип подстановки Барбары Лисков (строгая формулировка). Интерпретация в контрактном программировании.
33. Полиморфизм в иерархиях с одиночным и множественным наследованием. Комбинация методов на примере CLOS.
34. Обобщенные функции и посылка сообщений. Диспетчеризация по нескольким параметрам. Примеры использования.
35. Вспомогательные методы. Комбинация в иерархии наследования на примере CLOS. Примеры использования для разделения ответственостей.
36. Command-Query Separation. Применение в функциональных и императивных объектных моделях.
37. Аспектно-ориентированное программирование. Перехват, способы реализации. Применение для разделения ответственостей.
38. Понятие динамического лексического контекста. Реализация в Clojure. Связь с АОП. Применение для разделения ответственостей.

Пример билета к дифференцированному зачету

1. Интерпретация и компиляция. АОТ-, JIT- компиляция. Компиляция в динамических
2. Аспектно-ориентированное программирование. Перехват, способы реализации. Применение для разделения ответственостей.

Оценочные материалы по промежуточной аттестации, предназначенные для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС ВО, хранятся на кафедре-разработчике РПД в электронном виде.

Оценочные материалы по промежуточной аттестации (приложение 2), предназначенные для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, хранятся на кафедре-разработчике РПД в печатном и электронном виде.

Лист актуализации рабочей программы дисциплины «Разработка надежного ПО на языке Rust»

Приложение 1

Аннотация рабочей программы дисциплины «Разработка надежного ПО на языке Rust»

направление подготовки: 03.04.01 Прикладные математика и физика

Профиль: Прикладные математика и физика. Информационные процессы и системы

Программа дисциплины «Разработка надежного ПО на языке Rust»

составлена в соответствии с ФГОС ВО по направлению подготовки 03.04.01 **Прикладные математика и физика**, а также задачами, стоящими перед Новосибирским государственным университетом по реализации Программы развития НГУ. Дисциплина реализуется кафедрой автоматизации физико-технических исследований физического факультета Федерального государственного автономного образовательного учреждения высшего образования «Новосибирский национальный исследовательский государственный университет» (НГУ) для обучающихся магистратуры.

Цель дисциплины – обучение теоретическим основам организации, проектирования, разработки, эксплуатации и сопровождения Автоматизированных Систем Управления Технологическими Процессами (АСУ ТП), а также получение начальных навыков разработки управляющего программного обеспечения на языках технологического программирования.

Дисциплина нацелена на формирование у обучающихся универсальной компетенции:

Результаты освоения образовательной программы (компетенции)	Индикаторы	Результаты обучения по дисциплине
<p>ПК-2. Способность проводить опытно-конструкторские разработки при исследовании самостоятельных тем.</p>	<p>ПК-2.1. Проводит работы по обработке и анализу научно-технической информации и результатов исследований.</p> <p>ПК-2.2. Разрабатывает планы и методические программы проведения исследований и разработок.</p> <p>ПК-2.3. Оформляет результаты опытно-конструкторских работ</p>	<p>Знать: состав и порядок принятия мер по обеспечению надежности функционирования АСУ ТП и предотвращению техногенных катастроф</p> <p>Уметь: создавать программы контроля и управления технологическими объектами, используя среды для разработки технологического программного обеспечения</p> <p>Владеть: технологиями создания, внедрения, эксплуатации и сопровождения АСУ ТП, языками технологического программирования (промышленный стандарт IEC-61131).</p>

Курс рассчитан на один семестр. Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные занятия, самостоятельная работа студентов и дифференцированный зачет.

Программой дисциплины предусмотрены следующие виды контроля:

Текущий контроль успеваемости: контроль посещения занятий обучающимся и работа на семинарских занятиях.

Промежуточная аттестация: дифференцированный зачет.

Общая трудоемкость рабочей программы дисциплины составляет **108** академических часов / **3** зачетные единицы.