



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ИГУ»)**

Институт математики и информационных технологий  
Кафедра алгебраических и информационных систем

**«УТВЕРЖДАЮ»**

Директор ИМИТ ИГУ

*М. В. Фалалеев*  
М. В. Фалалеев

«19» мая 2021 г.



**Рабочая программа дисциплины**

**Наименование дисциплины (модуля)** Б1.В.16 Структуры данных

**Направление подготовки** 09.03.03 Прикладная информатика

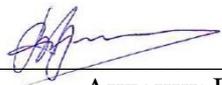
**Направленность (профиль) подготовки** Проектирование и разработка информационных систем

**Квалификация выпускника** – бакалавр

**Форма обучения** очная

Согласовано с УМК Института математики  
и информационных технологий  
Протокол № 4 от «19» мая 2021 г.

Председатель \_\_\_\_\_



Антоник В.Г.

Рекомендовано кафедрой Алгебраических и  
информационных систем ИМИТ ИГУ:  
Протокол № 11 от «29» апреля 2021 г.

Зав. кафедрой \_\_\_\_\_



Пантелеев В.И.

## СОДЕРЖАНИЕ

1.	ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ	4
2.	МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО	4
3.	ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	4
4.	СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ	7
4.1.	СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ	7
4.2.	План внеаудиторной самостоятельной работы обучающихся по дисциплине	8
4.3.	СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА	9
4.4.	МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ	13
4.5.	ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)	13
5.	УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)	14
6.	МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ	14
7.	ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	15
8.	ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	15

## 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

### Цель

Сформировать компетенции в области хранения данных на основе понимания принципов организации структур данных и оценок временной сложности методов доступа к данным

### Задачи:

Научится использовать методы прогнозирования времени работы алгоритмов на основе теории вычислительной сложности. Изучить способы организации памяти в структурах данных с произвольным доступом, понимать особенности реаллокации данных и корректного использования итераторов. Изучить устройство и приемы работы структур данных с ограниченным доступом: стек, очередь, приоритетная очередь. Изучить приемы выполнения групповых операций и отложенных вычислений на примере дерева отрезков. Изучит ссылочные структуры данных: списки и деревья. Изучить алгоритмы работы с различными видами деревьев сортировки. Научиться реализовывать множества и отображения на основе деревьев сортировки и хэш-таблиц.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Учебная дисциплина (модуль) относится к части программы, формируемой участниками образовательных отношений, и изучается на втором курсе.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы знания, умения и навыки, сформированные предшествующими дисциплинами: математика, информатика, программирование, основы алгоритмизации, дискретная математика, линейная алгебра, языки программирования.

2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной: проектирование информационных систем, алгоритмы и анализ сложности, интеллектуальный анализ данных.

## 3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование компетенций (элементов следующих компетенций) в соответствии с ФГОС ВО по соответствующему направлению подготовки.

### Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-2 Способность разрабатывать и адаптировать прикладное программное обеспечение.	ИДК ПК2.1 Способен записывать программный код в соответствии с требуемой парадигмой программирования	Знает виды структур данных, их особенности и характеристики Умеет записывать программный код с использованием структур данных. Владеет навыком использования контейнерных типов в программировании
	ИДК ПК2.2 Способен выбирать и анализировать способы	Знает оценки сложности выполнения запросов к данным при различных способах их организации

	решения прикладных задач,	Умеет выбирать и анализировать способы организации данных в прикладных программах Владеет навыком использования ассоциативных контейнеров, множеств и хэш-таблиц
	ИДК ПК2.3 Способен адаптировать и модифицировать программное обеспечение в соответствии с требованиями	Знает требования к интерфейсам контейнерных типов. Умеет адаптировать и модифицировать алгоритмы в соответствии с требованиями к интерфейсам контейнерных типов. Владеет навыком программирования итераторов и контейнерных типов на языке C++
ПК-3 Способность демонстрации общенаучных базовых знаний математических и естественных наук и информационных технологий; способность применять в профессиональной деятельности современные языки программирования, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии.	ИДК ПК3.1 Способен использовать математические знания в профессиональной деятельности	Знает математические основы аппарата теории вычислительной сложности Умеет определять класс сложности по алгоритму или тексту программы Владеет навыком оценки планируемого времени работы программы при выборе способа ее реализации.
	ИДК ПК3.2 Способен использовать теоретические принципы информационных технологий в профессиональной деятельности	Знает Теоретические основы построения случайных деревьев, хэш-таблиц и ленивых вычисление в дереве отрезков. Умеет реализовывать сложные структуры данных на основе знаний о принципах их организации Владеет техникой реализации итераторов для деревьев и хэш-таблиц
	ИДК ПК3.3 Способен применять в профессиональной деятельности современные языки программирования,	Знает особенности реализации контейнерных типов в различных языках программирования

	операционные системы пакеты программ и сетевые технологии	Умеет использовать в профессиональной деятельности языки программирования Java, Python и C++. Владеет навыком программирования с использованием стандартной библиотеки C++, контейнерных типов Java и Python
--	---	---

#### 4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Объем дисциплины составляет 3 зачетных единицы, 108 часов, практическая подготовка 108 часов.

Форма промежуточной аттестации: 4 семестр - зачет с оценкой.

##### 4.1. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ

№ п/п	Раздел дисциплины/темы	Се мес тр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоя тельная работа + контроль	
			Лекции	Семинарские (практические занятия)	Контроль обучения		
1	<b>Прогнозирование времени работы программ</b>	4	6	6		8	отчет
	Тема 1. Сравнение времени работы алгоритмов		3	3		4	отчет
	Тема 2. Введение в теорию вычислительной сложности		3	3		4	отчет
2	<b>Контейнеры с произвольным доступом</b>	4	6	6		8	отчет
	Тема 1. Массивы и контейнеры с концевой вставкой		2	2		2	отчет
	Тема 2. Способы организации памяти в контейнерах с произвольным доступом		2	2		4	отчет
	Тема 3. Дек.		2	2		2	отчет
3	<b>Структуры данных на основе полного бинарного дерева</b>	4	6	6		8	отчет
	Тема 1. Двоичная куча		2	2		4	отчет
	Тема 2. Дерево отрезков		4	4		4	отчет
4	<b>Ссылочные структуры данных</b>	4	10	10		16	отчет

№ п/п	Раздел дисциплины/темы	Се мес тр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)			Самостоя тельная работа + контроль	Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися				
			Лекции	Семинарские занятия (практические занятия)	Контроль обучения		
	Тема 1. Связный список		2	2		4	
	Тема 2. Дерево поиска		2	2		4	отчет
	Тема 3. AVL-дерево		3	3		4	отчет
	Тема 4. Случайные деревья		3	3		4	отчет
5	<b>Хэш-таблицы</b>	4	4	4		4	отчет
<b>Итого часов</b>			32	32		44	

#### 4.2. План внеаудиторной самостоятельной работы обучающихся по дисциплине

Семестр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятельной работы	Сроки выполнения	Затраты времени (час.)		
4	Прогнозирование времени работы программ	УИЛ	1-3 недели	8	отчет	доступно из ИОС Домик
4	Контейнеры с произвольным доступом	УИЛ	4-6 недели	8	отчет	доступно из ИОС Домик
4	Структуры данных на основе полного бинарного дерева	УИЛ	7-9 недели	8	отчет	доступно из ИОС Домик

Семестр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно-методическое обеспечение самостоятельной работы
		Вид самостоятельной работы	Сроки выполнения	Затраты времени (час.)		
4	Ссылочные структуры данных	УИЛ	10-14 недели	16	отчет	доступно из ИОС Домик
4	Хэш-таблицы	УИЛ	15-16 недели	4	отчет	доступно из ИОС Домик
Общая трудоемкость самостоятельной работы по дисциплине (час)				44		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				<b>44</b>		

### 4.3. СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

#### 1. Прогнозирование времени работы программ.

Тема 1. Сравнение времени работы алгоритмов

Профилировщики кода. Функции для измерения времени в различных языках программирования. Факторы, влияющие на время работы программы. Тестирование производительности.

Тема 2. Введение в теорию вычислительной сложности

Представление о вычислительной сложности. Средняя и максимальная сложность. Асимптотические оценки. Классы сложности. Методы доказательства оценок сложности. Исследование сложности некоторых алгоритмов.

#### 2. Контейнеры с произвольным доступом.

Тема 1. Массивы и контейнеры с концевой вставкой

Произвольный доступ с элементом. Итераторы произвольного доступа в языке C++. Массивы с изменяемым размером. Реаллокация элементов и допустимость итераторов. Реализация концевой вставки за амортизированное константное время. Реализация контейнеров `std::array` и `std::vector` стандартной библиотеки C++

Тема 2. Способы организации памяти в контейнерах с произвольным доступом.

Двухуровневая схема организации памяти. Реализация итераторов произвольного доступа при двухуровневой организации памяти. Сравнение схем организации памяти. Реализация контейнера с концевой вставкой с двухуровневой схемой организации памяти.

Тема 3. Дек.

Контейнеры с двухсторонней вставкой в Python и C++. Способы реализации двухсторонней вставки. Обзор реализации контейнера `std::deque` стандартной библиотеки C++. Реализация дека на основе циклического буфера.

#### 3. Структуры данных на основе полного бинарного дерева

Тема 1. Двоичная куча

Реализация хранения полного бинарного дерева в массиве. Процедуры движения по дереву снизу вверх и сверху вниз. Инвариант двоичной кучи. Алгоритмы построения двоичной кучи и их сложность. Алгоритмы вставки и извлечения элементов из двоичной кучи. Реализация контейнера `std::priority_queue` стандартной библиотеки C++. Реализация пирамидальной сортировки и ее сложность.

Тема 2. Дерево отрезков

Ассоциативные операции на отрезках. Сохранение результатов вычислений. Вычисление результатов операций на отрезке алгоритмом рекурсивного спуска и алгоритмом подъема. Алгоритмы поиска в дереве отрезков и их сложность. Алгоритмы выполнения запросов на изменение данных в дереве отрезков. Групповые отложенные операции. Реализация ленивых вычислений в дереве отрезков. Реализация дерева отрезков в стиле стандартной библиотеки C++.

#### **4. Ссылочные структуры данных**

Тема 1. Связный список

Реализация узла в ссылочной структуре данных. Узел двусвязного списка. Реализация итератора списка. Операции вставки, удаления, склейки и их сложность. Реализация контейнера `std::list` стандартной библиотеки C++. Взаимодействие методов `size()` и `splice()`.

Тема 2. Дерево поиска

Определение бинарного дерева поиска. Узел бинарного дерева. Построение дерева. Поиск вставка и удаление элементов в дереве. Способы реализации удаления узла. Итерация по дереву, реализация итератора. Сложность операций поиска предыдущего и следующего элемента. Реализация контейнера `std::set` стандартной библиотеки C++ на основе дерева поиска.

Тема 3. AVL-дерево

Проблема сложности операций в дереве поиска. Сбалансированные и почти сбалансированные деревья. Операции одинарного и двойного поворота. Некоторые виды почти сбалансированных деревьев. Инвариант красно-черного дерева и AVL-дерева. Алгоритмы балансировки AVL-дерева. Реализация контейнера `std::set` стандартной библиотеки C++ на основе AVL-дерева. Сравнение со стандартной реализацией.

Тема 4. Случайные деревья

Определение случайного дерева. Использование случайных деревьев в программировании. Оценки высоты случайного дерева. Рандомизированные алгоритмы разрезания и слияния деревьев. Рандомизированные алгоритмы удаления и вставки и их сложность. Дерево с неявными ключами. Неявный ключ последовательной индексации. Итератор произвольного доступа для дерева с неявными ключами. Реализация контейнера в стиле стандартной библиотеки C++ на основе случайного дерева с неявным ключом последовательной индексации.

#### **5. Хэш-таблицы**

Принцип хэширования. Хэш-функции, виды хэш-функций. Свойства хэш-функций. Структура `std::hash` стандартной библиотеки C++. Некоторые алгоритмы хэширования и их реализация. Хэш-таблицы. Коллизии в хэш-таблицах. Списочный и циклический способ разрешения коллизий. Итерация в хэш-таблицах. Характеристики и оптимизация хэш-таблиц. Построение контейнера на основе хэш-таблиц и связного списка. Реализация контейнера `std::unordered_set` стандартной библиотеки C++.

#### 4.3.1. Перечень семинарских, практических занятий и лабораторных работ

№ п/п	№ раздела и темы	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)		Оценочные средства	Формируемые компетенции (индикаторы)*
			Всего часов	Из них практическая подготовка		
1	2	3	4	5	6	7
1	1.1	Тема 1. Сравнение времени работы алгоритмов	3	3	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
1	1.2	Тема 2. Введение в теорию вычислительной сложности	3	3	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
2	2.1	Тема 1. Массивы и контейнеры с концевой вставкой	2	2	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
2	2.2	Тема 2. Способы организации памяти в контейнерах с произвольным доступом	2	2	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
2	2.3	Тема 3. Дек.	2	2	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
3	3.1	Тема 1. Двоичная куча	2	2	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)

3	3.2	Тема 2. Дерево отрезков	4	4	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
4	4.1	Тема 1. Связный список	2	2	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
4	4.2	Тема 2. Дерево поиска	2	2	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
4	4.3	Тема 3. AVL-дерево	3	3	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
4	4.4	Тема 4. Случайные деревья	3	3	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
5		<b>Хэш-таблицы</b>	4	4	отчет	ПК-2 (ИДК ПК2.1, ИДК ПК2.2, ИДК ПК2.3) ПК-3 (ИДК ПК3.1, ИДК ПК3.2, ИДК ПК3.3)
		<b>Всего</b>	32	32		

#### 4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение студентами в рамках самостоятельной работы (СР)

1. Прогнозирование времени работы программ.
2. Контейнеры с произвольным доступом.
3. Структуры данных на основе полного бинарного дерева
4. Ссылочные структуры данных
5. Хэш-таблицы

Полный перечень вопросов и заданий для самостоятельных работ доступен на странице курса в ИОС Домик.

#### **4.4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ**

Методические рекомендации по организации самостоятельной работы студентов доступны на странице курса в ИОС Домик.

#### **4.5. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)**

Не предусмотрено.

## **5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)**

### **а) основная литература**

1. Романенко, Т. А. Программные коллекции данных. Проектирование и реализация : учебное пособие для спо / Т. А. Романенко. — Санкт-Петербург : Лань, 2021. — 152 с. — ISBN 978-5-8114-8207-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/183217> (дата обращения: 21.04.2021). — Режим доступа: для авториз. пользователей.

### **б) дополнительная литература**

1. Тюкачев, Н. А. С#. Алгоритмы и структуры данных : учебное пособие для вузов / Н. А. Тюкачев, В. Г. Хлебостроев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2021. — 232 с. — ISBN 978-5-8114-8247-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/172708> (дата обращения: 21.04.2021). — Режим доступа: для авториз. пользователей.

2. Круценюк, К. Ю. Динамические структуры данных : учебное пособие / К. Ю. Круценюк. — Норильск : НГИИ, 2013. — 154 с. — ISBN 978-5-89009-552-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/155905> (дата обращения: 21.04.2021). — Режим доступа: для авториз. пользователей.

### **в) список авторских методических разработок:**

1. Полные материалы лекций, включая примеры программ и наглядно-иллюстративные материалы. Доступны на странице курса в ИОС Домик.

2. Указания по выполнению лабораторных работ и для самостоятельной работы. Доступны на странице курса в ИОС Домик.

3. Видеозаписи лекций. Доступны по ссылкам на странице курса в ИОС Домик.

### **г) базы данных, информационно-справочные и поисковые системы:**

1. ISO/IEC 14882:2020 Programming languages — C++ URL: <https://www.iso.org/standard/79358.html>

2. JDK 11 Documentation. URL: <https://docs.oracle.com/en/java/javase/11/>

3. Python 3.10.4 documentation. URL: <https://docs.python.org/3/>

## **6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **6.1. УЧЕБНО-ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ:**

1. Аудитория, оснащенная компьютерами и сетевым оборудованием.

### **6.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:**

1. Комплект разработчика приложений Java Platform (JDK) 11, Standard Edition (распространяется бесплатно).

2. Интегрированная среда разработки NetBeans IDE 12 (распространяется бесплатно, LGPLv2.1, GPLv2 with Classpatch exception).

3. Автоматическая проверяющая система Ejudge contest management system 3.7.9. (распространяется бесплатно).

4. Среда разработки Code::Blocks IDE 20.03 (распространяется бесплатно).

5. Язык программирования Python 3.9 (распространяется бесплатно).

### **6.3. ТЕХНИЧЕСКИЕ И ЭЛЕКТРОННЫЕ СРЕДСТВА:**

ИОС EDUCA, DOMIC, презентационное оборудование, персональный

компьютер с возможностью демонстрации презентаций в формате pdf.

## **7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ**

При реализации данного курса используются следующие образовательные технологии: технологии традиционного обучения, игровые технологии, технологии проблемного обучения, технологии обучения в сотрудничестве, технологии контекстного обучения, интерактивные технологии, технологии дистанционного обучения, активные педагогические технологии.

## **8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

### **8.1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ВХОДНОГО КОНТРОЛЯ**

Входной контроль обеспечивается выполнением учебного плана по предшествующим дисциплинам.

### **8.2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ**

Оценочные материалы представлены в виде задач по программированию, вопросов с развернутым ответом, заданий на понимание исходного кода. Все задания доступны на странице курса в ИОС Домик

### **8.3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ**

#### **Примерный перечень итоговых заданий для зачета с оценкой.**

1. Реализуйте хэш-таблицы с итерацией на основе однонаправленного списка.  
На основе дерева отрезков реализуйте дерево полиномиальных хэшей. Для заданной строки большой длины структура данных должна позволять изменять символы в произвольных позициях и сравнивать подстроки произвольной длины за  $O(\ln n)$ .
2. Реализуйте итератор произвольного доступа для двоичного дерева с неявными ключами. Кроме этого, реализуйте операторы присваивания и копирующие конструкторы с семантикой копирования и с семантикой перемещения.
3. Реализуйте методы `lower_bound(const T &x)` и `upper_bound(const T &x)` для AVL-дерева. Кроме этого, реализуйте операторы присваивания и копирующие конструкторы с семантикой копирования и с семантикой перемещения.
4. На основе класса `set` с реализацией в виде AVL-дерева разработайте класс `map`. Класс должен быть близок по возможностям и по интерфейсу к стандартному классу `map`.
5. Реализуйте класс приоритетной очереди `priority_queue` в соответствии со всеми методами и шаблонными параметрами, присутствующими в стандарте C++.
6. Разработайте внешнее расширение на языке C++ для Python, реализующее приоритетную очередь. Каждый элемент очереди будет парой из целого числа, задающего приоритет, и произвольного объекта Python. Очередь должна реализовывать методы `append()`, `pop()` и `top()`. Реализацию требуется выполнить без использования библиотечных классов и функций C++. Стандартный класс `vector` использовать можно. Для возврата значения методами `pop` и `top` следует использовать класс `tuple`.
7. Разработайте внешнее расширение на языке C++ для Python, реализующее дерево отрезков. Интерфейс можно разработать самостоятельно, однако он должен позволять решать все задачи, которые имеются в соответствующей лабораторной работе.
8. Разработайте внешнее расширение на языке C++ для Python, реализующее дек. В качестве реализации можно взять кольцевой дек из лабораторной работы. Дек должен позволять хранить произвольные объекты Python.
9. Разработайте внешнее расширение на языке C++ для Python, реализующее дерево с неявными ключами. В качестве значения дерево должно позволять сохранять произвольные объекты Python.

**Разработчики:**



(подпись)

доцент

(занимаемая должность)

Кириченко Константин Дмитриевич

(Ф.И.О.)

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 «Прикладная информатика» (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 г. № 922, зарегистрированный в Минюсте России «12» октября 2017 г. № 48531 с изменениями и дополнениями от 26.11.2020 г., 8.02.2021 г.

Программа рассмотрена на заседании кафедры Алгебраических и информационных систем ИМИТ ИГУ «29» апреля 2021 г.

Протокол № 11 Зав. кафедрой



Пантелеев В.И.

*Настоящая программа, не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.*