

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
биологической и медицинской
физики**

Д.В. Кузьмин

	Рабочая программа дисциплины (модуля)
по дисциплине:	Введение в программирование
по направлению:	Прикладные математика и физика
профиль подготовки:	Алгоритмическая биология
	Физтех-школа Биологической и Медицинской Физики
	центр образовательных программ Физтех-школы биологической и медицинской физики
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 1 (осенний) - Зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 60 час.

Всего часов: 90, всего зач. ед.: 2

Программу составил: А.Н. Голенкова, руководитель центра

Программа обсуждена на заседании центра образовательных программ Физтех-школы биологической и медицинской физики 12.07.2023

Аннотация

Дисциплина направлена на освоение базовых знаний в программировании. После освоения курса студент будет знать основы одного или нескольких алгоритмических языков программирования, общие характеристики языков программирования, фундаментальные структуры данных и алгоритмы.

1. Цели и задачи

Цель дисциплины

Изучение свойств, законов, методов и средств формирования, преобразования, хранения и распространения информации в природе и обществе при помощи ЭВМ.

Задачи дисциплины

- Формирование у обучающихся базовых знаний по информатике;
- Формирование информационной культуры: умение логически и алгоритмически мыслить, устанавливать логические связи между понятиями;
- Формирование умений и навыков применять полученные знания для решения задач при помощи ЭВМ, самостоятельного анализа полученных результатов.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Способен планировать и проводить научные исследования самостоятельно или в составе научного коллектива
	ПК-2.2 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- основы дискретной математики;
- основы теории алгоритмов;
- свойства алгоритмов, проблемы алгоритмической сложности и алгоритмической неразрешимости;
- основы одного или нескольких алгоритмических языков программирования, общие характеристики языков программирования, идеологию объектно-ориентированного подхода;
- приемы разработки программ;
- общие понятия о структурах данных: стеки, очереди, списки, деревья, таблицы;
- основы архитектуры электронно-вычислительной машины (ЭВМ), представления информации в ЭВМ и архитектурные принципы повышения их производительности.

уметь:

- выбирать оптимальные алгоритмы для современных программ;
- разрабатывать полные законченные программы на одном из языков программирования высокого уровня;
- разрабатывать программы на одном или нескольких языках программирования как индивидуально, так и в команде, с использованием современных средств написания и отладки программ;
- применять объектно-ориентированный подход для написания программ;
- использовать знания по информатике для приложения в инновационной, конструкторско-технологической и производственно-технологической сферах деятельности.

владеть:

- одним или несколькими современными языками программирования и методами создания программ с использованием библиотек и современных средств их написания и отладки;
- навыками освоения современных архитектур ЭВМ.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Алгоритмы сжатия данных (словарные методы)	2			2
2	Алгоритмы сжатия данных (статистические методы)	3			2
3	Алгоритмы сжатия изображений и звука	2			2
4	Введение в теорию алгоритмов	2			10
5	Графы и задачи на графах	2			2
6	Динамическое программирование	2			2
7	Жадные алгоритмы	3			2
8	Классы сложности алгоритмов	2			2
9	Метод ветвей и границ	3			2
10	Поиск с возвратом	2			2
11	Простые алгоритмы	2			10
12	Фундаментальные структуры данных и алгоритмы	2			12
13	Языки программирования	3			10
Итого часов		30			60
Подготовка к экзамену		0 час.			
Общая трудоёмкость		90 час., 2 зач.ед.			

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

Семестр: 1 (Осенний)

1. Алгоритмы сжатия данных (словарные методы)

Алгоритмы Лемпеля-Зива.

2. Алгоритмы сжатия данных (статистические методы)

Коды Хаффмана.

3. Алгоритмы сжатия изображений и звука

Алгоритмы сжатия без потерь и с потерями. Основные форматы хранения видео и аудио информации.

4. Введение в теорию алгоритмов

Интуитивное понятие алгоритма. Свойства алгоритмов. Понятие об исполнителе алгоритма. Алгоритм как преобразование слов из заданного алфавита. Связь понятия алгоритма с понятием функции. Машина Тьюринга. Нормальные алгорифмы Маркова. Вычислимые функции и их свойства. Невычислимые функции. Различные эквивалентные определения множества вычислимых функций. Алгоритмическая сложность.

5. Графы и задачи на графах

Представление графов (списки смежности, матрица смежности, матрица инцидентности). Обходы графов в ширину и в глубину. Минимальные пути в графе (алгоритм Дейкстры). Минимальное остовное дерево (алгоритмы Прима и Краскала). Топологическая сортировка графа.

6. Динамическое программирование

Разбиение задачи на пересекающиеся подзадачи. Получение рекуррентных соотношений. Получение решения исходной задачи. Задача о минимальных путях в графе. Задача о транзитивном замыкании.

7. Жадные алгоритмы

Описание жадной стратегии. Примеры задач, когда жадная стратегия дает глобально оптимальное решение и когда не дает.

8. Классы сложности алгоритмов

Описание различных классов сложности алгоритмов. Полиномиальная сложность алгоритмов. Неполиномиальная сложность алгоритмов. NP-полные задачи. Приведение.

9. Метод ветвей и границ

Описание метода ветвей и границ. Примеры задач решаемых этим методом.

10. Поиск с возвратом

Описание алгоритма поиска с возвратом. Примеры задач, решаемых с помощью поиска с возвратом.

11. Простые алгоритмы

Понятие внутренней и внешней сортировки. Устойчивая сортировка. Сортировка in-place. Сортировка простыми вставками, простым выбором, метод «пузырька». Шейкер сортировка. Метод Шелла. Быстрая сортировка Хора. Сортировка слиянием. Пирамидальная сортировка. Оценка трудоемкости.

12. Фундаментальные структуры данных и алгоритмы

Абстрактные структуры данных: список, стек, очередь, очередь с приоритетом, ассоциативный массив. Отображение абстрактных структур данных на структуры хранения: массивы, линейные списки, деревья. Различные реализации ассоциативного массива: двоичные деревья поиска (АВЛ-деревья, красно-чёрные деревья), перемешанные таблицы (с прямой и открытой адресацией, использование техники двойного хэширования при открытой адресации). Оценки алгоритмической сложности операций поиска, добавления и удаления элемента. Классические алгоритмы: перебор с возвратом, жадные алгоритмы. Примеры алгоритмов работы с графами: поиск минимального остового дерева, поиск кратчайшего пути, задача коммивояжера.

13. Языки программирования

Характеристика алгоритмических языков и их исполнителей. Понятие трансляции. Понятие о формальных языках. Способы строгого описания формальных языков, понятие о метаязыках. Алфавит, синтаксис и семантика алгоритмического языка. Описание синтаксиса языка с помощью металингвистических формул и синтаксических диаграмм. Языки программирования. Общие характеристики языков программирования. Алфавит, имена, служебные слова, стандартные имена, числа, текстовые константы, разделители. Препроцессор и комментарии. Типы данных, их классификация. Переменные и константы. Скалярные типы данных и операции над ними. Старшинство операций, стандартные функции. Выражения и правила их вычисления. Оператор присваивания. Файлы. Стандартные функции ввода-вывода. Простые и сложные операторы. Пустой, составной, условный операторы. Оператор варианта. Оператор перехода. Оператор цикла. Программирование рекуррентных соотношений. Составные типы данных. Массивы. Описание функций (процедур). Формальные и фактические параметры. Способы передачи параметров. Локализация имен. Побочные эффекты. Итерации и рекурсии. Ссылочный тип данных. Методы выделения памяти: статический, динамический и автоматический. Структуры. Битовые поля. Объединения. Перечисления. Декларация typedef.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Компьютерный класс, оснащенный видеопроектором или телевизором.

6. Перечень рекомендуемой литературы

Основная литература

1. Практика и теория программирования [Текст] : в 2 кн. : учеб. пособие для вузов / Н. А. Винокуров, А. В. Ворожцов .— М. : Физматкнига, 2008 .— (Серия "Информатика"). - ISBN 978-5-89155-182-4 (в пер.) .— Кн.2, Ч. 3-4. - 2008. - 288 с.
2. Язык программирования C [Текст] : [учеб. пособие для вузов] / Б. Керниган, Д. Ритчи ; пер. с англ. и ред. В. Л. Бродового .— 2-е изд., перераб. и доп. — М. : Вильямс, 2006, 2007, 2009, 2010, 2012, 2013, 2015 .— 304 с.

Дополнительная литература

1. Алгоритмы: построение и анализ [Текст] : [учебник для вузов] / Т. Кормен [и др.] ; [пер. с англ. И. В. Красикова и др.] .— 3-е изд. — М. : Вильямс, 2014 .— 1328 с.
2. Алгоритмы [Текст] : [учеб. пособие для вузов] / С. Дасгупта, Х. Пападимитриу, У. Вазирани ; пер. с англ. А. А. Куликова ; под ред. А. Шеня .— М. : МЦНМО, 2014 .— 320 с.
3. Структуры данных и алгоритмы [Текст] / А. В. Ахо, Д. Э. Хопкрофт, Д. Д. Ульман ; пер. с англ. А. А. Минько .— М. : Вильямс, 2001, 2007 .— 384 с.
4. Алгоритмы и структуры данных [Текст] / Н. Вирт ; пер. с англ. Д. Б. Подшивалова .— 2-е изд., испр. — СПб. : Невский Диалект, 2001, 2005 .— 352 с.
5. Алгоритмы на C++. Анализ, структуры данных, сортировка, поиск, алгоритмы на графах [Текст] / Р. Седжвик ; [пер. с англ. А. А. Моргунова ; под ред. Ю. Н. Артеменко] .— [Научное изд.] .— М. : Вильямс, 2011 .— 1056 с.
6. Элементы теории алгоритмов и язык программирования C [Текст] : учеб. пособие для вузов / В. Я. Митницкий, М-во образования РФ, Моск. физ.-техн. ин-т (гос. ун-т) .— М. : МФТИ, 2001 .— 180 с.

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

<http://cs.mipt.ru>
<http://judge.mipt.ru>
<http://acm.mipt.ru>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лекциях используются мультимедийные технологии, включая демонстрацию презентаций. Для контроля и коррекции знаний обучающиеся могут использовать компьютерное тестирование, в том числе на сайте www.judge.mipt.ru. В процессе самостоятельной работы обучающихся возможно использование любые среды программирования.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Студент, изучающий курс должен, с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения, понятия, алгоритмы, уметь писать многопроцессные и многопоточные приложения в среде операционной системы Linux, корректно организовывать взаимодействие процессов и потоков, как локальных, так и удаленных, работать с файлами и устройствами ввода-вывода.

Успешное освоение курса требует напряжённой работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы;
- проработку учебного материала (по конспектам лекций, учебной и научной литературе), доказательство отдельных утверждений, свойств;
- решение задач, предлагаемых студентам на лекциях и лабораторных работах;
- подготовку к лабораторным работам;
- решение заданий.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Показателем владения материалом служит умение решать теоретические и практические задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше практических задач. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения. Программы должны легко читаться и иметь подробные комментарии.

Важно добиться понимания изучаемого материала, а не механического его запоминания.

ПРИЛОЖЕНИЕ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладные математика и физика
профиль подготовки:	Алгоритмическая биология Физтех-школа Биологической и Медицинской Физики центр образовательных программ Физтех-школы биологической и медицинской физики
курс:	<u>1</u>
квалификация:	магистр
Семестр, формы промежуточной аттестации: 1 (осенний) - Зачет	
Разработчик:	А.Н. Голенкова, руководитель центра

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Способен планировать и проводить научные исследования самостоятельно или в составе научного коллектива
	ПК-2.2 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

2. Показатели оценивания компетенций

В результате изучения дисциплины «Введение в программирование» обучающийся должен:

знать:

- основы дискретной математики;
- основы теории алгоритмов;
- свойства алгоритмов, проблемы алгоритмической сложности и алгоритмической неразрешимости;
- основы одного или нескольких алгоритмических языков программирования, общие характеристики языков программирования, идеологию объектно-ориентированного подхода;
- приемы разработки программ;
- общие понятия о структурах данных: стеки, очереди, списки, деревья, таблицы;
- основы архитектуры электронно-вычислительной машины (ЭВМ), представления информации в ЭВМ и архитектурные принципы повышения их производительности.

уметь:

- выбирать оптимальные алгоритмы для современных программ;
- разрабатывать полные законченные программы на одном из языков программирования высокого уровня;
- разрабатывать программы на одном или нескольких языках программирования как индивидуально, так и в команде, с использованием современных средств написания и отладки программ;
- применять объектно-ориентированный подход для написания программ;
- использовать знания по информатике для приложения в инновационной, конструкторско-технологической и производственно-технологической сферах деятельности.

владеть:

- одним или несколькими современными языками программирования и методами создания программ с использованием библиотек и современных средств их написания и отладки;
- навыками освоения современных архитектур ЭВМ.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. Интуитивное понятие алгоритма.
2. Свойства алгоритмов.
3. Понятие об исполнителе алгоритма.
4. Алгоритм как преобразование слов из заданного алфавита.
5. Связь понятия алгоритма с понятием функции.
6. Машина Тьюринга.
7. Нормальные алгорифмы Маркова.
8. Вычислимые функции и их свойства.
9. Невычислимые функции.
10. Различные эквивалентные определения множества вычислимых функций.
11. Алгоритмическая сложность.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1. Абстрактные структуры данных: список, стек, очередь, очередь с приоритетом, ассоциативный массив.
2. Отображение абстрактных структур данных на структуры хранения: массивы, линейные списки, деревья.
3. Различные реализации ассоциативного массива: двоичные деревья поиска (АВЛ-деревья, красно-чёрные деревья), перемешанные таблицы (с прямой и открытой адресацией, использование техники двойного хэширования при открытой адресации).
4. Оценки алгоритмической сложности операций поиска, добавления и удаления элемента.
5. Классические алгоритмы: перебор с возвратом, жадные алгоритмы.
6. Примеры алгоритмов работы с графами: поиск минимального остового дерева, поиск кратчайшего пути, задача коммивояжера.
7. Представление графов (списки смежности, матрица смежности, матрица инцидентности).
8. Обходы графов в ширину и в глубину.
9. Минимальные пути в графе (алгоритм Дейкстры).
10. Минимальное остовное дерево (алгоритмы Прима и Краскала).
11. Топологическая сортировка графа.

Критерии оценивания

Оценка "зачтено" - ставится при правильном ответе на вопросы к зачету,

Оценка "не зачтено" - ставится при неправильном ответе.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.