

9. Цели и задачи учебной дисциплины

Цели дисциплины:

– формирование у студентов профессиональных компетенций, связанных с использованием знаний в области структур данных и теории алгоритмов, пониманием концепции абстрактных типов данных и подходов к их реализации на языке C++ на основе принципов объектно-ориентированного построения программ, оценки влияния выбора структур данных и/или алгоритмов на эффективность программы.

Задачи изучения дисциплины:

- получение практических навыков решения задач профессиональной деятельности с использованием разных структур данных (линейных списков, стеков, очередей, деревьев, хэш-таблиц), используя концепции абстракции данных и модульного программирования;
- развитие умений, основанных на полученных теоретических знаниях, предлагать и применять эффективные подходы к решению (алгоритмизации) поставленных задач с использованием данных простой и сложной структуры;
- получение навыков самостоятельной работы, предполагающей изучение специфических особенностей работы со структурами данных в рамках разработки подходов (алгоритмов) к решению поставленной задачи, вопросов управления памятью в C++ и использования компонентов стандартной библиотеки шаблонов (STL).

10. Место учебной дисциплины в структуре ООП: Дисциплина «Языки и методы программирования» входит в обязательную часть Блока1.

Изучение данного курса должно базироваться на знании обучающимися материала курса «Информатика и программирование». Дисциплина является базовой для изучения курсов: «Численные методы», «Компьютерная графика», «Программирование на C#», «Программирование встроенных систем», «Алгоритмы биоинформатики».

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОПК-2	Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности	ОПК-2.1	Владеет основными положениями и концепциями в области программирования, архитектуры языков программирования, теории коммуникации. Знаком с основной терминологией, перечнем ПО, включенного в Единый Реестр российских программ.	Знать: базовые абстрактные типы (структуры) данных (контейнеры), понимать их особенности, применяемые операции и методы реализации на языке C++; контейнеры, итераторы и алгоритмы как основные компоненты STL, приемы хэширования и разрешения коллизий. Уметь: разрабатывать программы на C++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий. Владеть: навыками процедурного и объектно-ориентированного программирования на C++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.
		ОПК-2.2	Анализирует типовые языки программирования, составляет программы.	

		ОПК-2.3 задач	Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникации	
--	--	------------------	--	--

12. Объем дисциплины в зачетных единицах/час.(в соответствии с учебным планом) — 10/360.

Форма промежуточной аттестации(зачет/экзамен) зачет, экзамен

13. Трудоемкость по видам учебной работы

Вид учебной работы		Трудоемкость		
		Всего	По семестрам	
			3 семестр	4 семестр
Контактная работа		160	80	80
в том числе:	лекции	64	32	32
	практические	32	16	16
	лабораторные	64	32	32
	курсовая работа			
Самостоятельная работа		128	64	64
Промежуточная аттестация (для экзамена)		72	36	36
Итого:		360	180	180

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
1. Лекции			
1.1	Модульное программирование	Понятие модульного программирования. Интерфейс и реализация модуля. Модули в С++	Курс «Языки и методы программирования_ФИИТ» https://edu.vsu.ru/course/view.php?id=10368
1.2	Динамические структуры данных	Линейные однонаправленные и двунаправленные списки. Интерфейс обработки списков	
1.3	Стеки	Понятие стека. Методы работы со стеком	
1.4	Очереди	Понятие очереди. Методы работы с очередью	
1.5	Бинарные деревья	Понятие бинарного дерева. Сбалансированные деревья, деревья поиска. Рекурсивная и некурсивная обработки бинарного дерева	
1.6	Сильноветвящиеся деревья	Способы представления сильноветвящихся деревьев. В-деревья. Trie-деревья	
1.7	STL-контейнеры	STL-контейнеры. Способы обработки элементов контейнера	
1.8	Сортировки	Внутренние сортировки. Внешние сортировки.	
1.9	Хеширование	Хеширование как метод преобразование массива входных данных произвольной длины в (выходную) битовую строку установленной длины	
1.10	Алгоритмы с возвратом	Поиск одного решения, поиск всех решений, поиск оптимального решения	
1.11	Основы параллельного программирования	Понятия процесса и потока. Независимые потоки, способы возврата результатов. Зависимые потоки, способы работы с разделяемыми ресурсами	

2. Практические занятия		
2.1	Динамические структуры данных	Линейные однонаправленные и двунаправленные списки. Интерфейс обработки списков
2.2	Стеки	Методы работы со стеком
2.3	Очереди	Методы работы с очередью
2.4	Бинарные деревья	Сбалансированные деревья, деревья поиска (добавление и удаление элементов), дерево-формула
2.5	Сильноветвящиеся деревья	Trie-деревья. Алгоритмы добавления, удаления, поиска
2.6	STL-контейнеры	STL-контейнеры. Способы обработки элементов контейнера
2.7	Сортировки	Внутренние сортировки. Внешние сортировки.
2.8	Хеширование	Хэш-функция, ее свойства. Способы разрешения коллизии
2.9	Алгоритмы с возвратом	Поиск одного решения, поиск всех решений, поиск оптимального решения
2.10	Основы параллельного программирования	Асинхронные задачи, независимые потоки, способы возврата результатов. Зависимые потоки, Interlocked-функции, атомарные типы, сигнальные объекты ядра
3. Лабораторные работы		
3.1	Динамические структуры данных	Линейные однонаправленные и двунаправленные списки. Интерфейс обработки списков
3.2	Стеки	Методы работы со стеком
3.3	Очереди	Методы работы с очередью
3.4	Бинарные деревья	Сбалансированные деревья, деревья поиска, дерево-формула. Рекурсивная и нерекурсивная обработки деревьев
2.5	Сильноветвящиеся деревья	Trie-деревья. Алгоритмы добавления, удаления, поиска.
2.6	STL-контейнеры	STL-контейнеры. Способы обработки элементов контейнера
2.7	Сортировки	Внутренние сортировки. Внешние сортировки.
2.8	Хеширование	Хэш-функция, ее свойства. Способы разрешения коллизии
2.9	Алгоритмы с возвратом	Поиск одного решения, поиск всех решений, поиск оптимального решения
2.10	Основы параллельного программирования	Асинхронные задачи, независимые потоки, способы возврата результатов. Зависимые потоки, Interlocked-функции, атомарные типы, сигнальные объекты ядра

* заполняется, если отдельные разделы дисциплины изучаются с помощью онлайн-курса. В колонке Примечание необходимо указать название онлайн-курса или ЭУМК. В других случаях в ячейки ставятся прочерки.

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				Всего
		Лекции	Практические	Лабораторные	Самостоятельная работа	
1	Модульное программирование	2			4	6
2	Динамические структуры данных	8	4	8	16	36
3	Стеки	4	2	4	8	18
4	Очереди	4	2	4	8	18
5	Бинарные деревья	8	4	8	16	36
6	Сильноветвящиеся деревья	6	4	8	12	30
7	STL-контейнеры	4	2	4	8	18
8	Сортировки	6	4	6	12	28
9	Хеширование	4	2	4	8	18
10	Алгоритмы с возвратом	6	4	6	12	28
11	Основы параллельного	12	4	12	24	52

	программирования					
	Итого:	64	32	64	128	288

14. Методические указания для обучающихся по освоению дисциплины

Работа с конспектами лекций, выполнение практических и лабораторных заданий, заданий текущей и промежуточной аттестаций.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей по работе на LMS-платформе, своевременно подключаться к online-занятиям, соблюдать рекомендации по организации самостоятельной работы

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	Павлов, Л. А. Структуры и алгоритмы обработки данных : учебник для вузов / Л. А. Павлов, Н. В. Первова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 256 с. — ISBN 978-5-8114-7259-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/156929
2	Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/119661
3	Свердлов, С. З. Языки программирования и методы трансляции : учебное пособие / С. З. Свердлов. — 2-е изд., испр. — Санкт-Петербург : Лань, 2019. — 564 с. — ISBN 978-5-8114-3457-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/116391

б) дополнительная литература:

№ п/п	Источник
4	Тюкачев, Н. А. С#. Алгоритмы и структуры данных : учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2018. — 232 с. — ISBN 978-5-8114-2566-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/104961 .
5	Бабушкина И.А. Практикум по объектно-ориентированному программированию / И.А. Бабушкина, С.М. Окулов. – М.: "Лаборатория знаний", 2012. – 366 с. Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=8781
6	Липпман С.Б. Язык программирования С++. Базовый курс / С.Б. Липпман., Ж. Лажойе, Б.Э. Му. – М. : Вильямс, 2014. – 1120 с.
7	Пахомов Б.И. С/С++ и MS Visual С++ 2010 для начинающих / Б.И. Пахомов. – СПб. ; БХВ-Петербург, 2012. – 736 с.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
8	www.lib.vsu.ru – ЗНБ ВГУ
9	Окулов С.М. Программирование в алгоритмах. – М. : "Лаборатория знаний", 2014. – 384 с. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=50562
10	Онлайн-курс «Языки и методы программирования_ФИИТ». – https://edu.vsu.ru/course/view.php?id=10368

* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы

16. Перечень учебно-методического обеспечения для самостоятельной работы (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)

№ п/п	Источник
1	Ускова, О. Ф. Информатика и программирование. Задачник-практикум по

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

При реализации дисциплины используются модульно-рейтинговая и личностно-ориентированные технологии обучения (ориентированные на индивидуальность студента, компьютерные и коммуникационные технологии). В рамках дисциплины предусмотрены следующие виды лекций: информационная, лекция-визуализация, лекция с применением обратной связи.

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий, для организации самостоятельной работы обучающихся используется онлайн-курс, размещенный на платформе Электронного университета ВГУ (LMS moodle), а также другие Интернет-ресурсы, приведенные в п.15в

18. Материально-техническое обеспечение дисциплины:

Лекционная аудитория должна быть оборудована учебной мебелью, компьютером, мультимедийным оборудованием (проектор, экран, средства звуковоспроизведения), допускается переносное оборудование.

Практические занятия должны проводиться в специализированной аудитории, оснащенной учебной мебелью и персональными компьютерами с доступом в сеть Интернет (компьютерные классы, студии), мультимедийным оборудованием (мультимедийный проектор, экран, средства звуковоспроизведения), Число рабочих мест в аудитории должно быть таким, чтобы обеспечивалась индивидуальная работа студента на отдельном персональном компьютере.

Для самостоятельной работы необходимы компьютерные классы, помещения, оснащенные компьютерами с доступом к сети Интернет.

Программное обеспечение:

- ОС Windows
- MS Office
- Microsoft Visual Studio Community Edition (свободное и/или бесплатное ПО)
- Adobe Reader (свободное и/или бесплатное ПО)

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Модуль 1. Программирование	ОПК-2	ОПК-2.1	КИМы (для проведения текущей аттестации) Задания для контрольных работ Задания для лабораторных работ
2	Динамические структуры данных	ОПК-2	ОПК-2.2, ОПК-2.3	
3	Стеки	ОПК-2	ОПК-2.2, ОПК-2.3	
4	Очереди	ОПК-2	ОПК-2.2, ОПК-2.3	
5	Бинарные деревья	ОПК-2	ОПК-2.2, ОПК-2.3	
6	Сбалансированные деревья	ОПК-2	ОПК-2.2, ОПК-2.3	

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
7	STL-контейнеры	ОПК-2	ОПК-2.2, ОПК-2.3	
8	Сортировки	ОПК-2	ОПК-2.2, ОПК-2.3	
9	Хеширование	ОПК-2	ОПК-2.2, ОПК-2.3	
10	Алгоритмы с возвратом	ОПК-2	ОПК-2.2, ОПК-2.3	
11	Основы параллельного программирования	ОПК-2	ОПК-2.2, ОПК-2.3	
Промежуточная аттестация форма контроля – экзамен				КИМы для проведения промежуточной аттестации

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Контрольных работ

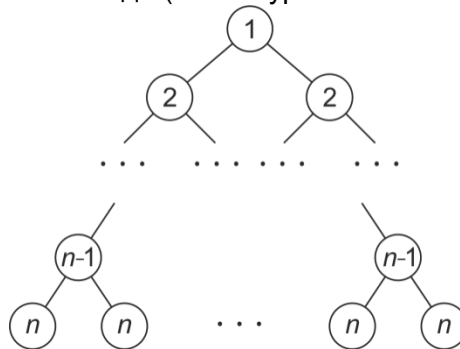
Примеры заданий по теме «Линейные списки»

Дан линейный однонаправленный список из различных целых чисел.

1. Перед первым двузначным числом вставить его квадрат.
2. Удалить все элементы кратные 7.
3. Переставить максимальный элемент в голову списка, переключая указатели.
4. Упорядочить список по возрастанию методом простых вставок.
5. Реализовать рекурсивную функцию для построения копии заданного списка.

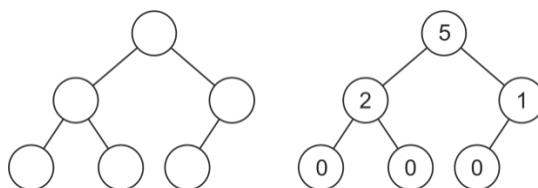
Примеры заданий по теме «Бинарные деревья»

1. Построить дерево заданного вида (n – натуральное число):



2. Найти сумму элементов на n -м уровне.
3. Найти глубину дерева.
4. Посчитать количество листьев, имеющих четное значение. Использовать нерекурсивную функцию.

5. Построить копию дерева, в узлах которой записать количество потомков у соответствующего узла в исходном дереве. Пример исходного дерева и его копии представлен на рисунке.



Лабораторных работ

Примеры заданий по теме «STL-контейнеры»



Разработать контейнерный класс «Библиотека», в котором список изданий упорядочен по возрастанию значений поля Издательство. Найти издательство, которое чаще всего публиковало журналы в заданном году.

Требования к выполнению заданий (или шкалы и критерии оценивания)

Задание считается выполненным, если разработанный программный продукт полностью решает поставленную задачу, решение предусматривает обработку исключительных ситуаций, обладает удобным пользовательским интерфейсом, соответствует требованиям оформления кода (структурированность, наличие комментариев).

Требования к выполнению заданий, шкалы и критерии оценивания

«Отлично»

Сформированные знания базовых абстрактных типов (структуры) данных (контейнеры), понимание их особенностей, применяемых операции и методы реализации на языке C++.

Сформированные умения разрабатывать программы на C++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.

Сформированные навыки процедурного и объектно-ориентированного программирования на C++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.

«Хорошо»

Сформированные, но содержащие отдельные пробелы, знания базовых абстрактных типов (структуры) данных (контейнеры), понимание их особенностей, применяемых операции и методы реализации на языке C++.

Сформированные, но содержащие отдельные пробелы, умения разрабатывать программы на C++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.

Сформированные, но содержащие отдельные пробелы, навыки процедурного и объектно-ориентированного программирования на C++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.

«Удовлетворительно»

Неполное представление о базовых абстрактных типах (структуры) данных (контейнеры), понимании их особенностей, применяемых операциях и методах реализации на языке C++.

Успешное, но не системное умение разрабатывать программы на C++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.

Неполное владение навыками процедурного и объектно-ориентированного программирования на C++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.

«Неудовлетворительно»

Фрагментарные знания или отсутствие знаний. Фрагментарные умения или отсутствие умений. Фрагментарные навыки или отсутствие навыков