

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»**
(БГТУ им. В.Г. Шухова)

УТВЕРЖДАЮ
Директор института



« 26 » 2022 г.

РАБОЧАЯ ПРОГРАММА
дисциплины

Основы программирования

направление подготовки:

09.03.04 «Программная инженерия»

Направленность программы (профиль):

Разработка программно-информационных систем

Квалификация

Бакалавр

Форма обучения

Очная

Институт энергетики, информационных технологий и управляющих систем

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Рабочая программа составлена на основании требований:

- Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.04 «Программная инженерия», утвержденного приказа Минобрнауки России от 19.09.2017 № 920
- учебного плана, утвержденного ученым советом БГТУ им. В.Г. Шухова в 2021 году.

Составители: к.т.н., доцент

(ученая степень и звание, подпись)

(Поляков В.М.)

(инициалы, фамилия)

(ученая степень и звание, подпись)

(Притчин И.С.)

(инициалы, фамилия)

Рабочая программа обсуждена на заседании кафедры

« 28 » 04 2022 г., протокол № 8

Заведующий кафедрой: к.т.н., доцент

(ученая степень и звание, подпись)

(Поляков В.М.)

(инициалы, фамилия)

Рабочая программа согласована с выпускающей кафедрой программного обеспечения вычислительной техники и автоматизированных систем

(наименование кафедры/кафедр)

Заведующий кафедрой: к.т.н., доцент

(ученая степень и звание, подпись)

(Поляков В.М.)

(инициалы, фамилия)

« 28 » 04 2022г.

Рабочая программа одобрена методической комиссией института

« 26 » 05 2022г., протокол № 9

Председатель к.т.н., доцент

(ученая степень и звание, подпись)

(Семернин А.Н.)

(инициалы, фамилия)

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Категория (группа) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Наименование показателя оценивания результата обучения по дисциплине
Разработка алгоритмов и программ	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1 Использует основные методологии программирования, языки программирования и принципы работы с базами данных, операционными системами и оболочками, современными программными средами разработки информационных систем и технологий	Знания
		ОПК-6.2 Применяет языки программирования для 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.	Умения
		ОПК-6.3 Демонстрирует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	Навыки
Применение в практической деятельности основных знаний в области информатики	ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.1 Понимает основные концепции, принципы, теории и факты, связанные с информатикой, базами данных, операционными системами и компьютерными сетями	Знания
		ОПК-7.2 Применяет на практике основные концепции, принципы и теории, связанные с информатикой, при решении стандартных задач, базами данных, операционными системами и компьютерными сетями	Умения
		ОПК-7.3 Демонстрирует навыками решения задач профессиональной деятельности с использованием основ информатики, базами данных, операционными системами и компьютерными сетями	Навыки

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1. **Компетенция ОПК-6.** Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов.

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1.	Основы программирования
2.	Основы алгоритмизации
3.	Спецификация, архитектура и проектирование программных систем
4.	Тестирование программных систем
5.	Производственная технологическая практика

2. **Компетенция ОПК-7.** Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой.

Данная компетенция формируется следующими дисциплинами.

Стадия	Наименования дисциплины
1.	Информатика
2.	Основы программирования
3.	Базы данных
4.	Операционные системы
5.	Компьютерные сети

3. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 10 зач. единиц, 360 часов.

Дисциплина реализуется в рамках практической подготовки: 6 зач. единиц.

Форма промежуточной аттестации зачёт, экзамен.

Вид учебной работы	Всего часов	Семестр № 1	Семестр № 2	Семестр № 3
Общая трудоемкость дисциплины, час	396	144	144	108
Контактная работа (аудиторные занятия), в т.ч.:	145	73	55	34
лекции	51	34	17	
лабораторные	68	34	34	
практические	34			34
групповые консультации в период теоретического обучения и промежуточной аттестации	9	5	4	
Самостоятельная работа студентов, включая индивидуальные и групповые консультации, в том числе:	234	71	89	74
Курсовой проект	—	—	—	—
Курсовая работа	36	—	—	36
Расчетно-графическое задание	—	—	—	—
Индивидуальное домашнее задание	—	—	—	—
Самостоятельная работа на подготовку к аудиторным занятиям (лекции, практические занятия, лабораторные занятия)	107	35	53	38
Экзамен	72	36	36	—

4.СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1 Наименование тем, их содержание и объем

Курс 1 Семестр 1

№ п/п	Наименование раздела (краткое содержание)	Объем на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа
1. Алгоритмы. Языки программирования.					
	Программы, как способ записи алгоритмов на языке программирования. История, тенденции развития и классификация языков программирования, их свойства	2			1
2. Введение в язык С.					
	Структура программы. Подключение библиотек. Сборка программы на языке программирования С. Ввод вывод в языке программирования С.	2		2	3
3. Типы данных языка программирования С.					
	Понятие «типа данных», классификация типов данных, согласно стандарту С90. Тип данных bool, определенный в <stdbool.h>.	2			1
4. Целочисленные типы данных. Вещественные типы данных. Указатели.					
	Представление целочисленных типов данных в памяти ЭВМ, вычисление диапазонов, принимаемых тем или иных типов данных. Представление вещественных типов данных в памяти ЭВМ. Проблемы вещественной арифметики. Операции, определенные над указателями.	2			1
5. Линейные алгоритмы. Разветвляющиеся алгоритмы.					
	Реализация линейных / разветвляющихся алгоритмов в языке программирования С. Знакомство с простейшими операциями.	2		10	6
6. Арифметические и побитовые операции языка программирования С.					
	Понятие «операция». Типы операций. Арифметические операции: унарные плюс и минус, сложение и вычитание, умножение, деление и остаток от деления, приоритеты арифметических операций. Логические побитовые операции, операции сдвига.	2			1
7. Логические операции и операции сравнения. Приведение типов.					
	Логические операции, операции сравнения и их приоритеты. Приоритеты операций. Приведение типов.	2			1
8. Циклические алгоритмы: цикл <i>for</i> .					
	Описание и использование цикла <i>for</i> . Рассмотрение решаемых при помощи цикла задач.	2		4	2
9. Циклические алгоритмы: циклы <i>while</i> , <i>do-while</i> . Операторы <i>break</i> , <i>goto</i> , <i>continue</i> .					
	Описание и использование циклов <i>while</i> , <i>do-while</i> . Рассмотрение решаемых при помощи циклов задач.	2		2	2

	Рассмотрение аспектов использования операторов <i>break</i> , <i>goto</i> , <i>continue</i> .				
10. Функции.					
	Понятие «Функция». Объявление и определение функций. Формальные и фактические параметры. Действия, производимые при вызове функций, возвращаемое значение функций. Передача аргументов.	4		2	2
11. Алгоритмы обработки одномерных массивов.					
	Алгоритмы обработки одномерных массивов, независимые от упорядоченности.	2		4	4
12. Алгоритмы обработки одномерных массивов. Алгоритмы поиска.					
	Алгоритмы, зависящие от упорядоченности. Алгоритмы поиска: линейный поиск, бинарный поиск, блочный поиск, тернарный поиск. Применение бинарного поиска в задачах оптимизации.	2		4	4
13. Рекурсивные функции.					
	Решение задач при помощи рекурсивных функций. Стек и рекурсия.	2		2	2
14. Алгоритмы сортировок.					
	Сортировки выбором, вставками, обменная сортировка, сортировка расческой. Сортировки подсчетом, цифровая, слиянием, кучей, быстрая.	4		2	2
15 Структуры.					
	Объявление структур. Передача структур в функции. Массивы структур.	2		2	3
	ВСЕГО	34		34	35

Курс1 Семестр 2

№ п/п	Наименование раздела (краткое содержание)	Объем на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа
1. Структуры данных на одномерных массивах: множества.					
	Представление множеств в памяти ЭВМ.	1		2	3
2. Структуры данных на одномерных массивах: вектор.					
	Работа с динамической памятью в языке программирования С. Обзор механизма работы вектора (саморасширяющегося массива).	1		2	3
3. Алгоритмы обработки многомерных массивов.					
	Представления многомерных массивов в памяти ЭВМ (схемы размещения). Функции, используемые для обработки многомерных массивов.			4	4
4. Строки.					
	Алгоритмы обработки строк стандартной библиотеки. Реализация пользовательских функций обработки.	1		4	5
5. Введение в C++: ссылки, изменения в управляющих конструкциях.					
	Ссылочный тип. Ссылки на константные и неконстантные	2		2	4

	данные. <i>Range-based for</i> . Инициализация локальных переменных в <i>if</i> и <i>switch</i> .				
6.	Изменения в типах данных, функциях.				
	Ключевое слово <i>auto</i> . Перегрузки функций, параметры функций по умолчанию, <i>lambda</i> -функции, введение в шаблоны.	2			2
7.	Структуры в C++.				
	Причины появления в языках программирования, ограничение доступа к полям структуры, константность функций-членов, конструкторы, инициализация полей структур и классов, деструкторы.	2		2	4
8.	Обзор контейнеров языка C++.				
	Массивы, векторы, дек, односвязный список, двусвязный список, стек, очередь, приоритетная очередь, отображения, множества, мультимножества, неупорядоченные отображения и множества.	2		4	8
9.	Итераторы. Алгоритмы на паре итераторов.				
	Подход без использования итераторов. Определение «итератор». Рассмотрение алгоритмов на паре итераторов.	2		2	4
10.	Работа с файловыми потоками.				
	Потоки ввода/вывода. Перенаправление потоков.	2		4	6
11.	Динамическое программирование.				
	Решение задач с подходов динамического программирования. Мемоизация и табуляция.	2		4	6
12.	Реализация структур данных с использованием ООП.				
	Реализация вектора, односвязного списка, несбалансированных бинарных деревьев, с использованием объектно-ориентированного подхода в программировании.			4	4
	ВСЕГО	17		34	53

Курс 2 Семестр 3

№ п/п	Наименование раздела (краткое содержание)	Объем на тематический раздел по видам учебной нагрузки, час			
		Лекции	Практические занятия	Лабораторные занятия	Самостоятельная работа
1.	Вектора				
	Реализация СД для хранения произвольного типа.		6		6
2.	Длинная арифметика				
	Способы представления чисел в памяти ЭВМ для решения задач длинной арифметики.		6		6
3.	Линейная алгебра.				
	Разработка модуля программных объектов для решения задач линейной алгебры (матрицы целочисленных дро-		6		6

	бей).				
4. Аналитическая геометрия					
	Разработка модуля программных объектов для решения задач аналитической геометрии на плоскости		6		8
5. Модульное программирование.					
	Системы сборки проектов (Make, CMake).		10		12
	ВСЕГО		34		38

4.2. Содержание практических (семинарских) занятий

№ п/п	Наименование раздела дисциплины	Тема практического (семинарского) занятия	К-во часов	Самостоятельная работа на подготовку к аудиторным занятиям
семестр № 3				
1	Вектора	Разработка модуля программных объектов для работы с векторами	6	6
2	Длинная арифметика	Разработка модуля программных объектов для работы с длинными целыми	6	6
3	Линейная алгебра.	Разработка модуля программных объектов для решения задач линейной алгебры	6	6
4	Аналитическая геометрия	Разработка модуля программных объектов для решения задач линейной алгебры (матрицы целочисленных дробей).	6	8
5	Модульное программирование.	Системы сборки проектов (Make, CMake).	10	12
	ВСЕГО		34	38

4.3. Содержание лабораторных занятий

№ п/п	Наименование раздела дисциплины	Тема лабораторного занятия	К-во часов	Самостоятельная работа на подготовку к аудиторным занятиям
семестр №1				
1	Введение в язык С.	Ввод/вывод в языке программирования С.	2	2
2	Линейные алгоритмы. Разветвляющиеся алгоритмы.	Программирование алгоритмов разветвляющейся структуры.	10	8
3	Циклические алгоритмы: цикл <i>for</i> . Циклические алгоритмы: циклы <i>while</i> , <i>do-while</i> . Операторы <i>break</i> , <i>goto</i> , <i>continue</i>	Программирование алгоритмов циклической структуры.	6	4
4	Функции.	Введение в функции.	2	2
5	Алгоритмы обработки одномерных массивов.	Алгоритмы обработки одномерных массивов.	4	2
6	Алгоритмы обработки одномерных массивов.	Бинарный поиска при решении задач оптимизации.	4	2
7	Рекурсивные функции.	Рекурсия.	2	2
8	Алгоритмы сортировок.	Анализ алгоритмов сортировки.	2	2
9	Структуры	Структуры. Функции для работы со структурами.	2	2
ИТОГО:			34	26
семестр №2				
1	Структуры данных на одномерных массивах: множества.	Множества	2	3
2	Структуры данных на одномерных массивах: вектор.	Реализация структуры данных «Вектор».	2	2
3	Алгоритмы обработки многомерных массивов.	Алгоритмы обработки многомерных массивов.	4	4
4	Строки.	Строки.	4	3
5	Введение в С++.	Введение в С++.	2	2
6	Структуры в С++.	Структуры в С++.	2	2
7	Обзор контейнеров языка С++.	Обзор контейнеров языка С++.	4	4
8	Итераторы. Алгоритмы на паре итераторов	Итераторы. Алгоритмы на паре итераторов	2	2

9	Работа с файловыми потоками.	Работа с файловыми потоками.	4	4
10	Динамическое программирование.	Динамическое программирование.	4	4
11	Реализация структур данных с использованием ООП.	Реализация структур данных с использованием ООП.	4	3
ИТОГО:			34	33
ВСЕГО:			34	59

4.4. Содержание курсовой работы

Целью выполнения курсовой работы является углубление и закрепление студентами знаний основных приемов, методов и принципов работы при решении на ЭВМ задач с использованием языков высокого уровня, соблюдая принципы структурного программирования. Для выполнения работы достаточно знаний основных разделов дисциплин «Основы программирования», «Математический анализ», «Алгебра и геометрия», «Дискретная математика». Курсовая работа заключается в разработке модулей программных объектов и приложений для решения задач заданного класса.

Курсовая работа оформляется в виде пояснительной записки (15-20 с.), в которой отражаются все этапы создания программного продукта.

В процессе выполнения курсовой работы осуществляется контактная работа обучающегося с преподавателем. Консультации проводятся в аудитории. Примеры тем курсовых работ.

1. Точное решение системы линейных уравнений произвольного порядка с рациональными коэффициентами.
2. Вычисления значения арифметического выражения, заданного строкой.
3. Решение головоломки «Судоку».
4. Решение головоломки «Какуро».
5. Обзор инструментов тестирования в языке программирования С.
6. Нахождение рациональных корней многочлена с рациональными коэффициентами.
7. Реализация игры «Морской бой».
8. Обзор алгоритмов обработки строк.
9. Реализация красно-черных и AVL-деревьев.
10. Разработка редактора кода в консоли.
11. Реализация алгоритма генерации уровней в игре.

Пояснительная записка оформляется в произвольной форме в зависимости от выбранной предметной области, однако обязательно включает в себя анализ предметной области, обзор имеющихся решений, этапы проектирования разработанного решения, результаты работы разработанного ПО.

4.5. Содержание расчетно-графического задания, индивидуальных домашних заданий

Не предусмотрено учебным планом.

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

5.1. Реализация компетенций

1. Компетенция ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов.

Наименование индикатора достижения компетенции	Используемые средства оценивания
ОПК-6.1 Использует основные методологии программирования, языки программирования и принципы работы с базами данных, операционными системами и оболочками, современными программными средами разработки информационных систем и технологий	защита лабораторных работ, зачет, экзамен
ОПК-6.2 Применяет языки программирования для 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.	защита лабораторных работ, зачет, экзамен
ОПК-6.3 Демонстрирует навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	защита лабораторных работ, зачет, экзамен

2 Компетенция ОПК-7. Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой

Наименование индикатора достижения компетенции	Используемые средства оценивания
ОПК-7.1 Понимает основные концепции, принципы, теории и факты, связанные с информатикой, базами данных, операционными системами и компьютерными сетями	защита лабораторных работ, зачет, экзамен
ОПК-7.2 Применяет на практике основные концепции, принципы и теории, связанные с информатикой, при решении стандартных задач, базами данных, операционными системами и компьютерными сетями	защита лабораторных работ, зачет, экзамен
ОПК-7.3 Демонстрирует навыками решения задач профессиональной деятельности с использованием основ информатики, базами данных, операционными системами и компьютерными сетями	защита лабораторных работ, зачет, экзамен

5.2. Типовые контрольные задания для промежуточной аттестации

5.2.1. Перечень контрольных вопросов (типовых заданий)

для экзамена

Курс 1 Семестр 1

№ п/п	Наименование раздела дисциплины	Содержание вопросов (типовых заданий)
1	Алгоритмы. Языки программирования.	<ol style="list-style-type: none">1. Дайте определение термину 'алгоритм'. Перечислите свойства алгоритмов с их определениями. (ОПК-6.1)2. Что является входными и выходными данными? (ОПК-6.1)3. Перечислите способы описания алгоритмов. Приведите примеры каждого способа. От чего зависит выбор способа записи алгоритма? (ОПК-6.1)
2	Введение в язык C.	<ol style="list-style-type: none">1. Перечислите основные достоинства и недостатки языка C. (ОПК-7.1)2. Препроцессор, компилятор и компоновщик. Этапы компиляции. (ОПК-7.1)3. При использовании директивы препроцессора <i>#include</i> в чем заключается разница между <code>< ></code> и <code>" "</code>. (ОПК-7.1)4. Перечислите принципы структурного программирования. (ОПК-6.1)5. Каскадная модель. Её особенности и этапы. (ОПК-7.1)
3	Типы данных языка программирования C.	<ol style="list-style-type: none">1. Что определяет тип данных? Перечислите базовые типы данных C. (ОПК-7.1)
4	Целочисленные типы данных. Вещественные типы данных. Указатели.	<ol style="list-style-type: none">1. Целочисленный тип <i>char</i>. Вывод диапазона значений <i>char</i> / <i>unsigned char</i>. (ОПК-7.2)2. Целочисленный тип <i>int</i>. Вывод диапазона значений. (ОПК-7.1)3. Вещественный тип <i>float</i> и <i>double</i>. В чём разница между ними? (ОПК-7.1)4. В каких системах счисления могут быть представлены целочисленные литералы? Приведите примеры. (ОПК-7.2)5. Приведите примеры <i>escape</i>-последовательностей. (ОПК-7.2)6. В каком случае инициализация переменной при её объявлении будет излишней? (ОПК-7.1)7. Массивы в C. Объявление. Инициализация. (ОПК-7.2)8. Указатели в C. Операции взятия адреса и косвенного доступа. Макрос <i>NULL</i>. (ОПК-7.2)9. В чем заключается разница между литералами <code>'x'</code> и <code>"x"</code>? (ОПК-7.1)10. В чем заключается разница между объявлением констант при помощи макроса <i>#define</i> и при помощи ключевого слова <i>const</i>? (ОПК-7.1)
5	Линейные алгоритмы. Разветвляющиеся алгоритмы.	<ol style="list-style-type: none">1. Что такое линейный алгоритм? Приведите примеры. (ОПК-6.1)2. Функция ввода <i>scanf</i> и её параметры. (ОПК-7.2)3. Функция вывода <i>printf</i> и её параметры. (ОПК-7.2)4. Подключение библиотеки в C. Какую библиотеку необходимо подключить для использования математических функций? (ОПК-7.3)5. Функции для генерации случайных чисел. Для чего используются функции <i>srand</i> и <i>rand</i>? (ОПК-7.2)

		<p>6. Правило выполнения деления в С. (ОПК-7.1)</p> <p>7. Какие алгоритмы называются разветвляющимися? (ОПК-6.1)</p> <p>8. Какие имеются требования к тестовым данным для тестирования программ с ветвлением? (ОПК-6.2)</p> <p>9. Что представляет собой логическое выражение? (ОПК-7.1)</p> <p>10. Как организовать бинарное ветвление? (ОПК-7.2)</p> <p>11. Как организовать множественное ветвление? (ОПК-7.2)</p> <p>12. В каком случае лучше использовать тернарный оператор? (ОПК-7.3)</p> <p>13. К какому <i>if</i> относится <i>else</i>? (ОПК-7.3)</p>
6	Арифметические и побитовые операции языка программирования С.	<p>1. Дайте определение терминам 'операция' и 'операнд'. Перечислите виды операций в зависимости от количества операндов. Приведите примеры. (ОПК-6.1, ОПК-7.1)</p> <p>2. Арифметические операции и их приоритеты (ОПК-7.1)</p> <p>3. Какие существуют способы в С увеличить или уменьшить значение переменной на единицу? (ОПК-7.2)</p> <p>4. В чем заключается разница между постфиксной и префиксной записи инкремента и декремента? (ОПК-7.2)</p> <p>5. Какая проблема может возникнуть при умножении чисел в С? (ОПК-7.3)</p> <p>6. Каким образом работает деление в языке программирования С. (ОПК-7.2)</p> <p>7. Побитовые операции и их приоритеты. (ОПК-7.1)</p>
7	Логические операции и операции сравнения. Приведение типов.	<p>1. Перечислите какие бывают приведения типов в С. В каких случаях они выполняются? (ОПК-7.1)</p> <p>2. Перечислите последствия при приведении типов. Приведите примеры. (ОПК-7.2)</p> <p>3. Логические операции и их приоритеты. Опишите как работает 'ленивая' схема вычислений. (ОПК-7.1)</p>
8	Циклические алгоритмы: цикл <i>for</i> .	<p>1. Какие алгоритмы называются циклическими? (ОПК-6.1)</p> <p>2. Секции цикла <i>for</i>. Какие из них могут быть опущены? (ОПК-7.2)</p>
9	Циклические алгоритмы: циклы <i>while</i> , <i>do-while</i> . Операторы <i>break</i> , <i>goto</i> , <i>continue</i> .	<p>1. В чем заключается разница между циклом <i>while</i> и циклом <i>do-while</i>? (ОПК-6.1, ОПК-7.1)</p> <p>2. В каком случае цикл <i>while</i> может стать бесконечным? (ОПК-7.2)</p> <p>3. Сколько раз обязательно выполнится тело цикла <i>do-while</i>? (ОПК-7.3)</p> <p>4. Инструкции <i>break</i> и <i>goto</i>. Когда имеет смысл их применять? (ОПК-7.1)</p> <p>5. Как заменить цикл <i>for</i> 'эквивалентным' через цикл <i>while</i>. (ОПК-7.3)</p> <p>6. Инструкция <i>continue</i>. В чём разница применения <i>continue</i> для цикла <i>for</i> и <i>while</i>? (ОПК-7.3)</p>
10	Функции.	<p>1. В каких случаях целесообразно использовать функции? (ОПК-6.1, ОПК-7.1)</p> <p>2. Принципы <i>DRY</i> и <i>WET</i>. (ОПК-6.1, ОПК-7.1)</p> <p>3. В чём разница между объявлением и определением функции? (ОПК-7.2)</p> <p>4. Какие параметры называются фактическими, а какие формальными? (ОПК-7.1)</p> <p>5. Может ли в результате вызова функции измениться значение фактического параметра функции? (ОПК-7.3)</p> <p>6. Действия, производимые при вызове функций. (ОПК-7.3)</p>

		<p>7. Для чего используется квалификатор <i>const</i> для формальных параметров функции? (ОПК-7.2)</p> <p>8. В чем заключается побочный эффект функции? (ОПК-7.2)</p> <p>9. Как изменить аргументы, передаваемые в функцию? (ОПК-7.2)</p>
11	Алгоритмы обработки одномерных массивов.	<p>1. Как описываются массивы в С? (ОПК-7.2)</p> <p>2. Как осуществляется ввод и вывод одномерных массивов? (ОПК-7.2)</p> <p>3. Какими способами может быть осуществлен поиск элемента в упорядоченном и неупорядоченном массиве? (ОПК-7.2)</p> <p>4. Опишите алгоритм однопроходного алгоритма удаления из последовательности членов, удовлетворяющих заданному условию. (ОПК-7.2)</p> <p>5. Как передавать в функцию адрес некоторой отличной от начальной части массива? Приведите пример. (ОПК-7.2)</p> <p>6. В чем заключается принцип работы стека? (ОПК-7.1)</p> <p>7. В чем заключается принцип работы кучи? (ОПК-7.1)</p> <p>8. В каких случаях используются динамические массивы? (ОПК-7.1)</p> <p>9. Какие функции отвечают в С за работу с динамической памятью? Как работают данные функции? (ОПК-7.2)</p> <p>10. Передача функции в функцию. Для чего необходим данный прием? Приведите примеры. (ОПК-7.3)</p> <p>11. Опишите последовательность решения задачи на одномерные массивы с использованием принципа пошаговой детализации. (ОПК-7.3, ОПК-6.3)</p>
12	Алгоритмы обработки одномерных массивов. Алгоритмы поиска.	<p>1. Опишите алгоритм линейного поиска. (ОПК-7.1)</p> <p>2. Опишите алгоритм блочного поиска. Каков оптимальный размер блока при блочном поиске? Приведите доказательство. (ОПК-7.1)</p> <p>3. Бинарный поиск в двух вариациях. Различия в реализации. (ОПК-7.2)</p>
13	Рекурсивные функции.	<p>1. Опишите действия, производимые на стеке вызовов при работе с рекурсивными функциями. (ОПК-7.1)</p> <p>2. Опишите виды рекурсии, приведите примеры. (ОПК-7.1, ОПК-6.1)</p> <p>3. Подходы к решению задач с использованием рекурсии. (ОПК-7.2)</p>
14	Алгоритмы сортировок.	<p>1. Реализация сортировки пузырьком. (ОПК-7.2)</p> <p>2. Реализация сортировки выбором. (ОПК-7.2)</p> <p>3. Реализация сортировки вставками. (ОПК-7.2)</p> <p>4. Реализация сортировки расческой. (ОПК-7.2)</p> <p>5. Реализация сортировки Шелла. (ОПК-7.2)</p> <p>6. Реализация пирамидальной сортировки. (ОПК-7.2)</p> <p>7. Реализация быстрой сортировки. (ОПК-7.2)</p> <p>8. Реализация цифровой сортировки. (ОПК-7.2)</p>
15	Структуры.	<p>1. Объявление структуры. (ОПК-7.2)</p> <p>2. Инициализация структуры. (ОПК-7.2)</p> <p>3. Создание массива структур. (ОПК-7.2)</p> <p>4. Создание динамического массива структур. (ОПК-7.2)</p> <p>5. Передача структур в функцию. (ОПК-7.2)</p> <p>6. Сортировка функций. Использование компаратора при сортировке. (ОПК-7.3)</p> <p>7. Битовые поля. (ОПК-7.1)</p>

Курс 1 Семестр 2

16	Структуры данных на одномерных массивах: множества.	<ol style="list-style-type: none"> 1. Способ представления множества на битах целого числа. (ОПК-7.1) 2. Способ представления множества на неупорядоченном массиве. (ОПК-7.1) 3. Способ представления множества на упорядоченном массиве. (ОПК-7.1)
17	Структуры данных на одномерных массивах: вектор.	<ol style="list-style-type: none"> 1. Политики расширения динамического массива при его заполнении. (ОПК-7.3) 2. Реализация функций для работы с векторами. (ОПК-7.2)
18	Алгоритмы обработки многомерных массивов.	<ol style="list-style-type: none"> 1. Способы размещения матрицы в динамической памяти. (ОПК-7.1) 2. Схемы размещения. (ОПК-7.2)
19	Строки.	<ol style="list-style-type: none"> 1. Стандартная библиотека <code><string.h></code>. Функции. Механизмы работы. (ОПК-7.1) 2. Ввод и вывод строк в языке программирования C. (ОПК-7.2)
20	Введение в C++: ссылки, изменения в управляющих конструкциях.	<ol style="list-style-type: none"> 1. Виды ссылок: ссылки на неконстантные данные, ссылки на константные данные. (ОПК-7.1) 2. <i>Range-based for</i>. (ОПК-7.2) 3. Вариации <i>range-based for</i>. (ОПК-7.2) 4. Ключевое слово <i>auto</i> и <i>decltype</i>. (ОПК-7.2)
21	Изменения в типах данных, функциях.	<ol style="list-style-type: none"> 1. <i>lambda</i>-функции. (ОПК-7.1) 2. Шаблоны функций. (ОПК-7.1) 3. Перегрузки функций. (ОПК-7.1) 4. Параметры функции по умолчанию. (ОПК-7.1)
22	Структуры в C++.	<ol style="list-style-type: none"> 1. Причины появления в языках программирования (ОПК-7.1) 2. Ограничение доступа к полям структуры. (ОПК-7.1) 3. Константность функций-членов. (ОПК-7.1) 4. Конструкторы. (ОПК-7.1) 5. Инициализация полей структур. (ОПК-7.1) 6. Деструкторы. (ОПК-7.1)
23	Обзор контейнеров языка C++.	<ol style="list-style-type: none"> 1. Массивы. (ОПК-7.1) 2. Векторы. (ОПК-7.1) 3. Дек. (ОПК-7.1) 4. Односвязный список. (ОПК-7.1) 5. Двусвязный список. (ОПК-7.1) 6. Стек. (ОПК-7.1) 7. Очередь. (ОПК-7.1) 8. Приоритетная очередь. (ОПК-7.1) 9. Отображения. (ОПК-7.1) 10. Множества. (ОПК-7.1) 11. Мультимножества. (ОПК-7.1) 12. Неупорядоченные отображения и множества. (ОПК-7.1)
24	Итераторы. Алгоритмы на паре итераторов.	<ol style="list-style-type: none"> 1. Итераторы. Определение. Предпосылки к созданию. (ОПК-7.1) 2. Алгоритмы на паре итераторов. (ОПК-7.2)
25	Работа с файловыми потоками.	<ol style="list-style-type: none"> 1. Понятия «файл», «поток». (ОПК-7.1) 2. Функции для работы с файлами. (ОПК-7.2)
26	Динамическое программирование.	<ol style="list-style-type: none"> 1. Динамическое программирование. Этапы решения задач с использованием динамического программирования. (ОПК-6.1, ОПК-7.1) 2. Задачи одномерного динамического программирования. (ОПК-7.1) 3. Задачи многомерного динамического программирования.

		(ОПК-7.1) 4. Техники кэширования при решении задач на динамическое программирование: мемоизация и табуляция. (ОПК-7.3)
27	Реализация структур данных с использованием ООП.	1. Структуры и классы. (ОПК-7.1) 2. Конструктор без параметров / с параметрами. (ОПК-7.1) 3. Деструктор. (ОПК-7.1) 4. Конструкторы копирования и перемещения. (ОПК-7.1) 5. Перегрузка операций. (ОПК-7.2)

5.2.2. Перечень контрольных материалов для защиты курсовой работы

1. Формулировка принципов структурного программирования
2. Какие программные объекты можно описывать в заголовочных файлах, а какие не рекомендуется?
3. По каким критериям выполняется выделение модулей при разработке программы?
4. Какими соображениями руководствуются при создании наборов тестовых данных?

5.3. Типовые контрольные задания (материалы) для текущего контроля в семестре

Текущий контроль осуществляется в течение семестра в форме защиты лабораторных работ.

В методических указаниях к выполнению лабораторных работ по дисциплине представлен перечень лабораторных работ, обозначены цель и задачи, представлены варианты заданий.

Защита лабораторной работы проводится в форме устного опроса студента и направлена на проверку степени усвоения материала и понимания теоретических сведений, используемых в процессе выполнения работы. Для защиты необходимо представить в печатной (рукописной) форме отчет по лабораторной работе, выполненный самостоятельно и в соответствии со всеми требованиями, приведёнными в методических указаниях к выполнению лабораторных работ. Примерный перечень контрольных вопросов для защиты лабораторных работ приведен в таблице:

Тематика лабораторной работы	Контрольные вопросы
Лабораторная работа №1 Ввод/вывод в языке программирования С.	1. Функция ввода <i>printf</i> и её параметры. (ОПК-7.1) 2. Функция вывода <i>scanf</i> и её параметры. (ОПК-7.1) 3. Буферизация вывода. Очистка потока вывода. (ОПК-7.2) 4. Спецификаторы преобразования для функций ввода и вывода. (ОПК-7.1) 5. Ввод и вывод строк. Строковые литералы. (ОПК-7.1) 6. Типы данных языка С: (ОПК-7.1) <ul style="list-style-type: none"> • Диапазоны значений базовых типов. • Количество значений, представимых переменными базового ти-

	<p>па.</p> <ul style="list-style-type: none"> • Литералы базовых типов.
Лабораторная работа №2а. Программирование алгоритмов разветвляющейся структуры.	
Лабораторная работа №2б. Программирование алгоритмов разветвляющейся структуры.	<ol style="list-style-type: none"> 1. Способы организации ветвления в языке программирования С. (ОПК-7.1, ОПК-7.2) 2. Условный оператор <i>if</i>. (ОПК-7.1) 3. Определение «логическое выражение». (ОПК-7.1) 4. Оператор множественного ветвления <i>switch</i>. (ОПК-7.1)
Лабораторная работа №2с. Программирование алгоритмов разветвляющейся структуры.	
Лабораторная работа №3а Разработка алгоритмов циклической структуры. Введение в тестирование.	<ol style="list-style-type: none"> 1. Циклы языка программирования С. (ОПК-7.1) 2. Описание и механизм работы цикла <i>for</i>. (ОПК-7.1, ОПК-7.2) 3. Описание и механизм работы цикла <i>while</i>. (ОПК-7.1, ОПК-7.2) 4. Описание и механизм работы цикла <i>do</i>. (ОПК-7.1, ОПК-7.2) 5. Оператор <i>break</i>. (ОПК-7.1)
Лабораторная работа №3б Разработка алгоритмов циклической структуры.	<ol style="list-style-type: none"> 6. Оператор <i>goto</i>. (ОПК-7.1) 7. Оператор <i>continue</i>. (ОПК-7.1) 8. В каких случаях оправдано использование <i>break</i>, <i>goto</i>, <i>continue</i>. (ОПК-7.3)
Лабораторная работа №4 Введение в функции.	<ol style="list-style-type: none"> 1. В каких случаях целесообразно использовать функции? (ОПК-7.1) 2. Принципы <i>DRY</i> и <i>WET</i>. (ОПК-7.1) 3. В чём разница между объявлением и определением функции? (ОПК-7.1) 4. Какие параметры называются фактическими, а какие формальными? (ОПК-7.1) 5. Может ли в результате вызова функции измениться значение фактического параметра функции? (ОПК-7.2) 6. Действия, производимые при вызове функций. (ОПК-7.1) 7. Для чего используется квалификатор <i>const</i> для формальных параметров функции? (ОПК-7.1) 8. В чем заключается побочный эффект функции? (ОПК-7.1) 9. Как изменить аргументы, передаваемые в функцию? (ОПК-7.2)
Лабораторная работа №5 Алгоритмы обработки одномерных массивов.	<ol style="list-style-type: none"> 1. Опишите алгоритм ввода / вывода массива. (ОПК-6.1, ОПК-7.2) 2. Поиск позиции элемента, удовлетворяющему условию. (ОПК-6.1, ОПК-7.2) 3. Поиск количества элементов, удовлетворяющих условию. (ОПК-6.1, ОПК-7.2) 4. Поиск максимального количества подряд идущих элементов, удовлетворяющих условию. (ОПК-6.1, ОПК-7.2) 5. Однопроходный алгоритм удаления. (ОПК-6.1, ОПК-7.2) 6. Вставка элемента с сохранением порядка остальных элементов. (ОПК-6.1, ОПК-7.2) 7. Вставка элемента без сохранения порядка остальных элементов. (ОПК-6.1, ОПК-7.2) 8. Удаление элемента с сохранением порядка остальных элементов. (ОПК-6.1, ОПК-7.2) 9. Удаление элемента без сохранения порядка остальных элементов.

	(ОПК-6.1, ОПК-7.2) 10. Передача функций, как параметров. (ОПК-7.3)
Лабораторная работа №6 Бинарный поиска при решении задач оптимизации.	1. Бинарный поиск в двух вариациях. Различия в реализации. (ОПК-6.1, ОПК-7.2) 2. Определение сложности алгоритмов бинарного поиска. (ОПК-6.1, ОПК-6.2)
Лабораторная работа №7 Рекурсия.	1. Опишите действия, производимые на стеке вызовов при работе с рекурсивными функциями. (ОПК-7.2) 2. Опишите виды рекурсии, приведите примеры. (ОПК-6.1, ОПК-7.1) 3. Подходы к решению задач с использованием рекурсии. (ОПК-6.3, ОПК-7.3)
Лабораторная работа №8 Алгоритмы сортировок.	1. Сортировка выбором. (ОПК-6.1, ОПК-7.2) 2. Сортировка вставками. (ОПК-6.1, ОПК-7.2) 3. Обменная сортировка (пузырьковая). (ОПК-6.1, ОПК-7.2) 4. Сортировка расческой. (ОПК-6.1, ОПК-7.2) 5. Сортировка слиянием. (ОПК-6.1, ОПК-7.2) 6. Быстрая сортировка. (ОПК-6.1, ОПК-7.2) 7. Поразрядная сортировка. (ОПК-6.1, ОПК-7.2) 8. Сортировка подсчетом. (ОПК-6.1, ОПК-7.2) 9. Цифровая сортировка. (ОПК-6.1, ОПК-7.2)
Лабораторная работа №9 Структуры. Функции для работы со структурами.	1. Объявление структуры. (ОПК-7.2) 2. Инициализация структуры. (ОПК-7.2) 3. Создание массива структур. (ОПК-7.2) 4. Создание динамического массива структур. (ОПК-7.2) 5. Передача структур в функцию. (ОПК-7.2) 6. Сортировка функций. Использование компаратора при сортировке. (ОПК-7.2) 7. Битовые поля. (ОПК-6.1, ОПК-7.2)
Лабораторная работа №10 Множества.	1. Способ представления множества на битах целого числа. (ОПК-7.2) 2. Способ представления множества на неупорядоченном массиве. (ОПК-7.2) 3. Способ представления множества на упорядоченном массиве. (ОПК-7.2)
Лабораторная работа №11 Реализация структуры данных «Вектор».	1. Политики расширения динамического массива при его заполнении. (ОПК-7.3) 2. Функции для работы с векторами. (ОПК-7.2)
Лабораторная работа №12 Работа с многомерными массивами.	1. Способы размещения матрицы в динамической памяти. (ОПК-7.2) 2. Схемы размещения. (ОПК-7.3)
Лабораторная работа №13 Алгоритмы обработки строк.	1. Стандартная библиотека <string.h>. Функции. Механизмы работы. (ОПК-7.1) 2. Ввод и вывод строк в языке программирования C. (ОПК-7.2)
Лабораторная работа №14 Введение в C++: ссылки, изменения в управляющих конструкциях.	1. Виды ссылок: ссылки на неконстантные данные, ссылки на константные данные. (ОПК-7.1) 2. <i>Range-based for</i> . (ОПК-7.2) 3. Вариации <i>range-based for</i> . (ОПК-7.2) 4. Ключевое слово <i>auto</i> и <i>decltype</i> . (ОПК-7.2)
Лабораторная работа №15 Структуры в C++.	1. Причины появления в языках программирования (ОПК-7.1) 2. Ограничение доступа к полям структуры. (ОПК-7.1) 3. Константность функций-членов. (ОПК-7.1) 4. Конструкторы. (ОПК-7.1) 5. Инициализация полей структур. (ОПК-7.1)

	6. Деструкторы. (ОПК-7.1)
Лабораторная работа №16 Обзор контейнеров языка C++.	1. Массивы. (ОПК-7.1) 2. Векторы. (ОПК-7.1) 3. Дек. (ОПК-7.1) 4. Односвязный список. (ОПК-7.1) 5. Двусвязный список. (ОПК-7.1) 6. Стек. (ОПК-7.1) 7. Очередь. (ОПК-7.1) 8. Приоритетная очередь. (ОПК-7.1) 9. Отображения. (ОПК-7.1) 10. Множества. (ОПК-7.1) 11. Мультимножества. (ОПК-7.1) 12. Неупорядоченные отображения и множества. (ОПК-7.1)
Лабораторная работа №17 Итераторы. Алгоритмы на паре итераторов.	1. Итераторы. Определение. Предпосылки к созданию. (ОПК-7.1) 2. Алгоритмы на паре итераторов. (ОПК-7.2)
Лабораторная работа №18 Работа с файловыми потоками.	1. Понятия «файл», «поток». (ОПК-7.1) 2. Функции для работы с файлами. (ОПК-7.2)
Лабораторная работа №19 Динамические переменные	1. Динамическое программирование. Этапы решения задач с использованием динамического программирования. (ОПК-6.1, ОПК-7.1) 2. Задачи одномерного динамического программирования. (ОПК-7.1) 3. Задачи многомерного динамического программирования. (ОПК-7.1) 4. Техники кэширования при решении задач на динамическое программирование: мемоизация и табуляция. (ОПК-7.3)
Лабораторная работа №20 Реализация структур данных с использованием ООП.	1. Структуры и классы. (ОПК-7.1) 2. Конструктор без параметров / с параметрами. (ОПК-7.1) 3. Деструктор. (ОПК-7.1) 4. Конструкторы копирования и перемещения. (ОПК-7.1) 5. Перегрузка операций. (ОПК-7.2)

Критерии оценки лабораторной работы: лабораторная работа считается защищенной, если студент выполнил задание к работе полностью и во время устного опроса по работе правильно ответил на заданные преподавателем дополнительные вопросы.

Оценочные материалы

Задания закрытого типа:

1. Какой из перечисленных типов является вещественным типом двойной точности?
 - а) int,
 - б) float,
 - в) double,
 - г) long.
2. Какое ключевое слово используется для создания цикла с фиксированным числом шагов?
 - а) repeat,
 - б) cycle,

в) while,

г) for.

3. Какой размер имеет указатель?

а) байт,

б) слово,

в) машинное слово,

г) двойное машинное слово.

4. Какое максимальное значение принимает беззнаковая однобайтовая переменная?

а) 128,

б) 127,

в) 256,

г) 255.

5. Как называется преобразование значения одного типа в значение другого типа?

а) определение,

б) приведение,

в) присвоение,

г) объявление.

Задания открытого типа:

1) Алгоритм – это ..., выполнение которых преобразует входные данные в выходные в соответствии с условием задачи.

2) ... - это свойство алгоритма получать результат за конечное время.

3) Разбиение алгоритма на части (подпрограммы) подразумевается в ... парадигме программирования.

4) ... называется единица данных, которая выбрана естественной для данной архитектуры процессора.

5) Каждый элемент массива характеризуется:

6) Для получения значения по адресу переменной используется операция

7) Для работы со стандартными потоками ввода/вывода используется библиотека

8) ... - это выражение, результатом вычисления значения которого является 'истина' или 'ложь'.

9) Схема вычисления выражения, называется ..., если в процессе вычисления выражения уже можно понять результат и расчеты прекращаются.

10) Принцип ... нарушается при дублировании кода.

11) Анализ кода, выполняемый IDE без исполнения программ, называется

12) Для работы с динамической памятью в языке C используется библиотека

....

13) Порядок функции временной сложности обменной сортировки равен

14) ... - это вид рекурсии, при которой аргумент рекурсивной функции определяется как результат другого рекурсивного вызова.

15) Объединение нескольких объектов, возможно, различного типа под одним именем называется

5.4. Описание критериев оценивания компетенций и шкалы оценивания

При промежуточной аттестации в форме экзамена и дифференцированного зачёта при защите курсовой работы используется следующая шкала оценивания: 2 – неудовлетворительно, 3 – удовлетворительно, 4 – хорошо, 5 – отлично.

Критериями оценивания достижений показателей являются:

Наименование показателя оценивания результата обучения по дисциплине	Критерий оценивания
Знания	Знание синтаксиса и семантики языка программирования
	Объем освоенного материала
	Полнота ответов на вопросы
	Четкость изложения и интерпретации знаний
Умения	Умение разрабатывать алгоритмы и программы с применением принципов структурного программирования
	Умение проверять решение и анализировать результаты
Навыки	Владение навыками отладки и тестирования программ
	Самостоятельность выполнения отладки и тестирования программ

Оценка преподавателем выставляется интегрально с учётом всех показателей и критериев оценивания.

Оценка сформированности компетенций по показателю Знания.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Знание синтаксиса и семантики языка программирования	Не знает синтаксиса и семантики языка программирования	Знает основные синтаксические конструкции языка программирования	Знает синтаксис языка программирования	Знает синтаксис и семантику языка программирования
Объем освоенного материала	Не знает значительной части материала дисциплины	Знает только основной материал дисциплины, не усвоил его деталей	Знает материал дисциплины в достаточном объеме	Обладает твердым и полным знанием материала дисциплины, владеет дополнительными знаниями
Полнота ответов на вопросы	Не дает ответы на большинство вопросов	Дает неполные ответы на все вопросы	Дает ответы на вопросы, но не все - полные	Дает полные, развернутые ответы на поставленные вопросы
Четкость изложения и интерпретации знаний	Излагает знания без логической последовательности	Излагает знания с нарушениями в логической последовательности	Излагает знания без нарушений в логической последовательности	Излагает знания в логической последовательности, самостоятельно их интерпретируя и анализируя
	Не иллюстрирует изложение поясняющими схемами, рисунками и примерами	Выполняет поясняющие схемы и рисунки небрежно и с ошибками	Выполняет поясняющие рисунки и схемы корректно и понятно	Выполняет поясняющие рисунки и схемы точно и аккуратно, раскрывая полноту усвоенных знаний
	Неверно излагает и интерпретирует знания	Допускает неточности в изложении и интерпретации знаний	Грамотно и по существу излагает знания	Грамотно и точно излагает знания, делает самостоятельные выводы

Оценка сформированности компетенций по показателю Умения.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Умение разрабатывать алгоритмы и программы с применением принципов структурного программирования	Не умеет разрабатывать алгоритмы и программы с применением принципов структурного программирования	Допускает неточности в разработке алгоритмов и программ с применением принципов структурного программирования	Умеет разрабатывать алгоритмы и программы с применением принципов структурного программирования	Безошибочно разрабатывает алгоритмы и программы с применением принципов структурного программирования
Умение проверять решение и анализировать результаты	Не умеет проверять решение и анализировать результаты	Умеет проверять решение некоторых задач	Умеет проверять решение некоторых задач и анализировать результаты	Умеет проверять решение и анализировать результаты

Оценка сформированности компетенций по показателю Навыки.

Критерий	Уровень освоения и оценка			
	2	3	4	5
Владение навыками отладки и тестирования программ	Не владеет навыками отладки и тестирования программ	Не достаточно хорошо владеет навыками отладки и тестирования программ	Владеет навыками отладки и тестирования программ	Профессионально владеет навыками отладки и тестирования программ
Самостоятельность выполнения отладки и тестирования программ	Не может самостоятельно выполнять исследования объектов профессиональной деятельности	Выполняет исследования объектов профессиональной деятельности с посторонней помощью	При выполнении исследования объектов профессиональной деятельности иногда требуется посторонняя помощь	Самостоятельно выполняет исследования объектов профессиональной деятельности

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

6.1. Материально-техническое обеспечение

№	Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы
1.	Учебная аудитория для проведения лекционных занятий	Специализированная мебель. Мультимедийная установка, экран, доски
2.	Учебная аудитория для проведения лабораторных занятий	Специализированная мебель. Компьютеры на базе процессоров Intel или AMD.
3.	Зал электронных ресурсов, здание библиотеки, № 302 Читальный зал учебной литературы, здание библиотеки, № 303	Специализированная мебель. Компьютерная техника, подключенная к сети интернет и имеющая доступ в электронно-образовательную среду

6.2. Лицензионное и свободно распространяемое программное обеспечение

№	Перечень лицензионного программного обеспечения.	Реквизиты подтверждающего документа
1.	Microsoft Windows 10 Корпоративная	(Соглашение Microsoft Open Value Subscription V9221014 Соглашение действительно с 01.11.2020 по 31.10.2023). Договор поставки ПО № 128-21 от 30.10.2021.
2.	Microsoft Office Professional Plus 2016	(Соглашение Microsoft Open Value Subscription V9221014 Соглашение действительно с 01.11.2020 по 31.10.2023). Договор поставки ПО № 128-21 от 30.10.2021.
3.	Kaspersky Endpoint Security «Стандартный Russian Edition»	Сублицензионный договор № 102 от 24.05.2018. Срок действия лицензии до 19.08.2020 Гражданско-правовой Договор (Контракт) № 27782 «Поставка продления права пользования (лицензии) Kaspersky Endpoint Security от 03.06.2020. Срок действия лицензии 19.08.2022г.
4.	Google Chrome	Свободно распространяемое ПО согласно условиям лицензионного соглашения
5.	Среды программирования Dev C++ , CodeBlocks, Visual Studio Community Edition	Свободно распространяемое ПО согласно условиям лицензионного соглашения
6.	ОС Linux	Свободно распространяемое ПО согласно условиям лицензионного соглашения

6.3. Перечень учебных изданий и учебно-методических материалов

1. Керниган, Б. В. Язык программирования С: учебник / Б. В. Керниган, Д. М. Ричи. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 313 с. — ISBN 978-5-4497-0918-9. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/102075.html>
2. Фридман, А. Л. Язык программирования С++: учебное пособие / А. Л. Фридман. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 217 с. — ISBN 978-5-4497-0920-2. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/102076.html>
3. Керниган, Б. Язык программирования Си: Пер. с англ./ Б. Керниган, Д. Ритчи. — 2-е изд., испр.- М.: Вильямс, 2017. — 288 с.
4. Брусенцева В. С. Язык программирования Си. Учебное пособие по дисциплине "Основы программирования" для студентов направлений подготовки 09.01.01 – Информатика и вычислительная техника и 09.01.04 – Программная инженерия, Белгород: Изд-во БГТУ им. В.Г. Шухова, 2018. – 64 с
5. Брусенцева В. С. Методические указания к выполнению курсовых работ по дисциплине «Основы программирования» для студентов, направлений бакалавриата 09.03.01 – Информатика и вычислительная техника, 09.03.04 – Программная инженерия. Белгород: Изд-во БГТУ им. В.Г. Шухова, 2014. – 11 с
6. Методы отладки и тестирования программных продуктов: учебное пособие к проведению исследовательских лабораторных работ / составители Е. О. Ткачук. — Ростов-на-Дону: Северо-Кавказский филиал Московского технического университета связи и информатики, 2018. — 102 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/89519.html>
7. Плаксин М.А. Тестирование и отладка программ для профессионалов будущих и настоящих [Электронный ресурс]/ Плаксин М.А.— Электрон. текстовые данные. — М.: БИНОМ. Лаборатория знаний, 2013.— 167 с.
<http://www.iprbookshop.ru/20704.html>
8. Лоспинозо Д. С++ для профи. — СПб.: Питер, 2021. — 816 с.: ил. ISBN 978-5-4461-1730-7.
9. Липпман С., Лажойе Ж. Язык программирования С++. Полное руководство / Пер. с англ. – 2-е изд. – Саратов: Профобразование, 2019. – 1104 с.: ил. ISBN 978-5-4488-0136-5.
10. Мартин Р. Чистый код: создание, анализ, рефакторинг. – СПб.: Питер, 2019. — 464с. ISBN 978-5-4461-0960-9.
11. Совершенный код. Мастер-класс / Пер. с англ. — М. : Издательство «Русская редакция», 2010. — 896 стр. : ил. ISBN 978-5-7502-0064-1.

6.4. Перечень интернет-ресурсов, профессиональных баз данных, информационно-справочных систем

1. Электронная библиотека (на базе ЭБС «БиблиоТех») — Режим доступа: <http://ntb.bstu.ru>.
2. Электронно-библиотечная система IPRbooks — Режим доступа: <http://www.iprbookshop.ru>.
3. Электронно-библиотечная система «Университетская библиотека ONLINE» — Режим доступа: <http://www.biblioclub.ru/>