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 /
подпись 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 Вычислительные машины, комплексы, системы и сети

Квалификация выпускника бакалавр

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

Форма обучения очная / заочная

Год начала подготовки 2021

Автор программы _____ /А.Н. Юров/

Заведующий кафедрой Компьютерных интеллектуальных технологий проектирования _____ /М.И. Чижов/

Руководитель ОПОП _____ / В.Ф. Барабанов /

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 обязательной части блока Б1.

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, обеспечивающем 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	Семестры
		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	Семестры 1
Аудиторные занятия (всего)	16	16
В том числе:		
Лекции	4	4
Лабораторные работы (ЛР)	12	12
Самостоятельная работа	155	155
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	Лекц	Лаб. зан.	СРС	Всего, час
1	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	6	10	8	24
2	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	6	10	8	24
3	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.	6	10	8	24

		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.				
4	Работа со строками Структуры, перечисления и объединения Контейнерные типы данных, итераторы Шаблонные функции, исключения, директивы предпроцессора	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.	6	8	10	24
5	Введение в классы, конструкторы и деструкторы Области доступа public, private и protected Простое и множественное наследование	Представление об объектах и объектно-ориентировочном 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.	6	8	10	24
6	Функторы и лямда-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	Лекц	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	26	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	2	2	26	30

		при использовании указателей и ссылок. Использование указателя на символьную строку. Указатели и массивы. Проблемные места при создании программ с использованием указателей и выявление ошибок.				
3	Работа со строками Структуры, перечисления и объединения Контейнерные типы данных, итераторы Шаблонные функции, исключения, директивы предпроцессора	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	26	28
4	Введение в классы, конструкторы и деструкторы Области доступа public, private и protected Простое и множественное наследование	Представление об объектах и объектно-ориентировочном 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 лямда-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	**4**	**12**	**155**	**171**

5.2 Перечень лабораторных работ

1. Интегрированные среды и средства разработки на языке программирования высокого уровня
2. Базовые конструкции и ключевые слова языка программирования высокого уровня
3. Структурный подход к созданию приложений

4. Дополнительные возможности языков высокого уровня
5. Введение в стандартную библиотеку
6. Введение в графическую разработку приложений
7. Решения для подготовки установочных пакетов

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 включат в себя программную часть и расчетно-пояснительную записку.

Темы курсовых проектов согласно заданному варианту:

1. Решение алгоритмических задач с условиями.
2. Решение алгоритмических задач с конструкциями повторений.
3. Решение алгоритмических задач с функциями.
4. Решение алгоритмических задач с файлами.
5. Разработка текстового редактора.
6. Разработка табличного редактора.
7. Разработка архиватора.
8. Разработка файлового менеджера.
9. Разработка векторного редактора.
10. Разработка графического редактора.
11. Разработка приложения по просмотру графических форматов.
12. Разработка приложения по просмотру 3D моделей.
13. Разработка приложения по анализу аппаратной части.
14. Разработка приложения по тестированию оборудования ЭВМ.
15. Разработка инженерного калькулятора.

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».

Компе-тенция	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.	Выполнение лабораторных работ	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	Выполнение лабораторных работ	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.	Выполнение лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-9	знать	Выполнение лабораторных	Выполнение работ в	Невыполнение

	фундаментальные алгоритмы и методики, которые могут быть востребованы для решения практических задач	работ	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	Выполнение лабораторных работ	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.	Выполнение лабораторных работ	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».

Компе-тенция	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	Решение стандартных	Задачи решены в	Продемонстр ирован	Продемонстр ирован верный	Задачи не решены

	решения для операционных систем 32 и 64-х разрядной архитектуры, работающих на ПК и мобильных платформах	практических задач	полном объеме и получены верные ответы	верный ход решения всех, но не получен верный ответ во всех задачах	ход решения в большинстве задач	
	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.Класс - это:

1.любой тип данных, определяемый пользователем;

2.описание объекта, который определяется пользователем и включает данные, а также методы по работе с ними;

3.структура, для которой в программе имеются методы работы с нею

Верный ответ: 2

2.Членами класса могут быть?
1.как переменные, так и методы, могут быть объявлены как private, protected, public;
2.только переменные, объявленные как private;
3.только методы, объявленные как private;
4.только переменные и методы, объявленные как private;
5.только переменные и методы, объявленные как public;

Верный ответ: 1

3.Что называется конструктором?
1.метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса);
2.метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса;
3.метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса;
4.метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса;

Верный ответ: 4

4.Объект — это?
1. переменная, содержащая указатель на класс;
2. экземпляр класса;
3. класс, который содержит в себе данные и методы их обработки;

Верный ответ: 2

5. Деструктором называется?
1.метод, который уничтожает объект;
2.метод, который удаляет объект;
3.метод, который освобождает память, занимаемую объектом;
4.системная функция, которая освобождает память, занимаемую объектом;

Верный ответ: 3

6. Выберите правильное утверждение?

1.деструктор - это метод класса, применяемый для удаления объекта;
2.деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом;
3.деструктор - это отдельная функция головной программы,

применяемая для освобождения памяти, занимаемой объектом;
4.деструктор наследуется, но должен быть перегружен;

Верный ответ: 2

7. Что называется наследованием?
1. это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы;
2. это механизм переопределения методов базового класса;
3. это механизм, посредством которого производный класс получает все поля базового класса;
4. это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить;

Верный ответ: 1

8. Возможность и способ обращения производного класса к элементам базового определяется?
1.ключами доступа: private, public, protected в теле производного класса;
2.только ключом доступа protected в заголовке объявления производного класса;
3.ключами доступа: private, public, protected в заголовке объявления производного класса;
4.ключами доступа: private, public, protected в теле базового класса;

Верный ответ: 3

9. Дружественная функция - это?
1.функция другого класса, среди аргументов которой есть элементы данного класса;
2.функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса;
3.функция, являющаяся членом класса и объявленная с атрибутом friend;
4.функция, которая в другом классе объявлена как дружественная данному;

Верный ответ: 2

10.Для доступа к элементам объекта используются:
1. при обращении через имя объекта – точка, при обращении через указатель – операция «->»;
2. при обращении через имя объекта – два двоеточия, при обращении

через указатель – операция «точка»;
3. при обращении через имя объекта – точка, при обращении через указатель – два двоеточия;
4. при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»;

Верный ответ: 1

7.2.2 Примерный перечень заданий для решения стандартных задач

1.
```
#include <iostream>
#include <algorithm>
#include <array>
#include <iterator>
int main()
{
    auto myArray = std::to_array({ 6,0,6,3,2,9 });
    std::sort(myArray.begin(), myArray.end());

        for (const auto& i : myArray)
        {
            std::cout << i << "\t";
        }
}
```

Программный код, приведенный выше, позволяет:

1.Подготовить целочисленный массив и вывести элементы массива на экран

2.Распечатать элементы массива абстрактного типа с заданного элемента

*3. Согласно новым стандартам с префиксом сборки C++latest, создать целочисленный массив, отсортировать его элементы и вывести на экран в порядке возрастания

Верный ответ: 3

2.
```
#include<iostream>
int main()
{
    auto myValue{5};
    auto *pnt = &myValue;
    std::cout<<*pnt;
    return 0;
}
```

В консольном приложении ответом будет один из вариантов:

1.0x0567FF1

*2.5

3.Произвольный адрес, который предоставила ОС для текущего сеанса

работы приложения.

4.Произвольное значение из диапазона целых чисел
Верный ответ: 2

3.
```
#include<iostream>
int main()
{
    size_t a{5},b{10};
    a^=b^=a^=b;
    std::cout<<a<<"\t"<<b;
    return 0;
}
```
В консольном приложении ответом будет один из вариантов:
1.a=5;b=10;
2.a=50;b=10;
3.a=10000;b=10;
*4.a=10;b=5;
Верный ответ: 4

4.
```
#include<iostream>
int main()
{
    if(size_t a{5};a>2)
        std::cout<<(a<<2);
    return 0;
}
```
В консольном приложении ответом будет один из вариантов:
1.a=5;
2.a=10;
*3.a=20;
4. нет ответа, так как условие содержит ошибку в синтаксисе
Верный ответ: 3

5.
```
#include<iostream>
int main()
{
  size_t a{0b11};
        std::cout<<a;
    return 0;
}
```
В консольном приложении ответом будет один из вариантов:
1.a=11;
2.a=3;
3.a=2833
4. нет ответа, так как условие содержит ошибку в синтаксисе

Верный ответ: 2

6. Приведен следующий фрагмент консольного приложения:

```
class MyClass
{
    int m_age;
    MyClass(int age):m_age(age){
    }
};
int main()
{
  MyClass(21);
  return 0;
}
```

Ошибка в программном коде вызвана:

1. Отсутствуют данные в теле конструктора

*2. Для конструктора не задан спецификатор доступа public

3. Неправильное присвоение данных в целочисленную переменную

4. Нельзя передавать в конструктор значение в формате десятичного числа

Верный ответ: 2

7. Приведен следующий фрагмент консольного приложения:

```
#include <iostream>
class A
{
    public:
    void Test(){
      std::cout<<"test method A";
    }
};
class B: public A
{
    public:
    void Test(){
      std::cout<<"test method B";
    }
};
int main()
{
  A obj1;
  B obj2;
  obj1=obj2;
  obj1.Test();
  return 0;
```

```
}
```
Какой метод Test() будет вызван:
*1. Метод класса A
2. Метод класса B
3. Программа содержит ошибочный программный код
Верный ответ: 1

```
8. class A
{
    public:
    A(){}
    ~A(){}
};
```
В классе объявлены:
*1. Конструктор и деструктор
2. Два пустых конструктора
3. Два абстрактных метода
Верный ответ: 1

```
9. #include <iostream>
class A
{
    int m_value;
    public:
    A(int value):m_value{value} {}
    const int& getValue()const{
          return m_value;
    }
};
int main()
{
   A obj(15);
   std::cout<<obj.getValue();
   return 0;
}
```
В классе метод объявлен константным:
1. Чтобы вернуть из метода целочисленное значение
2. Чтобы не создавать копию объекта
*3. Чтобы не копировать и не изменять целочисленный объект класса
Верный ответ: 3

```
10. #include <iostream>
class A
{
    public:
```

```
    A(size_t value)
    {
      value+=++value;
      std::cout<<value;
    }
};
int main()
{
  A obj(10);
  return 0;
}
```

В классе в конструкторе будет выведено:

1. 21
*2. 22
3. 11
4. inf

Верный ответ: 2

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Функции. Область видимости в функциях.
2. Передача переменных и массивов в функцию и возврат значений.
3. Использование функций с параметрами по умолчанию.
4. Перегруженные функции и функции-шаблоны.
5. Рекурсивные функции.
6. Понятие структур. Описание структур.
7. Перечисления. Примеры использования.
8. Контейнерные типы данных: стеки и очереди
9. Контейнерные типы данных: списки
10. Контейнерные типы данных: вектора
11. Применение исключений в разработке программ
12. Заголовочные файлы.
13. Понятие препроцессора. Назначение директив препроцессоров.
14. Макроопределения и макроподстановки.

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень заданий для подготовки к экзамену

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.

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, защита лабораторных работ, требования к курсовому проекту
2	Условные и циклические конструкции на ЯВУ	ОПК-8, ОПК-9	Тест, защита лабораторных работ, требования к курсовому проекту

3	Массивы данных, функции, указатели и ссылки	ОПК-8, ОПК-9	Тест, защита лабораторных работ, требования к курсовому проекту
4	Работа со строками Структуры, перечисления и объединения Контейнерные типы данных, итераторы Шаблонные функции, исключения, директивы предпроцессора	ОПК-8, ОПК-9	Тест, защита лабораторных работ, требования к курсовому проекту
5	Введение в классы, конструкторы и деструкторы Области доступа public, private и protected Простое и множественное наследование	ОПК-8, ОПК-9	Тест, защита лабораторных работ, требования к курсовому проекту
6	Функторы и лямда-выражения Разработка приложений с графическим интерфейсом Обработка событий в графическом приложении	ОПК-8, ОПК-9	Тест, защита лабораторных работ, 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 методике 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 методике 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 методике 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

1. Ковалевская, Е. В. Методы программирования : учебное пособие / Е. В. Ковалевская, Н. В. Комлева. — Москва : Евразийский открытый институт, 2011. — 320 c. — ISBN 978-5-374-00356-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/10784.html

2. Фарафонов, А. С. Программирование на языке высокого уровня : методические указания к проведению лабораторных работ по курсу «Программирование» / А. С. Фарафонов. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2013. — 32 c. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: http://www.iprbookshop.ru/22912.html

3. Юров А.Н. Методические указания к лабораторным работам № 1-3 по дисциплине “Программирование” 14-2013 2013, магн.

4. Юров А.Н. Методические указания к лабораторным работам № 4-5 по дисциплине “Программирование” 15-2013 2013, магн.

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:

Лицензионное ПО:

- Среда разработки приложений Visual Studio
- Qt SDK+ Creator

Ресурсы информационно-телекоммуникационной сети «Интернет»:

- http://www.edu.ru/

Современные профессиональные базы данных:

- eLIBRARY.RU
- База ГОСТ docplan.ru
- Образовательный портал ВГТУ

Информационные справочные системы:

- http://window.edu.ru
- https://wiki.cchgeu.ru/

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой

Учебные лаборатории:

1 “Лаборатория интеллектуальных систем проектирования”

2 “Лаборатория компьютерного моделирования и дизайна”

3 “ Интернет-лаборатория ”

Лаборатории расположены по адресу г. Воронеж, ул. Плехановская, д. 11

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 удается 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, ознакомится с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - 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.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП