МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий
Кафедра информационных технологий

«УТВЕРЖДАЮ»
Директор ИМИТ ИГУ
_____ М. В. Фалалеев
«29» апреля 2026 г.

Рабочая программа дисциплины (модуля)

Б1.О.23 Структуры данных

Направление подготовки 02.03.02 Фундаментальная информатика и информационные технологии

Направленность (профиль) подготовки Фундаментальная информатика и программная инженерия

Квалификация выпускника бакалавр

Форма обучения очная

Иркутск 2026 г.

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 обязательной части программы и изучается на втором курсе.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы знания, умения и навыки, сформированные предшествующими дисциплинами: математический анализ, 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, теория алгоритмической 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
ОПК-3 Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей,	ИДК ОПК3.1 Знает основные языки программирования и типы баз данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий	Знает виды структур данных, их особенности и характеристики Умеет записывать программный код с использованием структур данных. Владеет навыком использования контейнерных типов в программировании

образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям		
	ИДК $_{ОПК3.2}$ Применяет языки программирования и современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, создания информационных ресурсов глобальных сетей, ведения баз данных и информационных хранилищ	Знает оценки сложности выполнения запросов к данным при различных способах их организации Умеет выбирать и анализировать способы организации данных в прикладных программах Владеет навыком использования ассоциативных контейнеров, множеств и хэш-таблиц
	ИДК $_{ОПК3.3}$ Способен выполнять задачи программирования, отладки и тестирования прототипов программных средств и информационных систем	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++
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	ИДК $_{УК2.1}$ Формулирует цели, задач, значимости, ожидаемых результатов проекта	Знает понятийный аппарат для формулировки целей и задач исследования Умеет обосновывать актуальность исследования, формулировать на её основе цели и задачи Владеет приемами декомпозиции цели на логичный и последовательный набор задач.
	ИДК $_{УК2.2}$ В рамках поставленных задач определяет имеющиеся ресурсы и ограничения, действующие правовые нормы	Знает правовые нормы в том числе в области антиплагиата Умеет определять ресурсы, требуемые для решения поставленной задачи

		Владеет навыком распределения ресурсов, в том числе времени, для решения поставленных задач
	ИДК $_{УК2.3}$ Разрабатывает план реализации проекта	Знает принципы планирования Умеет оценивать сроки реализации этапов решения задачи Владеет навыками планирования работ.
ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	ИДК $_{ОПК2.1}$ Понимает базовые принципы и устройство современных информационных технологий и программных средств	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 $_{ОПК2.2}$ Способен применять современное программное обеспечение (в том числе отечественного производства) для решения задач 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 $_{ОПК2.3}$ Способен применять суперкомпьютерные методы для решения задач профессиональной деятельности	Знает особенности реализации контейнерных типов в различных языках программирования 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 единиц, 108 часов, практическая подготовка 108.
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	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоятельная работа + контроль	
			Лекции	Семинарские (практические занятия)	Контроль обучения		
1	**Прогнозирование времени работы программ**	4	6	6	1	6	Проверка отчетов по лабораторным работам
	Тема 1. Сравнение времени работы алгоритмов		3	3			
	Тема 2. Введение в теорию вычислительной сложности		3	3			
2	**Контейнеры с произвольным доступом**	4	6	6	2	6	Проверка отчетов по лабораторным работам
	Тема 1. Массивы и контейнеры с концевой вставкой		2	2			
	Тема 2. Способы организации памяти в контейнерах с произвольным доступом		2	2			
	Тема 3. Дек.		2	2			

№ п/п	Раздел дисциплины/темы	Семестр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоятельная работа + контроль	
			Лекции	Семинарские (практические занятия)	Контроль обучения		
3	**Структуры данных на основе полного бинарного дерева**	4	6	6	2	8	Проверка отчетов по лабораторным работам
	Тема 1. Двоичная куча		2	2			
	Тема 2. Дерево отрезков		4	4			
4	**Ссылочные структуры данных**	4	10	10	2	12	Проверка отчетов по лабораторным работам
	Тема 1. Связный список		2	2			
	Тема 2. Дерево поиска		2	2			
	Тема 3. АВЛ-дерево		3	3			
	Тема 4. Случайные деревья		3	3			
5	**Хэш-таблицы**	4	4	4	1	4	Проверка отчетов по лабораторным работам
Итого часов 108			32	32	8	36	

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	УИЛП	1-3 недели	6	Отчеты по лабораторным работам	Материалы курса на платформе ИОС DOMIC
4	Контейнеры с произвольным доступом	УИЛП	4-6 недели	6	Отчеты по лабораторным работам	Материалы курса на платформе ИОС DOMIC
4	Структуры данных на основе полного бинарного дерева	УИЛП	7-9 недели	8	Отчеты по лабораторным работам	Материалы курса на платформе ИОС DOMIC
4	Ссылочные структуры данных	УИЛП	10-14 недели	12	Отчеты по лабораторным работам	Материалы курса на платформе ИОС DOMIC
4	Хэш-таблицы	УИЛП	15-16 недели	4	Отчеты по лабораторным работам	Материалы курса на платформе ИОС DOMIC
Общая трудоемкость самостоятельной работы по дисциплине (час)				36		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				**36**		

Виды самостоятельной работы: У – выполнение упражнений; И – информационный поиск; Л – изучение литературы; П – написание компьютерных программ.

4.3. Содержание учебного материала

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 элементам. 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 двусторонней 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 сортировка 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 вычислении 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 дерево. 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. АВЛ-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 АВЛ-дерева. Алгоритмы балансировки АВЛ-дерева. Реализация контейнера std::set стандартной библиотеки C++ на основе АВЛ-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 ключем 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/н	№ 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	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)

1	1.2	Тема 2. Введение в теорию вычислительной сложности	3	3	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
2	2.1	Тема 1. Массивы и контейнеры с концевой вставкой	2	2	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
2	2.2	Тема 2. Способы организации памяти в контейнерах с произвольным доступом	2	2	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
2	2.3	Тема 3. Дек.	2	2	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
3	3.1	Тема 1. Двоичная куча	2	2	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
3	3.2	Тема 2. Дерево отрезков	4	4	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$)

						ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
4	4.1	Тема 1. Связный список	2	2	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
4	4.2	Тема 2. Дерево поиска	2	2	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
4	4.3	Тема 3. АВЛ-дерево	3	3	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
4	4.4	Тема 4. Случайные деревья	3	3	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
5		**Хэш-таблицы**	4	4	Лабораторные работы	УК-2 (ИДК $_{УК2.1}$, ИДК $_{УК2.2}$, ИДК $_{УК2.3}$) ОПК-2 (ИДК $_{ОПК2.1}$) ОПК-3 (ИДК $_{ОПК3.1}$, ИДК $_{ОПК3.2}$, ИДК $_{ОПК3.3}$)
		Всего	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. Павлов, Л. А. Структуры и алгоритмы обработки данных / Л. А. Павлов, Н. В. Первова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 256 с. — ISBN 978-5-507-44105-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/207563 (дата обращения: 21.03.2022). — Режим доступа: для авториз. пользователей.

2. 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: 21.03.2022). — Режим доступа: для авториз. пользователей.

б) дополнительная литература

1. Тюкачев, Н. А. C#. 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: 21.03.2022). — 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: 21.03.2022). — 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.

Входной контроль не предусмотрен в виде отдельных мероприятий.

8.2. Оценочные материалы текущего контроля

Текущий контроль производится путем проверки отчетов по лабораторным работам. Имеется 14 лабораторных работ в каждой из которых от 5 до 12 заданий одного из двух видов: вопросы с развернутым ответом, задания на написание программ на указанном языке программирования (C++, Python). Все задания доступны на странице курса в ИОС DOMIC. Оценка за выполнение лабораторной работы выставляется пропорционально числу выполненных заданий.

В качестве примера далее приводятся задания лабораторной работы 3

1. *Ответьте на следующие вопросы.*
   - *Почему в классе std::array итераторы и ссылки на элементы всегда допустимы,а в классе std::vector - нет.*
   - *Почему массивы всегда хранят элементы одного типа?*
   - *Предположим, что в программе вы собираетесь использовать достаточно много динамических массивов, причем их размер в ходе работы программы может сильно изменяться как в сторону увеличения так и в сторону уменьшения. Стоит ли в этом случае использовать класс std::vector для работы? Обоснуйте ответ.*
2. *В работе по изучению времени сортировки на C++ замените vector на array и сравните время работы. Сделайте выводы.*
3. *В работе по изучению времени сортировки на Python замените list на ndarray и сравните время работы. Сделайте выводы. Сортировку следует выполнять методом numpy.sort(x).*
4. *Реализуйте на Python функцию спуска до нуля с использованием библиотеки numpy. Задана прямоугольная матрица из целых неотрицательных чисел. С этой матрицей выполняются следующие преобразования. В каждой строке находится минимальный элемент, после чего, этот элемент вычитается из всех элементов строки. В результате в каждой строке должен получиться хотя бы один ноль.Далее аналогичное преобразование выполняется со столбцами.При реализации требуется максимально использовать встроенные функции. В частности, нельзя использовать циклы, рекурсию и т.д. Иными словами, реализация должна заключаться только в*

вызовах встроенных функций. Подсказка. Надо использовать функцию min из класса ndarray с параметром axis, вычитание матриц и транспонирование.

5. *Это задание для языка C++. Создайте вектор, используя, аллокатор из примера в лекции. Проверьте, при каких условиях происходит реаллокация элементов и как определяется размер выделяемой области. Запишите алгоритм определения размера резервируемой памяти при выполнении метода resize*
6. *Разработайте шаблонный класс, близкий по возможностям к std::vector. В качестве итераторов в нем можно использовать указатели. Требуется реализовать методы resize, begin, end, push_back, pop_back, конструкторы перемещения и копирования, операторы присваивания с семантикой перемещения и копирования, оператор индексации. При реаллокации элементов не забывайте использовать функцию move.*

В архив поместите тексты программ на C++ и Python, файл с отчетом.

В качестве результата исполнения работы требуется предоставить отчет с ответами на вопросы и текстами программ. Программы прикрепляются в виде файлов с исходным кодом. Ответы на вопросы можно записывать в тетради или в электронном формате.

8.3. Оценочные материалы промежуточного контроля

Итоговая сумма первичных баллов складывается из баллов, полученных за выполнение лабораторных работ (14 работ с максимальной оценкой в 5 баллов) и баллов, полученных за выполнение итоговой (зачетной) работы (максимум 10 баллов).

Для достижения соответствия с бально-рейтинговой системой ИГУ первичные баллы пересчитываются в аттестационные баллы по формуле 3*max(x,20)+0.5*max(x-20,20), где x – первичные баллы за курс. Далее оценка выставляется в соответствии с бально-рейтинговой системой ИГУ (80 баллов – отлично, 70 баллов – хорошо, 60 баллов – удовлетворительно).

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 АВЛ-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 АВЛ-дерева разработайте класс map. Класс должен быть близок по возможностям и по интрефейсу 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. Интрефейс 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:

доцент Кириченко Константин Дмитриевич

(подпись) (занимаемая должность) (Ф.И.О.)

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии» (уровень бакалавриата), утвержденный приказом Министерства образования и науки Российской Федерации от 23 августа 2017 г. N 808, зарегистрированный в Минюсте России «14» сентября 2017 г. № 48185 с изменениями и дополнениями с изменениями и дополнениями от: 26 ноября 2020 г., 8 февраля 2021 г.