

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

УТВЕРЖДАЮ

Проректор по учебной работе

_____ Василенко В.Н.
(подпись) (Ф.И.О.)

«25» мая 2023 г.

РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ

Программирование на языках высокого уровня

Направление подготовки

09.03.02 Информационные системы и технологии

Направленность (профиль) подготовки

Моделирование и проектирование информационных технологий и систем

Квалификация выпускника

Бакалавр

Воронеж

1. Цели и задачи дисциплины

Целью освоения дисциплины «Программирование на языках высокого уровня» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

Об Связь, информационные и коммуникационные технологии в сфере исследования, разработки, внедрения и сопровождения информационных технологий и систем.

Дисциплина направлена на решение задач профессиональной деятельности следующих типов:

- производственно-технологический;
- организационно-управленческий;
- проектный.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 «Информационные системы и технологии».

2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	ИД1 _{УК-2} – Определяет (исходя из действующих правовых норм) совокупность взаимосвязанных задач, решение которых обеспечивает достижение поставленной цели.
2	ПКв-9	Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения	ИД1 _{ПКв-9} - Демонстрирует знания современных методов предпроектного анализа, методов сбора требований к ПО; основные подходы к проектированию ПО; методов моделирования бизнес процессов и спецификации требований; методологию и технологию и средства проектирования программного обеспечения.

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{УК-2} – Определяет (исходя из действующих правовых норм) совокупность взаимосвязанных задач, решение которых обеспечивает достижение поставленной цели.	Знает: основные понятия языков высокого уровня
	Умеет: выполнять декомпозицию задачи на отдельные модули.
	Владеет: навыками выполнения математической формулировки конкретных прикладных задач.
ИД1 _{ПКв-9} - Демонстрирует знания современных методов предпроектного анализа, методов сбора требований к ПО; основные подходы к проектированию ПО; методов моделирования бизнес процессов и спецификации требований; методологию и технологию и средства проектирования программного обеспечения.	Знает: основные приемы и методы структурирования программы.
	Умеет: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.
	Владеет: возможностями инструментальной среды Visual Studio 2010 для написания и отладки программ.

3. Место дисциплины в структуре ОП ВО

Дисциплина относится к части, формируемой участниками образовательных отношений Блока 1 ООП. Дисциплина является обязательной к изучению.

Изучение дисциплины основано на знаниях, умениях и навыках, полученных при изучении обучающимися следующих дисциплин: Компьютерные технологии.

Дисциплина является предшествующей для следующих дисциплин: Технологии программирования, Программирование на Java, Веб-дизайн.

4. Объем дисциплины и виды учебных занятий

Общая трудоёмкость дисциплины составляет 7 зачётных единиц.

Виды учебной работы	Всего ак. ч	Распределение трудоёмкости по семестрам, ак. ч	
		1 семестр	2 семестр
Общая трудоёмкость дисциплины (модуля)	252	108	144
Контактная работа в т. ч. аудиторные занятия:		30,85	76,1
Лекции		15	18
<i>в том числе в форме практической подготовки</i>		15	18
Практические занятия		15	36
<i>в том числе в форме практической подготовки</i>		15	36
Лабораторные занятия		-	18
<i>в том числе в форме практической подготовки</i>		-	18
Консультации текущие	2,65	0,75	1,9
Консультации перед экзаменом	2	-	2
Вид аттестации (зачет/экзамен)		0,1	0,2
Самостоятельная работа:	112,25	77,15	35,1
Проработка материалов по лекциям, учебникам, учебным пособиям	40,1	37,15	15,1
Подготовка к практическим/лабораторным занятиям	22,65	10	10
Курсовой проект/работа		10	
Домашнее задание, реферат		10	10
Другие виды самостоятельной работы		10	
Контроль	33,8	33,8	33,8

5. Содержание дисциплины и виды учебной работы

5.1 Содержание разделов дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела (указываются темы и дидактические единицы)	Трудоёмкость час
1 семестр			
1	Основные понятия языка C++	Понятия структуры программы. Типы данных: представление в памяти ЭВМ, диапазон значений, операции; простые и структурные типы (массив, структура, файл) типы данных. Переменные, константы, операции (логические, арифметические, над строками и т.п.), операторы: простые, ветвления, цикла	51,15
2	Алгоритмы	Понятие алгоритма, разработка способы записи и реализация алгоритмов	56
2 семестр			
3	Декомпозиция программ.	Декомпозиция программ: модули, процедуры, функции, их структура, формальные – фактические параметры. Рекурсия.	60
4	Среда Visual Studio 2010	Стандартные модули. Возможности отладки.	47,1
		<i>Консультации текущие</i>	2,65
		<i>зачет</i>	0,1
		<i>экзамен</i>	0,2

5.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, час	ЛЗ, час	ПЗ, час	СРО, час
1 семестр					
1.	Основные понятия языка C++	7	-	7	37,15
2.	Алгоритмы	8	-	8	40
2 семестр					
5.	Декомпозиция программ	10	10	20	20
6.	Среда Visual Studio 2010	8	8	16	15,1
<i>Консультации текущие</i>					2,65
<i>зачет</i>					0,1
<i>экзамен</i>					0,2

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость, ак. ч
1 семестр			
1	Основные понятия языка C++	<p>Общее понятие о языках высокого уровня, их описание при помощи лексем. Основные понятия: идентификатор, константа, комментарий и т.д. Общая характеристика языка C++</p> <p>Типы данных языка C++, <code>typedef</code>. Пространство имен. Область видимости имени. Классы хранения.</p> <p>Пользовательские и стандартные типы данных Простые и структурные типы. Указатели Представление в ЭВМ чисел и символов</p> <p>Структурные типы: массивы (ввод/вывод массива), строки, записи.</p> <p>Файлы, их классификация и организация в базах данных. Работа с текстовыми и бинарными файлами</p> <p>Классификация операций. Таблица истинности логических операций. Тип операндов и результата для операций. Приоритет выполнения операций в выражении</p>	7
2	Алгоритмы	<p>Алгоритмы ветвления и повтора (<code>for</code>, <code>While</code>, <code>do_while</code>)</p> <p>Модели решения типовых задач</p>	8
2 семестр			
5	Декомпозиция программ	<p>Основные принципы структурного программирования Программные структуры языка C++, синтаксис описания процедур и функций. Глобальные и локальные переменные. Область видимости имен, обмен данными между подпрограммами, дополнительные возможности при использовании директив.</p> <p>Классы. Объектно-ориентированная парадигма программирования. Режимы доступа к элементам класса</p>	10
6	Среда Visual Studio 2010	Возможности отладки среды Visual Studio	8

		2010. Пользовательские и стандартные модули. Компоновка программы.	
--	--	--	--

5.2.2 Практические занятия

№ п/п	Наименование раздела дисциплины	Тематика практических занятий	Трудоемкость, ак. ч
1	2	3	4
1 семестр			
1.	Основные понятия языка C++	Практическая работа по написанию программы в структурном стиле	3
		Практическая работа по написанию программы с использованием массивов	2
		Практическая работа по написанию программы с использованием динамически выделяемой памяти	2
2	Алгоритмы	Практическая работа по написанию программы с использованием классов	4
		Практическая работа по написанию программы с созданием конструкторов и деструкторов класса	2
		Практическая работа по написанию программы с созданием экземпляров класса	2
2 семестр			
3	Декомпозиция программ	Практическая работа по написанию программы, использующей механизм множественного наследования	4
		Практическая работа по написанию программы с дружественными классами и методами	4
		Практическая работа по написанию программы с виртуальными методами	4
		Практическая работа по написанию программы с использованием перегрузки операторов	4
		Практическая работа по написанию программы с применением шаблонов функций	4
4	Среда Visual Studio 2010	Практическая работа по написанию программы с перехватом исключительной ситуации	8
		Практическая работа по написанию программы с созданием пользовательского класса исключительной ситуации	8

5.2.3 Лабораторные работы

№ п/п	Наименование раздела дисциплины	Тематика лабораторных работ	Трудоемкость, час
3	Декомпозиция программ.	Получение навыков в составлении программ на языке С++ Структуризация программ. Функции. Параметры-указатели, параметры ссылки, параметры –значения	10
4	Среда Visual Studio 2010	Отладчик среды Visual Studio 2010	8

5.2.4. Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Труд., час
1 семестр			
1	Основные понятия языка С++	Проработка материалов по конспекту лекций.	7,15
		Проработка материалов по учебникам.	10
		Подготовка к коллоквиуму.	5
		Подготовка к практическим работам	5
		Написание РГР	20
2	Алгоритмы	Проработка материалов по конспекту лекций.	5
		Проработка материалов по учебникам.	15
		Подготовка к коллоквиуму.	5
		Подготовка к практическим работам	5
2 семестр			
3	Декомпозиция программ.	Проработка материалов по конспекту лекций.	3,1
		Проработка материалов по учебникам.	5
		Подготовка к коллоквиуму.	5
		Подготовка к практическим/лабораторным занятиям	5
4	Среда Visual Studio 2010	Проработка материалов по конспекту лекций.	2
		Проработка материалов по учебникам.	5
		Подготовка к практическим/лабораторным занятиям	5
		Написание РГР	5

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Основная литература

- Б. Страуструп., Язык программирования С++, 3-е изд./Пер. с англ., М.: «Невский Диалект» — «Издательство БИНОМ», 2015 г. — 991 с., ил.
- С/С++Программирование на языке высокого уровня/ Т.А. Павловская, - СПб.: Питер 2013 – 461 с.:

6.2. Дополнительная литература

- Тондо К., Гиммел С. Язык Си. Книга ответов. М.: Финансы і статистика, 2014. <http://biblos.vsu.ru/MegaPro/Web/SearchResult/MarcFormat/890>
- Ирина Евгеньевна **Медведкова** **Языки программирования** [Электронный ресурс] : методические указания по выполнению контрольной работы для студентов, обучающихся по специальностям 090105.65 и 090303.65, очной формы обучения / Ирина Евгеньевна **Медведкова** , Ольга Владимировна Авсеева; ВГУИТ, Кафедра информационных технологий, моделирования и управления. - Воронеж, 2013. - 20 с <http://biblos.vsu.ru/MegaPro/Web/SearchResult/ToPage/1>

6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся.

Методические указания к самостоятельной работе по дисциплине "Программирование на языках высокого уровня" [Электронный ресурс] : для студентов, обучающихся по направлениям 09.03.02 - "Информационные системы" любой формы обучения / И. Е. Медведкова; ВГУИТ, Кафедра информационных и управляющих систем. - Воронеж : ВГУИТ, 2016. - 19 с.

<http://biblos.vsu.ru/ProtectedView/Book/ViewBook/2540>

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

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	http://www.edu.ru/index.php
Научная электронная библиотека	http://www.elibrary.ru/defaulttx.asp?
Федеральная университетская компьютерная сеть России	http://www.runnet.ru/
Информационная система «Единое окно доступа к образовательным ресурсам»	http://www.window.edu.ru/
Электронная библиотека ВГУИТ	http://biblos.vsu.ru/megapro/web
Сайт Министерства науки и высшего образования РФ	http://minobrnauki.gow.ru
Портал открытого on-line образования	http://npoed.ru
Информационно-коммуникационные технологии в образовании. Система федеральных образовательных порталов	http://www.ict.edu.ru/
Электронная образовательная среда ФГБОУ ВО «ВГУИТ	http://education.vsu.ru
Поисковая система «Google»	http://www.google.ru

6.5. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

При освоении дисциплины используется лицензионное и открытое программное обеспечение и информационные справочные системы: ОС MS Windows, MS Office (MS Word, MS PowerPoint), программа для создания и редактирования видео Corel VideoStudio, ПС Google, СУБД «Рейтинг студентов», информационная среда для дистанционного обучения «Moodle», виртуальная машина Oracle VM Virtual Box. Локальная сеть университета и глобальная сеть Internet.

7. Материально-техническое обеспечение дисциплины

Ауд. 334 для проведения лекционных занятий, оснащенная комплектом мебели для учебного процесса – 50 шт., проектором Epson EH-TW650.

Ауд. 339 для проведения практических и лабораторных работ:

Количество ПК – 16 (IntelCore i5 – 4570), проектор – 1 (WiewSonicPJD5255).

Microsoft Windows 7 Microsoft Open License Microsoft Windows Professional 7 Russian Upgrade Academic OPEN 1 License No Level#47881748 от 24.12.2010г. <http://eopen.microsoft.com>.

Microsoft VisualStudio 2010 Сублицензионный договор № 42082/VRN3 от 21 августа 2013 г. на право использование программы DreamSparkElectronicSoftwareDeliver;

Microsoft Office 2007 Standar Microsoft Open License Microsoft Office 2007 Russian Academic OPEN No Level #44822753 от 17.11.2008<http://eopen.microsoft.com>.

1С: Предприятие. Бухгалтерия 8 Лицензионное соглашение с ЗАО «1С» Регистрационный номер 9985964 1С: Предприятие 8. Комплект для обучения в высших и средних учебных заведениях.

8. Оценочные материалы для промежуточной аттестации обучающихся по дисциплине

Оценочные материалы (ОМ) для дисциплины включают в себя:

- перечень компетенций с указанием индикаторов достижения компетенций, этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности.

ОМ представляются отдельным комплектом и **входят в состав рабочей программы дисциплины**.

Оценочные материалы формируются в соответствии с П ВГУИТ «Положение об оценочных материалах».

к рабочей программе
«Программирование на языках высокого уровня»

1. Организационно-методические данные дисциплины для заочной формы обучения

Виды учебной работы	Всего ак. ч	Распределение трудоемкости по семестрам, ак. ч	
		1 семестр	2 семестр
Общая трудоемкость дисциплины (модуля)	252	108	144
Контактная работа в т. ч. аудиторные занятия:			
Лекции	12	6	6
<i>в том числе в форме практической подготовки</i>	12	6	6
Практические/лабораторные занятия	12/8	6	6/8
<i>в том числе в форме практической подготовки</i>	12/8	6	6/8
Консультации текущие	3,4	1,7	1,7
Вид аттестации (зачет/экзамен)	0,3	0,1	0,2
Самостоятельная работа:	172,1	90,3	113,3
Проработка материалов по лекциям, учебникам, учебным пособиям	53	50,3	63,3
Подготовка к практическим/лабораторным занятиям	44,1	10	10
Выполнение контрольной работы	50	20	20
Другие виды самостоятельной работы		10	10
Подготовка к экзамену (контроль)	10,7	6,8	3,9

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

по дисциплине

Программирование на языках высокого уровня

1 Перечень компетенций с указанием этапов их формирования

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	ИД1 _{УК-2} – Определяет (исходя из действующих правовых норм) совокупность взаимосвязанных задач, решение которых обеспечивает достижение поставленной цели.
2	ПКв-9	Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения	ИД1 _{ПКв-9} - Демонстрирует знания современных методов предпроектного анализа, методов сбора требований к ПО; основные подходы к проектированию ПО; методов моделирования бизнес процессов и спецификации требований; методологию и технологию и средства проектирования программного обеспечения.

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 _{УК-2} – Определяет (исходя из действующих правовых норм) совокупность взаимосвязанных задач, решение которых обеспечивает достижение поставленной цели.	Знает: основные понятия языков высокого уровня
	Умеет: выполнять декомпозицию задачи на отдельные модули.
	Владеет: навыками выполнения математической формулировки конкретных прикладных задач.
ИД1 _{ПКв-9} - Демонстрирует знания современных методов предпроектного анализа, методов сбора требований к ПО; основные подходы к проектированию ПО; методов моделирования бизнес процессов и спецификации требований; методологию и технологию и средства проектирования программного обеспечения.	Знает: основные приемы и методы структурирования программы.
	Умеет: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.
	Владеет: возможностями инструментальной среды Visual Studio 2010 для написания и отладки программ.

2 Паспорт фонда оценочных средств по дисциплине

№ п/п	Разделы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные средства		Технология оценивания (способ контроля)
			наименование	№№ заданий	
1	Основные понятия языка C++	УК-2	Вопросы к экзамену	170 - 197	Собеседование с преподавателем
			Тесты (тестовые задания)	1-29	Компьютерное тестирование
			Вопросы к коллоквиуму	101 - 130	Собеседование или письменный опрос
			Задания для самостоятельного выполнения на практических занятиях	74-85	Контроль преподавателя
			Написание РГР	222-230	Контроль преподавателя
2	Алгоритмы	УК-2	Тестовые вопросы	49- 60	Компьютерное тестирование
			Задания для самостоятельного выполнения на практических занятиях	69 - 73	Контроль преподавателя
			Вопросы к экзамену	198 - 204	Собеседование с преподавателем
			Вопросы к коллоквиуму	1231 - 157	Собеседование или письменный опрос
3	Декомпозиция программ.	УК-2	Тестовые вопросы	49- 60	Компьютерное тестирование
			Вопросы к коллоквиуму	158 - 165	Собеседование или письменный опрос
			Задания для самостоятельного выполнения на практических занятиях	86 - 100	Контроль преподавателя
			Вопросы к экзамену	205 - 216	Собеседование с преподавателем
4	Среда Visual Studio 2010	ПКв-9	Вопросы к экзамену	213	Собеседование с преподавателем
			Вопросы к коллоквиуму	166- 169	Собеседование или письменный опрос
			Написание РГР	222-230	Контроль преподавателя

3 Оценочные средства для промежуточной аттестации

3.1 Тесты(тестовые задания)

УК 2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

№	Тестовое задание с вариантами ответов
1.	<p>Функции работы с датой и временем (отметить все правильные ответы):</p> <p><input type="checkbox"/> memmove (p1?p2?n) <input checked="" type="checkbox"/> char *ctime(time_t *t) <input checked="" type="checkbox"/> clock_t clock(void) <input type="checkbox"/> memchr (p,ch,n)</p>
2.	<p>Функции работы с блоками памяти (отметить все правильные ответы):</p> <p><input checked="" type="checkbox"/> memmove (p1?p2?n) <input type="checkbox"/> char *ctime(time_t *t) <input type="checkbox"/> clock_t clock(void) <input checked="" type="checkbox"/> memchr (p,ch,n)</p>
3.	<p>Начальный элемент массива (M) имеет индексы</p> <p>MDG J L X Z C B M Q E W R T U O P S F H K I Y</p> <p><input type="checkbox"/> [0][1]; <input checked="" type="checkbox"/> [0][0]; <input type="checkbox"/> [1][0]; <input type="checkbox"/> [1][1].</p>
4.	<p>Элемент массива A [1][2] имеет значение:</p> <p>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</p> <p><input checked="" type="checkbox"/> 8 ; <input type="checkbox"/> 2 ; <input type="checkbox"/> 12; <input type="checkbox"/> 7;</p>
5.	<p>При использовании в программе: <code>cout<<"12 + 12 ="<<12 +12<<"/n";</code> на экран будет выведено ...</p> <p><input type="checkbox"/> 24=24 <input checked="" type="checkbox"/> 12+12=24 <input type="checkbox"/> 12+12=12+12 <input type="checkbox"/> 48</p>
6.	<p>Синтаксически верное обозначение элемента массива:</p> <p><input checked="" type="checkbox"/> M [2][4] <input type="checkbox"/> A [1].[2] <input type="checkbox"/> P [2..4] <input type="checkbox"/> F (1)(3)</p>
7.	<p>При выполнении: <code>Int N= 20/3;</code> переменная N примет значение:</p> <p><input type="checkbox"/> 6.66666 <input checked="" type="checkbox"/> 6 <input type="checkbox"/> 7 <input type="checkbox"/> диагностика о несоответствии типов</p>
8.	<p>Индекс элемента одномерного массива – это:</p> <p><input type="checkbox"/> наибольший номер элемента массива; <input checked="" type="checkbox"/> порядковый номер элемента массива; <input type="checkbox"/> размерность массива; <input type="checkbox"/> имя массива</p>

9. Если элементы массива `intF[5]` равны соответственно 1,2, 3, 4, 5 то значение выражения `F[4] + F[2]`:

- (!) 8
- (?) 6
- (?) 0
- (?) 5

10. После выполнения последовательности операторов:

```
A [3]=5;
A [4]=6;
A [3]=A [4];
A[4]= A [3];
```

элементы массива `A [3]` и `A [4]` примут значения:

- (!) `A [3]=6; A [4]=6;`
- (?) `A [3]=5; A [4]=6 ;`
- (?) `A [3]=6; A [4]=5;`
- (?) `A [3]=5; A [4]=5`

11. Синтаксически корректными являются (отметить все правильные ответы):

```
[!] for (i= 1; i!=0; i- -) cout<<i;
[?] for(inti= 1; i< 20; i++; i:=i*2;
[?] for (inti= sin(1) ; i!=1; i++) f:=i;
[!] for (inti= 10;i> 2; cin>>i)f:=i;
```

12. Оператором в языке C/C++ является...

- (!) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения
- (?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение
- (?) Константа, которое не создает нового символического имени
- (?) Базовый элемент программы , в конце которого стоит символ точка с запятой

13. Операндом в языке C/C++ является...

- (?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения
- (!) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение
- (?) Константа, которое не создает нового символического имени
- (?) Базовый элемент программы , в конце которого стоит символ точка с запятой

14. Литералом в языке C/C++ является...

- (?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения
- (?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение
- (!) Константа, которое не создает нового символического имени
- (?) Базовый элемент программы , в конце которого стоит символ точка с запятой

15. Инструкцией в языке C/C++ является...

- (?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения
- (?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение
- (?) Константа, которое не создает нового символического имени
- (!) Базовый элемент программы , в конце которого стоит символ точка с запятой

16. Исключением в языке C/C++ является...

- (?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения

- (?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение
- (!) Необычное состояние программы, требуемое немедленного вмешательства
- (?) Базовый элемент программы, в конце которого стоит символ точка с запятой

17. Операторами сравнения являются операторы (отметить все правильные ответы):

==

sizeof

>>=

!=

18. Размер выражения или типа данных определяется с помощью оператора

sizeof

scanf

|=

(?) /

19. Значение выражения 4/2 будет иметь тип:

(?) Цельный

(!) Тот, каким обладает переменная, какой это выражение присваивается

(?) Вещественный

(?) Логический

20. Массив – это:

(!) проиндексированный набор данных одного типа с фиксированной конфигурацией;

(?) ограниченная двоеточием последовательность любых символов;

(?) совокупность разнородных данных, описываемых и обрабатываемых как единое целое;

(?) именованный набор однотипных данных на диске.

21. Инструкция выбора ... :

(!) switch

(?) While

(?) If

(?) default

22. Для определения новых типов данных используется оператор ...

(!) typedef

(?) switch;

(?) char

(?) default

23. Для объявления функций, которые не возвращают никаких значений используется спецификатор:

(?) virtual

(!) void

(?) static

(?) namespace

24. Для определения пространства имен функций, классов, определений используется спецификатор:

(?) virtual

(?) void

(?) static

(!) namespace

25. Из предложенных значений индекс массива

Int R [55] может принимать (отметить все верные значения):

[!]0

[?]55

[!]54

[?] -54

26. В массиве изменять значение элементов ...

(?) Нельзя ;

(!) можно;

(?) можно только в числовых массивах; (?) можно только в символьных массивах.

27. Инструкция цикла с предусловием

`while`

28. Модификаторами доступа из перечисленного являются:

[?] void

[!] protected

[?] static

[!] public

29. Для объявления закрытых членов класса используется модификатор

(!) private

(?) protected

(?) public

(?)void

30. Логическими являются операторы (отметить все правильные ответы):

[!] !

[?] ++

[?] scanf

[!] >>

31. Инструкция цикла с предусловием:

(!) While

(?) do

(?) If

(?) default

32. `int *d, gr=10, *pr=&gr;` Здесь описаны переменные (отметить верные утверждения):
`d` – статическая переменная для хранения адреса на величину типа `int`

`gr` – статическая переменная типа `int`, инициализирована значением 10

`pr` – статическая переменная (указатель на тип `int`), инициализирована

`pr` – динамическая переменная (указатель на тип `int`), инициализирована ссылкой на переменную `gr`.

33. Что такое указатель?

Статическая переменная в описании типа которой добавлен знак “*”.

Динамическая переменная в описании типа которой добавлен знак “*”.

Переменная, хранящая адрес в памяти.

Класс памяти

34. Операция разыменования – это

Обращение по адресу, содержащемуся в указателе

Присваивание начального значения переменной
Переименование типа переменной

Присваивание указателю значения NULL

35. Создание динамической переменной возможно с помощью операции new

36. Уничтожение динамической переменной возможно с помощью операции delete

37. К переменным типа указатель применимы операции:

Разыменование
Инкремент

(++)

)Деление

Поразрядовый сдвиг

38. К переменным типа указатель применимы операции:

Взятие адреса (&)

Декремент (--)
Умножение
Логические операции

39. К переменным типа указатель применимы операции сравнения:

==

!=

>=

<=

>

<

40. Какие операторы увеличивают в 2 раза значение последнего элемента массива `int a[5]` ?
`a[4] *=2;`

`*(a+4) *=2;`

`a[5] = a[5]*2;`

`*a+4 *=2;`

41. Отметьте все правильные утверждения

Ссылка есть синоним имени, указанного при инициализации ссылки. После инициализации ссылке не может быть присвоена др. переменная. Можно создавать массивы ссылок.

Тип ссылки должен совпадать с типом величины, на которую она ссылается.

42. Отметьте все правильные утверждения

Динамические массивы создаются с помощью операции new. Указатель на динамический массив можно изменять

Имя статического массива является указателем на его начальный элемент и его нельзя изменить

Имя статического массива является указателем на его начальный элемент и его можно изменять

43. Чему равен элемент A[2][1] массива A, если он инициализирован следующим образом:

`int A[3][3] = {1,2,3,4,5}`

5

4

2

0

44. При обращении к элементу A[4] массива A, если он описан как двумерный массив:

`int A[5][3] = {1,2,3,4,5}`

(!) 12+12=24

(?) 12+12=12+12

(?) 48

45. Синтаксически верное обозначение элемента массива:

(!) M [2][4]

(?) A [1].[2]

(?) P [2..4]

(?) F (1)(3)

46. При выполнении:

`Int N= 20/3;`

переменная N примет значение:

(?) 6.66666

(!) 6

(?) 7

(?) диагностика о несоответствии типов

47. После выполнения последовательности операторов:

`A [3]=5;`

`A [4]=6;`

`A [3]=A [4];`

`A[4]= A [3];`

элементы массива A [3] и A [4] примут значения:

(!) A [3]=6; A [4]=6;

(?) A [3]=5; A [4]=6 ;

(?) A [3]=6; A [4]=5;

(?) A [3]=5; A [4]=5

48. Индекс элемента одномерного массива – это:

(?) наибольший номер элемента массива;

(!) порядковый номер элемента массива;

(?) размерность массива;

(?) имя массива

49. Если элементы массива `int F[5]` равны соответственно 1,2, 3, 4, 5 то значение выражения

`F[4] + F[2];`

(!) 8

(?) 6

(?) 0

50. Оператором в языке C/C++ является...

(!) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения

- (?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение
- (?) Константа, которое не создает нового символического имени
- (?) Базовый элемент программы, в конце которого стоит символ точка с запятой

51. Операндом в языке C/C++ является...

(?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения

(!) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение

- (?) Константа, которое не создает нового символического имени

(?) Базовый элемент программы , в конце которого стоит символ точка с запятой

52. Литералом в языке C/C++ является...

(?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения

(?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение

(!) Константа, которое не создает нового символического имени

(?) Базовый элемент программы , в конце которого стоит символ точка с запятой

53. Инструкцией в языке C/C++ является...

(?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения

(?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение

(?) Константа, которое не создает нового символического имени

(!) Базовый элемент программы , в конце которого стоит символ точка с запятой

54. Исключение в языке C/C++ является...

(?) Символ, при помощи которого подвыражения (называемые операндами) комбинируются в более сложные выражения

(?) Подвыражение, которое при помощи операторов комбинируется в более сложное выражение

(!) Необычное состояние программы, требуемое немедленного вмешательства

(?) Базовый элемент программы , в конце которого стоит символ точка с запятой

55. Операторами сравнения являются операторы (отметить все правильные ответы):

[!] ==

[?] sizeof

[?] >>=

[!] !=

56. Размер выражения или типа данных определяется с помощью оператора

(!) sizeof

(?) scanf

(?) |=

(?) /

57. Значение выражения 4/2 будет иметь тип:

(?) Цельный

(!) Тот, каким обладает переменная, какой это выражение присваивается

(?) Вещественный

(?) Логический

58. Массив – это:

(!) проиндексированный набор данных одного типа с фиксированной конфигурацией;

(?) ограниченная двоеточием последовательность любых символов;

(?) совокупность разнородных данных, описываемых и обрабатываемых как единое целое;

(?) именованный набор однотипных данных на диске.

59. Инструкция выбора ... :

(!) switch

(?) While

(?) If

(?) default

60. Для определения новых типов данных используется оператор ...

(!) typedef

(?) switch;

(?) char

(?) default

61. Для объявления функций, которые не возвращают никаких значений используется спецификатор:

(?) virtual

(!) void

(?) static

(?) namespace

62. Из предложенных значений индекс массива

`int R [55]` может принимать (отметить все верные значения):

0

155

63. Для определения пространства имен функций, классов, определений используется спецификатор:

`virtual`

`void`

`static`

`namespace`

64. В массиве изменять значение элементов ...

Нельзя ;

можно;

можно только в числовых массивах;

можно только в символьных массивах.

65. Модификаторами доступа из перечисленного являются:

`void`

`protected`

`static`

`public`

66. Для объявления закрытых членов класса используется модификатор

`private`

`protected`

`public`

`void`

67. Логическими являются операторы (отметить все правильные ответы):

!

++

`scanf`

<<

3.2 Задания для самостоятельного выполнения на практических занятиях

ПКВ 9 Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения

69. В кинозале 20 рядов по 25 мест в каждом. Посетитель просит продать ему k мест подряд. Определите, возможно ли это на момент обращения и в каких рядах есть требуемое количество мест подряд.
70. Считайте из файла текст и определите, сколько в нем слов и в каком слове максимальное количество одинаковых символов подряд. Слова отделены друг от друга пробелами, запятыми и точками.
71. Напечатайте все простые числа от 1 до N . N - ввести с терминала.
72. Введите с терминала число N . И найдите произведение цифр из которых оно состоит.
73. Считайте из файла текст и определите, сколько в нем цифр и распечатайте все строки текста перенеся цифры в начало строки.
74. Вычислить $\sum_{n=1}^{10} (a_n - d_n) * n!$, где a_n и d_n - элементы векторов $A(10)$ и $D(10)$ соответственно.
75. Найти произведение положительных элементов вектора $A(100)$.
76. Вычислить $\sum_{n=1}^{10} (a_n - n!)$, где a_n - элемент вектор $A(10)$
77. Вычислить $\sum_{n=1}^{10} (a_n)$, где a_n - элемент вектор $A(10)$
78. Вычислить $\sum_{n=1}^{10} (n^2 / n!)$.
79. Вычислить с точностью $\varepsilon = 10^{-3}$ сумму ряда с общим членом $a_n = (1 + n^3) / n!$
80. Вычислить с точностью $\varepsilon = 10^{-3}$ сумму ряда с общим членом $a_n = 1 / n!$
81. Вычислить с точностью $\varepsilon = 10^{-3}$ сумму ряда с общим членом $a_n = 3n / n!$
82. Вычислить с точностью $\varepsilon = 10^{-3}$ сумму ряда с общим членом $a_n = 5 / n!$
83. Найти скалярное произведение 2-х векторов $A[10]$ и $D[10]$
84. Напечатать столбец матрицы $A(10*20)$ с \min суммой элементов
85. Вычислить $\sum_{n=1}^{10} \prod_{i=2}^{20} (1 - i / n!)$.

86. Найти max элемент вектора $A(100)$ и поменять его местами с 1-м.

87. Найти max элемент главной диагонали матрицы $A(20*20)$ и поменять его местами с $A[1,1]$.
88. Найти сумму всех нечетных элементов вектора $A(100)$ и обнулить все его четные элементы.
89. Найти min элементы 1-го столбца и 1-й строки матрицы $A(20*20)$, и поменять их местами.
90. Найти max нечетный элемент вектора $A(100)$ и поменять его местами с 1-м.
91. Напечатать суммы элементов для всех столбцов матрицы $A(10*20)$
92. Найти произведение элементов вектора $A(100)$, кратных 8.
93. Найти столбец матрицы $A(20*20)$, где находится ее max элемент и поменять местами этот столбец с 1-м.
94. Найти сумму четных элементов n-го столбца матрицы $A(10*20)$
95. Найти количество всех нечетных элементов вектора $A(100)$.
96. Найти max элементы 1-го столбца и k-й строки матрицы $A(10*10)$, и поменять их местами.
97. Найти произведение элементов вектора $A(100)$, кратных 4, и заменить им 1-й элемент вектора.
98. Найти столбец матрицы $A(20*20)$, где находится ее min элемент распечатать этот столбец.
99. Все элементы вектора $A(100)$, кратные 7, перенести в массив $D(100)$, и найти сумму всех остальных элементов вектора $A(100)$.
100. Найти столбец матрицы $A(20*20)$, где находится ее min элемент и все элементы этого столбца умножить на 5.

3.3 Вопросы коллоквиума

УК 2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

№	Текст вопроса
101.	Является ли обязательным заголовок программы в языке C++?
102.	Какова структура C- программы?
103.	Какие директивы препроцессора вы знаете?
104.	Что такое потоки ввода/вывода? Какие манипуляторы вывода вы знаете?
105.	В чем разница использования строк класса string и встроенного строкового массива.
106.	Как получить значение кода требуемого символа?
107.	Что такое алгоритм?
108.	Какие формы отображения алгоритма вы знаете?
109.	Приведите пример отображения алгоритма разными способами
110.	Каким образом отображаются в блок-схеме циклические операторы?
111.	Каким образом отображается в блок-схеме условный оператор?
112.	Какая разновидность строк введена в язык C++?
113.	Что представляют собой пользовательские типы?
114.	Что такое заголовочные файлы?
115.	Что представляют собой константы?
116.	Какие типы строк допустимы в C++?

117.	Что определяет тип данных?
118.	Приведение типов данных в выражении.
119.	Приведите полный перечень типов данных в C++ с примерами величин каждого типа.
120.	Поясните понятия (и приведите примеры) выражение, инструкция, операнд, операция
121.	Охарактеризуйте целочисленные типы данных, какие они могут принимать значения, в каких операциях могут принимать участие, сколько места занимают в памяти.
122.	Какие операции и стандартные функции определены над данными целого типа
123.	Чем отличаются вещественные числа от целых?
124.	Какие функции преобразуют вещественный аргумент в целое число? Чем они отличаются?
125.	Охарактеризуйте символьный тип данных.
126.	Где применяется булевский тип данных, какие он принимает значения, сколько места требуется для его размещения в памяти?
127.	Что такое пользовательские типы данных, чем они отличаются от стандартных типов данных? Приведите примеры данных перечисляемого и интервального типов.
128.	Что такое выражение, операция, операнд? Какие операции в языке C++ вы знаете?
129.	Охарактеризуйте каждую арифметическую операцию.
130.	Какие операции называются операциями отношения? Какой тип имеет результат операций отношения?
131.	Охарактеризуйте каждую логическую операцию. Приведите таблицу истинности для каждой.
132.	Что представляет собой выражение?
133.	Как делятся операции по количеству операндов?
134.	Какими факторами определяется последовательность выполнения операций?
135.	Как подразделяются операции C++ по приоритету?
136.	Какое различие между операциями / и %?
137.	Какой тип результата имеют операции отношения?
138.	К операндам каких типов могут быть применены поразрядные операции?
139.	Какова максимальная длина результирующей строки при выполнении строковых операций?
140.	Какие типы операндов допустимы при выполнении операций над множествами?
141.	Что такое оператор? Чем отличаются простые и структурные операторы?
142.	Оператор присваивания, назначение и порядок выполнения.
143.	Оператор безусловного перехода, его назначение и особенности применения.
144.	Назначение оператора вызова процедуры.
145.	В чем особенности пустого оператора? Его назначение?
146.	Что представляет собой составной оператор?
147.	Назначение, формы записи и порядок выполнения оператора условия if.
148.	Особенности использования вложенных условных операторов.
149.	Каковы отличия оператора выбора switch от оператора условия if?
150.	Какие правила должны выполняться при использовании оператора выбора switch?
151.	Как оперативно получить подсказку по языку программирования в интегрированной среде программирования?
152.	Каково назначение операторов повтора (цикла)?
153.	Какие требования предъявляются к выражениям, управляющим повторениями?
154.	В чем отличия операторов повтора while и do_while?
155.	В каких случаях предпочтительнее использовать для организации циклов оператор повтора for? Что записывается в заголовке этого оператора?
156.	Каким образом в операторе цикла for описывается направление изменения значе-

	ния параметра цикла?
157.	Какие ограничения налагаются на использование управляющей переменной (параметра цикла) в цикле for?
158.	Что такое вложенные циклы? Какие дополнительные условия необходимо соблюдать при организации вложенных циклов?
159.	Что представляет собой массив? Какой тип данных не могут иметь элементы массива?
160.	Как индексируются элементы массива?
161.	Каким образом многомерные массивы хранятся в памяти?
162.	Как декларируются многомерные массивы?
163.	Приведите примеры заполнения и вывода на экран массивов
164.	Приведите пример инициализации 2-мерного массива
165.	Зачем нужна отладка программ? Какие возможности для отладки программ предусмотрены в интегрированной среде программирования?
166.	Возможности отладчика среды Visual Studio 2010
167.	Каким образом в среде Visual Studio 2010 можно установить и отменить точки останова?
168.	Каким образом можно проследить изменение значений переменных с помощью отладчика среды Visual Studio 2010?
169.	Горячие клавиши отладчика среды Visual Studio 2010?

Коллоквиум проходит в устной форме или в форме письменного опроса. Студенту предлагается 3-5 вопросов. Вопросы носят конкретный характер. При ответе на них надо либо дать определение (пример: вопросы № 131 – 133...) либо дать краткие пояснения, например, вопрос 154: «В чем отличия операторов повтора while и do_while?»

3.4 Вопросы экзамена

ПКв-9 Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения

№	Текст экзаменационного вопроса
170	Машинное представление чисел и символов. Системы счисления.
171	Понятие алгоритма. Способы описания алгоритмов.
172	Языки программирования, понятие синтаксиса и семантики. Форма Бэкуса-Наура.
173	Основные понятия языка С: язык, лексема, алфавит, идентификатор, константа, переменная.
174	Системы счисления (СС). Перевод чисел из одной СС в другую.
175	Логические и арифметические основы работы ЭВМ.
176	Двоичная СС. Действия над целыми и вещественными числами в двоичной СС.
177	Трансляция, компиляция и интерпретация исходного текста программы.
178	Структура С- программы. Область видимости имен. <i>Пространство имен.</i>
179	Базовые типы данных языка С++ и модификаторы типа.
180	Типы данных. Правила приведения типа. Оператор <i>typedef</i> .
181	Операции sizeof, ‘,’, ‘?:’
182	Символьные данные: представление в ЭВМ, операции над ним.
183	Целочисленные данные: представление в ЭВМ, операции над ними.

184	<i>Булевы величины. Их машинное представление. Логические операции.</i>
185	Вещественные типы данных: машинное представление, операции над ними.
186	Выражения, операнды и операции. Приоритет выполнения операций в выражении.
187	Операции: арифметические, логические, поразрядные, отношения. Их приоритет.
188	Арифметические и поразрядные операции, тип их операндов и результата.
189	<i>Директивы препроцессора.</i>
190	Операции присваивания, составной оператор.
191	Условный оператор if, if_else.
192	Условный оператор switch.
193	Оператор цикла for.
194	Операторы цикла while и do_while.
195	Сравнительный анализ операторов цикла.
196	Массивы одномерные и многомерные. Обращение к элементам массива, ввод-вывод массива, инициализация массива.
197	Работа с массивом с использованием указателя.
198	Сортировка массива. Алгоритм пузырька.
199	Алгоритм сортировки массива выбором.
200	Сложные циклы. Инструкция break.
201	Структуры (struct): описание, инициализация, обращение к полям.
202	Потоки ввода/вывода. Управление форматом вывода в потоке, манипуляторы вывода.
203	Файлы текстовые и бинарные. Последовательность работы с ними.
204	Работа с файлами посредством файловых потоков.
205	Функции. Их объявление и определение. Перегрузка функций. Спецификатор INLINE.
206	Формальные и фактические параметры. Значения аргументов по умолчанию.
207	Передача массива в качестве параметра функции.
208	Константы, строковые литералы, работа со строками.
209	Передача массива в качестве параметра функции.
210	Передача параметров в виде ссылки, указателя, значения.
211	Указатели и ссылки. Основные операции работы с указателями. Операторы new, delete.
212	Понятие объекта и Lvalue. Классы хранения (памяти).
213	Динамические массивы, их создание и работа с ними.
214	Перегрузка функций, значения аргументов функции по умолчанию
215	Передача параметров в виде ссылки, указателя, значения.
216	Охарактеризуйте возможности отладчика среды Visual Studio 2010

3.4.1 Примеры экзаменационных задач

УК 2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений

217. Заполнить матрицу $A[n][n]$ $a_{ij} = \sum_{r=1}^5 (r + |i - j|)!$, и найти номер столбца с min-м элементом.
218. Ввести координаты трех точек на плоскости, и, напечатать номера тех из них, которые ближе всего находятся друг к другу.
219. Вычислить с точностью 10^{-3} сумму ряда (формулы приближенного вычисления для возведения в степень не применять):
- $$1 - \frac{x^2}{2!} + \frac{x^3}{3!} - \frac{x^4}{4!} + \dots$$
220. Определить, в каком столбце матрицы больше всего элементов, кратных 7, и обнулить его элементы.
221. Организовать ввод последовательности целых чисел, признаком конца которой служит 0, и найти среднее арифметическое ее нечетных чисел.

Экзамен Проводится в устной форме. В билете 2 вопроса (№170-216) и задача.

3.5 Примерные задания для расчетно – графической работы (РГР)

ПКв 9 Способность проводить анализ требований к программному обеспечению, выполнять работы по проектированию программного обеспечения

Составьте программу, которая заполняет квадратную матрицу по спирали. Вариант: заполнение через 1, по часовой стрелке, а потом против часовой стрелки заполняются пропущенные позиции. Процесс должен быть проиллюстрирован ани-мацией на экране.

222. Составьте программу, которая вводит по выбору пользователя с терминала или считывает из файла текстовую строку, признаком конца которой является точка, и печатает столбиком слова, начинающиеся с эталонной комбинации символов, вводимой с терминала. Эталонная комбинация в каждом слове должна быть выделена цветом.
223. Считать из файла текст. Вывести на экран все слова текста, упорядочив их по алфавиту. Процесс должен быть проиллюстрирован анимацией на экране.
224. Считать из файла текст. Вывести на экран все слова текста, упорядочив их по размеру. Процесс должен быть проиллюстрирован анимацией на экране.
225. Указать натуральное число из заданного интервала, в двоичном представлении которого больше всего единиц. Процесс должен быть проиллюстрирован анимацией на экране. Результат записан в файл.
226. Считать из файла целочисленную матрицу $A[n,m]$ и заменить в ней нулями элементы матрицы, стоящие на пересечении строк и столбцов, в которых имеется хотя бы по одному 0. Процесс должен быть проиллюстрирован анимацией на экране.

227. Считать из файла текст и распечатать его, исключив из него те символы, которые находятся между скобками (). Сами скобки не удалять. Если хотя бы одной скобки нет – сообщить об этом. Процесс должен быть проиллюстрирован анимацией на экране.

228. Считать из файла целочисленный массив $A[n]$. Подсчитать длину самой длинной последовательности чисел одного знака. Например, для массива

1, 4, 6, **-3, -11, -6, -32, -4**, 1, 90, -44, 7

искомая длина равна 5 (в примере эта последовательность выделена жирным шрифтом). Процесс должен быть проиллюстрирован анимацией на экране.

229. Считать из файла целочисленный массив и вывести его на экран, выделяя цветом те его элементы, которые больше своих «соседей», т.е. элементов, стоящих справа и слева.

Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания для каждого результата обучения по дисциплине/практике

Результаты обучения по этапам формирования компетенции	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
ОПК 1					
ЗНАТЬ: <ul style="list-style-type: none"> ▪ Основные понятия языков высокого уровня (структуры и типы данных, операции, операторы, модули; структура программ, модулей, процедур и функций) ▪ Основные приемы и методы структурирования программы 	Коллоквиум	Собеседование или письменный опрос	Обучающийся знает суть прочитанного материала, без грубых ошибок отвечает на вопросы	Зачтено/балл	Освоена
			Обучающийся неверно представляет суть основных понятий дисциплины, не может проиллюстрировать свой ответ каким-либо примером, сам ответ принципиально неверен.	Не зачтено /балл	Не освоена
	Тестовое задание	Результат компьютерного тестирования	Набрано менее 50% правильных ответов	не удовлетворительно	не освоена (недостаточный)
			Набрано 50% - 74,99% правильных ответов	удовлетворительно	освоена (базовый)
			Набрано 75% - 89.99% правильных ответов	хорошо	освоена (повышенный)
			Набрано 90% - 100% правильных ответов	отлично	освоена (повышенный)
	Вопросы к экзамену	Экзамен (собеседование)	Обучающийся решил задачу верно, полностью раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности, ответил на дополнительные вопросы преподавателя	отлично	освоена (повышенный)
			Обучающийся решил задачу, твердо знает материал, но допускает в ответе некоторые неточности или отказался от доп. вопроса	хорошо	освоена (повышенный)
			Обучающийся не смог решить задачу, но решил упрощенный вариант задачи. Материал билета ответил неполно или непоследовательно, но показал общее понимание вопроса,	удовлетворительно	освоена (базовый)
			Обучающийся не решил даже предельно упрощенный вариант задачи, ответы на вопросы билета не знает или допускает грубые ошибки в фор-	не удовлетворительно	не освоена (недостаточный)

			мулировках основных понятий дисциплины		
УМЕТЬ: <ul style="list-style-type: none"> ▪ выполнять декомпозицию задачи на отдельные модули; ▪ разрабатывать и/или адаптировать алгоритмы под конкретную задачу; ▪ составлять программы на языке C++; 	Задачи для самостоятельного выполнения	Контроль преподавателя	Верно выполнено задание практической работы, возможно с помощью преподавателя, но студент внятно может детально пояснить смысл кода.	Зачтено	Освоена
			Практическая работа не выполнена, либо выполнена не в полном объеме и допущены грубые ошибки, либо студент не может пояснить смысл кода.	Не зачтено	Не освоена
ВЛАДЕТЬ: <ul style="list-style-type: none"> ▪ навыками выполнения математической формулировки конкретных прикладных задач; ▪ возможностями инструментальной среды VisualStudio 2010 для написания и отладки программ 	Расчетно графическая работа	Контроль преподавателя	Задача РГР решена, построенная обучающимся блок- схема без ошибок отражает алгоритм решения, правильно оформлен отчет,	Зачтено	освоена (базовый)
			Задача не решена, обучающийся не имеет навыков построения блок схем, им избран нерациональный алгоритм решения. При решении не проведена декомпозиции программы.	Не зачтено	не освоена (недостаточный)