

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
радиотехники и компьютерных
технологий**

Д.А. Гаврилов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Язык программирования Rust
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра перспективных вычислительных технологий
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Программу составил: В.А. Шиповский, преподаватель

Программа обсуждена на заседании кафедры перспективных вычислительных технологий 16.05.2023

Аннотация

В курсе рассматриваются основные вопросы, связанные с разработкой программного обеспечения на языке программирования Rust. Курс затрагивает основные понятия и концепции языка, рассмотрение элементов стандартной библиотеки, а также средств экосистемы языка для компиляции, сборки и тестирования кода.

1. Цели и задачи

Цель дисциплины

Формирование у студентов знаний и навыков в области разработки программ на языке программирования Rust.

Задачи дисциплины

- обучение студентов основным правилам и особенностям разработки программного обеспечения на языке Rust;
- обучение студентов основным интерфейсам, предоставляемым стандартной библиотекой языка;
- освоение студентами материала на уровне, достаточном для разработки собственных программ на языке программирования Rust.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- Основные конструкции языка Rust;
- особенности модели памяти языка Rust;
- возможности стандартной библиотеки;
- возможности экосистемы Rust.

уметь:

- Создавать программы, используя язык программирования Rust;
- организовывать и осуществлять тестирование кода средствами экосистемы Rust;
- осуществлять компиляцию и сборку кода на языке Rust.

владеть:

- Основными приемами программирования с использованием языка Rust;
- Возможностями систем сборки, анализа кода и тестирования.

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

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Введение в Rust. Общие концепции программирования.	4			2
2	Владение. Ссылки и указатели.	4			2
3	Контейнеры в стандартной библиотеке Rust.	2			1
4	Трейты. Времена жизни.	4			2
5	Организация кода в Rust. Пакеты, крейты, модули. Cargo – система сборки и менеджер пакетов Rust. Организация тестирования.	4			2
6	Функциональные возможности языка: итераторы и замыкания.	4			2
7	Параллельное программирование в Rust.	4			2
8	Асинхронное программирование в Rust.	4			2
Итого часов		30			15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 7 (Осенний)

1. Введение в Rust. Общие концепции программирования.

- История появления и развития Rust.

- Преимущества и недостатки языка.
- Установка окружения разработки. Hello world application.
- Переменные и изменяемость.
- Типы данных:
- Скалярные типы данных: целые и вещественные числа, логический тип и символы.
- Сложные типы данных: tuple & array.
- Функции.
- Управляющие конструкции: if, loop, while, for.
- Структуры.
- Перечисления и сопоставления с образцом.
- Обобщенные типы данных.
- Обработка ошибок.

2. Владение. Ссылки и указатели.

- Концепция владения.
- Правила владения.
- Borrow checker.
- Ссылки и заимствование.
- Drop.
- Указатель на данные в куче Box<T>.
- Указатели с подсчетом ссылок: Rc<T> & Arc<T>.
- Внутренняя изменяемость с помощью RefCell<T>.

3. Контейнеры в стандартной библиотеке Rust.

- Vec, VecDeque.
- LinkedList.
- BTreeMap, BTreeSet.
- HashMap, HashSet.
- BinaryHeap.
- String + &str.

4. Трейты. Времена жизни.

- Понятие трейта.
- Имплементация трейтов и ограничения типажа.
- Объекты трейтов и шаблоны.
- Понятие времени жизни.
- Higher-Rank Trait Bounds (HRTBs).

5. Организация кода в Rust. Пакеты, крейты, модули. Cargo – система сборки и менеджер пакетов Rust. Организация тестирования.

- Пакеты и крейты.
- Модули.
- Области видимости и ключевое слово use.
- Разделение модулей на разные файлы.
- Возможности cargo.
- Организация тестирования кода.

6. Функциональные возможности языка: итераторы и замыкания.

- Итераторы
- Замыкания (closures).

- Сравнение производительности: циклы и итераторы.

7. Параллельное программирование в Rust.

- Использование потоков для параллельного исполнения кода.
- Пересылка сообщений для передачи данных между потоками.
- Параллелизм с общим состоянием.
- Трейты Sync и Send. Расширяемый параллелизм.

8. Асинхронное программирование в Rust.

- Отложенное выполнение кода. Futures. Pin<Future>.
- Async / await.
- Tokio crate – де-факто стандарт асинхронного программирования на Rust.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор).

6.Перечень рекомендуемой литературы

Основная литература

1. Luca Palmieri: «Zero To Production In Rust: An introduction to backend development».
2. Tim McNamara: «Rust in Action»

Дополнительная литература

1. Mara Bos: «Rust Atomics and Locks», -O'Reilly

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

1. <https://doc.rust-lang.org/book/>
2. <https://doc.rust-lang.org/cargo/>
3. https://rust-lang.github.io/async-book/01_getting_started/01_chapter.html
4. <https://doc.rust-lang.org/rust-by-example/index.html>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Текстовый редактор, утилиты экосистемы Rust: rustc, cargo, rustup.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

При работе со студентами важную роль играет преподавательский контроль за освоением и усвоением ими учебного материала. Для этого непосредственно во время занятий устанавливается «обратная связь» со студентами. Преподаватель задает вопросы, направленные на контроль усвоения и понимание пройденного материала. При выявлении «тупиковых» ситуаций (правильных ответов нет или вообще нет ответов), материал обсуждается повторно.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Информатика и вычислительная техника
профиль подготовки:	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра перспективных вычислительных технологий
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 7 (осенний) - Дифференцированный зачет

Разработчик: В.А. Шиповский, преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

2. Показатели оценивания компетенций

В результате изучения дисциплины «Язык программирования Rust» обучающийся должен:

знать:

- Основные конструкции языка Rust;
- особенности модели памяти языка Rust;
- возможности стандартной библиотеки;
- возможности экосистемы Rust.

уметь:

- Создавать программы, используя язык программирования Rust;
- организовывать и осуществлять тестирование кода средствами экосистемы Rust;
- осуществлять компиляцию и сборку кода на языке Rust.

владеть:

- Основными приемами программирования с использованием языка Rust;
- Возможностями систем сборки, анализа кода и тестирования.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлой лекции или в конце занятия по пройденной теме.

Обучающемуся в течение курса будет предложено выполнить домашнюю работу - написать короткую программу, используя язык программирования Rust.

По результату выполнения и с учетом общей активности может быть проставлена оценка в соответствии с таблицей критериев оценивания.

Пример варианта домашней работы: "Реализовать многопоточный аналог консольной утилиты Linux «cat», используя язык программирования Rust".

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Концепция владения и правила владения.
2. Времена жизни.
3. Трейты.
4. Замыкания (closures).
5. Внутренняя изменяемость.
6. Организация многопоточности и примитивы синхронизации.
7. Асинхронное программирование.
8. Понятие трейта. Имплементация трейтов и ограничения типажа.
9. Организация кода в Rust. Пакеты, крейты, модули.
10. Параллельное программирование в Rust.

Критерии оценивания

Оценка отлично (10) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (9) выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка отлично (8) выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка хорошо (7) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка хорошо (6) выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка хорошо (5) выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка удовлетворительно (4) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка удовлетворительно (3) выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка неудовлетворительно (2) выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка неудовлетворительно (1) выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Дифференцированный зачет проводится в устной форме. При проведении дифференцированного зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 30 минут. Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины.