

МИНОБРНАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ

Директор института прикладной
математики и компьютерных наук

А.В. Замятин

« 04 » *исафта* 2022 г.

Языки программирования

рабочая программа дисциплины

Закреплена за кафедрой	<i>Теоретических основ информатики</i>
Учебный план	<i>02.03.02 Фундаментальная информатика и информационные технологии, профиль «Искусственный интеллект и разработка программных продуктов»</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 контроля в семестрах	
экзамен/зачет/зачет с оценкой	<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



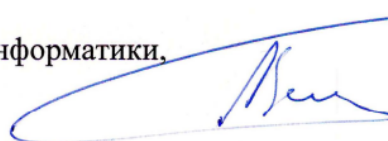
Ю.Л. Костюк

Рабочая программа дисциплины «Языки программирования» разработана в соответствии с образовательным стандартом высшего образования – бакалавриат, самостоятельно устанавливаемым федеральным государственным автономным образовательным учреждением высшего образования «Национальный исследовательский Томский государственный университет» по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии (Утвержден Ученым советом НИ ТГУ, протокол от 27.10.2021 г. № 08).

Рабочая программа одобрена на заседании кафедры 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 24 февраля 2022 г. № 01

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



С.П. Сущенко

Цель освоения дисциплины – 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».

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 универсальной компетенции	Код и наименование результатов обучения (планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций)
<p>ОПК-2. Способен применять компьютерные/супер компьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной</p>	<p>ИОПК-2.1. Обладает необходимыми знаниями основных концепций современных вычислительных систем</p> <p>ИОПК-2.2. Использует методы высокопроизводительных вычислительных технологий, современного программного обеспечения, в том числе отечественного происхождения</p> <p>ИОПК-2.3. Использует инструментальные средства высокопроизводительных вычислений в научной и практической деятельности</p>	<p>ОР-2.1.1 Обучающийся будет: - Знать классификации языков программирования по поколениям, парадигмам, принципы построения программ с использованием различных концептуальных подходов.</p> <p>ОР-2.2.1 Обучающийся будет: - Знать принципы работы знаковых систем, основы функционирования вычислительных процессов, основной функционал стандартных библиотек, предоставляющих алгоритмы и средства сетевого взаимодействия, API к БД, математических вычислений</p> <p>ОР-2.2.2 Обучающийся сможет: - программировать с использованием компилируемых и интерпретируемых языков;</p> <p>ОР-2.3.1 Обучающийся сможет: - разрабатывать алгоритмические и программные решения на современных интерпретируемых и компилируемых языках, составлять программы, взаимодействующие с устройствами на низком уровне.</p>
<p>ПК-1. Способен осуществлять программирование, тестирование и опытную</p>	<p>ИПК-1.3. Кодировать на языках программирования и проводит модульное тестирование ИС</p>	<p>ОР-1.3.1 Обучающийся сможет: - кодировать на языках программирования и проводить модульное тестирование ИС</p>

эксплуатацию ИС с использованием технологических и функциональных стандартов, современных моделей и методов оценки качества и надежности программных средств		
--	--	--

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		
Контактная работа:	54,7	54,7
Лекции (Л):	16	16
Практики (ПЗ)	32	32
Лабораторные работы (ЛР)	0	0
Семинары (СЗ)	0	0
Групповые консультации	4,4	4,4
Индивидуальные консультации	0	0
Промежуточная аттестация	2,3	2,3
Самостоятельная работа обучающегося:	89,3	89,3
- изучение учебного материала	17,6	17,6
- подготовка к практическим занятиям	20	20
- подготовка к рубежному контролю по теме	20	20
- подготовка к рубежному контролю по дисциплине	31,7	31,7
Вид промежуточной аттестации - 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. Общее введение в теорию компиляции		7			1,2,3,4,5	
1.1.	1. Классификация подходов при создании трансляторов; 2. Компиляторы; 3. Интерпретаторы	Лекции			1		ОР-2.1.1
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
1.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к рубежному контролю по теме; подготовка реферата	СРС			8		
	Текущий контроль успеваемости	Письменный опрос			1		
	Раздел 2. Ликбез по компиляторам и принципам трансляции		7			1,2,3,4,5	
2.1.	1. Фазы компиляции 2. Лексический анализ 3. Синтаксический анализ	Лекции			1		ОР-2.1.1 ОР-2.2.1
2.2.	1. Семантический анализ 2. Промежуточные языки 3. Оптимизация 4. Генерация кода	Лекции			1		ОР-2.2.2
2.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к рубежному контролю по теме; подготовка реферата	СРС			8		
	Текущий контроль успеваемости	Письменный опрос			1		

	Раздел 3. Обзор современных языков		7			1,2,3,4,5	
3.1.	1. Парадигмы языков программирования 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	лекции			1		OP-2.2.1 OP-2.2.2
		практическая работа			1		
3.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата	СРС			8		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос			4		
	Раздел 4. Языки программирования, парадигмы и классификация		7			1,2,3,4,5	
4.1.	1. Языки с множественными парадигмами 2. Стандартизация языков 3. Интернационализация	лекции			1		OP-2.1.1 OP-2.2.1
		практическая работа			1		
4.2.	1. Типы данных 2. Управление последовательностью действий 3. Управление памятью 4. Синтаксический сахар/соль	лекции			2		OP-2.2.2 OP-2.3.1
		практическая работа			2		
4.3.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата	СРС			8		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос			3		

	Раздел 5. Скриптовый язык Python 3		7			1,2,3,4,5	
5.1.	<ol style="list-style-type: none"> 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 Exceptions 7. Форматирование строк 8. Объекты и классы 9. Методы, поля и property 	лекции			2		OP-2.2.2 OP-2.3.1
		практическая работа			2		
5.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата	СРС			7		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос			4		
	Раздел 6. Скриптовый язык Javascript		7			1,2,3,4,5	
6.1.	<ol style="list-style-type: none"> 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-2.2.2 OP-2.3.1 OP-1.3.1
		практическая работа			2		
6.2.	<i>Форма СРС:</i> изучение учебного материала и подготовка к практической работе; подготовка реферата	СРС			4		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос			3		
	Раздел 7. Компилируемый язык с сборщиком мусора Go (golang)		7		2	1,2,3,4,5	

7.1.	1. Типы данных 2. Функции 3. Конструкции потока управления 4. Структуры и «объекты» 5. Методы и интерфейсы 6. Прimitives многозадачности: light-weight processes (goroutines), channels, и select statement	лекции					OP-2.2.1
		практическая работа			2		OP-2.2.2 OP-2.3.1
7.3.	Форма СРС: изучение учебного материала и подготовка к практической работе; подготовка реферата	СРС			6		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос			2		
	Раздел 8. Компилируемый язык Rust		7			1,2,3,4,5	
8.1.	1. Объявление переменных, mutable-данные и жизненный цикл 2. Базовые типы и вывод типов 3. Конструкции потока управления 4. Владение переменными 5. Классы, примеси и методы Многозадачность без состояния гонки	лекции			2		OP-2.2.1
		практическая работа			2		OP-2.2.2 OP-2.3.1 OP-1.3.1
8.2.	Форма СРС: изучение учебного материала и подготовка к практической работе; подготовка реферата	СРС			8,6		
	Текущий контроль успеваемости	Отчет по практической работе. Письменный опрос.			2		
	Консультации в период теоретического обучения и промежуточной аттестации	Консультация			4,4		
	Подготовка к промежуточной аттестации в форме экзамена	СРС	7		31,7		

	Прохождение промежуточной аттестации в форме экзамена	экзамен	7		2,3		
--	---	---------	---	--	-----	--	--

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.

Типовые задания или иные материалы, необходимые для оценки результатов обучения, характеризующих этапы формирования компетенций, и методические материалы, определяющие процедуры оценивания результатов обучения, приведены в Приложении 1 к рабочей программе «Фонд оценочных средств».

4.1. Рекомендуемая литература и учебно-методическое обеспечение

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

4.2. Перечень лицензионного и программного обеспечения

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.

4.3. Оборудование и технические средства обучения

При освоении дисциплины используются компьютерные классы ФИнф ТГУ с доступом к ресурсам Научной библиотеки ТГУ, в том числе отечественным и зарубежным периодическим изданиям и Интернету.

5. Методические указания обучающимся по освоению дисциплины

Лекционный курс и практические задания по курсу Языки программирования [Электронный ресурс] / Электронный университет – MOODLE.– ТГУ 2016. – URL: <https://moodle.tsu.ru/course/view.php?id=2162/> (дата обращения: 14.10.2016).

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.