

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

УТВЕРЖДАЮ:

Директор ИУЦТ



С.П. Вакуленко

10 июля 2020 г.

Кафедра «Цифровые технологии управления транспортными процессами»

Автор Разживайкин Игорь Станиславович

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Функциональное программирование

Направление подготовки:	09.03.01 – Информатика и вычислительная техника
Профиль:	Программное обеспечение средств вычислительной техники и автоматизированных систем
Квалификация выпускника:	Бакалавр
Форма обучения:	очная
Год начала подготовки	2020

Одобрено на заседании Учебно-методической комиссии института Протокол № 4 30 апреля 2020 г. Председатель учебно-методической комиссии  Н.А. Клычева	Одобрено на заседании кафедры Протокол № 1 27 апреля 2020 г. Доцент  В.Е. Нутович
---	--

Москва 2020 г.

1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Знакомство с основами функционального программирования. Знания и навыки, полученные в рамках данного курса необходимы для:

- понимания функциональных конструкций в других языках программирования;
- применения лямбда-выражений при разработке и написании алгоритмов;
- понимание принципов работ с большими данными;
- совершенствование навыков работы со сложными структурами данных.

Функциональное программирование часто используется не только в прикладном программировании, но и для проведения аналитических, научно-исследовательских и проектных работ в различных областях науки.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Учебная дисциплина "Функциональное программирование" относится к блоку 1 "Дисциплины (модули)" и входит в его вариативную часть.

2.1. Наименования предшествующих дисциплин

Для изучения данной дисциплины необходимы следующие знания, умения и навыки, формируемые предшествующими дисциплинами:

2.1.1. Программирование :

Знания: Знать и понимать: существующие языки программирования и их методологии, принципы разработки процедурных программ и подпрограмм.

Умения: Уметь: формализовывать алгоритмическую и функциональную структуру программного обеспечения; разрабатывать блок схемы алгоритмов, подпрограмм.

Навыки: Владеть: основными инструментами разработки, отладки, тестирования и распространения программного обеспечения.

2.1.2. Технологии программирования:

Знания: принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения, типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения, методы и средства проектирования

Умения: разрабатывать метрики (количественные показатели) работы ИС, анализировать исходные данные

Навыки: навыками разработки и согласования технических спецификаций на программные компоненты и их взаимодействие с архитектором программного обеспечения

2.1.3. Языки программирования высокого уровня:

Знания:

Умения:

Навыки:

2.2. Наименование последующих дисциплин

Результаты освоения дисциплины используются при изучении последующих учебных дисциплин:

2.2.1. Разработка мобильных приложений

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В результате освоения дисциплины студент должен:

№ п/п	Код и название компетенции	Ожидаемые результаты
1	ПКР-1 Способность выполнять работы и управлять работами по разработке архитектур и прототипов информационных систем (ИС)	ПКР-1.1 Знать инструменты и методы проектирования архитектуры ИС; инструменты и методы верификации архитектуры ИС; возможности ИС; предметную область автоматизации; архитектуру, устройство и функционирование вычислительных систем; коммуникационное оборудование; сетевые протоколы; основы современных операционных систем; основы современных систем управления базами данных; устройство и функционирование современных ИС; современные стандарты информационного взаимодействия систем; программные средства и платформы инфраструктуры информационных технологий организаций; современные подходы и стандарты автоматизации организации (например, CRM, MRP, ERP..., ITIL, ITSM); системы классификации и кодирования информации, в том числе присвоение кодов документам и элементам справочников; отраслевую нормативную техническую документацию; источники информации, необходимой для профессиональной деятельности; современный отечественный и зарубежный опыт в профессиональной деятельности; основы бухгалтерского учета и отчетности организаций; основы налогового законодательства Российской Федерации; основы управленческого учета; основы международных стандартов финансовой отчетности (МСФО); основы управления торговлей, поставками и запасами; основы организации производства; основы управления персоналом, включая вопросы оплаты труда; основы финансового учета и бюджетирования; основы управления взаимоотношениями с клиентами и заказчиками (CRM); современные инструменты и методы управления организацией, в том числе методы планирования деятельности, распределения поручений, контроля исполнения, принятия решений; методологию ведения документооборота в организациях; инструменты и методы определения финансовых и производственных показателей деятельности организаций; культуру речи; правила деловой переписки. ПКР-1.2 Уметь проектировать архитектуру ИС; проверять (верифицировать) архитектуру ИС; кодировать на языках программирования; тестировать результаты прототипирования; проводить презентации; проводить переговоры. ПКР-1.3 Владеть навыками разработки архитектурной спецификации ИС; согласования архитектурной спецификации ИС с заинтересованными сторонами; разработки прототипа ИС в соответствии с требованиями; тестирования прототипа ИС на проверку корректности архитектурных решений; анализа

№ п/п	Код и название компетенции	Ожидаемые результаты
		результатов тестов; принятие решения о пригодности архитектуры; согласования пользовательского интерфейса с заказчиком.
2	УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений	УК-2.1 Знать действующие правовые нормы. УК-2.2 Уметь выбирать оптимальные способы решения поставленных задач. УК-2.3 Владеть приемами решения поставленных задач исходя из действующих правовых норм, имеющихся ресурсов и ограничений.

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ И АКАДЕМИЧЕСКИХ ЧАСАХ

4.1. Общая трудоемкость дисциплины составляет:

4 зачетные единицы (144 ак. ч.).

4.2. Распределение объема учебной дисциплины на контактную работу с преподавателем и самостоятельную работу обучающихся

Вид учебной работы	Количество часов	
	Всего по учебному плану	Семестр 6
Контактная работа	42	42,15
Аудиторные занятия (всего):	42	42
В том числе:		
лекции (Л)	14	14
практические (ПЗ) и семинарские (С)	14	14
лабораторные работы (ЛР)(лабораторный практикум) (ЛП)	14	14
Самостоятельная работа (всего)	66	66
Экзамен (при наличии)	36	36
ОБЩАЯ трудоемкость дисциплины, часы:	144	144
ОБЩАЯ трудоемкость дисциплины, зач.ед.:	4.0	4.0
Текущий контроль успеваемости (количество и вид текущего контроля)	КП (1), ПК1, ПК2	КП (1), ПК1, ПК2
Виды промежуточной аттестации (экзамен, зачет)	ЭК	ЭК

4.3. Содержание дисциплины (модуля), структурированное по темам (разделам)

№ п/п	Семестр	Тема (раздел) учебной дисциплины	Виды учебной деятельности в часах/ в том числе интерактивной форме						Формы текущего контроля успеваемости и промежуточной аттестации
			Л	ЛР	ПЗ	КСР	СР	Всего	
1	2	3	4	5	6	7	8	9	10
1	6	Раздел 1 Раздел 1 Введение в функциональное программирование	2	2	2		6	12	
2	6	Тема 1.1 Тема 1.1 Основы функционального программирования	1	1	1		3	6	
3	6	Тема 1.2 Тема 1.2 Рекурсивные структуры данных	1	1	1		3	6	
4	6	Раздел 2 Раздел 2 Типовые приемы функционального программирования	6		12		14	32	ПК1, ПК2
5	6	Тема 2.1 Тема 2.1 Замыкания	1		2		2	5	
6	6	Тема 2.2 Тема 2.2 Динамическое связывание	1		2		2	5	
7	6	Тема 2.3 Тема 2.3 Mutable-переменные	1		2		2	5	
8	6	Тема 2.4 Тема 2.4 Генераторы и ссылочные переменные	1		2		2	5	
9	6	Тема 2.5 Тема 2.5 Ленивые последовательности	1		2		3	6	
10	6	Тема 2.6 Тема 2.6 Ленивые и энергичные вычисления	1		2		3	6	
11	6	Раздел 3 Раздел 3 Императивное и объектно-ориентированное функциональное программирование	3	4			22	29	КП
12	6	Тема 3.1 Тема 3.1 Мультипарадигмальность F#	1				8	9	
13	6	Тема 3.2 Тема 3.2 Элементы императивного программирования	1	2			7	10	
14	6	Тема 3.3 Тема 3.3 Объектно-ориентированное программирование	1	2			7	10	

№ п/п	Семестр	Тема (раздел) учебной дисциплины	Виды учебной деятельности в часах/ в том числе интерактивной форме						Формы текущего контроля успеваемости и промежу- точной аттестации
			Л	ЛР	ПЗ	КСР	СР	Всего	
1	2	3	4	5	6	7	8	9	10
15	6	Раздел 4 Раздел 4 Метапрограммирование	3	8			24	35	ПК2
16	6	Тема 4.1 Тема 4.1 Основы метапрограммирования	1	2			8	11	
17	6	Тема 4.2 Тема 4.2 Монады	1	3			8	12	
18	6	Тема 4.3 Тема 4.3 Монадические выражения	1	3			8	12	
19	6	Раздел 5 Раздел 5 Экзамен						36	ЭК
20		Всего:	14	14	14		66	144	

4.4. Лабораторные работы / практические занятия

Лабораторные работы предусмотрены в объеме 14 ак. ч.

№ п/п	№ семестра	Тема (раздел) учебной дисциплины	Наименование занятий	Всего ча- сов/ из них часов в интерак- тивной форме
1	2	3	4	5
1	6	Раздел 1 Введение в функциональное программирование	Тема 1.1 Основы функционального программирования	1
2	6	Раздел 1 Введение в функциональное программирование	Тема 1.2 Рекурсивные структуры данных	1
3	6	Раздел 3 Императивное и объектно-ориентированное функциональное программирование	Тема 3.2 Элементы императивного программирования	2
4	6	Раздел 3 Императивное и объектно-ориентированное функциональное программирование	Тема 3.3 Объектно-ориентированное программирование	2
5	6	Раздел 4 Метaprogramмирование	Тема 4.1 Основы метaprogramмирования	2
6	6	Раздел 4 Метaprogramмирование	Тема 4.2 Монады	3
7	6	Раздел 4 Метaprogramмирование	Тема 4.3 Монадические выражения	3
ВСЕГО:				14 / 0

Практические занятия предусмотрены в объеме 14 ак. ч.

№ п/п	№ семестра	Тема (раздел) учебной дисциплины	Наименование занятий	Всего ча- сов/ из них часов в интерак- тивной форме
1	2	3	4	5
1	6	Раздел 1 Введение в функциональное программирование	Тема 1.1 Основы функционального программирования	1
2	6	Раздел 1 Введение в функциональное программирование	Тема 1.2 Рекурсивные структуры данных	1
3	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.1 Замыкания	2
4	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.2 Динамическое связывание	2

№ п/п	№ семестра	Тема (раздел) учебной дисциплины	Наименование занятий	Всего ча- сов/ из них часов в интерак- тивной форме
1	2	3	4	5
5	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.3 Mutable-переменные	2
6	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.4 Генераторы и ссылочные переменные	2
7	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.5 Ленивые последовательности	2
8	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.6 Ленивые и энергичные вычисления	2
ВСЕГО:				14 / 0

4.5. Примерная тематика курсовых проектов (работ)

Разработка приложения на функциональном языке F#

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Преподавание дисциплины «СФункциональное и логическое программирование» осуществляется в форме лекций, лабораторных работ, самостоятельной работы студентов. Лекции проводятся в форме мультимедиа-лекций, на которых демонстрируются презентации. Студенты имеют возможность ознакомиться с материалами презентации до начала лекции.

Лабораторные занятия проводятся в компьютерном классе с установленным программным обеспечением, необходимым для разработки индивидуальных проектов. На лабораторных работах выполняются индивидуальные задания, демонстрируются готовые части выполненных заданий и отчета по заданию. Разработка проектов по индивидуальным заданиям ведется на языке F#.

Самостоятельная работа студента организована с использованием традиционных видов работы.

Оценка полученных знаний, умений и навыков основана на модульно-рейтинговой технологии. Весь курс разбит на 6 разделов, представляющих собой логически заверченный объем учебной информации. Фонды оценочных средств освоенных компетенций включают как вопросы теоретического характера для оценки знаний, так и задания практического содержания (индивидуальные задания на разработку лабораторной работы) для оценки умений и навыков. Теоретические знания проверяются путём решения тестов с использованием компьютеров и в ходе проверки отчетов по выполненным индивидуальным работам.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

№ п/п	№ семестра	Тема (раздел) учебной дисциплины	Вид самостоятельной работы студента. Перечень учебно-методического обеспечения для самостоятельной работы	Всего часов
1	2	3	4	5
1	6	Раздел 1 Введение в функциональное программирование	Тема 1.1 Основы функционального программирования	3
2	6	Раздел 1 Введение в функциональное программирование	Тема 1.2 Рекурсивные структуры данных	3
3	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.1 Замыкания	2
4	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.2 Динамическое связывание	2
5	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.3 Mutable-переменные	2
6	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.4 Генераторы и ссылочные переменные	2
7	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.5 Ленивые последовательности	3
8	6	Раздел 2 Типовые приемы функционального программирования	Тема 2.6 Ленивые и энергичные вычисления	3
9	6	Раздел 3 Императивное и объектно-ориентированное функциональное программирование	Тема 3.1 Мультипарадигмальность F#	8
10	6	Раздел 3 Императивное и объектно-ориентированное функциональное программирование	Тема 3.2 Элементы императивного программирования	7
11	6	Раздел 3 Императивное и объектно-ориентированное функциональное программирование	Тема 3.3 Объектно-ориентированное программирование	7
12	6	Раздел 4 Метапрограммирование	Тема 4.1 Основы метапрограммирования	8
13	6	Раздел 4 Метапрограммирование	Тема 4.2 Монады	8
14	6	Раздел 4 Метапрограммирование	Тема 4.3 Монадические выражения	8
ВСЕГО:				66

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Основная литература

№ п/п	Наименование	Автор (ы)	Год и место издания Место доступа	Используется при изучении разделов, номера страниц
1	Функциональное программирование	А. Филд, П. Харрисон	Мир, 1993 НТБ (фб.); НТБ (чз.2)	Все разделы

7.2. Дополнительная литература

№ п/п	Наименование	Автор (ы)	Год и место издания Место доступа	Используется при изучении разделов, номера страниц
2	Функциональное программирование на F#	Сошников Д.В.	Издательство "ДМК Пресс", 2011 https://e.lanbook.com/book/1274	Все разделы

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ "ИНТЕРНЕТ", НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1. <https://ru.stackoverflow.com/> - Q&A-портал программистов
2. <https://ru.wikipedia.org> – online энциклопедия
3. <https://fsharpforfunandprofit.com/site-contents/> - online-учебник по F# на английском языке

9. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ИСПОЛЪЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Используемые информационные технологии:

- Объектно-ориентированное программирование
- Модульный подход к проектированию программного обеспечения

Для выполнения лабораторных требуется следующее программное обеспечение:

- Microsoft Visual Studio (не позднее 2010)

10. ОПИСАНИЕ МАТЕРИАЛЬНО ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Для лекционных занятий – наличие проектора и экрана. Для лабораторных занятий – наличие персональных компьютеров вычислительного класса.

11. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

1. В лекционном курсе рассматриваются основные вопросы по данной дисциплине. Дополнительные вопросы, необходимые студентам при выполнении своих индивидуальных заданий, изучаются студентами самостоятельно и контролируются преподавателем.
2. Задания по всем лабораторным работам выдаются студентам в начале семестра, чтобы студенты имели возможность самостоятельно изучить дополнительные теоретические

сведения, необходимые им при выполнении индивидуальных заданий, и спланировать график выполнения заданий с учетом их специфики.

3. Прежде чем приступить к выполнению конкретного задания студент должен изучить: материалы лекций по теме задания;

дополнительные материалы, относящиеся к специфике индивидуального задания; программные средства, используемые при выполнении задания.

4. Выполнение индивидуальных заданий и их сдача осуществляется по определенному графику и учитывается при периодической аттестации студентов.

5. Лекции по дисциплине, подготовленные в электронном виде, рекомендуется выдавать студентам в начале семестра с целью лучшего освоения материала и возможности досрочного изучения вопросов, необходимых для выполнения индивидуальных заданий.

6. Индивидуальные задания, требующие разработки сложных программных систем, могут выдаваться на группу студентов, но при этом необходимо контролировать знание каждым студентом всего задания в целом.

7. Для полноценного освоения дисциплины необходимо:

- Посещение лекций и практических занятий;
- Изучение лекционного материала;
- Освоение теоретического материала, вынесенного на самостоятельное изучение, по предложенным источникам (литература, интернет-ресурсы);
- Изучение программного обеспечения, необходимого, для выполнения индивидуальных заданий;
- Консультации с преподавателем в ходе выполнения индивидуальных заданий и обсуждение промежуточных результатов выполнения индивидуальных заданий;
- Своевременное выполнение индивидуальных заданий;
- Своевременное предоставление отчетов по индивидуальным заданиям и защита выполненных работ.