

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук



Языки программирования

рабочая программа дисциплины

Закреплена за кафедрой	<i>теоретических основ информатики</i>
Учебный план	<i>02.03.03 Математическое обеспечение и администрирование информационных систем, профиль «DevOps-инженерия в администрировании инфраструктуры ИТ-разработки»</i>
Форма обучения	<i>очная</i>
Общая трудоёмкость	<i>4 з.е.</i>
Часов по учебному плану	<i>144</i>
в том числе:	
аудиторная контактная работа	<i>54,7</i>
самостоятельная работа	<i>89,3</i>
Вид(ы) контроля в семестрах	
<i>экзамен/зачет/зачет с оценкой</i>	<i>Семестр 7 – экзамен</i>

Программу составил:
старший преподаватель кафедры теоретических основ информатики М.С. Овсянников

Рецензент:
д-р техн. наук, профессор,
профессор кафедры теоретических основ информатики Ю.Л. Костюк

Рабочая программа дисциплины «Языки программирования» разработана в соответствии с самостоятельно устанавливаемым образовательным стандартом высшего образования – бакалавриат – федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский Томский государственный университет» по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (Утвержден Ученым советом НИ ТГУ, протокол от 27.10.2021 г. № 08).

Рабочая программа одобрена на заседании теоретических основ информатики

Протокол от 17 февраля 2022 г. №02

Заведующий кафедрой теоретических основ информатики
д-р техн. наук, профессор А.В. Замятин

Рабочая программа одобрена на заседании учебно-методической комиссии института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от от 24 февраля 2022 г. № 01

Председатель УМК ИПМКН,
д-р техн. наук, профессор С.П. Суценко

Цель 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.

1. Место дисциплины в структуре ОПОП

Дисциплина «Языки программирования» относится к обязательной части Блока 1 «Дисциплины», входит в модуль «Разработка программного обеспечения».

Для освоения дисциплины необходимо знать основы программирования.

Пререквизиты дисциплины: «Основы программирования», «Алгоритмы и структуры данных», «Объектно-ориентированное программирование».

Постреквизиты дисциплины: «Разработка приложений для мобильных платформ»

2. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Таблица 1.

Компетенция	Индикатор компетенции	Код и наименование результатов обучения (планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций)
ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности ОПК	ИОПК-2.1 Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем - ИОПК-2.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий - ИОПК-2.3 Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности -	Знать: ОП-2.1.1. Освоить данные C# и Python. ОП-2.2.1. Ознакомиться с библиотеками numpy, pandas, matplotlib для работы с искусственным интеллектом на языке Python. Уметь: ОП-2.3.1. Освоить работу с online компиляторами как средствами редактирования, отладки, компиляции и выполнения программ. ОП-1.3.1. Научиться использовать библиотеки для работы с большими данными и искусственным интеллектом. ОП-1.3.2. Применять на практике структуры данных для хранения и обработки данных. ОП-1.3.3. Научиться создавать пользовательские классы на C#.
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 ПК	ИПК-1.3 Кодирует на языках программирования и проводит модульное тестирование ИС -	

3. Структура и содержание дисциплины

3.1. Структура и трудоемкость видов учебной работы по дисциплине

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часов.

Таблица 2.

Вид учебной работы	Трудоемкость в академических часах	
	7 семестр	всего
Общая трудоемкость	144	144
Контактная работа:	54,7	54,7
Лекции (Л):	16	16
Практики (ПЗ)	32	32
Лабораторные работы (ЛР)		
Семинары (СЗ)		
Групповые консультации	2	2
Индивидуальные консультации	0,4	0,4
Промежуточная аттестация	4,3	4,3
Самостоятельная работа обучающегося:	89,3	89,3
- изучение учебного материала	23	23
- подготовка к практическим занятиям	23	23
- подготовка к рубежному контролю по теме	5,6	5,6
- подготовка реферата	6	6
- подготовка к рубежному контролю по теме/разделу	31,7	31,7
Вид промежуточной аттестации (зачет, зачет с оценкой, экзамен)	Экзамен	Экзамен

3.2. Содержание и трудоемкость разделов дисциплины

Таблица 3.

Код занятия	Наименование разделов и тем и их содержание	Вид учебной работы, занятий, контроля	С е м е с т р	Часы в электронной форме	Всего (час.)	Литература	Коды результатов обучения
	Раздел 1. Общие введение в теорию компиляции		7			1,2,3,4,5,6	
1.1.	1. Классификация подходов при создании трансляторов; 2. Компиляторы; 3. Интерпретаторы	Лекции			1		ОР-2.1.1, ОР-2.2.1, ОР-2.3.1.
1.2.	1. Виртуальные машины; 2. Jit-компиляторы; 3. Ассемблер и объектный код; 4. Compile-ahead трансляторы;	Лекции			1		ОР-2.1.1, ОР-2.2.1, ОР-2.3.1.
1.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к рубежному контролю по теме; подготовка реферата				2		
	Текущий контроль успеваемости	Письменный опрос					
	Раздел 2. Ликбез по компиляторам и принципам трансляции		7			1,2,3,4,5,6	
2.1.	1. Фазы компиляции 2. Лексический анализ 3. Синтаксический анализ	Лекции			1		ОР-2.1.1, ОР-2.2.1, ОР-2.3.1.
2.2.	1. Семантический анализ 2. Промежуточные языки 3. Оптимизация 4. Генерация кода	Лекции			1		ОР-2.1.1, ОР-2.2.1, ОР-2.3.1.
2.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к рубежному контролю по теме; подготовка реферата				2		
	Текущий контроль успеваемости	Письменный опрос					
	Раздел 3. Обзор современных языков		7			1,2,3,4,5,6	
3.1.	1. Парадигмы языков программирования	лекции			1		ОР-2.1.1, ОР-2.2.1,

	2. Поколения языков программирования 3. Рейтинг ТЮВЕ 4. Динамика популярности языков с 2000г. 5. Рейтинг RedMonk 6. Популярные языки по дивизионам 7. Популярные языки по применению 8. Перспективные языки	практическая работа			4		ОП-2.3.1.
3.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата				2		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос					
	Раздел 4. Языки программирования, парадигмы и классификация		7			1,2,3,4,5,6	
4.1.	1. Языки с множественными парадигмами 2. Стандартизация языков 3. Интернационализация	лекции			1		ОП-2.1.1, ОП-2.2.1, ОП-2.3.1.
		практическая работа			4		
4.2.	1. Типы данных 2. Управление последовательностью действий 3. Управление памятью 4. Синтаксический сахар/соль	лекции			2		ОП-2.1.1, ОП-2.2.1, ОП-2.3.1.
		практическая работа			4		
4.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата				3		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос					
	Раздел 5. Скриптовый язык Python 3		7			1,2,3,4,5,6	
5.1.	1. Обзор базовых типов Python 3 2. Списки, Comprehensions, операции map и filter 3. Словари и Dict Comprehensions 4. Функции и Lambda функции 5. Функции с переменным числом параметров 6. Ошибки и Excerptions 7. Форматирование строк 8. Объекты и классы 9. Методы, поля и property	лекции			2		ОП-2.1.1, ОП-2.2.1, ОП-2.3.1.
		практическая работа			5		

5.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата				3		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос					
	Раздел 6. Скриптовый язык Javascript		7			1,2,3,4,5,6	
6.1.	1. Обзор интерпретаторов для браузеров и серверов 2. Базовые типы, строки, числа, NaN, undefined 3. Типы-объекты: Функции, Ошибки, Даты 4. Конструкторы и прототипы Ajax запросы, XMLHttpRequest и fetch	лекции			2		OP-2.1.1, OP-2.2.1, OP-2.3.1.
		практическая работа			5		
6.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата				3		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос					
	Раздел 7. Компилируемый язык с сборщиком мусора Go (golang)		7			1,2,3,4,5,6	
7.1.	1. Типы данных 2. Функции 3. Конструкции потока управления 4. Структуры и «объекты» 5. Методы и интерфейсы 6. Примитивы многозадачности: light-weight processes (goroutines), channels, и select statement	лекции			2		OP-2.1.1, OP-2.2.1, OP-2.3.1.
		практическая работа			5		
7.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата				3		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос					
	Раздел 8. Компилируемый язык Rust		7			1,2,3,4,5,6	
8.1.	1. Объявление переменных, mutable-данные и жизненный цикл	лекции			1		OP-2.1.1, OP-2.2.1,

	2. Базовые типы и вывод типов 3. Конструкции потока управления 4. Владение переменными 5. Классы, примеси и методы Многозадачность без состояния гонки	практическая работа			5		ОР-2.3.1.
8.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата				3,6		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос.			42,7		
	Промежуточная аттестация	экзамен			31,7		

4. Образовательные технологии, учебно-методическое и информационное обеспечение для освоения дисциплины

Теоретический материал по дисциплине дается в виде лекций с применением стандартных средства демонстрации мультимедиа в формате .ppt. На лабораторных занятиях студенты реализуют предложенные алгоритмы вычислительной математики. Текущий контроль по лабораторным работам осуществляется в виде обсуждения алгоритма и результатов его работы.

Самостоятельная работа студентов по дисциплине организуется в следующих формах:

- 1) самостоятельное изучение основного теоретического материала, ознакомление с дополнительной литературой, Интернет-ресурсами;
- 2) подготовка к выполнению лабораторных работ.

Текущий контроль по лабораторным работам осуществляется в виде обсуждения алгоритма и результатов его работы. Текущий контроль успеваемости по теоретическому материалу осуществляется на контрольных неделях семестра.

Итоговая оценка выставляется как среднеарифметическое по результатам контрольных и лабораторных работ с округлением до ближайшего целого.

4.1. Рекомендуемая литература и учебно-методическое обеспечение

№ п/п	Авторы / составители	Заглавие	Издательство	Год издания, количество страниц
Основная литература				
1.	Альфред В. Ахо, Миника С. Лам, Рави Сети, Джеффри Д. Ульман	Компиляторы: принципы, технологии и инструментарий	Москва [и др.]: Вильямс	2011 г.
2.	Вирт Н.	Построение компиляторов	М.: ДМК Пресс	2010 г.
4.	А. Ахо, Р. Сети, Д. Ульман	Компиляторы. Принципы, технологии, инструменты	М.: "Вильямс",	2003 г.
5.	Мозговой М.В. –	Классика программирования. Алгоритмы, языки, автоматы, компиляторы. Практический подход.	СПб.: Наука и Техника	2006 г.
6.	Fernández M. Programming Languages and Operational Semantics	Programming Languages and Operational Semantics.	London:Springer London: Imprint: Springer	2014 г.

4.2. Базы данных и информационно-справочные системы, в том числе зарубежные

1. Электронная библиотека (репозиторий) ТГУ [Электронный ресурс] / Электронная библиотека (репозиторий) ТГУ: [сайт]. – [Томск, 2011–2016]. – URL: <http://vital.lib.tsu.ru/vital/access/manager/Index>.

4.3. Перечень лицензионного и программного обеспечения

MS Windows; MS Office. 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.

4.4. Оборудование и технические средства обучения

Для реализации дисциплины необходимы лекционные аудитории и аудитории для проведения практических занятий. Специальные технические средства (проектор, компьютер и т.д.) требуются для демонстрации материала в рамках изучаемых разделов, проведения защиты проектов в конце семестра. Вся основная и дополнительная литература, необходимая для самостоятельной работы и подготовки к экзамену, имеется в научной библиотеке ТГУ.

5. Методические указания обучающимся по освоению дисциплины

Лекционный курс и практические задания по курсу Языки программирования [Электронный ресурс] / Электронный университет – MOODLE. – ТГУ 2021. – URL: <https://moodle.tsu.ru/course/view.php?id=2162/>.

Сопроводительные материалы в локальной сети факультета информатики в каталоге X:\Workspace\Бакалавриат\Языки программирования

Лабораторные работы.

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 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).

6. Преподавательский состав, реализующий дисциплину

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

7. Язык преподавания – русский язык.