

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета информационных
технологий и компьютерной безопасности



/ П.Ю. Гусев /
И.О. Фамилия

«31» августа 2021 г.

**РАБОЧАЯ ПРОГРАММА
дисциплины**

«Программирование на языках высокого уровня»

Направление подготовки 09.03.01 Информатика и вычислительная техника

Профиль Программное обеспечение автоматизированных систем

Квалификация выпускника бакалавр

Нормативный период обучения 4 года / 4 года и 11 м.

Форма обучения очная / заочная

Год начала подготовки 2019

Автор программы

А.Н. Юров Юров А.Н.

Заведующий кафедрой
Компьютерных
интеллектуальных
технологий проектирования

М.И. Чижов Чижов М.И.

Руководитель ОПОП

П.Ю. Гусев Гусев П.Ю.

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью изучения дисциплины является изучение и применение языка программирования для реализации алгоритмов, обеспечение фундаментальной математической подготовки, адаптированной к решению задач на ЭВМ, позволяющей будущим специалистам ориентироваться в научно-технической информации, использовать численные методы и представление информации для создания программных решений в тех областях и сферах деятельности, в которых они будут трудиться.

Изучение дисциплины должно способствовать формированию у студентов основ научного мышления, в том числе: владение основными методами, способами и средствами управления информацией, создание алгоритмов и освоение методик при разработке программ для решения практических задач.

1.2. Задачи освоения дисциплины

- знакомство с направлениями развития программного обеспечения вычислительной техники;
- знакомство с методами структурного и объектно-ориентированного программирования как наиболее распространенными и эффективными методами разработки программных продуктов;
- разработка алгоритмов на основе структурного и объектно-ориентированного подхода;
- ознакомление студентов с тенденцией развития программного обеспечения и указание перспективных направлений при решении практических задач;
- освоение языков программирования высокого уровня, а также стандартов кодирования, спецификаций и последующих решений на их основе;
- создание практической базы для изучения других учебных дисциплин, таких, как "Среды визуального программирования", "Компьютерная графика" и ряда других.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на языках высокого уровня» относится к дисциплинам базовой части блока Б1.

Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

- Объектно-ориентированное программирование;
- Среды визуального программирования;
- Компьютерная графика

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование на языках высокого уровня» направлен на формирование следующих компетенций:

ОПК-8 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-9 - Способен осваивать методики использования программных средств для решения практических задач.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-8	знать: - синтаксис, ключевые слова, операторы, конструкции, типы данных, функциональные элементы языка программирования высокого уровня; - производственные среды разработки, понятие и функциональные возможности отладочных средств, тестирования и сборки приложений; - основные положения теории алгоритмизации; - основные принципы конструирования алгоритмов.
	уметь создавать программные решения для операционных систем 32 и 64-х разрядной архитектуры, работающих на ПК и мобильных платформах
	владеть: - навыками работы в различных средах программирования; - навыками работы в операционных системах Windows и Linux, готовить инсталляционные пакеты разработанных программ для указанных ОС.
ОПК-9	знать фундаментальные алгоритмы и методики, которые могут быть востребованы для решения практических задач
	уметь и проводить сборку ПО, а также различных элементов информационных систем из готовых компонентов
	владеть программным обеспечением, обеспечивающим разработку программных систем для практических нужд.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программирование на языках высокого уровня» составляет 5 з.е.

**Распределение трудоемкости дисциплины по видам занятий
очная форма обучения**

Виды учебной работы	Всего часов	Семестр
		ы 1
Аудиторные занятия (всего)	90	90
В том числе:		
Лекции	36	36
Лабораторные работы (ЛР)	54	54
Самостоятельная работа	54	54
Курсовой проект	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость: академические часы зач.ед.	180 5	180 5

заочная форма обучения

Виды учебной работы	Всего часов	Семестр
		ы 1
Аудиторные занятия (всего)	18	18
В том числе:		
Лекции	6	6
Лабораторные работы (ЛР)	12	12
Самостоятельная работа	153	153
Курсовой проект	+	+
Часы на контроль	9	9
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость: академические часы зач.ед.	180 5	180 5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

**5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий
очная форма обучения**

№ п/п	Наименование темы	Содержание раздела	Лек ц	Лаб. зан.	СРС	Всего, час
-------	-------------------	--------------------	----------	--------------	-----	---------------

1	Введение в разработку на ЯВУ. Производственные среды для создания программ. Операции языка программирования	<p>Обзор функциональных возможностей языка программирования. Первая программа с описанием. Компиляторы, интерпретаторы, отладчики, производственные среды.</p> <p>Переменные и константы. Типы данных. Математические, логические, битовые операторы, операторы присвоения и перечисления. Типовые конструкции и области применения.</p> <p>Простые и составные условные конструкции. Проверка условного выражения, операции ветвления, многовариантное построение условий, предопределенные циклические конструкции. Включение условий в повторяющийся процесс при разработке программ</p>	6	10	8	24
2	Условные и циклические конструкции на ЯВУ	<p>Простые и составные условные конструкции. Проверка условного выражения, операции ветвления, многовариантное построение условий, предопределенные циклические конструкции. Включение условий в повторяющийся процесс при разработке программ. Конструкции циклов с предусловием и пост решением, использованием конструкции циклов с известными параметрами. Расширение функциональных возможностей благодаря использованию новых стандартов для ЯВУ</p>	6	10	8	24
3	Массивы данных, функции, указатели и ссылки	<p>Основные положения. Инициализация элементов при объявлении. Доступ к элементам массива. Многомерные массивы. Введение. Объявление и вызов функций в программе. Необязательные аргументы функции. Рекурсивный вызов функций. Передача массивов в функции. Базовые концепции при использовании указателей и ссылок. Использование указателя на символьную строку. Указатели и массивы. Проблемные места при создании программ с использованием указателей и выявление ошибок.</p>	6	10	8	24
4	Работа со строками Структуры, перечисления и объединения Контейнерные типы данных, итераторы Шаблонные функции, исключения, директивы предпроцессора	<p>Объявление строковых типов, основные операции, преобразование строк в целые и вещественные диапазоны, возможности стандартной библиотеки по работе со строками</p> <p>Создание и использование структур, объединений и перечислений. Работа с битовыми полями структур данных. Применение структур, позволяющих динамически хранить данные, добавлять, удалять и вести поиск по заданному значению, применение итераторов</p> <p>Директива подключения заголовочных файлов стандартных библиотек, определение пространства имен, конструкции подстановок, операции предусловий, подключение специфических особенностей компиляторов. Шаблоны и абстрактные данные.</p>	6	8	10	24
5	Введение в классы, конструкторы и деструкторы Области доступа public, private и protected Простое и	<p>Представление об объектах и объектно-ориентированном программировании. Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов.</p> <p>Конструкторы классов. Создание простого конструктора. Стандартные и</p>	6	8	10	24

	множественное наследование	дополнительные конструкторы. Деструкторы. Вызов методов вне класса. Области видимости данных для классов, защищенные области, разрешение и запрет доступа к данным Простое наследование классов. Множественное наследование и абстрактные классы. Использование в программах виртуальных и чисто-виртуальных методов в классах.				
6	Функторы и лямбда-выражения Разработка приложений с графическим интерфейсом Обработка событий в графическом приложении	Использование параметрических модулей в разработке приложений, применение специального синтаксиса для определения функциональных объектов Оконный режим работы приложений, основные классы для создания интерфейсов программ События от основных элементов управления приложения, взаимосвязь событий между элементами, примеры использования	6	8	10	24
Итого			36	54	54	144

заочная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лек ц	Лаб. зан.	СРС	Всего, час
1	Условные и циклические конструкции на ЯВУ	Простые и составные условные конструкции. Проверка условного выражения, операции ветвления, многовариантное построение условий, предопределенные циклические конструкции. Включение условий в повторяющийся процесс при разработке программ. Конструкции циклов с предусловием и пост решением, использованием конструкции циклов с известными параметрами. Расширение функциональных возможностей благодаря использованию новых стандартов для ЯВУ	2	2	24	28
2	Массивы данных, функции, указатели и ссылки	Основные положения. Инициализация элементов при объявлении. Доступ к элементам массива. Многомерные массивы. Введение. Объявление и вызов функций в программе. Необязательные аргументы функции. Рекурсивный вызов функций. Передача массивов в функции. Базовые концепции при использовании указателей и ссылок. Использование указателя на символьную строку. Указатели и массивы. Проблемные места при создании программ с использованием указателей и выявление ошибок.	2	2	26	30
3	Работа со строками Структуры, перечисления и объединения Контейнерные типы данных, итераторы Шаблонные функции, исключения, директивы препроцессора	Объявление строковых типов, основные операции, преобразование строк в целые и вещественные диапазоны, возможности стандартной библиотеки по работе со строками Создание и использование структур, объединений и перечислений. Работа с битовыми полями структур данных. Применение структур, позволяющих динамически хранить данные, добавлять, удалять и вести поиск по заданному значению, применение итераторов Директива подключения заголовочных файлов стандартных библиотек, определение пространства имен, конструкции подстановок, операции предусловий, подключение специфических особенностей компиляторов. Шаблоны и абстрактные данные.	2	2	26	30

4	Введение в классы, конструкторы и деструкторы Области доступа public, private и protected Простое и множественное наследование	Представление об объектах и объектно-ориентированном программировании. Способы описания классов. Создание объектов. Обращение к атрибутам и методам объектов. Конструкторы классов. Создание простого конструктора. Стандартные и дополнительные конструкторы. Деструкторы. Вызов методов вне класса. Области видимости данных для классов, защищенные области, разрешение и запрет доступа к данным Простое наследование классов. Множественное наследование и абстрактные классы. Использование в программах виртуальных и чисто-виртуальных методов в классах.	-	2	26	28
5	Функторы и лямбда-выражения Разработка приложений с графическим интерфейсом Обработка событий в графическом приложении	Использование параметрических модулей в разработке приложений, применение специального синтаксиса для определения функциональных объектов Оконный режим работы приложений, основные классы для создания интерфейсов программ События от основных элементов управления приложения, взаимосвязь событий между элементами, примеры использования	-	2	26	28
6	Введение в разработку на ЯВУ. Производственные среды для создания программ. Операции языка программирования	Обзор функциональных возможностей языка программирования. Первая программа с описанием. Компиляторы, интерпретаторы, отладчики, производственные среды. Переменные и константы. Типы данных. Математические, логические, битовые операторы, операторы присвоения и перечисления. Типовые конструкции и области применения. Простые и составные условные конструкции. Проверка условного выражения, операции ветвления, многовариантное построение условий, предопределенные циклические конструкции. Включение условий в повторяющийся процесс при разработке программ	-	2	25	27
Итого			6	12	153	171

5.2 Перечень лабораторных работ

Очная форма обучения

Неделя семестра	Наименование лабораторной работы	Объем часов	Виды контроля
1/1	Интегрированные среды и средства разработки на языке программирования высокого уровня	4	Отчет и защита
1/2-1/3	Базовые конструкции и ключевые слова языка программирования высокого уровня	8	Отчет и защита
1/4-1/6	Структурный подход к созданию приложений	8	Отчет и защита
1/7-1/8	Дополнительные возможности языков высокого уровня	8	Отчет и защита
1/9-1/10	Введение в стандартную библиотеку	6	Отчет и защита
1/11-1/17	Введение в графическую разработку приложений	16	Отчет и

			защита
1/18	Решения для подготовки установочных пакетов	4	Отчет и защита
Итого часов:		54	

Заочная форма обучения

Неделя семестра	Наименование лабораторной работы	Объем часов	Виды контроля
	Интегрированные среды и средства разработки на языке программирования высокого уровня	2	Отчет и защита
	Базовые конструкции и ключевые слова языка программирования высокого уровня	2	Отчет и защита
	Структурный подход к созданию приложений	2	Отчет и защита
	Дополнительные возможности языков высокого уровня	2	Отчет и защита
	Введение в стандартную библиотеку	2	Отчет и защита
	Введение в графическую разработку приложений	2	Отчет и защита
		12	

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 1 семестре для очной формы обучения, в 1 семестре для заочной формы обучения.

Примерная тематика курсового проекта: «Расчетные задачи с применением графического интерфейса»

Задачи, решаемые при выполнении курсового проекта:

- расчетные задачи;
- простейшие геометрические преобразования;
- построение деревьев, введение в графы;
- сортировка данных;
- программирование математических выражений и структур;
- численные методы;
- вывод графических объектов на экран (графики функций, простейшие геометрические объекты);
- файловые операции;
- работа со строковыми и табличными данными;
- создание многооконных кроссплатформенных приложений;
- интеграция со сторонними приложениями.

Курсовой проект включает в себя графическую часть и расчетно-пояснительную записку.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-8	знать синтаксис, ключевые слова, операторы, конструкции, типы данных, функциональные элементы языка программирования высокого уровня; -производственные среды разработки, понятие и функциональные возможности отладочных средств, тестирования и сборки приложений; - основные положения теории алгоритмизации; - основные принципы конструирования алгоритмов.	текст	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь создавать программные решения для операционных систем 32 и 64-х разрядной архитектуры, работающих на ПК и мобильных платформах	текст	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть -навыками работы в различных средах программирования; -навыками работы в операционных системах Windows и Linux, готовить инсталляционные пакеты разработанных программ для указанных ОС.	текст	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-9	знать	текст	Выполнение	Невыполнение

	фундаментальные алгоритмы и методики, которые могут быть востребованы для решения практических задач		работ в срок, предусмотренный в рабочих программах	работ в срок, предусмотренный в рабочих программах
	уметь проводить сборку ПО, а также различных элементов информационных систем из готовых компонентов	текст	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть программным обеспечением, обеспечивающем разработку программных систем для практических нужд.	текст	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 1 семестре для очной формы обучения, 1 семестре для заочной формы обучения по четырехбалльной системе:

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-8	знать синтаксис, ключевые слова, операторы, конструкции, типы данных, функциональные элементы языка программирования высокого уровня; -производственные среды разработки, понятие и функциональные возможности отладочных средств, тестирования и сборки приложений; - основные положения теории алгоритмизации; - основные принципы конструирования алгоритмов.	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов

	уметь создавать программные решения для операционных систем 32 и 64-х разрядной архитектуры, работающих на ПК и мобильных платформах	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть -навыками работы в различных средах программирования; -навыками работы в операционных системах Windows и Linux, готовить инсталляционные пакеты разработанных программ для указанных ОС.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-9	знать фундаментальные алгоритмы и методики, которые могут быть востребованы для решения практических задач	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	уметь проводить сборку ПО, а также различных элементов информационных систем из готовых компонентов	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть программным обеспечением, обеспечивающем разработку программных систем для практических нужд.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Тип данных int представляет собой:

а) целочисленный тип

- б) строковый тип
- в) вещественный тип
- г) логический тип

2. Ключевое слово for:

- а) означает использование в программе конструкции повторения
- б) означает использование в программе конструкции с условием
- в) слово не является ключевым

3. bool flag=false:

- а) объявлена и задана строка
- б) объявлена и задана логическая переменная
- в) объявлена и задана логическая константа

4. main():

- а) точка входа в приложение
- б) заголовок функции
- в) утверждения в предыдущих ответах верны

5. Применение условий в составе программы требуется:

- а) когда необходимо сравнить несколько значений
- б) когда надо присвоить результат переменной
- в) когда программа запускается

6. Указатель - это:

- а) адрес в памяти
- б) ключевое слово
- в) переменная, заданная именем

7. Директивы предпроцессора:

- а) расширяют возможности приложения
- б) обеспечивают особые варианты сборки приложения
- в) все ответы правильные

8. Интегрированная среда разработки содержит:

- а) средства отладки
- б) средства авто дополнения кода
- в) средства анализа памяти
- г) все ответы правильные

9. Структуры позволяют:

- а) группировать данные по заданному признаку
- б) используются для хранения разнотипных данных
- в) обеспечивать взаимосвязь между записями

г) все вышеперечисленное

10. Вещественные типы данных float и double отличаются:

- а) размером хранения данных
- б) это одно и то же
- в) не используются при создании программ

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Дана непустая последовательность неотрицательных целых чисел, оканчивающаяся отрицательным числом. Найти среднее арифметическое всех чисел последовательности (без учета отрицательного числа).

2. Среди чисел 1, 4, 9, 16, 25, ... найти первое число, большее n .

3. Дана непустая последовательность целых чисел, оканчивающаяся нулем.

Найти:

а) сумму всех чисел последовательности, больших числа x ;

б) количество всех четных чисел последовательности

4. Дана последовательность ненулевых целых чисел. Определить, сколько раз в этой последовательности меняется знак. Например, в последовательности 10, -4, 12, 56, -4 знак меняется 3 раза.

5. Найти количество делителей каждого из целых чисел от 120 до 140.

6. Дано слово. Получить и вывести на экран буквосочетание, состоящее из его третьего и последнего символа.

7. Дано слово. Получить его часть, образованную второй, третьей и четвертой буквами

8. Найти периметр треугольника, заданного координатами своих вершин. (Определить функцию для расчета длины отрезка по координатам его вершин.)

9. Дан двумерный массив. Найти:

а) сумму элементов, расположенных в левом верхнем и правом нижнем углах;

б) среднее арифметическое элементов, расположенных в четырех углах

10. Имеется файл, в котором записаны 12 вещественных чисел. Переписать все числа файла в массив в том же порядке.

7.2.3 Примерный перечень заданий для решения прикладных задач

№	Задача
1	Разработка текстового редактора
2	Разработка табличного редактора
3	Разработка архиватора
4	Разработка файлового менеджера
5	Разработка векторного редактора

6	Разработка графического редактора
7	Разработка приложения по просмотру графических форматов
8	Разработка приложения по просмотру 3D моделей
9	Разработка приложения по анализу аппаратной части
10	Разработка приложения по тестированию оборудования ЭВМ

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень заданий для решения прикладных задач

1. Структура и синтаксис простейшего консольного приложения.
2. Консольный ввод-вывод, использование функций из математической библиотеки.
3. Типы данных. Пример консольного приложения
4. Беззнаковые целочисленные типы данных, представление в памяти.
5. Вещественные типы данных, их представление в памяти.
6. Символьный тип данных, представление символов в памяти.
7. Преобразование типов данных.
8. Арифметические и логические операции. Приоритеты операций.
9. Битовые операции.
10. Переменные-константы.
11. Условный оператор if ... else. Примеры
12. Оператор выбора switch.
13. Оператор цикла for.
14. Оператор цикла do-while.
15. Оператор цикла while.
16. Вспомогательные операторы: break, continue.
17. Строковые переменные.
18. Стандартные функции для работы со строками.
19. Операция присваивания и ее разновидности.
20. Префиксное и постфиксное использование операций.
21. Понятие массивов. Одномерные массивы.
22. Понятие массивов. Многомерные массивы.
23. Указатели. Описание указателей.
24. Арифметические операции с указателями.
25. Ссылки.
26. Операторы выделения и освобождения памяти для переменной и массива.
27. Функции. Область видимости в функциях.
28. Передача переменных и массивов в функцию и возврат значений.
29. Использование функций с параметрами по умолчанию.
30. Перегруженные функции и функции-шаблоны.
31. Рекурсивные функции.
32. Понятие структур. Описание структур.
33. Перечисления. Примеры использования.
34. Контейнерные типы данных: стеки и очереди
35. Контейнерные типы данных: списки

- 36. Контейнерные типы данных: вектора
- 37. Применение исключений в разработке программ
- 38. Заголовочные файлы.
- 39. Понятие препроцессора. Назначение директив препроцессоров.
- 40. Макроопределения и макроподстановки.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.)

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Введение в разработку на ЯВУ. Производственные среды для создания программ. Операции языка программирования	ОПК-8, ОПК-9	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
2	Условные и циклические конструкции на ЯВУ	ОПК-8, ОПК-9	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
3	Массивы данных, функции, указатели и ссылки	ОПК-8, ОПК-9	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
4	Работа со строками Структуры, перечисления и объединения Контейнерные типы данных, итераторы Шаблонные функции, исключения, директивы	ОПК-8, ОПК-9	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....

	предпроцессора		
5	Введение в классы, конструкторы и деструкторы Области доступа public, private и protected Простое и множественное наследование	ОПК-8, ОПК-9	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....
6	Функторы и лямда-выражения Разработка приложений с графическим интерфейсом Обработка событий в графическом приложении	ОПК-8, ОПК-9	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту....

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

№ п/п	Фонд учебной и учебно-методической литературы с учетом качества содержания литературы (наличие грифа)				Книгообес-печенность
	Тип носителя (печ./эл.)	Наименование	Автор, название, место издания, издательство, год издания учебной и учебно-методической литературы, URL (для фонда ЭБС)	Кол-во экз. / точек доступа	
1	3	4	5	6	7

1	печ.	Программирование на языке высокого уровня. Программирование на языке C++: учебное пособие	Л.Г. Гагарина –ИД: ФОРУМ, 2012		0,5
2	печ.	Язык программирования С: учеб. пособие	Брайан У.К., Деннис М.Р –М.: Вильямс, 2015		0,5
3	печ.	Язык программирования С. Лекции и упражнения: учеб. пособие	С. Прата –М.: Вильямс, 2016		0,6
4	печ.	С/С++. Процедурное и объектно-ориентированное программирование: учебник для вузов	Т.А Павловская. –СПб.: Питер, 2014		0,3

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

8.2. 1	Методические указания к выполнению лабораторных работ представлены на сайте: http://calsvstu.ru/index.php/project/uchebnaya-literatura
8.2. 2	Компьютерные лабораторные работы: <ul style="list-style-type: none"> – Разработка консольных приложений – Разработка графических приложений – Отладочный процесс приложений – Развертывание приложений – Сборка приложений для ряда операционных систем
8.2. 3	Мультимедийные видеофрагменты: <ul style="list-style-type: none"> – Среды разработки ПО – Отладочные средства ПО – Базовые конструкции языка – Синтаксис и операции – Условные конструкции и циклы – Массивы и функции – Структуры – Введение в классы и объекты – Подготовка графических приложений – Работа с файлами – События в приложении – Работа с клавиатурой и мышью – Поиск зависимостей в приложении – Подготовка установочных пакетов
8.2. 4	Мультимедийные лекционные демонстрации: <ul style="list-style-type: none"> – Сборка и настройка персональных компьютеров – Работа с командной строкой в операционных системах

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

9.1	Специализированная лекционная аудитория , оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой
9.2	Учебные лаборатории: <ul style="list-style-type: none"> – “Лаборатория интеллектуальных систем проектирования” – “Лаборатория компьютерного моделирования и дизайна” – “Лаборатория телекоммуникационных и сетевых технологий” – “Интернет-лаборатория ” – ”Учебный центр ВГТУ, академия Софтлайн, сетевой академии CISCO”
9.3	Дисплейный класс , оснащенный компьютерными программами для проведения лабораторного практикума
9.4	Кабинеты , оборудованные проекторами и интерактивными досками
9.5	Натурные лекционные демонстрации: <ul style="list-style-type: none"> – Порядок работы в IDE средах разработки – Работа с отладочными средствами – Профилирование, оптимизация и поиск зависимостей в приложении.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Программирование на языках высокого уровня» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если

	самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none"> - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.