

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тверской государственный технический университет»
(ТвГТУ)

УТВЕРЖДАЮ

Проректор
по учебной работе

_____ Э.Ю. Майкова
« _____ » _____ 2019 г.

РАБОЧАЯ ПРОГРАММА
дисциплины обязательной части Блока 1

«Теоретические основы программирования»

Направление подготовки магистров 09.04.02 Информационные системы и технологии

Направленность (профиль) – Разработка, внедрение и сопровождение информационных систем

Тип задач профессиональной деятельности – организационно-управленческий

Форма обучения - очная

Факультет информационных технологий
Кафедра «Информационных систем»

Тверь 2019

Рабочая программа дисциплины соответствует ОХОП подготовки магистров в части требований к результатам обучения по дисциплине и учебному плану.

Разработчик программы: к.т.н., доцент каф. ИС

И.А. Егерова

Программа рассмотрена и одобрена на заседании кафедры ИС
«13» мая 2019 г., протокол № 5.

Заведующий кафедрой ИС, д.т.н., профессор

Б.В. Палюх

Согласовано
Начальник учебно-методического
отдела УМУ

Д.А. Барчуков

Начальник отдела
комплектования
зональной научной библиотеки

О.Ф. Жмыхова

1. Цели и задачи дисциплины

Целью изучения дисциплины «Теоретические основы программирования» является формирование у обучающегося компетенции разрабатывать алгоритмы и программные средства, модернизировать программное обеспечение информационных и автоматизированных систем.

Задачами дисциплины являются:

- формирование знаний о различных подходах и методах в области программирования;
- формирование знаний методов трансляции, формальных моделей, методов и алгоритмов синтаксически управляемого разбора и перевода;
- формирование умений разрабатывать и реализовывать на языках программирования алгоритмы программ в соответствии с поставленной задачей.

2. Место дисциплины в структуре ООП

Дисциплина относится к обязательной части Блока 1 «Дисциплины (модули)». Для изучения курса требуются знания, умения и навыки, полученные в процессе изучения дисциплин «Математика», «Информатика», «Алгоритмизация», «Технология программирования».

Приобретенные знания в рамках данной дисциплины помимо их самостоятельного значения являются основой для изучения курсов «Системная инженерия», «Методы и средства разработки архитектуры информационных систем», «Прикладные аспекты управления ИТ-проектами» и других дисциплин, профессиональная подготовка по которым предполагает применение теоретических знаний в области программирования, применение навыков разрабатывать и реализовывать алгоритмы программ, использование программных средств для решения прикладных задач, а также при выполнении выпускной квалификационной работы.

3. Планируемые результаты обучения по дисциплине

3.1. Планируемые результаты обучения по дисциплине

Компетенция, закрепленная за дисциплиной в ОХОП:

ОПК-2. *Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач.*

Индикаторы компетенции, закреплённые за дисциплиной в ОХОП:

ИОПК-2.1. Демонстрирует знания современных информационно-коммуникационных и интеллектуальных технологий, инструментальных сред, программно-технических платформ для решения профессиональных задач.

Показатели оценивания индикаторов достижения компетенций:

Знать:

31.1. Этапы развития теории программирования, базовые основополагающие понятия теории программирования.

31.2. Принципы надежного программирования.

31.3. Современные и перспективные языки программирования.

Уметь:

У1.1. Анализировать и систематизировать знания в области информационных технологий.

У1.2. Пользоваться литературой и информационными ресурсами при поиске решений прикладных задач.

ИОПК-2.2. Обосновывает выбор современных информационно-коммуникационных и интеллектуальных технологий, разрабатывает оригинальные программные средства для решения профессиональных задач.

Показатели оценивания индикаторов достижения компетенций:

Знать:

32.1. Современные направления информационных технологий.

32.2. Особенности объектно-ориентированного, функционального, логического программирования.

Уметь:

У2.1. Анализировать и систематизировать требования основных направлений различных отраслей информационных технологий.

У2.2. Формализовать требования к языку программирования в соответствии с поставленной задачей.

У2.3. Аргументировать применение инструментальных средств при решении прикладных задач.

ИОПК-2.3. Применяет навыки разработки оригинальных программных средств, в том числе с использованием современных информационно-коммуникационных и интеллектуальных технологий для решения профессиональных задач.

Показатели оценивания индикаторов достижения компетенций:

Знать:

33.1. Современные информационные технологии и сферы их применения.

33.2. Основы модульного и структурного программирования.

33.3. Разновидности архитектуры программного средства.

33.4. Конструкции используемых языков программирования.

33.5. Синтаксис языков программирования.

Уметь:

У3.1. Реализовывать алгоритмы программ на конкретном языке программирования.

У3.2. Осуществлять оптимизацию алгоритмов и реализовывать в виде программ на конкретном языке программирования.

У3.3. Определять типы связности и виды сцепления модулей в ПС.

У3.4. Использовать различные приемы, методы, подходы при создании программного обеспечения.

Компетенция, закреплённая за дисциплиной в ОХОП:

ОПК-5. *Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем.*

Индикаторы компетенции, закреплённые за дисциплиной в ОХОП:

ИОПК-5.3. Разрабатывает программное и аппаратное обеспечение

информационных и автоматизированных систем для решения профессиональных задач.

Показатели оценивания индикаторов достижения компетенций:

Знать:

34.1. Требования к оформлению программной документации.

34.2. Теорию схем программ.

34.3. Избранные формальные теории.

34.4. Языки программирования.

34.5. Синтаксис и семантику языков программирования. Функции для синтаксического разбора простых языковых конструкций.

Уметь:

У4.1. Грамотно формулировать постановку задачи с учетом внешних и внутренних спецификаций.

У4.2. Составлять необходимую документацию, отражающую логические процессы в информационных и автоматизированных системах.

У4.3. Разрабатывать схемы алгоритмов программ.

У4.4. Разрабатывать программные модули.

У4.5. Формировать с применением различных методов тестовые наборы данных.

3.2. Технологии, обеспечивающие формирование компетенций

Проведение лекционных и практических занятий, проведение лабораторных работ, выполнение курсовой работы, самостоятельная работа под руководством преподавателя.

4. Трудоемкость дисциплины и виды учебной работы

Таблица 1а. Распределение трудоемкости дисциплины по видам учебной работы

Вид учебной работы	Зачетные единицы	Академические часы
Общая трудоемкость дисциплины	4	144
Аудиторные занятия (всего)		36
В том числе:		
Лекции		12
Практические занятия (ПЗ)		12
Лабораторные работы (ЛР)		12
Самостоятельная работа обучающихся (всего)		72+36 (экз)
В том числе:		
Курсовая работа (КР)		52
Курсовой проект (КП)		не предусмотрен
Расчетно-графические работы		не предусмотрены
Другие виды самостоятельной работы: - подготовка к защите лабораторных работ		20
Текущий контроль успеваемости и промежуточная аттестация (зачет)		не предусмотрен
Текущий контроль успеваемости и промежуточная аттестация (экзамен)		36 (экз)
Практическая подготовка при реализации дисциплины (всего)		0

5. Структура и содержание дисциплины

5.1. Структура дисциплины

Таблица 2. Модули дисциплины, трудоемкость в часах и видах учебной работы

№	Наименование модуля	Труд-ть часы	Лекции	Практич. занятия	Лаб. практикум	Сам. работа
1	Классификация знаний в области программирования.	7	1		2	2+2(экз)
2	Разработка языков программирования и компиляторов.	13	2			4+8(экз)
3	Специализированные вычислительные машины. Языки программирования для специализированных вычислительных машин.	9	1		2	2+4(экз)
4	Теория программирования.	76	2	10	4	52+6(экз)
5	Надежное программирование. Тьюринг-полные и эзотерические языки программирования. Перспективные языки программирования. Каркас нового языка программирования.	7	1			2+4(экз)
6	Модульное программирование	12	1	2	4	2+4(экз)
7	Функциональное программирование	10	2			4+4(экз)
8	Логическое программирование	10	2			4+4(экз)
<i>Всего часов за 1 семестр</i>		<i>144</i>	<i>12</i>	<i>12</i>	<i>12</i>	<i>72+36(экз)</i>

5.2. Содержание дисциплины

МОДУЛЬ 1 «Классификация знаний в области программирования»:

Системность в области программировании. Граф дисциплин, относящихся к программированию. Теоретический минимум для программиста. Матрица компетентности программиста.

МОДУЛЬ 2 «Разработка языков программирования и компиляторов»:

Понятие программирующей программы. Трансляция. Системное и теоретическое программирование. Схемы Янова. Библиотеки стандартных программ. Программирование на машинном языке. Примеры некоторых языков программирования, их особенности (Ada, Algol-x, C, CLU, Forth, Fortran, Lisp, Modula-2, Pascal, PL/I, SETL, Simula, Snobol и другие).

МОДУЛЬ 3 «Специализированные вычислительные машины. Языки программирования для специализированных вычислительных машин»:

История создания и развития языков программирования для специализированных вычислительных машин. Специализированные бортовые вычислительные машины, особенности программных разработок для данного класса вычислительных машин. Специализированные цифровые вычислительные машины.

МОДУЛЬ 4 «Теория программирования»:

Языки программирования, синтаксис, семантика, прагматика. Когнитивные особенности человеческого мышления и их влияние на развитие языков программирования.

Языки программирования в ретроспективе. Процедурное, объектно-ориентированное, логическое и функциональное программирование. Предметно-ориентированные языки. Языки вне классификации.

Абстрактный и конкретный синтаксис. Статическая и динамическая семантика. Компиляция и интерпретация. Проекция Футамуры-Ершова.

Генеративный и аналитический подходы к описанию синтаксиса. Формальные грамматики, иерархия Хомского.

Регулярные языки и конечные автоматы. Применение регулярных выражений в народном хозяйстве (grep/sed/awk) и для лексического анализа (lex/flex). Отсутствие бесконтекстной лексики в реальных языках программирования.

Контекстно-свободные грамматики. Нормальные формы Хомского и Грейбах. Алгоритмы Эрли и Кока-Янгера-Касами. Неконтекстосвободность реальных языков программирования.

Нисходящий анализ. Возврат и заглядывание вперед. Класс языков LL(k). Рекурсивный спуск, магазинные автоматы, парсер-комбинаторы, PEG, "скаредный" разбор. GLL. Инструменты нисходящего анализа (Parsec, ANTLR и пр.)

Восходящий анализ, классы LR(k) и LALR(k). GLR. Инструменты восходящего анализа (yacc/bison).

Двухуровневые и атрибутные грамматики, вопросы применения на практике.

Идентификация. Область видимости и область действия. Статическое и динамическое, раннее и позднее связывание.

Энергичность и ленивость. Call-by-name, call-by-value, call-by-reference.

Строгость, чистота, прозрачность по ссылкам.

Языки с типами и языки без типов. Статическая и динамическая типизация.

Номинальная и структурная эквивалентность типов. Простейшие конструкторы.

Типы с кванторами и что они означают. Универсальные и экзистенциальные типы.

Subtyping. Структурный и номинальный subtyping.

Динамическая семантика языков. Операционная семантика большого и малого шага.

Денотационный подход к описанию семантики.

Аксиоматическая семантика. Верификация программ. Design by contract.

Когерентность языков программирования и машинных архитектур. Языково-специфичные архитектуры, виртуальные машины и JIT-компиляция.

МОДУЛЬ 5 «Надежное программирование»:

Формализация требований к языку программирования. Каркас нового языка программирования. Тьюринг-полный и эзотерический языки программирования.

Перспективные языки программирования.

МОДУЛЬ 6 «Модульное программирование»:

Цель модульного программирования. Основные характеристики программного модуля. Типы связности и виды сцепления модулей в ПС. Методы разработки структуры программы. Контроль структуры программы.

МОДУЛЬ 7 «Функциональное программирование»:

Суть функционального программирования. Приемы и методы, используемые в функциональном программировании. Понятие функции. Побочные эффекты, чистые функции. Лямбда-функции. Абстрактность, декларативное программирование, императивное программирование. Теория категорий. Категория, морфизм, монада, функтор, аппликативный функтор, каррирование. Лямбда-выражения. Замыкания. Эффект. Примеры языков ФП.

МОДУЛЬ 8 «Логическое программирование»:

Суть функционального программирования. Приемы и методы, используемые в логическом программировании. Answer Set Programming (ASP). Комбинаторные задачи, комбинаторная оптимизация. Логическое программирование на классической логике FO(.) и IDP.

5.3. Лабораторные работы

Таблица 3а. Лабораторные занятия и их трудоемкость

Порядковый номер модуля. Цели лабораторных работ	Наименование Лабораторных работ	Трудоем кость в часах
Модуль 1. «Классификация знаний в области программирования» Цель: Изучить требования к знаниям и умениям специалистов по различным современным направлениям информационных технологий.	Теоретический минимум для программистов различных отраслей ИТ.	2
Модуль 3. «Специализированные вычислительные машины. Языки программирования для специализированных вычислительных машин» Цель: Изучить особенности функционирования специализированных вычислительных машин. Систематизировать подходы к разработке ПО для данного класса машин.	Систематизация подходов к разработке ПО для специализированных вычислительных машин.	2
Модуль 4. «Теория программирования» Цель: Выбрать и обосновать стратегию тестирования алгоритма работы программы. Изучить методы тестирования логики программы, формализованные описания результатов тестирования и стандарты по составлению схем программ. Научиться использовать динамические структуры данных при создании программного обеспечения.	Разработка тестов программы.	4
	Динамические структуры данных.	
Модуль 6. «Модульное программирование» Цель: Определить уровень внутренней связности модулей. Определить вид сцепления модулей.	Типы связности и виды сцепления модулей в ПС.	4

5.4. Практические занятия

Таблица 4а. Тематика, форма практических занятий (ПЗ) и их трудоемкость

Порядковый номер модуля. Цели практических работ	Примерная тематика занятий и формах их проведения	Трудоемкость в часах
Модуль 1 (4). «Теория программирования» Цель: Изучить избранные формальные теории, синтаксис и семантику языков программирования.	Теория схем программ. Избранные формальные теории. Синтаксис языков программирования. Семантика языков программирования. Функции для синтаксического разбора простых языковых конструкций. Подготовка студентами презентаций по заранее выданным темам, обсуждение приведенных сведений на занятии.	10
Модуль 2 (6). «Модульное программирование» Цель: Описать функции разрабатываемого в рамках подготовки ВКР программного обеспечения. Представить схему взаимодействия модулей.	Разработка структуры ПС. Аудиторно.	2

6. Самостоятельная работа обучающихся и текущий контроль их успеваемости

6.1. Цели самостоятельной работы

Формирование способностей к самостоятельному изучению, исследованию, поиску информации, а также обобщению, оформлению и представлению полученных результатов, их анализу, поиску новых решений, аргументированному отстаиванию своих предложений.

6.2. Организация и содержание самостоятельной работы

Самостоятельная работа заключается в изучении отдельных тем курса по заданию преподавателя по рекомендуемой им учебной литературе, в подготовке к лабораторным работам, к текущему контролю успеваемости, в выполнении курсовой работы и подготовке к экзамену.

После вводных лекций, в которых обозначается содержание дисциплины, ее проблематика и практическая значимость, студентам выдаются задания на лабораторные работы. Лабораторные работы охватывают модули 1,3-4,6.

В рамках дисциплины выполняется 5 лабораторных работы, которые защищаются устным опросом.

Выполнение всех лабораторных работ обязательно.

В случае невыполнения лабораторной работы по уважительной причине студент должен выполнить пропущенные лабораторные занятия в часы, отведенные на консультирование с преподавателем.

Так же при невыполнении лабораторной работы по уважительной причине студент может выполнить письменный реферат, по согласованной с преподавателем теме по модулю, по которому пропущена лабораторная работа. Возможная тематическая направленность реферативной работы для каждого учебно-образовательного модуля представлена в таблице 5.

Таблица 5. Темы рефератов

№ п/п	Модули	Возможная тематика самостоятельной реферативной работы
1.	Модуль 1	<p>Основные понятия направления «мультитрендность».</p> <p>Примерный список (выбрать для рассмотрения 3-5): обедающие философы, deadlock/livelock/race condition/starvation, атомарность, lock инструкции процессора, memory model/barrier/ordering, CAS или LL/SC, wait/lock/obstruction-free, ABA problem, написание lock-free контейнеров, spin-lock, TLS/per-thread data, закон Амдала, OpenMP, MPI, map-reduce, critical section/mutex/semaphore/condition variable, WaitForSingleObject/WaitForMultipleObjects, green thread/coroutine, pthreads, future/deferred/promise, модель акторов.</p> <p>Основные понятия направления «Криптография».</p> <p>Примерный список (выбрать для рассмотрения 3-5): Шнайер/Яценко, Принцип Керкгоффа, симметричная (DES, AES), асимметричная (RSA), качество ГПСЧ, алгоритм Диффи-Хеллмана, эллиптические кривые, хэширование (MD5, SHA, CRCn), DHT, криптостойкость, криптоатаки (атака гроссмейстера), WEP/WPA/WPA2 и атаки на них, цифровая подпись и сертификаты, PKI, HTTPS/SSL, доказательство с нулевым разглашением, пороговая схема.</p> <p>Аналогично другие направления: машинное обучение, архитектура и стиль кода, тестирование, фреймворки, инструментальные средства разработки и т.д.</p>
2.	Модуль 3	<p>Специализированные вычислительные машины, особенности программных разработок для специализированных вычислительных машин на примере.</p> <p>Возможные варианты:</p> <ol style="list-style-type: none"> 1. для самолетов и ракет: «Аргон 12С», «Орбита 10», ЦВМ-263, ЦВМ-264, ВМ 15Л579, 4К75 (для ракет), 8К67 (для ракет) и др.; 2. для военно-морского флота: «Море», «Корень», «Туча», система модулей «Азов», «Карат» и др.; 3. наземные стационарные и подвижные ВМ: 5Э51, 5765, «Кадр», 5Э89, ВНИИЭМ-3 (В-3М), Аргон-1, Клён, М4-2М, 5Э53 и др.
3.	Модуль 4	<p>Структура рабочей программы. Код, данные, библиотеки, поддержка времени исполнения.</p> <p>Задача генерации кода. Генерация кода путем интерпретации.</p> <p>Восходящее переписывание деревьев и динамическое программирование (BURS).</p> <p>Алгоритмы распределения регистров. Распределение регистров и раскраска графов.</p> <p>Параллелизм на уровне инструкций. Планирование инструкций.</p> <p>Анализ потока управления. Глубинное остовное дерево, доминирование, анализ циклической структуры программ. Сводимость. Устранение недостижимого кода, оптимальная линейаризация.</p> <p>Анализ потока данных. Полурешеточная модель. RD, LV, AE, UEU. Устранение мертвого кода, экономия общих подвыражений, понижение силы операций, чистка циклов.</p>
4.	Модуль 6	<p>Методы разработки структуры программы.</p> <p>Архитектура ИС.</p>

Оценивание в этом случае осуществляется путем устного опроса проводится по содержанию и качеству выполненного реферата.

Также, студентам выдается задание на курсовую работу. Курсовая работа выполняется в соответствии с методическими указаниями по выполнению курсовой работы, разработанными на кафедре.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература по дисциплине

1. Введение в теорию программирования. Объектно-ориентированный подход : учебное пособие / Зыков С.В.. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 187 с. — ISBN 978-5-4497-0926-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102007.html>. Текст : электронный. - (ID=139626)

2. Программирование. Основы алгоритмизации и программирования [Текст]: учебник для вузов по напр. подготовки "Информатика и вычислительная техника"; в составе учебно-методического комплекса / Парфилова, Н.И., Пылькин, А.Н., Трусов, Б.Г. ; под ред. Б.Г. Трусова - Москва: Академия, 2014. - 240 с. - (114353-2)

7.2. Дополнительная литература по дисциплине

1. Алгоритмизация и программирование. Практикум : учебное пособие для спо / А. А. Андрианова, Л. Н. Исмагилов, Т. М. Мухтарова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 240 с. — ISBN 978-5-8114-8948-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/186390>.

2. Унифицированные системы документации. Унифицированная система организационно-распорядительной документации. Требования к оформлению документов : ГОСТ Р 6.30-2003 : введ. 2003-07-01 : в составе учебно-методического комплекса. - Москва : Стандартинформ, 2003. - (УМК-ДМ). - Сервер. - Текст : электронный. - URL: <http://elib.tstu.tver.ru/MegaPro/GetDoc/Megapro/130864>. - (ID=130864-0)

3. Теория и практика объектно-ориентированного программирования : учебное пособие / Лебедева Т.Н.. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 221 с. — ISBN 978-5-4486-0663-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/81498.html>. Текст : электронный (DOI: <https://doi.org/10.23682/81498>)

4. Экономико-математические методы и прикладные модели : учебник для бакалавриата и магистратуры / А.Н. Гармаш, И.В. Орлова, В.В. Федосеев; под ред. В.В. Федосеева. - 4-е изд. ; перераб. и доп. - Москва : Юрайт, 2019. - (Бакалавр и магистр. Академический курс). - ЭБС Юрайт. - Текст : электронный. - ISBN 978-5-9916-3698-8. - (ID=94990-0)URL: <https://www.biblio-online.ru/book/ekonomiko-matematicheskie-metody-i-prikladnye-modeli-406453>

5. Программирование. Функциональный подход : учебник и практикум для академического бакалавриата / С.В. Зыков; Национальный исследовательский ун-т - Высшая Школа Экономики. - Москва : Юрайт, 2019. - (Бакалавр. Академический курс). - ЭБС Юрайт. - Текст : электронный. - ISBN 978-5-534-00844-9. - (ID=100866-0)URL: <https://www.biblio-online.ru/book/programmirovanie-funkcionalnyy-podhod-434613>

7.3. Методические материалы

Электронный курс по дисциплине «Теоретические основы программирования»:
<https://elearning.tstu.tver.ru/course/view.php?id=936>

7.4. Программное обеспечение по дисциплине

Операционная система Microsoft Windows: лицензии № ICM-176609 и № ICM-176613 (Azure Dev Tools for Teaching).

Microsoft Office 2007 Russian Academic: OPEN No Level: лицензия № 41902814.

7.5. Специализированные базы данных, справочные системы, электронно-библиотечные системы, профессиональные порталы в Интернет

ЭБС и лицензионные ресурсы ТвГТУ размещены:

1. Электронно-библиотечная система ТвГТУ lib.tstu.tver.ru
2. База данных учебно-методических комплексов cdokp.tstu.tver.ru/emc
3. Подсистема расчета и анализа показателей книгообеспеченности учебного процесса, включая книгообеспеченность кафедр и специальностей на период до 2019 года: cdokp.tstu.tver.ru/site2/wsite/ws_supply.asp?p=ws_supply.asp
4. ЭБС «Юрайт» www.biblio-online.ru
5. ЭБС «Лань» e.lanbook.com
6. ЭБС «Университетская библиотека онлайн» www.biblioclub.ru
7. ЭБС «IPRbooks» www.iprbookshop.ru
8. НЭБ ELIBRARY.RU elibrarv.ru
9. Гарант и Консультант Плюс

УМК размещен: <https://elib.tstu.tver.ru/MegaPro/GetDoc/Megapro/146312>

8. Материально-техническое обеспечение дисциплины

Кафедра «Информационных систем» имеет аудитории для проведения лекций и лабораторных занятий по дисциплине; специализированные учебные классы, оснащенные современной компьютерной техникой, необходимым программным обеспечением, электронными учебными пособиями для проведения лабораторных работ и самостоятельной работы.

В наличии имеются презентационные мультимедийные лекционные материалы по курсу «Теоретические основы программирования».

Для проведения лабораторных работ имеются лаборатории с персональными компьютерами (наличие локальной вычислительной сети необязательно).

9. Оценочные средства для проведения промежуточной аттестации

9.1. Оценочные средства для проведения промежуточной аттестации в форме экзамена

1. Экзаменационный билет соответствует форме, утвержденной Положением о рабочих программах дисциплин, соответствующих федеральным государственным образовательным стандартам высшего образования с учетом профессиональных стандартов. Типовой образец экзаменационного билета приведен в Приложении.

Обучающемуся даётся право выбора заданий из числа, содержащихся в билете, принимая во внимание оценку, на которую он претендует.

Число экзаменационных билетов – 20. Число вопросов (заданий) в экзаменационном билете – 3 (1 вопрос для категории «знать» и 2 вопроса для категории «уметь»).

Продолжительность экзамена – 60 минут.

2. Шкала оценивания промежуточной аттестации в форме экзамена – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

3. Критерии оценки за экзамен:

для категории «знать»:

выше базового – 2;

базовый – 1;

ниже базового – 0;

критерии оценки и ее значение для категории «уметь»:

отсутствие умения – 0 баллов;

наличие умения – 2 балла.

«отлично» - при сумме баллов 5 или 6;

«хорошо» - при сумме баллов 4;

«удовлетворительно» - при сумме баллов 3;

«неудовлетворительно» - при сумме баллов 0, 1 или 2.

4. Вид экзамена – письменный экзамен.

5. База заданий, предъявляемая обучающимся на экзамене.

1. Классификация знаний в области программирования.
2. Теоретический минимум для программиста (рассмотреть 2-3 области знаний).
3. История разработки языков программирования и компиляторов.
4. Понятие модульного программирования.
5. Надежное программирование. Основные требования к языкам программирования.
6. Методы тестирования ПС - классификация.
7. Эзотерические языки программирования (привести примеры).
8. Тьюринг-полные языки программирования (привести примеры).
9. Перспективные языки программирования (привести примеры).
10. Привести примеры языков функционального программирования.
11. Отличия объектно-ориентированного и функционального языков программирования.
12. Понятие функции. Побочные эффекты, чистые функции. Лямбда-функции.
13. Функциональное программирование. Абстрактность, декларативное программирование, императивное программирование.
14. Теория категорий. Категория, морфизм, монада, функтор, аппликативный функтор, каррирование.
15. Лямбда-выражения. Замыкания. Эффект.
16. Определить уровень внутренней связности модулей ПС (на основе выданного примера).
17. Определить тип сцепления модулей ПС (на основе выданного примера).

18. Тестирование ПС методами "белого ящика" (реализация комбинаторного метода по примеру).
19. Прагматика языков программирования.
20. Отличия низкоуровневых ЯП от высокоуровневых ЯП.
21. Отличия статической типизации от динамической.
22. Понятия компилятора и интерпретатора.
23. Проекция Футауры (назначение, область применения).
24. Пояснить понятия конкретного и абстрактного синтаксиса.
25. Определение динамической и статической семантики.
26. Грамматика Хомского.
27. Регулярные выражения (привести пример).
28. Алгоритмы Эрли и Кока-Янгера-Касами.
29. Понятие восходящего анализа. Инструменты восходящего анализа.
30. Возможности GLR-парсера.
31. Статическое и динамическое связывание.
32. Виды стратегий вычислений.
33. Нестрогие вычисления.
34. Свойства чистой функции.
35. Ссылочно-прозрачное выражение.
36. Классификация языков по способу типизации.
37. Отличия языков с явной и неявной типизацией.
38. Объяснить понятия структурной эквивалентности и именной эквивалентности. Привести примеры языков, в которых используется именная эквивалентность.
39. Типы с кванторами. Зашифруйте фразу "в каждой аудитории есть проектор".
40. Типы с кванторами. Расшифровать квантор - (пример).
41. В чем различие между семантикой малого и большого шага.
42. Денотационная семантика. Определение. Суть понятия.
43. Аксиоматическая семантика.
44. Когерентность языков программирования.
45. Определение понятия "библиотека", схематично представить структуру библиотеки.
46. Напишите пример рабочей программы, приведите описание каждого блока программы.
47. Оптимизация кода.
48. Стадия синтеза кода.
49. Понятие динамического программирования.
50. Использование JIT-компиляции.
51. Предназначение BURS (Bottom-Up Rewriting Systems).
52. Подход "распределение раскраски графа".
53. Анализ потока данных.
54. Устранение мертвого кода.
55. Понятие параллелизма на уровне инструкций. Планирование инструкций.
56. Практическое применение параллелизма.
57. Логическое программирование.

58. Теория схем программ.
59. Полурешеточная модель. RD, LV, AE, UEU.
60. Структура рабочей программы. Код, данные, библиотеки, поддержка времени исполнения.

Пользование различными техническими устройствами, кроме ЭВМ компьютерного класса и программным обеспечением, необходимым для решения поставленных задач, не допускается. При желании студента покинуть пределы аудитории во время экзамена экзаменационный билет после его возвращения заменяется.

Преподаватель имеет право после проверки письменных ответов на экзаменационные вопросы и решенных на компьютере задач задавать студенту в устной форме уточняющие вопросы в рамках содержания экзаменационного билета, выданного студенту.

Иные нормы, регламентирующие процедуру проведения экзамена, представлены в Положении о текущем контроле успеваемости и промежуточной аттестации студентов.

9.2. Оценочные средства для проведения промежуточной аттестации в форме зачета

Учебным планом зачет по дисциплине не предусмотрен.

9.3. Оценочные средства для проведения промежуточной аттестации в форме курсовой работы

1. Шкала оценивания курсовой работы (проекта) – «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

2. Тема курсовой работы: «Исследование специальных разделов теории программирования».

3. Критерии итоговой оценки за курсовую работу.

Таблица 5. Оцениваемые показатели для проведения промежуточной аттестации в форме курсовой работы

№ раздела	Наименование раздела	Баллы по шкале уровня
	Термины и определения	Выше базового – 2 Базовый – 1 Ниже базового – 0
	Введение	Выше базового – 2 Базовый – 1 Ниже базового – 0
1	Общая часть (обзор литературы по выбранной теме курсовой работы)	Выше базового – 6 Базовый – 3 Ниже базового – 0
2	Специальная часть	Выше базового – 10 Базовый – 6 Ниже базового – 0
	Заключение	Выше базового – 2 Базовый – 1 Ниже базового – 0

	Список использованных источников	Выше базового – 2 Базовый – 1 Ниже базового – 0
--	----------------------------------	---

Критерии итоговой оценки за курсовую работу (проект):

«отлично» – при сумме баллов от 22 до 24;

«хорошо» – при сумме баллов от 17 до 20;

«удовлетворительно» – при сумме баллов от 12 до 16;

«неудовлетворительно» – при сумме баллов менее 11, а также при любой другой сумме, если по разделу «Специальная часть», работа имеет 0 баллов.

4. В процессе выполнения курсовой работы руководитель осуществляет систематическое консультирование.

5. Дополнительные процедурные сведения:

- студенты выбирают тему для курсовой работы самостоятельно из предложенного списка и согласовывают свой выбор с преподавателем в течение двух первых недель обучения;

- проверку и оценку работы осуществляет руководитель, который доводит до сведения обучающего достоинства и недостатки курсовой работы и ее оценку. Оценка проставляется в зачетную книжку обучающегося и ведомость для курсовой работы. Если обучающийся не согласен с оценкой руководителя, проводится защита работы перед комиссией, которую назначает заведующий кафедрой;

- защита курсовой работы проводится в течение двух последних недель семестра и выполняется в форме устной защиты в виде доклада и презентации на 5-7 минут с последующим ответом на поставленные вопросы, в ходе которых выясняется глубина знаний студента и самостоятельность выполнения работы;

- работа не подлежит обязательному внешнему рецензированию;

- курсовые работы хранятся на кафедре в течение трех лет.

10. Методические рекомендации по организации изучения дисциплины

Студенты перед началом изучения дисциплины ознакомлены с системами кредитных единиц и балльно-рейтинговой оценки.

Студенты, изучающие дисциплину, обеспечиваются электронными изданиями или доступом к ним, учебно-методическим комплексом по дисциплине, включая методические указания к выполнению лабораторных, курсовых работ, всех видов самостоятельной работы

В учебный процесс рекомендуется внедрение субъект-субъектной педагогической технологии, при которой в расписании каждого преподавателя определяется время консультаций студентов по закрепленному за ним модулю дисциплины.

11. Внесение изменений и дополнений в рабочую программу дисциплины

Содержание рабочих программ дисциплин ежегодно обновляется протоколами заседаний кафедры по утвержденной «Положением о структуре, содержании и оформлении рабочих программ дисциплин по образовательным программам, соответствующим ФГОС ВО с учетом профессиональных стандартов» форме.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Тверской государственный технический университет»

Направление подготовки магистров – 09.04.02 Информационные системы и технологии

Направленность (профиль) – Разработка, внедрение и сопровождение информационных систем

Кафедра «Информационные системы»

Дисциплина «Теоретические основы программирования»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Вопрос для проверки уровня «ЗНАТЬ» – 0 или 1 или 2 балла:
Абстрактность, декларативное программирование, императивное программирование.

2. Задание для проверки уровня «УМЕТЬ» по разделу «Программирование» - 0 или 2 балла:

Типы с кванторами. Зашифруйте фразу «в каждой аудитории есть проектор».

3. Задание для проверки уровня «УМЕТЬ» по разделу «Модульное программирование» - 0 или 2 балла:

Определить тип связности, силу связности, сопровождаемость, роль модуля для следующих модулей:

**Модуль "Прием и проверка записи",
включает следующие элементы:**

- 1 - Прочитать запись из файла;
- 2 - Проверить контрольные данные в записи;
- 3 - Удалить контрольные поля в записи;
- 4 - Вернуть обработанную запись;

Конец модуля.

Критерии итоговой оценки за экзамен:

«отлично» - при сумме баллов 5 или 6;

«хорошо» - при сумме баллов 4;

«удовлетворительно» - при сумме баллов 3;

«неудовлетворительно» - при сумме баллов 0, 1 или 2.

Составитель: доц. каф. ИС _____ И.А. Егерова

Заведующий кафедрой: д.т.н., проф. каф. ИС _____ Б.В. Палюх