

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ:
Директор

 А. В. Замятин

« 14 » июня 20 23 г.

Рабочая программа дисциплины

Языки программирования

по направлению подготовки

09.03.03 Прикладная информатика

Направленность (профиль) подготовки :

Разработка программного обеспечения в цифровой экономике

Форма обучения

Очная

Квалификация

Бакалавр

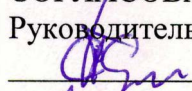
Год приема

2023

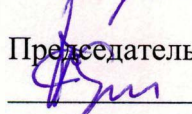
Код дисциплины в учебном плане: Б1.О.04.12

СОГЛАСОВАНО:

Руководитель ОП

 С.П. Сущенко

Председатель УМК

 С.П. Сущенко

Томск – 2023

1. Цель и планируемые результаты освоения дисциплины

Целью освоения дисциплины является формирование следующих компетенций:

– ОПК-2 – Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.

– ПК-1 – Способен осуществлять программирование, тестирование и опытную эксплуатацию ИС с использованием технологических и функциональных стандартов, современных моделей и методов оценки качества и надежности программных средств.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.3 Использует инструментальные средства высокопроизводительных вычислений в научной и практической деятельности.

ИОПК-2.2 Использует методы высокопроизводительных вычислительных технологий, современного программного обеспечения, в том числе отечественного происхождения.

ИОПК-2.1 Обладает необходимыми знаниями основных концепций современных вычислительных систем.

ИПК-1.3 Кодировать на языках программирования и проводить модульное тестирование ИС.

2. Задачи освоения дисциплины

Цель – научить студентов программировать с использованием компилируемых и интерпретируемых языков; научить студентов разрабатывать алгоритмические и программные решения на современных интерпретируемых и компилируемых языках, составлять программы, взаимодействующие с устройствами на низком уровне.

Задачи дисциплины:

- изучить классификации языков программирования по поколениям, парадигмам;
- освоить принципы построения программ с использованием различных концептуальных подходов;
- изучить принципы работы знаковых систем, основы функционирования вычислительных процессов;
- освоить основной функционал стандартных библиотек, предоставляющих алгоритмы и средства сетевого взаимодействия, API к БД, математических вычислений.

3. Место дисциплины в структуре образовательной программы

Дисциплина относится к обязательной части образовательной программы. Дисциплина входит в модуль Разработка программного обеспечения.

4. Семестр(ы) освоения и форма(ы) промежуточной аттестации по дисциплине

Седьмой семестр, экзамен

5. Входные требования для освоения дисциплины

Для успешного освоения дисциплины требуются результаты обучения по следующим дисциплинам: «Основы программирования», «Алгоритмы и структуры данных», «Объектно-ориентированное программирование».

6. Язык реализации

Русский

7. Объем дисциплины

Общая трудоемкость дисциплины составляет 4 з.е., 144 часов, из которых:

-лекции: 32 ч.

-практические занятия: 32 ч.

-практическая подготовка: 32 ч.

Объем самостоятельной работы студента определен учебным планом.

8. Содержание дисциплины, структурированное по темам

Тема 1. Общее введение в теорию компиляции

Классификация подходов при создании трансляторов. Компиляторы. Интерпретаторы. Виртуальные машины. Jit-компиляторы. Ассемблер и объектный код. Compile-ahead трансляторы;

Тема 2. Ликбез по компиляторам и принципам трансляции

Фазы компиляции. Лексический анализ. Синтаксический анализ. Семантический анализ. Промежуточные языки. Оптимизация. Генерация кода;

Тема 3. Обзор современных языков

Парадигмы языков программирования. Поколения языков программирования. Рейтинг ТЮВЕ. Динамика популярности языков с 2000г. Рейтинг RedMonk. Популярные языки по дивизионам. Популярные языки по применению. Перспективные языки;

Тема 4. Языки программирования, парадигмы и классификация

Языки с множественными парадигмами. Стандартизация языков. Интернационализация. Типы данных. Управление последовательностью действий. Управление памятью. Синтаксический сахар/соль;

Тема 5. Скриптовый язык Python 3

Обзор базовых типов Python 3. Списки, Comprehensions, операции map и filter. Словари и Dict Comprehensions. Функции и Lambda функции. Функции с переменным числом параметров. Ошибки и Exceptions. Форматирование строк. Объекты и классы. Методы, поля и property;

Тема 6. Скриптовый язык Javascript

Обзор интерпретаторов для браузеров и серверов. Базовые типы, строки, числа, NaN, undefined. Типы-объекты: Функции, Ошибки, Даты. Конструкторы и прототипы. Ajax запросы, XMLHttpRequest и fetch;

Тема 7. Компилируемый язык с сборщиком мусора Go (golang)

Типы данных. Функции. Конструкции потока управления. Структуры и «объекты». Методы и интерфейсы. Примитивы многозадачности: light-weight processes (goroutines), channels, и select statement;

Тема 8. Компилируемый язык Rust

Объявление переменных, mutable-данные и жизненный цикл. Базовые типы и вывод типов. Конструкции потока управления. Владение переменными. Классы, примеси и методы. Многозадачность без состояния гонки;

9. Текущий контроль по дисциплине

Текущий контроль по дисциплине проводится путем контроля посещаемости, проведения контрольных работ, выполнения лабораторных заданий и фиксируется

в форме контрольной точки не менее одного раза в семестр. Практическая подготовка оценивается по результатам выполненных практических работ.

10. Порядок проведения и критерии оценивания промежуточной аттестации

Теоретические и практические результаты формируются компетенциями ИОПК-2.1; ИОПК-2.2; ИОПК-2.3; ИПК-1.3 и результатами обучения:

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Тема 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 1, 2, 3, лабораторная работа
2.	Тема 2. Ликбез по компиляторам и принципам трансляции	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 1, 2, лабораторная работа
3.	Тема 3. Обзор современных языков	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 3, лабораторная работа
4.	Тема 4. Языки программирования, парадигмы и классификация	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 3, лабораторная работа
5.	Тема 5. Скриптовый язык Python 3	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 4, 5. лабораторная работа
6.	Тема 6. Скриптовый язык Javascript	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 4, лабораторная работа
7.	Тема 7. Компилируемый язык с сборщиком мусора Go (golang). Тема 8. Компилируемый язык Rust	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 2, лабораторная работа

Форма промежуточной аттестации – экзамен.

Оценка по дисциплине вычисляется как среднеарифметическая по итогам текущего контроля успеваемости (по результатам выполнения пяти лабораторных работ, каждая на 2 языках). Для получения положительной оценки необходимо сдать не менее 4 лабораторных работ.

Текущий контроль успеваемости осуществляется на протяжении семестра путем проведения устного опроса при сдаче лабораторных работ. Для всех лабораторных работ установлены предельные сроки их сдачи, при их превышении оценка снижается на 0.5 балла.

Период текущей аттестации	Виды текущей/промежуточной аттестации	Влияние оценки текущей аттестации на оценку промежуточной аттестации
Контрольная точка 2 (10 неделя семестра)	Лабораторные работы №1 и №2	Предельный срок сдачи лабораторных работ №1 и №2 без снижения оценки
Контрольная точка 3 (15 неделя семестра)	Лабораторные работы №3, №4 и №5	Предельный срок сдачи лабораторных работ №3, №4 и №5 без снижения оценки
Сессия (20-21 недели семестра)	Экзамен	Заключается в возможности защиты реферата и сдачи лабораторных работ по выбору для улучшения итоговой оценки

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) 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/.

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 строчка – координаты начала полупрямой и точка, через которую она проходит. На каждой следующей строчке 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.

11. Учебно-методическое обеспечение

- а) Электронный учебный курс по дисциплине в электронном университете «Moodle»
- б) Оценочные материалы текущего контроля и промежуточной аттестации по дисциплине.

12. Перечень учебной литературы и ресурсов сети Интернет

- а) основная литература:
 - Альфред В. Ахо, Миника С. Лам, Рави Сети, Джеффри Д. Ульман Компиляторы : принципы, технологии и инструментарий [пер. с англ. и общ. ред. И. В. Красикова]. – Москва [и др.] : Вильямс , 2011, 1175 с.: рис.
 - Вирт Н. Построение компиляторов [пер. с англ. Борисов Е. В., Чернышов Л. Н.]. – М.: ДМК Пресс , 2010, 190, [1] с.: ил. 1 электрон. опт. диск– ...
- б) дополнительная литература:
 - А. Ахо, Р. Сети, Д. Ульман Компиляторы. Принципы, технологии, инструменты.– М.: "Вильямс", 2003, 768 с.
 - Мозговой М.В. – Классика программирования. Алгоритмы, языки, автоматы, компиляторы. Практический подход. – СПб.: Наука и Техника, 2006. 320с.
 - Fernández M. Programming Languages and Operational Semantics.[Электронный ресурс]/ . –London:Springer London: Imprint: Springer, 2014. 209 p. 10 illus.: online resource. – URL: <http://dx.doi.org/10.1007/978-1-4471-6368-8/> (дата обращения: 14.10.2016).
- в) ресурсы сети Интернет:
 - открытые онлайн-курсы
 - Журнал «Эксперт» - <http://www.expert.ru>
 - Официальный сайт Федеральной службы государственной статистики РФ - www.gsk.ru
 - Официальный сайт Всемирного банка - www.worldbank.org
 - Общероссийская Сеть КонсультантПлюс Справочная правовая система. <http://www.consultant.ru>

13. Перечень информационных технологий

а) лицензионное и свободно распространяемое программное обеспечение:

При осуществлении образовательного процесса используется онлайн интерпретатор языков Python3 и Javascript – Repl.it, онлайн интерпретатор языка Javascript – Tryit, онлайн интерпретатор языка Go – play.golang.org, онлайн интерпретатор языка Rust – play.rust-lang.org.

б) информационные справочные системы:

– Электронный каталог Научной библиотеки ТГУ – <http://chamo.lib.tsu.ru/search/query?locale=ru&theme=system>

– Электронная библиотека (репозиторий) ТГУ – <http://vital.lib.tsu.ru/vital/access/manager/Index>

– ЭБС Лань – <http://e.lanbook.com/>

– Электронно-библиотечная система Znanium.com [Электронный ресурс] / Научно-издательский центр Инфра-М. – Электрон. дан. – М., 2016-. URL: <http://znanium.com/>

14. Материально-техническое обеспечение

Аудитории для проведения занятий лекционного типа.

Аудитории для проведения лабораторных занятий, индивидуальных и групповых консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы, оснащенные компьютерной техникой и доступом к сети Интернет, в электронную информационно-образовательную среду и к информационным справочным системам.

15. Информация о разработчиках

Овсянников Михаил Сергеевич, старший преподаватель кафедры теоретических основ информатики ТГУ.