

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ

Директор института прикладной  
математики и компьютерных наук



А.В. Замятин

2022 г.

**Фонд оценочных средств по дисциплине:**

Языки программирования

Направление подготовки

**02.03.03 Математическое обеспечение и администрирование информационных систем**

*код и наименование направления подготовки*

**DevOps-инженерия в администрировании инфраструктуры ИТ-разработки**

*наименование специальности*

Томск–2021

ФОС составил(и):

канд. физ.-мат. наук, доцент

доцент кафедры компьютерной безопасности

старший преподаватель

кафедры теоретических

основ информатики ТГУ

С.И. Самохин

М.С. Овсянников

Рецензент:

канд. техн. наук, доцент,

зав. кафедрой компьютерной безопасности

С.А. Останин

Фонд оценочных средств одобрен на заседании учебно-методической комиссии института прикладной математики и компьютерных наук (УМК ИПМКИ)

Протокол от от 24 февраля 2022 г. № 01

Председатель УМК ИПМКИ,  
д-р техн. наук, профессор

С.И. Сущенко

**Фонд оценочных средств (ФОС)** является элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ФОС разрабатывается в соответствии с рабочей программой (РП) дисциплины и включает в себя набор оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине.

### 1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Компетенция	Индикатор компетенции	Код и наименование результатов обучения (планируемые результаты обучения, характеризующие этапы формирования компетенций)	Критерии оценивания результатов обучения			
			Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
ОПК-2 – способность применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной	ИОПК-2.1 Обладает необходимыми знаниями основных концепций современных вычислительных систем. ИОПК-2.2 Использует методы высокопроизводительных вычислительных технологий, современного программного обеспечения, в том числе отечественного происхождения. ИОПК-2.3 Использует инструментальные средства высокопроизводительных вычислений в научной и практической деятельности.	<b>Знать:</b> ОП-2.1.1. Освоить данные C# и Python. ОП-1.2.2. Ознакомиться с библиотеками numpy, pandas, matplotlib для работы с искусственным интеллектом на языке Python. <b>Уметь:</b> ОП-2.3.1. Освоить работу с online компиляторами как средствами редактирования, отладки, компиляции и выполнения программ.	Знает основы языков программирования C# и Python. Способен составить и отладить программу. Умеет пользоваться библиотеками Python. Умеет создавать пользовательские классы на языке C#. Владеет работой с online компиляторами как средствами редактирования, отладки, компиляции и выполнения программ.	Знает основы языков программирования C# и Python. Способен составить и отладить программу. Умеет пользоваться библиотеками Python. Умеет создавать пользовательские классы на языке C#. Владеет работой с online компиляторами как средствами редактирования, отладки, компиляции и выполнения программ. Однако, при создании программы	Знает основы языков программирования C# и Python. Способен составить и отладить программу. Умеет пользоваться библиотеками Python. Умеет создавать пользовательские классы на языке C#. Владеет работой с online компиляторами как средствами редактирования, отладки, компиляции и выполнения программ. Однако, при выборе средств и создании программ	Не знает основы языков программирования C# и Python. Не способен составить и отладить программу. Не умеет пользоваться библиотеками Python. Владеет работой с online компиляторами как средствами редактирования, отладки, компиляции и выполнения программ. Не умеет создавать пользовательские классы на языке C#.

				допускает неточности.	допускает значительные ошибки.	
ПК-1 – способность осуществлять программирование, тестирование и опытную эксплуатацию ИС с использованием технологических и функциональных стандартов, современных моделей и методов оценки качества и надежности программных средств	ИПК-1.3 Кодирует на языках программирования и проводит модульное тестирование ИС.	ОР-1.3.1. Научиться использовать библиотеки для работы с большими данными и искусственным интеллектом. ОР-1.3.2. Применять на практике структуры данных для хранения и обработки данных. ОР-1.3.3. Научиться создавать пользовательские классы на С#.	Уметь использовать библиотеки для работы с большими данными и искусственным интеллектом. Владеет навыками применения на практике структуры данных для хранения и обработки данных и созданием пользовательских классов на С#.	Уметь использовать библиотеки для работы с большими данными и искусственным интеллектом. Допускается наличие отдельных пробелов в умении. Владеет навыками применения на практике структуры данных для хранения и обработки данных и созданием пользовательских классов на С#. Допускается наличие отдельных пробелов во владении.	Частично освоенное умение использовать библиотеки для работы с большими данными и искусственным интеллектом. Частично освоенное владение навыками применения на практике структуры данных для хранения и обработки данных и созданием пользовательских классов на С#.	Отсутствие умений использовать библиотеки для работы с большими данными и искусственным интеллектом. Отсутствие владения навыками применения на практике структуры данных для хранения и обработки данных и созданием пользовательских классов на С#.

## 2. Этапы формирования компетенций и виды оценочных средств

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Раздел 1. Общее введение в теорию компиляции	ОР-2.1.1. ОР-2.2.1. ОР-2.3.1. ОР-1.3.1. ОР-1.3.2 ОР-1.3.3	Задание, лабораторная работа
2.	Раздел 2. Ликбез по компиляторам и принципам трансляции		Задание, лабораторная работа
3.	Раздел 3. Обзор современных языков		Задание, лабораторная работа
4.	Раздел 4. Языки программирования, парадигмы и классификация		Задание, лабораторная работа
5.	Раздел 5. Скриптовый язык Python 3		Задание, лабораторная работа
6.	Раздел 6. Скриптовый язык Javascript		Задание, лабораторная работа
7.	Раздел 7. Компилируемый язык с сборщиком мусора Go (golang)		Задание, лабораторная работа
8.	Раздел 8. Компилируемый язык Rust		Задание, лабораторная работа

## 3. Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

3.1. Типовые задания для проведения текущего контроля успеваемости по дисциплине

**Задание 1.** Составить программу на C# для перевода чисел из одной системы счисления в другую. Системы счисления произвольные. Для этого сначала реализуем алгоритм перевода чисел из системы счисления N в десятичную систему счисления, потом из десятичной переводим в систему счисления M.

Если  $N = 10$  или  $M = 10$  это частный случай, его надо проверить.

Составляем подробный отчёт, в котором должен быть код с подробными комментариями и скрины результатов выполнения программы во всех нужных случаях, граничных и обычных. Отчёт прикрепляем в Moodle, лучше в формате pdf.

**Задание 2.** Реализовать алгоритм Ханойские башни для любого N.

Прикрепляем подробный отчёт, в котором должен быть код с подробными комментариями и скрины результатов выполнения программы, здесь тоже подробно и с комментариями

**Задание 3.** Дано арифметическое выражение в виде строки. Найти его значение.

Прикрепляем подробный отчёт, в котором должен быть код с подробными комментариями и скрины результатов выполнения программы во всех нужных случаях, показать, как вычисляются выражения с одинарным минусом, скобками, ошибками.

**Задание 4.** Создать небольшой телеграм-бот для указанной рассылки, используя язык программирования Python. Подобрать нужные библиотеки. Прикрепляем подробный отчёт, в котором должен быть код с подробными комментариями и ссылка на разработанный бот.

**Задание 5. Междисциплинарная, задача из теории графов.** Реализовать алгоритм раскраски графа для графов большой размерности. Подобрать нужные библиотеки. Прикрепляем подробный отчёт, в котором должно быть обоснование выбранных средств реализации, код с подробными комментариями и скрины результатов.

### Примерные вопросы для устного опроса при текущем контроле

1. Какие виды типизации применяются в современных языках?
2. Как получить список всех атрибутов объекта в Python?
3. Что такое GIL и как он работает?
4. Чем отличается наследование в JavaScript от наследования в Python?
5. Какие средства управления исключительными ситуациями есть в Go?
6. Что такое и как применять channels и select statement?
7. Какие механизмы приведения типов доступны в Rust?
8. Что такое типаж (trait) и как его использовать в Rust?

Примеры заданий для лабораторных работ :

**Лабораторная работа №1. «Спамеры».** Цель работы – написать скрипт, выполняющий рекурсивный обход сайта (напр. [www.csd.tsu.ru](http://www.csd.tsu.ru)) и вывести без дубликатов все адреса электронной почты, содержащиеся на страницах. Для ускорения работы добавьте ограничитель на переходы (напр. 10) по ссылкам – сайт может содержать очень много страниц. Для извлечения email и url следует использовать регулярные выражения.

Базовый язык - Python 2.7 или Python 3.5. Требуется использовать библиотеки requests для http запросов и re для RegEx.

Второй язык - любой скриптовый с динамической типизацией (напр. Ruby, Lua, Perl, Javascript). Не допускается семейство .Net, языки на основе JVM и все компилируемые языки.

**Лабораторная работа №2 «Сисадмины».** Цель работы – Необходимо написать скрипт, обрабатывающий лог-файл Nginx и выводящий список IP адресов, с которых производились запросы. Адреса из общей подсети \24 необходимо группировать при выводе (напр. 10.40.0.4 и 10.40.0.231 относятся к одной подсети).

Базовый язык - Python 2.7 или Python 3.5. Требуется использовать библиотеку re для RegEx.

Второй язык - любой скриптовый с динамической типизацией (напр. Ruby, Lua, Perl, Javascript). Не допускается семейство .Net, языки на основе JVM и все компилируемые языки.

**Лабораторная работа №3 «Тусовщики».** Цель работы – Необходимо написать скрипт, генерирующий html страницу со списком мероприятий (событий), связанных с IT полученных с сайта [https://www.meetup.com/meetup\\_api/](https://www.meetup.com/meetup_api/).

Необходимо заранее зарегистрироваться и получить ключ для доступа API.

В скрипте константами задаете город (любой, где много событий, напр. Boston). Диапазон дат вычисляется автоматически в виде следующей недели относительно времени запуска скрипта.

Необходимо вывести на каждый день недели список событий в виде даты, заголовка, адреса и аннотации. Для фильтрации тематики используйте ключевые слова или темы (topics).

Базовый язык - Javascript. Допускается реализация как под node.js, так и в виде скрипта в браузере.

Второй язык - любой скриптовый с динамической типизацией (напр. Ruby, Lua, Perl, Python). Не допускается семейство .Net, языки на основе JVM и все компилируемые языки.

**Лабораторная работа №4 «Братство кольца».** Цель работы – написать программу, имитирующую сеть TokenRing. Требуется запустить N потоков, где каждый связан с последующим. Основной поток отдает первому потоку экземпляр структуры/класса Token (data:string, recipient:int). Потоки передают токен по цепочке, пока сообщение не достигнет адресата.

Базовый язык - Go (golang). Контрольный срок сдачи - 23:59 (UTC+7) 9 декабря.

Второй язык по выбору - любой компилируемый, с динамической сборкой мусора. Допускаются языки семейства .Net и JVM

**Лабораторная работа №5 «Пальцем в небо».** Цель работы – написать программу, реализующую расчет пересечений полупрямых с набором отрезков в двумерной плоскости. Отрезки не пересекаются, но вершины могут совпадать. Полупрямая пересекается с ближайшим к ней отрезком, т.е. имеет 1 или 0 пересечений.

Формат ввода – первая строка – координаты начала полупрямой и точка, через которую она проходит. На каждой следующей строке координаты отрезков. Каждая точка задается вещественным числом (разделитель – точка) через запятую. Пары точек разделены пробелом. Например:

1.5,2.7 2.896,3

2.68,3 4,8.6666661

Базовый язык - Rust. Контрольный срок сдачи - 23:59 (UTC+7) 9 декабря.

Второй язык по выбору - любой компилируемый, с явным выделением памяти и без динамической сборки мусора (напр. C, C++, D, Pascal).

Для промежуточной аттестации используется традиционная шкала оценивания.

Оценка	Критерии оценивания
--------	---------------------

Отлично	Обучающийся показал творческое отношение к обучению, в совершенстве овладел всеми теоретическими вопросами обязательного материала в области теории компиляции и принципов построения и классификации языков, показал все требуемые умения и навыки в работе с скриптовыми языками Python и Javascript, а также с компилируемыми языками Go и Rust
Хорошо	Обучающийся овладел всеми теоретическими вопросами обязательного материала в области теории компиляции и принципов построения и классификации языков, частично овладел навыками использования современных языков Python, Javascript, Go и Rust, показал основные умения и навыки в работе с онлайн интерпретаторами.
Удовлетворительно	Обучающийся имеет недостаточно глубокие знания по теоретическим разделам обязательного материала дисциплины, недостаточно владеет навыками разработки программ, показал не все основные умения и навыки в работе со скриптовыми языками Python и Javascript, а также с компилируемыми языками Go и Rust.
Неудовлетворительно	Обучающийся имеет существенные пробелы по отдельным теоретическим разделам специальной дисциплины и не владеет навыками написания программ на скриптовых и компилируемых языках.