

I. Аннотация

1. Цель и задачи дисциплины:

Выработать у обучающихся практические навыки по проектированию алгоритмов и их программной реализации на языках высокого уровня (C/C++).

2. Место дисциплины в структуре ООП

Дисциплина входит в раздел «Информатика и коммуникационно-информационные технологии» обязательной части блока 1.

Предварительные знания и навыки. знание школьных курсов математики и информатики

Дальнейшее использование. Полученные знания используются в последующем при изучении предметов: «Технология программирования», «Базы данных». Знания, умения и навыки полученные при изучении дисциплины закрепляются в конце каждого семестра учебной практикой.

3. Объем дисциплины: 5зач.ед., 180акад. ч., в том числе:

контактная аудиторная работа лекций 0 ч., практических занятий 47 ч., лабораторных занятий 62 ч.;

контактная внеаудиторная работа контроль самостоятельной работы 0 ч., в том числе курсовая работа 0ч.; **самостоятельная работа** 71 ч., в том числе контроль 0 ч.

4. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы:

	Планируемые результаты обучения по дисциплине
Планируемые результаты освоения образовательной программы (формируемые компетенции)	
ОПК-2, Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.1, Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности ОПК-2.2, Выбирает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности

<p>Планируемые результаты освоения образовательной программы (формируемые компетенции)</p>	<p>Планируемые результаты обучения по дисциплине</p> <p>ОПК-2.3, Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности</p>
<p>ОПК-7, Способен разрабатывать алгоритмы и программы, пригодные для практического применения</p>	<p>ОПК-7.1 Демонстрирует знание основных языков программирования и работы с базами данных, операционных систем и оболочек, современных программных сред разработки информационных систем и технологий</p> <p>ОПК-7.2 Применяет языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ</p> <p>ОПК-7.3 Демонстрирует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>
<p>ОПК-9, Способен принимать участие в реализации профессиональных коммуникаций заинтересованными участниками проектной деятельности и в рамках проектных групп</p>	<p>ОПК-9.1 Демонстрирует знание инструментов и методов коммуникаций в проектах; каналов коммуникаций в проектах; моделей коммуникаций в проектах; технологий межличностной и групповой коммуникации в деловом взаимодействии, основ конфликтологии, технологии подготовки и проведения презентаций</p> <p>ОПК-9.2 Осуществляет взаимодействие с заказчиком в процессе реализации проекта; принимает участие в командообразовании и развитии персонала</p> <p>ОПК-9.3 Демонстрирует навыки проведения презентаций, переговоров, публичных выступлений</p>

5. Форма промежуточной аттестации и семестр прохождения:

зачет в 1-2 семестрах

6. Язык преподавания:

русский

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

Для студентов очной формы обучения

Учебная программа — наименование разделов и тем	Всего	Лк	Пр	Лб	КСР	Сам
Синтаксис, базовые типы переменных, операции, операторы, стандарт языка С	4	0	2	2	0	0
Системы счисления. Поразрядные операции.	15	0	6	6	0	3
Функции в С	27	0	6	10	0	11
Указатели и ссылки	26	0	6	10	0	10
Массивы и строки в С	22	0	6	8	0	8
Файловый ввод/вывод.	15	0	2	2	0	11
Работа с текстом	22	0	6	8	0	8
Сортировка и поиск	26	0	6	8	0	12
Динамические структуры данных	27	0	9	8	0	10
Итого	180	0	47	62	0	71

Учебная программа дисциплины

1. Синтаксис, базовые типы переменных, операции, операторы, стандарт языка С
 - Переменные и константы.
 - Функции. Форматный ввод/вывод на консоль.
 - Операции. Оператор присваивания.
 - Операторы. Условные операторы. Операторы выбора. Операторы цикла.
2. Системы счисления. Поразрядные операции.
 - Двоичная, восьмеричная, шестнадцатеричная система счисления.
 - Поразрядные операции. Двоичный дополнительный код.
 - Преобразование типов при присваивании. Переполнение целых.
3. Функции в С
 - Функции, возвращаемые значения, параметры и аргументы.
 - Объявление и определение функций. Локальные и глобальные переменные.
 - Дополнительные сведения о функциях. Рекурсия. Стек и рекурсия.
4. Указатели и ссылки
 - Указатель - средство хранения адреса. Имена указателей. Оператор разыменовывания. Использование адреса, хранящегося в указателе.

- Память стековая и динамически распределяемая. Утечка памяти.
 - Вычисления с указателями. Передача аргументов функций как указателей.
 - Ссылки в C++
5. Массивы и строки в C
- Одномерные массивы. Инициализация массива. Индексирование.
 - Массивы и указатели. Передача одномерного массива в функцию.
 - Двоичный поиск в отсортированном массиве.
 - Строки. Библиотечные функции для работы со строками.
 - Многомерные массивы. Передача многомерного массива в функцию.
 - Массивы указателей. Динамические массивы.
6. Файловый ввод/вывод.
- Потоки и файлы. Основы файловой системы.
 - Указатель файла, открытие файла, закрытие файла, запись символа, чтение символа и т.д.
 - Буферный ввод/вывод. Функции fread() и fwrite().
 - Ввод/вывод при прямом доступе. Функция fseek().
 - Структуры и бинарные файлы.
7. Работа с текстом
- Методы кодирования символов. Кодировки ASCII, ANSI, OEM, UTF, Unicode
 - Указатели, действия с указателями. Работа с массивами при помощи указателей
 - Простейшие алгоритмы: определение длины, копирование, конкатенация
 - Поиск и замена текста. Работа с различными элементами текста: символами, словами, строками
8. Сортировка и поиск
- Задача хранения информации. Линейный поиск. максимальные и средние оценки времени для линейного поиска
 - Двоичный поиск и его разновидности. Оценки времени для двоичного поиска. Поиск методом Фибоначчи, золотого сечения. Интерполяционный поиск и его обобщения
 - Задача сортировки. Сортировка пузырьком. Временные оценки пузырьковой сортировки
 - Сортировка при помощи поиска максимального элемента. Временные оценки
 - Сортировка вставками. Оценки времени.

- Сортировка почти упорядоченных массивов. Модификации методов.
- Сортировки Шелла. Задача выбора последовательности шагов. Оценки для различных последовательностей
- Слияние упорядоченных массивов. Модификации алгоритма для нахождения пересечения, объединения и разности множеств
- Сортировка слиянием. Временные оценки
- Быстрая сортировка Хоара. Оценки времени

9. Динамические структуры данных

- Операции динамического распределения памяти. Временные оценки операций
- Динамические структуры данных
- Линейные списки. Односвязные и двусвязные списки
- Основные операции со списками: добавление и удаление элемента
- Деревья. Двоичные деревья поиска
- Операции над двоичными деревьями поиска: вставка и удаление элемента, поиск
- Сбалансированные деревья. Методы балансировки
- Хеш-таблицы. Основная концепция, выбор хеш-функции
- Методы разрешения коллизий: метод цепочек, линейного опробования, двойного хеширования

III. Образовательные технологии

Учебная программа — наименование разделов и тем	Вид занятия	Образовательные технологии
Синтаксис, базовые типы переменных, операции, операторы, стандарт языка C	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Системы счисления. Поразрядные операции.	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Функции в C	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Указатели и ссылки	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Массивы и строки в C	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Файловый ввод/вывод.	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Работа с текстом	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ

Учебная программа — наименование разделов и тем	Вид занятия	Образовательные технологии
Сортировка и поиск	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ
Динамические структуры данных	практические занятия, лабораторные занятия	Работа с программным обеспечением, разработка программ

IV. Оценочные материалы для проведения текущей и промежуточной аттестации

Типовые контрольные задания и/или критерии для проверки индикаторов ОПК-2.1, ОПК-2.2, ОПК-2.3

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Уметь писать программы, используя синтаксис, базовые типы переменных, операции, операторы соответствующие стандарту языка C	<p>Примеры задач для контрольных работ:</p> <p>написать функцию, которая подсчитывает количество единиц в двоичной записи целого числа, используя битовые операции.</p> <p>написать функцию, которая находит элемент линейного массива, наиболее приближенный к среднему арифметическому элементов массива.</p> <p>написать функцию, которая вычисляет в одномерном массиве, состоящем из n вещественных элементов, сумму элементов массива, расположенных между первым и последним нулевыми элементами.</p> <p>написать программу, которая находит количество простых чисел на интервале $[x, y)$.</p> <p>написать функцию, получающую в качестве аргумента имя файла, содержащего последовательность вещественных чисел неизвестной длины, и возвращающую целое число, равное количеству максимальных элементов этой последовательности. Функция должна возвращать -1, -2 и т.д., если она не смогла открыть файл, прочитать элемент и т.д.. Основная программа должна вызывать эту функцию и выводить на экран результат ее работы.</p>	зачтено — умеет писать функции, выполнять задания по чтению записи текстового файла.
Уметь создавать массивы и строки.	<p>Примеры задач для контрольных работ:</p> <p>написать функцию, которая находит элемент линейного массива, наиболее приближенный к среднему арифметическому элементов массива.</p> <p>написать функцию, которая вычисляет в одномерном массиве, состоящем из n вещественных элементов, сумму элементов массива, расположенных между первым и последним нулевыми элементами.</p> <p>написать функцию, которая переворачивает строку. Например: "главрыба" — "абырвалг".</p> <p>функция «обрезает» строку на первом символе, совпадающем с данным символом.</p>	зачтено — умеет создавать массивы и строки, может передать массив и строку в функцию.
Уметь самостоятельно использовать библиотеку	<p>Примеры задач для контрольных работ:</p> <p>написать программу, которая считывает текст из файла и</p>	зачтено — умеет пользоваться основ-

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>лиотечные функции для работы со строками.</p>	<p>выводит на экран только строки, содержащие двузначные числа.</p> <p>аписать программу, которая считывает английский текст из файла и выводит на экран слова, начинающиеся с гласных букв.</p> <p>аписать программу, которая считывает текст из файла и выводит его на экран, меняя местами каждые два соседних слова.</p>	<p>ными функциями из библиотеки string.h</p>
<p>Уметь работать со стеклом, знать методы кодирования символов, уметь работать с кодировками ASCII, ANSI, OEM, UTF, Unicode</p>	<p>Примеры задач для контрольных работ:</p> <p>аписать функцию void f(char * str), которая в строке str убирает все лишние пробелы между словами, а также в начале и конце строки. Например, из строки « Quando tu sei vicina » должно получиться «Quando tu sei vicina». Вычислить максимальное время работы функции в зависимости от длины входных данных.</p> <p>аписать функцию, которая считывает текст из файла и выводит на экран только предложения, не содержащие запятых.</p>	<p>зачтено — умеет работать с текстом в кодировке ASCII, может работать с кодировкой Unicod.</p>
<p>Знать и уметь применять алгоритмы сортировки и поиска, а именно сортировки вставками, сортировки пузырьком, сортировки Шелла. Знать основные алгоритмы поиска в отсортированном массиве</p>	<p>Примеры задач для контрольных работ:</p> <p>аписать программу, сортирующую методом вставок главную диагональ двумерного массива целых чисел. Массив считать из текстового файла, содержащего в первой строке кол-во строк и кол-во столбцов, а далее в каждой строке целые числа, соответствующие элементам строки массива.</p> <p>аписать программу, сортирующую методом вставок двумерный массив целых чисел. При этом самое маленькое число должно оказаться в первом элементе первой строки, а самое большое в последнем элементе последней строки.</p> <p>аписать программу, ищущую методом половинного деления строку с заданным значением 1-го элемента строки в двумерном массиве целых чисел. Выполнить все необходимые для этого операции. Не использовать стандартные функции.</p>	<p>зачтено — умеет применять алгоритмы сортировки и поиска.</p>
<p>Уметь применять алгоритмы сортировки пузырьком, вставками, сортировку Шелла, созданные в командной разработке, для упорядочения массивов и списков.</p>	<p>Примеры задач для самостоятельных работ:</p> <p>тсортировать методом вставок список, содержащий целые числа.</p> <p>тсортировать сортировкой Шелла, созданной в группе, список, содержащий целые числа.</p> <p>Примеры задач для самостоятельных работ (задание для групповой работы):</p> <p>дать и протестировать контейнер для хранения данных о студенческой группе. Предусмотреть возможность быстрого ска информации о студенте группы :</p> <p>структурировать информацию о группе и о студентах группы (выполняет руководитель группы)</p> <p>ыбрать способ хранения данных (выполняет руководитель группы)</p> <p>ыбрать способ написания сортировки (с функциями сравнения или без)(выполняет руководитель группы)</p> <p>аспределить среди участников группы задания, которые необходимо выполнить для решения поставленной задачи (выполняет руководитель группы)</p>	<p>зачтено — умеет применять сортировки для упорядочения массива целых чисел, умеет применять шаблонные сортировки для упорядочивания массива, умеет создавать программы в группе разработки.</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	<ul style="list-style-type: none"> • Создать программу для хранения информации о студентах(выполняет участник группы) • Создать программу, которая заполняет контейнер данными из текстового файла (выполняет участник группы) • Создать функции, реализующие сортировку пузырьком, вставками, сортировку Шелла (выполняют участники группы) • Создать тесты, которые проверяют правильность работы программы (выполняет участник группы) • Выполнить сравнение эффективности алгоритмов сортировки (выполняет участник группы) • Создать функции, реализующие поиск информации в массиве данных о студентах (по фамилии студента) • Создать функции, реализующие поиск информации в массиве данных о студентах (по фамилии дате рождения студента) • Выполнить тестирование контейнера для хранения данных о студенческой группе (выполняют все участники группы) 	

Типовые контрольные задания и/или критерии для проверки индикаторов ОПК-7.1, ОПК-7.2

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>Уметь применять алгоритмы сортировки пузырьком, вставками, сортировку Шелла, созданные в командной разработке, для упорядочения массивов и списков.</p>	<p>Примеры задач для самостоятельных работ:</p> <ul style="list-style-type: none"> • Отсортировать методом вставок список, содержащий целые числа. • Отсортировать сортировкой Шелла, созданной в группе, список, содержащий целые числа. <p>Примеры задач для самостоятельных работ (задание для групповой работы):</p> <p>Создать и протестировать контейнер для хранения данных о студенческой группе. Предусмотреть возможность быстрого поиска информации о студенте группы :</p> <ul style="list-style-type: none"> • Структурировать информацию о группе и о студентах группы (выполняет руководитель группы) • Выбрать способ хранения данных (выполняет руководитель группы) • Выбрать способ написания сортировки (с функциями сравнения или без)(выполняет руководитель группы) • Распределить среди участников группы задания, которые необходимо выполнить для решения поставленной задачи (выполняет руководитель группы) • Создать программу для хранения информации о студентах(выполняет участник группы) • Создать программу, которая заполняет контейнер данными из текстового файла (выполняет участник группы) 	<p>зачтено — умеет применять сортировки для упорядочения массива целых чисел, умеет применять шаблонные сортировки для упорядочивания массива, умеет создавать программы в группе разработки.</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	<p>создать функции, реализующие сортировку пузырьком, вставками, сортировку Шелла (выполняют участники группы)</p> <p>создать тесты, которые проверяют правильность работы программы (выполняет участник группы)</p> <p>выполнить сравнение эффективности алгоритмов сортировки (выполняет участник группы)</p> <p>создать функции, реализующие поиск информации в массиве данных о студентах (по фамилии студента)</p> <p>создать функции, реализующие поиск информации в массиве данных о студентах (по фамилии дате рождения студента)</p> <p>выполнить тестирование контейнера для хранения данных о студенческой группе (выполняют все участники группы)</p>	
<p>Знать динамические структуры данных, а именно линейные списки, односвязные и двусвязные списки. Уметь использовать динамические данные, созданные в группе</p>	<p>вопросы вопросов к зачету:</p> <p>линейные списки. Основные операции со списками: добавление и удаление элемента</p> <p>двусвязные списки. Основные операции со списками: добавление и удаление элемента</p> <p>цепление и расцепление списков.</p> <p>ольцевые списки</p> <p>двумерные списки. Представление матриц двумерными списками.</p> <p>Примеры задач для самостоятельных работ (задание для групповой работы):</p> <p>дать и протестировать контейнер для хранения данных о студенческой группе:</p> <p>структурировать информацию о группе и о студентах группы (выполняет руководитель группы)</p> <p>выбрать способ хранения данных (выполняет руководитель группы)</p> <p>распределить среди участников группы задания, которые необходимо выполнить для решения поставленной задачи (выполняет руководитель группы)</p> <p>создать шаблонный список, структуру которого определил руководитель (выполняет участник группы)</p> <p>создать программу, которая заполняет контейнер данными из текстового файла (выполняет участник группы)</p> <p>создать тесты, которые проверяют правильность работы программы (выполняет участник группы)</p> <p>выполнить тестирование контейнера для хранения данных о студенческой группе (выполняют все участники группы)</p>	<p>зачтено — знает что такое динамические структуры данных, может создать линейный список, умеет создавать программы в группе разработки.</p>
<p>Знать как создать двоичное дерево поиска</p>	<p>вопросы вопросов к зачету:</p> <p>операции над двоичными деревьями поиска: вставка и удаление элемента, поиск</p> <p>балансированные деревья. Методы балансировки</p>	<p>оценка 3 — умеет создать двоичное дерево поиска, оценка 4 — умеет выполнять операции над двоичными деревьями поиска, оценка 5 — умеет</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
		создавать сбалансированные деревья поиска.
Знать понятие хеш-таблица, основную концепцию, как выбрать хеш-функцию	Примеры вопросов к зачету: <ul style="list-style-type: none"> • Хеш-таблицы. Основная концепция, выбор хеш-функции • Методы разрешения коллизий: метод цепочек, линейного опробования, двойного хеширования. 	зачтено — может осуществить поиск данных в массиве с помощью хеш-таблицы.

Типовые контрольные задания и/или критерии для проверки индикатора ОПК-7.3

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Уметь тестировать алгоритмы сортировки и поиска, созданные в группе разработки	Примеры задач для самостоятельных работ: <ul style="list-style-type: none"> • протестировать алгоритм сортировки, созданный в группе, методом вставок список, содержащий целые числа, и массив целых чисел того же размера. Оценить время работы алгоритма. • протестировать алгоритм сортировки методом Шелла список, содержащий целые числа, и массив целых чисел того же размера. Оценить время работы алгоритма. 	зачтено — может протестировать алгоритм сортировки.
Уметь тестировать динамические структуры данных	Примеры вопросов к зачету: <ul style="list-style-type: none"> • протестировать алгоритм создания линейного списка, содержащего целые числа. Оценить время поиска элемента, время вставки элемента. • протестировать алгоритм создания двоичного дерева поиска целых чисел. Оценить время поиска элемента, время вставки элемента. 	зачтено — может протестировать алгоритмы работы с динамическими структурами данных.
Уметь тестировать двоичное дерево поиска по заданному сценарию	Примеры задач для самостоятельных работ: <ul style="list-style-type: none"> • протестировать алгоритм создания двоичного дерева поиска целых чисел. Применить двоичное дерево для сортировки набора целых чисел. Оценить время сортировки в сравнении с сортировкой вставками. Сформировать сценарий тестирования. • протестировать алгоритм создания двоичного дерева поиска целых чисел. Оценить время поиска элемента в сравнении с алгоритмом поиска в отсортированном массиве. Сформировать сценарий тестирования. 	зачтено — умеет сформировать сценарий тестирования..

Типовые контрольные задания и/или критерии для проверки индикаторов ОПК-9.1, ОПК-9.2, ОПК-9.3

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Уметь применять алгоритмы сортировки пузырьком,	Примеры задач для самостоятельных работ: <ul style="list-style-type: none"> • Отсортировать методом вставок список, содержащий целые 	зачтено — умеет применять сортировки для упорядочения

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>вставками, сортировку Шелла, созданные в командной разработке, для упорядочения массивов и списков.</p>	<p>числа.</p> <p>тсортировать сортировкой Шелла, созданной в группе, список, содержащий целые числа.</p> <p>Примеры задач для самостоятельных работ (задание для групповой работы):</p> <p>Создать и протестировать контейнер для хранения данных о студенческой группе. Предусмотреть возможность быстрого поиска информации о студенте группы :</p> <p>структурировать информацию о группе и о студенах группы (выполняет руководитель группы)</p> <p>ыбрать способ хранения данных (выполняет руководитель группы)</p> <p>ыбрать способ написания сортировки (с функциями сравнения или без)(выполняет руководитель группы)</p> <p>аспределить среди участников группы задания, которые необходимо выполнить для решения поставленной задачи (выполняет руководитель группы)</p> <p>оздать программу для хранения инфрмации о студен- тах(выполняет участник группы)</p> <p>оздать программу, которая заполняет контейнер данными из текстового файла (выполняет участник группы)</p> <p>оздать функции, реализующие сортирови пузырьком, вставками, сортировку Шелла (выполняют участники группы)</p> <p>оздать тесты, которые проверяют правильность работы программы (выполняет участник группы)</p> <p>ыполнить сравниние эффективности алгоритмов сортировки (выполняет участник группы)</p> <p>оздать функции, реализующие поиск информации в массиве данных о студентах (по фамилии студента)</p> <p>оздать функции, реализующие поиск информации в массиве данных о студентах (по фамилии дате рождения студента)</p> <p>ыполнить тестирование контейнера для хранения данных о студенческой группе (выолняют все участники группы)</p>	<p>массива целых чисел, умеет применять шаблонные сортировки для упорядочивания массива, умеет создавать программы в группе разработки.</p>
<p>Знать динамические структуры данных,а именно линейные списки, односвязные и двусвязные списки. Уметь использовать динамические данные, созданные в группе</p>	<p>Примеры вопросов к зачету:</p> <p>инейные списки.Основные операции со списками: добавление и удаление элемента</p> <p>вусвязные списки.Основные операции со списками: добавление и удаление элемента</p> <ul style="list-style-type: none"> • Сцепление и расцепление списков. • Кольцевые списки <p>вумерные списки. Представление матриц двумерными списками.</p> <p>Примеры задач для самостоятельных работ (задание для групповой работы):</p> <p>Создать и протестировать контейнер для хранения данных о студенческой группе:</p>	<p>зачтено— знает что такое динамические структуры данных, может создать линейный список, умеет создавать программы в группе разработки.</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	<p>структурировать информацию о группе и о студенах группы (выполняет руководитель группы)</p> <p>выбрать способ хранения данных (выполняет руководитель группы)</p> <p>распределить среди участников группы задания, которые необходимо выполнить для решения поставленной задачи (выполняет руководитель группы)</p> <p>создать шаблонный список, структуру которого определил руководитель (выполняет участник группы)</p> <p>создать программу, которая заполняет контейнер данными из текстового файла (выполняет участник группы)</p> <p>создать тесты, которые проверяют правильность работы программы (выполняет участник группы)</p> <p>выполнить тестирование контейнера для хранения данных о студенческой группе (выполняют все участники группы)</p>	
Уметь тестировать алгоритмы сортировки и поиска, созданные в группе разработки	<p>Примеры задач для самостоятельных работ:</p> <p>тестировать алгоритм сортировки, созданный в группе, методом вставок список, содержащий целые числа, и массив целых чисел того же размера. Оценить время работы алгоритма.</p> <p>тестировать алгоритм сортировки методом Шелла список, содержащий целые числа, и массив целых чисел того же размера. Оценить время работы алгоритма.</p>	зачтено — может протестировать алгоритм сортировки.

V. Учебно-методическое и информационное обеспечение дисциплины

1. Рекомендованная литература а)

Основная литература

- [1] Немцова Т. И. Программирование на языке высокого уровня. Програм- мир. на языке C++: Уч. пос. / Т.И.Немцова и др.; Под ред. Л.Г.Гагариной - М.: ИД ФОРУМ: ИНФРА-М, 2012. - 512 с.: ил.; 60x90 1/16 + CD- ROM. — (Проф. обр.). (п) ISBN 978-5-8199-0492-3 — Режим доступа: <http://www.znanium.com/bookread.php?book=244875> (ЭБС znanium.com)
- [2] Вирт Н. Построение компиляторов [Электронный ресурс] / Никлаус Вирт; пер. с англ. Е. В. Борисов, Л. Н. Чернышов. — М.: ДМК Пресс, 2010. — 192 с.: ил. — Режим доступа: <http://www.znanium.com/bookread.php?book=408433> (ЭБС ИНФРА-М)
- [3] Окулов, С.М. Динамическое программирование [Электронный ресурс] : учеб. пособие / С.М. Окулов, О.А. Пестов. — Электрон. дан. — Москва : Издательство "Лаборатория знаний 2015. — 299 с. — Режим доступа: <https://e.lanbook.com/book/66114>. — Загл. с экрана.

б) Дополнительная литература

- [4] Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона + CD [Электронный ресурс] / Никлаус Вирт; пер. с англ. Ф. В. Ткачев. - М.: ДМК Пресс, 2010. - 272 с.: ил. — Режим доступа: <http://www.znanium.com/bookread.php?book=408420> (ЭБС znanium.com)

2. Программное обеспечение

Наименование помещений	Программное обеспечение
Ауд. 4б (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Adobe Acrobat Reader DC — бесплатно; Apache Tomcat 8.0.27 — бесплатно; Cadence SPB/OrCAD 16.6 — Государственный контракт на поставку лицензионных программных продуктов 103 — ГК/09 от 15.06.2009; GlassFish Server Open Source Edition 4.5.3 — бесплатно; Google Chrome — бесплатно; Java SE Development Kit 8 Update 45 (64-bit) — бесплатно; JetBrains PyCharm Community Edition 4.5.3 — бесплатно; JetBrains PyCharm Edu 3.0 — бесплатно; Kaspersky Endpoint Security 10 для Windows — Акт на передачу прав №2129 от 25 октября 2016 г.; Lazarus 1.4.0 — бесплатно; Mathcad 15 M010 — Акт предоставления прав ИС00000027 от 16.09.2011; MATLAB R2012b — Акт предоставления прав № Us000311 от 25.09.2012; Microsoft SQL Server 2012 Express LocalDB — бесплатно; Microsoft Visio Professional 2013 — Акт приема-передачи № 369 от 21 июля 2017; Microsoft Visual Studio Ultimate 2013 — Акт предоставления прав № Tr035055 от 19.06.2017; MiKTeX 2.9 — бесплатно; MSXML 4.0 SP2 Parser and SDK — бесплатно; NetBeans IDE 8.0.2 — бесплатно; Notepad++ — бесплатно; Oracle VM VirtualBox 5.0.2 — бесплатно; Origin 8.1 Sr2 — договор №13918/M41 от 24.09.2009 с ЗАО «СофтЛайн Трейд»; Python 3.4.3 — бесплатно; WCF RIA Services V1.0 SP2 — бесплатно; WinDjView 2.1 — бесплатно; Microsoft Windows 10 Enterprise — Акт приема-передачи № 369 от 21 июля 2017; MS Office 365 pro plus — Акт приема-передачи № 369 от 21 июля 2017
Ауд. 201а (компьютерная лаборатория ПМИК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Перечень программного обеспечения (со свободными лицензиями): Linux OpenSuse Tumbleweed, KDE, TeXLive, Mozilla Firefox, TeXStudio, Qt, QtCreator, Gcc, Python, Eric, LibreOffice, Cervisia, Kdbg, Umbrello, wxMaxima, Blender, digikam, GIMP, Gwenview, hugin, Inkscape, Okular, showFoto, Kmail, Konqueror, Konversation, Kopete, TigerVNC viewer, Amarok, K3b, Kdenlive, VLC media player, Kontact, Korganizer, Yast, Ark, Dolphin, Info Center, Kget, Konsole, Krusader, Midnight commander, OpenJDK, pgadmin3, Xterm, Emacs, Kate, Kcalc, Kpgg, Kleopatra, Kompare, Sweeper, Perl, Apache, PostgreSQL, MariaDB, SQLite, PHP

3. Современные профессиональные базы данных и информационные справочные системы

- [1] ЭБС «ZNANIUM.COM» <http://www.znanium.com>
[2] ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru>
[3] ЭБС IPRbooks <http://www.iprbookshop.ru>

- [4] ЭБС «Лань» <http://e.lanbook.com>
[5] ЭБС ВООk.ru <https://www.book.ru>
[6] ЭБС ТвГУ <http://megapro.tversu.ru/megapro/Web>
[7] Научная электронная библиотека eLIBRARY.RU (подписка на журналы)
https://elibrary.ru/projects/subscription/rus_titles_open.asp
[8] Репозитарий ТвГУ <http://eprints.tversu.ru>

4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- [1] C++ Tutorial, <https://code-live.ru/tag/cpp-manual/>

VI. Методические материалы для обучающихся по освоению дисциплины

Примеры задач для подготовки к контрольным работам

1. Примеры задач для контрольных работ:

- Написать функцию `void f(char * str)`, которая в строке `str` убирает все кратные пробелы между словами, а также в начале и конце строки. Например, из строки « Quando tu sei vicina » должно получиться «Quando tu sei vicina». Вычислить максимальное время работы функции в зависимости от длины входных данных.
- Написать шаблонную функцию `template<class T> int f(T * a, int m, T * b, int n)` которая принимает на вход массивы `a` и `b` длиной `m` и `n` соответственно. Функция должна вернуть количество элементов, которые встречаются только в одном из этих массивов. Максимальное время работы функции должно быть $O(n \log n + m \log m)$.

Требования к рейтинг контролю (1 семестр)

Самостоятельная работа 1. Темы: программы, использующие синтаксис, базовые типы переменных, операции, операторы соответствующие стандарту языка C.

Пример задания:

1. Написать функцию, которая находит элемент линейного массива, наиболее приближенный к среднему арифметическому элементов массива.
2. Написать функцию, которая выводит на печать шестнадцатеричное представление десятичного числа, используя двоичные операции
3. Написать программу, которая находит количество простых чисел на интервале $[x, y)$.

За решение каждой задачи выставляется максимум 5 баллов.

Самостоятельная работа 2. Темы: Библиотечные функции для работы со строками. Пример задания:

1. Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие двузначные числа.
 2. Написать программу, которая считывает английский текст из файла и выводит на экран слова, начинающиеся с гласных букв.
 3. Написать функцию `void f(char * str)`, которая в строке `str` убирает все кратные пробелы между словами, а также в начале и конце строки. Например, из строки «`Quando tu sei vicina`» должно получиться «`Quando tu sei vicina`».
- За решение каждой задачи выставляется максимум 5 баллов.

Самостоятельная работа 3. Темы: файловый ввод/вывод. Пример задания:

1. Написать функцию, получающую в качестве аргумента имя файла, содержащего последовательность вещественных чисел неизвестной длины, и возвращающую целое число, равное количеству максимальных элементов этой последовательности. Функция должна возвращать -1, -2 и т.д., если она не смогла открыть файл, прочитать элемент и т.д.. Основная программа должна вызывать эту функцию и выводить на экран результат ее работы.
2. Написать функцию, получающую в качестве аргумента имя файла, содержащего некоторую двоичную информацию, каждая запись которой представляется структурой `struct TRUNchar name[6];int ntrun;int lTime;`(название пункта назначения, номер поезда, время отправления) и возвращающую количество записей в файле. Функция должна возвращать —1, если она не смогла открыть файл. Основная программа должна сформировать файл, вызывать функцию и вывести на экран результат ее работы. Ввод данных произвести из программы, например `struct TRUNchar name[5];int ntrun;int lTime;myTrunc[]="tver",1,11,"Moscow",2,13,...`

За решение каждой задачи выставляется максимум 5 баллов.

Общая сумма В сумме за все задачи выставляет не более 40 баллов.

За работу на практических занятиях (решение задач у доски, выполнение домашних заданий) выставляется максимум 20 баллов.

За ответ на зачёте выставляется максимум 40 баллов.

Требования к рейтинг контролю (2 семестр)

Самостоятельная работа 1. Темы: алгоритмы сортировки и поиска, а именно сортировки вставками, сортировки пузырьком, сортировки Шелла. Пример задания:

1. Написать программу, сортирующую методом вставок главную диагональ двумерного массива целых чисел. Массив считать из текстового файла, содержащего в первой строке кол-во строк и кол-во столбцов, а далее в каждой строке целые числа, соответствующие элементам строки массива.
 2. Написать программу, сортирующую методом вставок двумерный массив целых чисел. При этом самое маленькое число должно оказаться в первом элементе первой строки, а самое большое в последнем элементе последней строки.
 3. Написать программу, ищущую методом половинного деления строку с заданным значением 1-го элемента строки в двумерном массиве целых чисел. Выполнить все необходимые для этого операции. Не использовать стандартные функции.
- За решение каждой задачи выставляется максимум 5 баллов.

Самостоятельная работа 2. Темы: динамические структуры данных, а именно линейные списки, односвязные и двусвязные списки. Пример задания:

1. Написать функцию, которая переворачивает список L, т.е. изменяет ссылки в этом списке так, чтобы его элементы оказались расположенными в обратном порядке.
 2. Написать функцию сортировки по убыванию массива целых чисел, использующую двоичное дерево поиска. Результат записать в массив.
- За решение каждой задачи выставляется максимум 5 баллов.

Самостоятельная работа 3. Темы: хеш-таблица, тестирование динамических структур данных.

Пример задания:

1. Для класса "группа студентов" построить хеш-таблицу по ключу - фамилия, написать функцию поиска в хеш-таблице номера записи по ключу, проверить работу хеш-поиска.
 2. Протестировать алгоритм создания линейного списка, содержащего целые числа. Оценить время поиска элемента, время вставки элемента.
 3. Протестировать алгоритм создания двоичного дерева поиска целых чисел. Оценить время поиска элемента, время вставки элемента.
- За решение каждой задачи выставляется максимум 5 баллов.

Общая сумма В сумме за все задачи выставляет не более 40 баллов.

За работу на практических занятиях (решение задач у доски, выполнение домашних заданий) выставляется максимум 20 баллов.

За ответ на зачёте выставляется максимум 40 баллов.

VII. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине Для аудиторной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 1л (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Ауд. 1л приспособлена для проведения лекционных и практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации и оснащена набором учебной мебели и меловой доской.
Ауд. 4б (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Ауд. 4б оснащена персональными ЭВМ (компьютер RAMEC STORM C2D 4600/160Gb/DVD-RW+Монитор LG TFT 17" L1753S-SF silver - 24 шт.) с доступом к сети Интернет, программным обеспечением общего назначения и специализированным ПО, мультимедийным проектором BenQ MP 724 с потолочным креплением и экраном 1105, кондиционером General Climate — 2 шт., коммутатором D-Link 10/100/1000mbps 16-port DGS-1016D Коммутатор D-Link 10/100/1000mbps 16-port DGS-1016D — 2 шт.

Для самостоятельной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 201а (компьютерная лаборатория ПМиК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Ауд. 201а (компьютерная лаборатория ПМиК) оснащена персональными ЭВМ (компьютер ПЭВМ «ХОПЕР» IS09001: 1.1/Intel Core i3-540/IntelH55-MLX/Unix-11.4/DVD RW Sony/Монитор 21,5" АОС TFT/клавиатура/мышь — 10 штук) с доступом к сети Интернет и необходимым программным обеспечением, системным блоком BASE P4 3200MHz 800 512K/1024 Мб DDR400/400Gb, концентратором сетевым DFE-916 DX HUB 16x10/100.

VIII. Сведения об обновлении рабочей программы дисциплины

№ Обновленный раздел рабочей дисциплины	Описание внесённых изменений	Дата и протокол п/п программы заседания кафедр, утвердившего изменения
---	------------------------------	--