Белорусский государственный университет

УТВЕРЖДАЮ
Проректор по учебной работе

_____ А.Л. Толстик
29.07.2016

Регистрационный № УД- 3006 / уч.

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Учебная программа учреждения высшего образования по учебной дисциплине для специальности второй ступени высшего образования (магистратуры) с углубленной подготовкой специалиста:

1-31 81 09 «Алгоритмы и системы обработки больших объемов информации».

2016 г.

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 81 09-2014 и учебного плана G31-219/уч. от 30.05.2016.

Составители:

А.А. Толстиков – старший преподаватель кафедры вычислительной математики Белорусского государственного университета.

А.М. Удовиченко – инженер-программист ООО СофтекФлешСолюшнс.

Рекомендована к утверждению:

Кафедрой дискретной математики и алгоритмики Белорусского государственного университета (протокол № 14 от 19 мая 2016 г.);

Методической комиссией факультета прикладной математики и информатики Белорусского государственного университета (протокол № 6 от 24 мая 2016 г.).

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа по учебной дисциплине «Языки программирования» разработана в соответствии с учебным планом и образовательным стандартом второй ступени высшего образования (магистратуры) с углубленной подготовкой специалиста по специальности 1-31 81 09 «Алгоритмы и системы обработки больших объемов информации».

Учебная дисциплина «Языки программирования» знакомит магистрантов с теоретическими основами и принципами проектирования и реализации языков программирования общего назначения, а также классификацией и отличительными особенностями языков.

Основой для изучения учебной дисциплины являются следующие учебные дисциплины первой ступени высшего образования: «Методы трансляции», «Программирование», «Технологии программирования» и «Теория алгоритмов».

Цель преподавания учебной дисциплины «Языки программирования»: создание базы для использования современных функциональных и объектно-ориентированных языков и формирование у магистрантов умения анализировать язык программирования как основной инструмент разработки программного обеспечения с целью выбора наиболее оптимального языка для решения конкретной задачи,
При изложении материала учебной дисциплины важно показать многообразие современных языков высокого уровня, классифицируя языки программирования по таким критериям, как парадигма декомпозиции и система типов.

Основные задачи, решаемые при изучении учебной дисциплины «Языки программирования»:

изучение 3 современных языков программирования из различных групп: функциональные статически-типизированные, функциональные динамически-типизированные, объектно-ориентированные динамически-типизированные;

сравнительный анализ изученных языков программирования;

использование изученных языков программирования в качестве метаязыка для разработки различных алгоритмов реализации языка программирования.

В результате изучения дисциплины магистрант должен:

знать:

– классификацию языков программирования;

– разнообразные свойства языка и их влияние на реализацию и использование языка;

– основные парадигмы и идиомы, применяемые при разработке с использованием каждого из изучаемых языков;

уметь:
– разрабатывать различные алгоритмы реализации языков;
– реализовывать эти алгоритмы на одном или нескольких изученных языках;
– оценивать положительные и отрицательные последствия использования конкретного языка программирования в контексте решаемой задачи.
владеть:
– 3 современными языками программирования из различных групп: функциональные статически-типизированные, функциональные динамически-типизированные, объектно-ориентированные динамически-типизированные

Освоение образовательной программы магистратуры должно обеспечить формирование следующих групп компетенций:

академических компетенций – углубленных научно-теоретических, методологических знаний и исследовательских умений, обеспечивающих разработку научно-исследовательских, инновационной деятельности, непрерывного самообразования (АК-1. Способность к самостоятельной профессиональной деятельности (анализ, сопоставление, систематизация, абстрагирование, моделирование, проверка достоверности данных, принятие решений и др.), готовность генерировать и использовать новые идеи. АК-2. Методологические знания и исследовательские умения, обеспечивающие решение прикладных задач и инновационной деятельности. АК-3. Способность к постоянному самообразованию);

социально-личностных компетенций – личностных качеств и умений следовать социально-культурным и нравственным ценностям; способностей к социальному, межкультурному взаимодействию, критическому мышлению; социальной ответственности, позволяющих решать социально-профессиональные, организационно-управленческие, воспитательные задачи (Магистр должен: СЛК-1. Учитывать социальные и нравственно-этические нормы в социально-профессиональной деятельности. СЛК-2. Быть способным к сотрудничеству и работе в команде. СЛК-3. Владеть коммуникативными способностями для работы в междисциплинарной и международной среде. СЛК-4. Совершенствовать и развивать свой интеллектуальный и общекультурный уровень, добиваться нравственного и физического совершенствования своей личности. СЛК-5. Формировать и аргументировать собственные суждения и профессиональную позицию. СЛК-6. Логично, аргументированно и ясно строить устную и письменную речь, использовать навыки публичной речи, ведения дискуссии и полемики. СЛК-7. Проявлять инициативу и креативность, в том числе в нестандартных ситуациях);

профессиональных компетенций – углубленных знаний по специальным

дисциплинам и способностей решать сложные профессиональные задачи, задачи научно-исследовательской и научно-педагогической деятельности, разрабатывать и внедрять инновационные проекты, осуществлять непрерывное профессиональное самосовершенствование (Магистр должен быть способен: ПК-1. Квалифицированно использовать современные достижения по разработке и анализу алгоритмов и современные информационные технологии. ПК-2. Строить математические модели для прикладных задач и проводить теоретическое и экспериментальное их исследование. ПК-3. Разрабатывать эффективные численные алгоритмы и интегрировать их в компьютерные системы. ПК-4. Обосновывать выбор методов и инструментов для решения прикладных задач. ПК-5. Обосновывать достоверность полученных результатов. ПК-6. Осваивать и реализовывать управленческие инновации в профессиональной деятельности. ПК-7. Формулировать выводы и рекомендации по применению современных достижений науки в инновационной деятельности).

Учебная программа рассчитана на 112 часов, из них 54 аудиторных часа, в том числе 18 лекционных часов и 36 часов лабораторных занятий.

Рекомендуемая форма текущей аттестации – зачет.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Введение

История языков программирования, классификация языков программирования, краткий обзор курса.

Раздел I. Статическая типизация в функциональных языках

Тема 1.1.Введение в язык F#. Система типов языка F#

Переменные и выражения. Синтаксис и проверка типов. Вычисление выражений. Статическое и динамическое окружение. Функции, let-выражения, условные выражения. Типы. Пары, кортежи, списки, option. Базовые и составные типы, основные способы сконструировать сложный тип данных. Записи. Типы-объединения.

Тема 1.2.Сопоставление по шаблонам и полиморфизм

Выражения match, шаблоны (patterns). Преимущества сопоставления по шаблонам. Примеры использования шаблонов. Рекурсивные типы. Семантика выражения match. Сопоставление кортежей и записей. Шаблоны в аргументах функций и связывании. Полиморфные шаблоны. Объявление и использование полиморфных типов. Вложенные шаблоны. Исключения.

Тема 1.3. Функции высшего порядка

Понятие функции первого класса и высшего порядка. Безымянные функции. Примеры функций высшего порядка: map, filter и fold. Применение функций первого класса. Понятие замыкания. Лексический и динамический контекст вызова. Понятие связанных и свободных переменных. Время вычисления аргументов. Композиция функций, конвейеризация (pipelining), каррирование. Изменяемое состояние в F#.

Тема 1.4. Автоматический вывод типов и модульность.

Проверка типов и неявная типизация. Автоматический вывод типов: обобщённый алгоритм и примеры. Вывод типов и полиморфизм. Особенности языка, усложняющие вывод типов.

Важность обеспечения модульности. Модули и пространства имён. Сигнатуры модулей. Сокрытие деталей реализации и абстракция. Сигнатуры и проверка типов. Эквивалентные реализации и понятие эквивалентности. Разные точки зрения на эквивалентность.

Раздел II. Динамическая типизация в функциональных языках.

Тема 2.1. Введение в язык Racket. Отложенные вычисления.

Краткая характеристика языка Racket. Определения, функции, ветвление. Функции с несколькими аргументами и каррирование. Работа со списками. Особенности синтаксиса языков семейства Lisp. Динамическая типизация. Выражение cond. Локальные связывания - let,

let*, letrec. Изменение переменных - set!. Пары cons и mcons. Отложенные вычисления и ленивые вычисления, понятие thunk, механизм delay и force. Стримы. Мемоизация. Макросы.

Тема 2.2. Пользовательские типы данных. Реализация языка программирования на Racket.

Способы организации пользовательских данных: гетерогенные списки и struct. Преимущества использования struct.

Способы реализации языка. Компиляция, интерпретация и гибридные реализации. Реализация языка без синтаксического разбора. Понятие метаязыка. Реализация вычисления арифметических выражений и работы с переменными. Передача окружения и замыкания. Вызов функций. Эффективность реализации замыканий. Вычисление свободных переменных.

Тема 2.3. Сравнение F# и Racket. Преимущества и недостатки статической и динамической типизации.

Ключевые отличия Racket и F#. Понятие статических проверок. Время обнаружения ошибки. Корректность и полнота системы типов. Неполнота системы типов F#. Система типов и проблема неразрешимости. Некорректность системы типов. Слабая типизация.

Преимущества и недостатки статической и динамической типизации в контексте удобства разработки, производительности, повторного использования кода, времени выявления ошибок, удобства поддержки и прототипирования.

Раздел III. Динамическая типизация в объектно-ориентированных языках

Тема 3.1. Основы языка Ruby. Особенности ООП в Ruby.

Краткая характеристика языка Ruby. Определение классов и методов. Создание и использование объектов. Переменные. Состояние объекта. Инициализация объектов. Переменные, константы и методы класса. Доступ, видимость и приватное состояние. Рефлексия. Утиная типизация. Блоки и замыкания. Коллекции (массивы, хэши, диапазоны). Наследование классов и переопределение методов. Альтернативы наследования: расширение класса и композиция. Динамическая диспетчеризация. Сравнение замыканий и виртуальных методов.

Тема 3.2. Сравнение функциональной и объектно-ориентированной декомпозиции.

Ортогональность функционального и объектно-ориентированного подхода. Влияние подхода на дальнейшую разработку и сопровождение. Бинарные операции. Двойная диспетчеризация. Мультиметоды. Проблемы множественного наследования. Миксины как альтернатива множественному наследованию. Сравнение миксинов и интерфейсов.

Тема 3.3 Системы типов в объектно-ориентированных языках. Мини-язык для изучения особенностей систем типов. Базовая система типов и расширение типов. Безопасное порождение новых типов. Допущения, нарушающие корректность системы типов. Ковариантность и контравариантность. Классы и типы. Дженерики и полиморфизм.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

№ п/п	Название раздела, темы	Количество часов				Коли-чество часов УСР	Форма контроля знаний
		Аудиторные					
		Лекции	Пр.и сем.зан.	Лаб. зан.	Иное		
1	**Введение. Статическая типизация в функциональных языках**	**6**		**18**			
1.1	Введение в язык F#. Система типов языка F#.	3					Устный опрос
	Лабораторная работа 1 Основы языка программирования F#			6			Защита лабораторной работы 1
1.2	Сопоставление по шаблонам и полиморфизм	1					Устный опрос
	Лабораторная работа 2 Сложные типы данных и алгоритм автомати-ческого вывода типов в языке F#			6			Защита лабораторной работы 2
1.3	Функции высшего порядка	1					Устный опрос
	Лабораторная работа 3 Функции высшего порядка в языке F#			6			Защита лабораторной работы 3
1.4	Автоматический вывод типов и модульность	1					Устный опрос
2	**Динамическая типизация в функциональных языках**	**6**		**12**			
2.1	Введение в язык Racket. Отложенные вычисления.	2					Устный опрос
2.2	Пользовательские типы данных. Реализация языка программирования на Racket	2					Устный опрос
	Лабораторная работа 4 Основы языка программирования Racket. Функции высшего порядка, стримы и отложенные вычисления на языке Racket.			8			Защита лабораторной работы 4
2.3	Сравнение F# и Racket. Преимущества и недостатки статической и динамической типизации	2					Устный опрос

	Лабораторная работа 5 Сложные типы данных в языке Racket. Реализация языка программирования на языке Racket.			4			Защита лабораторной работы 5
3.	**Динамическая типизация в объектно-ориентированных языках**	**6**		**6**			
3.1	Основы языка Ruby. Особенности ООП в Ruby.	2					Устный опрос
3.2	Сравнение функциональной и объектно-ориентированной декомпозиции	2					Устный опрос
3.3	Системы типов в объектно-ориентированных языках	2					Устный опрос
	Лабораторная работа 6 Двойная диспетчеризация на языке Ruby.			6			Защита лабораторной работы 6
ИТОГО		**18**		**36**			

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Рекомендуемаялитература

Основная

1. Thomas D., Fowler C., Hunt A. Programming Ruby 1.9: The Pragmatic Programmers' Guide– Pragmatic, 2010. – 916 p.
2. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, and ShriramKrishnamurthi, How to Design Programs: An Introduction to Programming and Computing, MIT Press, Cambridge, MA, 2002
3. Chris Smith, Programming F# 3.0, 2nd Edition: A comprehensive guide for writing simple code to solve complex problems - O'Reilly Media, 2012. - 476 p.
4. Peter Sestoft, Programming Language Concepts - Springer-Verlag London 2012. –278 p.
5. Michael L. Scott, Programming Language Pragmatics - Morgan Kaufmann 2015 – 992 p.

Дополнительная

1. Bruce Tate, Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages - Pragmatic Bookshelf 2010 – 328 p.
2. John C. Mitchell, Concepts in Programming Languages - Cambridge University Press 2004 – 529p.
3. Benjamin C. Pierce, Types and Programming Languages – MIT press 2012 – 645 p.

Рекомендации по контролю качества усвоения знаний и проведению аттестации

На лекционных занятиях по учебной дисциплине «Языки программирования» рекомендуется использование элементов проблемного обучения: проблемное изложение некоторых аспектов, использование частично-поискового метода.

Перечни используемых средств диагностики результатов учебной деятельности

Для аттестации обучающихся на соответствие их персональных достижений поэтапным и конечным требованиям образовательной программы создаются фонды оценочных средств, включающие типовые задания, контрольные работы и тесты. Оценочными средствами предусматривается оценка способности обучающихся к творческой деятельности, их готовность вести поиск решения новых задач, связанных с

недостаточностью конкретных специальных знаний и отсутствием общепринятых алгоритмов.

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: опросы, устная защита лабораторных работ.
2. Письменная форма: отчеты по лабораторным работам, оценивание на основе модульно-рейтинговой системы.

Контрольные мероприятия проводятся в соответствии с учебно-методической картой дисциплины. В случае неявки на контрольное мероприятие по уважительной причине студент вправе по согласованию с преподавателем выполнить его в дополнительное время. Для студентов, получивших неудовлетворительные оценки за контрольные мероприятия, либо не явившихся по неуважительной причине, по согласованию с преподавателем и с разрешения заведующего кафедрой мероприятие может быть проведено повторно.

Оценка текущей успеваемости рассчитывается как среднее за отчеты по домашним практическим упражнениям и оценки за итоговый тест.

Итоговая аттестация предусматривает проведение зачета. При этом рекомендуется использовать оценивание успеваемости на основе модульно-рейтинговой системы.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Методы трансляции	Многопроцессорных систем и сетей	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 14 от 19.05.2016 г
Программирование	Технологий программирования	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 14 от 19.05.2016 г
Технологии программирования	Информационных систем управления	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 14 от 19.05.2016 г
Теория алгоритмов	Дискретной математики и алгоритмики	Нет	Оставить содержание учебной дисциплины без изменения, протокол № 14 от 19.05.2016 г

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
на _____/_____ учебный год

№№ Пп	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры дискретной математики и алгоритмики (протокол № ____ от _____ 201_ г.)
Заведующий кафедрой

_____ _____ _____
(ученая степень, звание) (подпись) (И.О. Фамилия)

УТВЕРЖДАЮ
Декан факультета

_____ _____ _____
(ученая степень, звание) (подпись) (И.О.Фамилия)