

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

**УТВЕРЖДЕНО**

**Директор высшей школы  
программной инженерии  
А.В. Малеев**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Практикум по программированию и алгоритмам
<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 0 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 42 час.

Всего часов: 72, всего зач. ед.: 2

Программу составил: А.В. Созыкин, канд. техн. наук

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.04.2023

## Аннотация

Практикум предназначен для освоения практических навыков по программированию на языке Си и реализации базовых алгоритмов.

Во время изучения практикума студенты научатся реализации алгоритмов, и освоют основные техники для получения работоспособных программ, а также способы оценки сложности алгоритмов и времени их работы.

### 1. Цели и задачи

#### Цель дисциплины

- дисциплина направлена на формирование, закрепление и развитие практических навыков и компетенций по профилю образовательной программы.

#### Задачи дисциплины

Перед студентами ставятся следующие задачи:

- освоение инструментов и языка программирования;
- овладение навыками выбора подходящих алгоритмов для решения прикладных задач;
- овладение навыками реализации алгоритмов на языке программирования C++.

### 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
	ОПК-5.3 Владеет навыками установки программного и аппаратного обеспечения информационных и автоматизированных систем

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- оценки сложности стандартных алгоритмов;
- алгоритмы сортировки массивов;
- алгоритмы поиска подстрок в строке.

уметь:

- разбираться в описаниях алгоритмов;
- приближенно оценивать сложности алгоритмов;
- реализовывать стандартные алгоритмы на языке программирования C++;
- оформлять программный код в соответствии с принятыми требованиями;
- выполнять тестирование работоспособности написанных программ.

владеть:

- навыками анализа технической информации в области, соответствующей профилю образовательной программы;
- методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов;
- приемами сведения общих задач к более конкретным и простым.

#### 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	Введение		6		3
2	Решение задач на отработку навыков программирования		8		13
3	Решение алгоритмических задач с учетом вычислительной сложности		8		14
4	Контрольные мероприятия		8		12
Итого часов			30		42
Подготовка к экзамену		0 час.			
Общая трудоёмкость		72 час., 2 зач.ед.			

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

Семестр: 1 (Осенний)

###### 1. Введение

Проведение инструктажа по технике безопасности, ознакомление с правилами внутреннего распорядка. Установка необходимого программного обеспечения. Написание простейшего программного кода. Разбор стадий компиляции программ. Использование пошагового отладчика.

###### 2. Решение задач на отработку навыков программирования

Разбор алгоритмов поиска численных решений уравнений. Реализация алгоритмов на языке C++ с использованием функций. Реализация алгоритмов для решения геометрических задач.

###### 3. Решение алгоритмических задач с учетом вычислительной сложности

Формализация сложности алгоритма. Оценка сложности простых алгоритмов. Разбор различных алгоритмов сортировки и их сложности. Решение задач на реализацию этих алгоритмов. Представление строк в языках программирования Си и C++. Задача поиска подстроки в строке, ее алгоритмическая сложность. Понятие префикса подстроки. Реализация алгоритмов поиска подстрок в строках.

###### 4. Контрольные мероприятия

Демонстрация работоспособности реализованных решений. Решение итоговой контрольной.  
Подготовка отчета о проделанных задачах.

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

Для проведения дисциплины необходимы: рабочее место для самостоятельной работы, содержащее персональный компьютер, с доступом к сети Интернет и электронной образовательной среде МФТИ, учебные аудитории для проведения групповых и индивидуальных консультаций, проведения промежуточной аттестации (по месту прохождения практики).

## **6. Перечень рекомендуемой литературы**

### Основная литература

1. Введение в теорию алгоритмов и структур данных, Электронная версия печатной публикации / М. А. Бабенко, М. В. Левин. — Москва, МЦНМО, 2016
2. Алгоритмы и структуры данных. Новая версия для Оберона, Электронная версия печатной публикации / Н. Вирт. — Москва, ДМК Пресс, 2010

### Дополнительная литература

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

Не используются

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

Программное обеспечение: набор компиляторов GNU GCC, либо среда Microsoft Visual Studio.

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

В преподавании дисциплины сочетаются практические занятия в учебном классе и домашние задания, состоящие из теоретических задач.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Программная инженерия
<b>профиль подготовки:</b>	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр
Семестр, формы промежуточной аттестации: 1 (осенний) - Дифференцированный зачет	
<b>Разработчик:</b>	А.В. Созыкин, канд. техн. наук

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
	ОПК-5.3 Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Практикум по программированию и алгоритмам» обучающийся должен:

### знать:

- оценки сложности стандартных алгоритмов;
- алгоритмы сортировки массивов;
- алгоритмы поиска подстрок в строке.

### уметь:

- разбираться в описаниях алгоритмов;
- приближенно оценивать сложности алгоритмов;
- реализовывать стандартные алгоритмы на языке программирования C++;
- оформлять программный код в соответствии с принятыми требованиями;
- выполнять тестирование работоспособности написанных программ.

### владеть:

- навыками анализа технической информации в области, соответствующей профилю образовательной программы;
- методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов;
- приемами сведения общих задач к более конкретным и простым.

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

1. Информационные носители. Передача информации. Обработка информации. Меры информации
2. Представление информации в разных языковых системах. Кодирование текстовых данных.
3. Понятие системы счисления. Двоичная система счисления. Алгоритмы перевода из одной системы счисления в другую.
4. Алгоритм и его свойства. Этапы решения задач с помощью компьютера.
5. Понятие рекурсивных алгоритмов. Глубина рекурсии. Возможности использования рекурсивных алгоритмов для решения задач математики и информатики.
6. Понятие сортировки. Формулировка задачи сортировки. Возможности применения алгоритмов сортировки данных при решении задач.
7. Алгоритм сортировки методом простого обмена.
8. Алгоритм сортировки шейкер-методом.

9. Алгоритм сортировки методом прямого выбора.
10. Алгоритм сортировки вставками.
11. Метод сортировки Хоара.
12. Задача поиска. Понятие ключа. Проблема дублирующихся ключей.
13. Методы поиска. Линейный (последовательный) поиск.
14. Метод половинного деления (дихотомический поиск).
15. Бинарный поиск в массиве.
16. Возможности разработки программ (с использованием визуальных сред) на основе известных алгоритмов и методов обработки информации для осуществления профессиональной деятельности
17. Реализовать алгоритм численного решения заданного уравнения
18. Реализовать алгоритм поиска количества пар точек с заданным расстоянием на плоскости
19. Реализовать алгоритм поиска количества пар точек с заданным расстоянием в пространстве
20. Доказать, что сложность алгоритма сортировки вставками равна  $O(N^2)$
21. Реализовать алгоритм сортировки слиянием
22. Реализовать алгоритм быстрой сортировки
23. Реализовать алгоритм поиска количества заданных подпоследовательностей в строке
24. Реализовать алгоритм поиска подстроки в строке.

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

1. Бинарные деревья поиска. Определение, базовые свойства.
2. B+-дерево. Простейшие операции. Операции Split и Merge.
3. Splay-дерево. Операции. Доказательство амортизированного времени работы.
4. Декартово дерево. Операции Split и Merge. Оценка матожидания глубины вершины. Декартово дерево по неявному ключу.
5. Запросы на отрезках. Дерево отрезков, общие идеи. Sparse table. Disjoint sparse table.
6. Персистентность. Общие идеи, примеры частично и полностью персистентных структур данных. Применение в задачах.
7. Методы Path copying и Fat nodes. Их комбинирование для получения частично персистентного списка без штрафа по времени и памяти.
8. LCA. Метод двоичных подъёмов. Алгоритм Тарьяна с СНМ. Алгоритм Фараха-Колтона и Бендера.
9. Запросы на деревьях. Heavy-light декомпозиция, оптимизация времени работы до  $O(\log n)$  на запрос.
10. Переборные алгоритмы. Подходы через поиск в глубину и поиск в ширину. Способы хеширования состояний. Iterative deepening. Примеры отсечений.
11. Кратчайшие пути в графах. Алгоритмы Дейкстры и Форда-Беллмана. Двусторонний алгоритм Дейкстры.
12. Алгоритм Contraction Hierarchies для поиска кратчайших путей в графах, возникающих на практике.
13. Альфа-бета отсечение в антагонистических играх.
14. Мосты, точки сочленения. Построение деревьев компонент рёберной и вершинной двусвязности.
15. Конденсация ориентированного графа. Алгоритм Тарьяна.
16. Система непересекающихся множеств. Доказательство амортизированного времени работы  $O(\log^* n)$  на запрос.
17. Минимальные остовные деревья. Алгоритмы Прима, Краскала, Борувки.
18. Линейный вероятностный алгоритм построения MST (в предположении существования алгоритма проверки минимальности за линейное время). Проверка остовного дерева на минимальность за  $O(m \log \log n)$ .
19. Потоки в сетях. Теорема Форда-Фалкерсона. Примеры решения задач через сведение к минимальному разрезу или максимальному потоку.
20. Теорема Кёнига-Эгервари, лемма Холла. Доказательство через теорему Форда-Фалкерсона.

21. Алгоритм Эдмондса-Карпа. Алгоритм Диница, доказательство времени работы  $O(E \sqrt{V})$  при поиске максимального двудольного паросочетания.

#### Критерии оценивания

отлично

10 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач. Решения задач оформлены в едином удобочитаемом стиле.

9 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач

8 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач

хорошо

7 Полностью решены все задачи. Допущены несущественные ошибки.

6 Полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

5 Полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

удовлетворительно

4 Полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

3 Полностью решено более половины задач.

неудовлетворительно

2 Решено менее половины задач

1 Не решено ни одной задачи.

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

Обучающийся решает задачи и защищает их у семинариста. Процесс защиты заключается в демонстрации кода решения задачи и объяснения его работы.

Неправильно решенная задача или задача, имеющая ошибки, отправляется на доработку с возможностью повторной защиты. Количество повторных защит регламентируется преподавателем (семинаристом).

Защита может выполняться удаленно с использованием электронной почты, внешним репозиторием системы контроля версий и др.