

**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**

**Разработка приложений на C++**  
Направление подготовки

**09.04.03 Прикладная информатика**

Направленность (профиль) подготовки

**Математическое и компьютерное моделирование информационных и бизнес-процессов**

Квалификация выпускника

**магистр**

---

Воронеж

## 1. Цели и задачи дисциплины (модуля)

Целью освоения дисциплины «Разработка приложений на С++» является формирование у обучающегося теоретических знаний о современных информационных системах и технологиях, моделях, методах и средствах решения функциональных задач и организации информационных процессов, изучение организационной, функциональной и физической структуры информационных систем и базовых информационных процессов, рассмотрение перспектив использования информационных технологий в условиях перехода к информационному обществу.

Задачи дисциплины заключаются в подготовке обучающихся к решению следующих профессиональных задач:

- концептуальное проектирование информационных систем и технологий;
- подготовка заданий на проектирование компонентов информационных систем и технологий на основе методологии системной инженерии.

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

В результате освоения дисциплины в соответствии с предусмотренными компетенциями обучающийся должен:

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	ПКв-1	Способность применять современные методы и инструментальные средства прикладной информатики для автоматизации и информатизации решения прикладных задач различных классов и создания ИС	ИД1 <sub>ПКв-1</sub> Изучает современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС и сервисов, современные методы, средства, стандарты информатики для решения прикладных задач различных классов, логические методы и приемы системного научного анализа
			ИД2 <sub>ПКв-1</sub> Применяет современные методы и инструментальные средства для решения прикладных задач различных классов и создания информационных систем, логические методы и приемы системного научного анализа
2	ПКв-2	Способность проектировать информационные процессы и системы с использованием инновационных инструментальных средств	ИД1 <sub>ПКв-2</sub> – Анализирует современные способы и методы проектирования информационных процессов и систем с использованием инновационных инструментальных средств, современных интеллектуальных технологий; современное программное и аппаратное обеспечение ИС и автоматизированных систем; современные СУБД, принципы организации БД, их ведение и корректировку
			ИД2 <sub>ПКв-2</sub> – Использует методы, способы и средства проектирования информационных систем, процессов и баз данных на основе инновационных инструментальных средств создания, корректировки и обеспечения жизненного цикла ИС

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 <sub>ПКв-1</sub> Изучает современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС и сервисов, современные методы, средства, стандарты инфор-	Знает: современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС с использованием языка программирования С++
	Умеет: разрабатывать и проектировать приложения с использованием языка программирования С++

матики для решения прикладных задач различных классов, логические методы и приемы системного научного анализа	Владеет: методикой решения прикладных задач различных классов с использованием языка программирования С++
ИД2 <sub>ПКв-1</sub> Применяет современные методы и инструментальные средства для решения прикладных задач различных классов и создания информационных систем, логические методы и приемы системного научного анализа	Знает: современные методы и инструментальные средства разработки с использованием языка программирования С++
	Умеет: решать прикладные задачи различных классов и создавать информационные системы с использованием языка программирования С++
	Владеет: методикой решения научных задач с использованием языка программирования С++
ИД1 <sub>ПКв-2</sub> – Анализирует современные способы и методы проектирования информационных процессов и систем с использованием инновационных инструментальных средств, современных интеллектуальных технологий; современное программное и аппаратное обеспечение ИС и автоматизированных систем; современные СУБД, принципы организации БД, их ведение и корректировку	Знает: современные способы и методы проектирования информационных процессов и систем с использованием языка С++.
	Умеет: проектировать информационные процессы и системы с использованием инновационных с использованием языка С++.
	Владеет: методологией проектирования информационных процессов и систем с использованием языка С++.
ИД2 <sub>ПКв-2</sub> – Использует методы, способы и средства проектирования информационных систем, процессов и баз данных на основе инновационных инструментальных средств создания, корректировки и обеспечения жизненного цикла ИС	Знает: особенности современных СУБД
	Умеет: создавать основные объекты БД
	Владеет: методологией проектирования современных БД

### 3. Место дисциплины (модуля) в структуре ООП ВО

Дисциплина «Разработка приложений на С++» относится к вариативной части цикла Б1 ВУЗа учебного плана подготовки студентов по направлению 09.04.03 «Прикладная информатика». Дисциплина «Разработка приложений на С++» позволяет подготовить студентов для решения практических задач в рамках изучения последующих дисциплин.

### 4. Объем дисциплины (модуля) и виды учебных занятий

Общая трудоемкость дисциплины (модуля) составляет 8 зачетных единиц.

Виды учебной работы	Всего часов	Семестр
		3
	<b>акад.</b>	<b>акад.</b>
Общая трудоемкость дисциплины (модуля)	288	288
<b>Контактная работа, в т.ч. аудиторные занятия:</b>	51	51
Лекции	16	16
Практические занятия (ПЗ)	32	32
<i>в том числе в форме практической подготовки</i>	32	32
Семинары (С)		
Лабораторные работы (ЛБ)		
Консультации текущие	0,8	0,8
Проведение консультаций перед экзаменом	2	2
Виды аттестации (зачет, экзамен)	0,2	0,2
<b>Самостоятельная работа:</b>	203,2	203,2
Проработка материалов по конспекту лекций	100	100
Проработка материалов по учебникам, учебным пособиям	100	100

Виды учебной работы	Всего часов	Семестр
		3
	акад.	акад.
Курсовой проект (работа)		
Реферат		
и (или) другие виды самостоятельной работы	3,2	3,2
Подготовка к экзамену (контроль)	33,8	33,8

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

**5.1 Содержание разделов дисциплины (модуля)**

№ п/п	Наименование раздела дисциплины	Содержание раздела	Трудоемкость раздела, ак.ч
1	Структура программного обеспечения и этапы его разработки	Системное и специальное ПО. Инструментальная среда программирования. Языки программирования и их краткая характеристика.	46
2	Основы ООП в C++	Понятие класса. Специальные функции-члены класса (конструкторы, деструкторы, преобразования). Статические члены класса. Краткий обзор. Функции-члены и данные-члены. Интерфейсы и реализация. Конструкторы и деструкторы. Инициализация и очистка. Конструктор без параметров (по умолчанию). Конструктор копирования. Указатель this. Статические члены: функции и данные. Указатели на члены. Структуры и объединения.	50
3	Наследование в C++	Наследование классов и производные классы. Конструкторы, деструкторы и наследование. Иерархия классов. Виртуальные функции. Полиморфизм. Абстрактные классы и чистые виртуальные функции. Множественное наследование. Виртуальные базовые классы. Контроль доступа. Определение типа объектов во время выполнения программы (RTTI).	40
4	Обработка исключений в C++	Обработка ошибок в стандартном С. Распознавание ситуаций. Использование assert, кодов возврата, сигналов, setjmp и longjmp. Использование set_new_handler(). Объектно-ориентированная обработка исключений. Применение try, catch, throw. Раскрутка стека. Стандартные исключения в C++. Работа с конструкторами и исключениями. Функции terminate(), unexpected().	38
5	Потоки ввода-вывода в C++	Заголовочные файлы. Предопределенные объекты и потоки. Опе-	38

		рации помещения и извлечения. Форматирование. Флаги форматирования. Манипуляторы. Ошибки потоков. Файловый ввод-вывод с применением потоков С++. Конструкторы файловых потоков. Открытие файлов в разных режимах. Ввод вывод в файлы. Форматирование в памяти	
6	Шаблоны в С++	Шаблоны функций. Шаблоны классов. Параметры шаблонов. Наследование и шаблоны. Примеры построения шаблонов. Итераторы.	39,2
<i>Консультации текущие</i>			0,8
<i>Консультации перед экзаменом</i>			2
<i>Экзамен</i>			0,2

## 5.2 Разделы дисциплины (модуля) и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, час	Практические занятия, ак. ч.	СРО, час
1	Структура программного обеспечения и этапы его разработки	2	4	40
2	Основы ООП С++	4	6	40
3	Наследование в С++	4	6	30
4	Обработка исключений в С++	2	6	30
5	Потоки ввода-вывода в С++	2	6	30
6	Шаблоны в С++	2	4	33,2
<i>Консультации текущие</i>				0,8
<i>Консультации перед экзаменом</i>				2
<i>Экзамен</i>				0,2

### 5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость, час
1	Структура программного обеспечения и этапы его разработки	Системное и специальное ПО. Инструментальная среда программирования. Языки программирования и их краткая характеристика.	2
2	Основы ООП в С++	Понятие класса. Специальные функции-члены класса (конструкторы, деструкторы, преобразования). Статические члены класса. Краткий обзор. Функции-члены и данные-члены. Интерфейсы и реализация. Конструкторы и деструкторы. Инициализация и очистка. Конструктор без параметров (по умолчанию). Конструктор копирования. Указатель this. Статические члены: функции и данные. Указатели на члены. Структуры и объединения.	4
4	Наследование в С++	Наследование классов и производные классы. Конструкторы, деструкторы и наследование. Иерархия классов. Виртуальные функции. Полиморфизм. Абстрактные классы и чистые виртуальные функции. Множественное наследование. Виртуальные базовые классы. Контроль доступа. Определение типа объектов во время выполнения программы (RTTI).	4
5	Обработка исключений в С++	Обработка ошибок в стандартном С. Распознавание ситуаций. Использование assert, кодов возврата, сигналов, setjmp и longjmp. Использование set_new_handler().	2

		Объектно-ориентированная обработка исключений. Применение try, catch, throw. Раскрутка стека. Стандартные исключения в C++. Работа с конструкторами и исключениями. Функции terminate(), unexpected().	
5	Потоки ввода-вывода в C++	Заголовочные файлы. Предопределенные объекты и потоки. Операции помещения и извлечения. Форматирование. Флаги форматирования. Манипуляторы. Ошибки потоков. Файловый ввод-вывод с применением потоков C++. Конструкторы файловых потоков. Открытие файлов в разных режимах. Ввод-вывод в файлы.	2
6	Шаблоны в C++	Шаблоны функций. Шаблоны классов. Параметры шаблонов. Наследование и шаблоны. Примеры построения шаблонов. Итераторы.	2

### 5.2.2 Практические занятия

№ п/п	Наименование раздела дисциплины	Тематика практических занятий (семинаров)	Трудоемкость, час
1	Структура программного обеспечения и этапы его разработки	Практическая работа по написанию программы в структурном стиле. Практическая работа по написанию программы с использованием массивов. Практическая работа по написанию программы с использованием динамически выделяемой памяти.	4
2	Основы ООП в C++	Практическая работа по написанию программы с использованием классов.	6
3	Наследование в C++	Практическая работа по написанию программы, использующей механизм наследования	6
4	Обработка исключений в C++	Практическая работа по написанию программы с перехватом исключительной ситуации	6
5	Потоки ввода-вывода в C++	Практическая работа по написанию Программы файлового ввода-вывода	6
6	Шаблоны в C++	Практическая работа по написанию программы с применением шаблонов функций.	4

### 5.2.3 Лабораторный практикум не предусмотрен.

### 5.2.4 Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость, час
1	Структура программного обеспечения и этапы его разработки	Проработка материалов по курсу лекций	40
		Проработка материалов по учебнику	
		Оформление отчетов	
2	Основы ООП в C++	Проработка материалов по курсу лекций	40
		Проработка материалов по учебнику	
		Оформление отчетов	
3	Наследование в C++	Проработка материалов по курсу лекций	30
		Проработка материалов по учебнику	
		Оформление отчетов	
4	Обработка исключений в C++	Проработка материалов по курсу лекций	30

		Проработка материалов по учебнику	
		Оформление отчетов	
		Оформление отчетов	
		Проработка материалов по учебнику	
		Оформление отчетов	
5	Потоки ввода-вывода в C++	Проработка материалов по курсу лекций	30
		Проработка материалов по учебнику	
		Оформление отчетов	
6	Шаблоны в C++	Проработка материалов по курсу лекций	33,2
		Проработка материалов по учебнику	
		Оформление отчетов	
		Оформление отчетов	
		Проработка материалов по учебнику	
		Оформление отчетов	

## 6 Учебно-методическое и информационное обеспечение дисциплины (модуля)

### 6.1 Основная литература

1. Рысин, М. Л. Основы программирования на языке C++ : учебное пособие / М. Л. Рысин, М. В. Сартаков, О. В. Макеева. — Москва : РТУ МИРЭА, 2022. — 118 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/239957> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

2. Ахмедова, З. А. Лабораторный практикум по дисциплине «Программирование на языке Си» для направления подготовки «Информационная безопасность» : учебное пособие / З. А. Ахмедова ; составитель З. А. Ахмедова. — Махачкала : ДГУНХ, 2022. — 26 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/246560> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

3. Тюкачев, Н. А. C#. Алгоритмы и структуры данных : учебное пособие для спо / Н. А. Тюкачев, В. Г. Хлебостроев. — Санкт-Петербург : Лань, 2021. — 232 с. — ISBN 978-5-8114-6817-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/154117> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

4. Жулабова, Ф. Т. Системное программирование. Лабораторные работы : учебное пособие для спо / Ф. Т. Жулабова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 208 с. — ISBN 978-5-8114-7721-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/164955> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

### 6.2 Дополнительная литература

1. . Биллиг В. А. Объектное программирование в классах на C# 3.0 - Москва: Национальный Открытый Университет «ИНТУИТ», 2016 Биллиг, В.А. Объектное программирование в классах на C# 3.0 / В.А. Биллиг. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 391 с. : ил. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428945>.

2. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие для спо / Л. А. Залогова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 192 с. — ISBN 978-5-8114-7722-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/164956> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

3. Юрьева, А. А. Математическое программирование : учебное пособие / А. А. Юрьева. — 2-е изд., испр. и доп. — Санкт-Петербург : Лань, 2021. — 432 с. — ISBN 978-5-8114-1585-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/168878> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

4. Солдатенко, И. С. Практическое введение в язык программирования Си : учебное пособие / И. С. Солдатенко, И. В. Попов. — Санкт-Петербург : Лань, 2021. — 132 с. — ISBN 978-5-8114-3150-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/169287> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

### 6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся

1. Ламонина, Л. В. Практикум по алгоритмизации и программированию : учебное пособие / Л. В. Ламонина, Т. Ю. Степанова. — Омск : Омский ГАУ, 2021. — 123 с. — ISBN 978-5-89764-947-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/170276> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

2. Тюкачев, Н. А. С#. Алгоритмы и структуры данных : учебное пособие для вузов / Н. А. Тюкачев, В. Г. Хлебостроев. — 4-е изд., стер. — Санкт-Петербург : Лань, 2021. — 232 с. — ISBN 978-5-8114-8247-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/172708> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

3. Курбанисмаилов, З. М. Разработка интерактивных приложений на языке С# : учебно-методическое пособие / З. М. Курбанисмаилов. — Москва : РТУ МИРЭА, 2021. — 135 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176523> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

4. Скворцова, Л. А. Объектно-ориентированное программирование на языке C++: Практикум : учебное пособие / Л. А. Скворцова, А. А. Бирюкова, К. В. Гусев. — Москва : РТУ МИРЭА, 2021. — 146 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176540> (дата обращения: 03.07.2022). — Режим доступа: для авториз. пользователей.

### 6.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	<a href="https://www.edu.ru/">https://www.edu.ru/</a>
Научная электронная библиотека	<a href="https://elibrary.ru/defaultx.asp?">https://elibrary.ru/defaultx.asp?</a>
Национальная исследовательская компьютерная сеть России	<a href="https://niks.su/">https://niks.su/</a>
Информационная система «Единое окно доступа к образовательным ресурсам»	<a href="http://window.edu.ru/">http://window.edu.ru/</a>
Электронная библиотека ВГУИТ	<a href="http://biblos.vsu.ru/megapro/web">http://biblos.vsu.ru/megapro/web</a>
Сайт Министерства науки и высшего образования РФ	<a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>
Портал открытого on-line образования	<a href="https://npoed.ru/">https://npoed.ru/</a>
Электронная информационно-образовательная среда ФГБОУ ВО «ВГУИТ»	<a href="https://education.vsu.ru/">https://education.vsu.ru/</a>

### 6.5 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При изучении дисциплины используется программное обеспечение и информационные справочные системы: информационная среда для дистанционного обучения «Moodle», автоматизированная информационная база «Интернет-тренажеры», «Интернет-экзамен» и пр. (указать средства, необходимы для реализации дисциплины).

При освоении дисциплины используется лицензионное и открытое программное обеспечение – ОС Windows, MS Office. Локальная сеть университета и глобальная сеть Internet.

### **7 Материально-техническое обеспечение дисциплины (модуля)**

Аудитория для проведения лекционных занятий: ауд. 334 - комплект мебели для учебного процесса – 30 шт.; переносной проектор Acer с настольным проекционным экраном, ноутбук Lenovo; наборы демонстрационного оборудования и учебных наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие примерным программам дисциплин рабочим учебным программам.

Аудитории для проведения практических занятий: ауд. 332 – учебная лаборатория для лабораторных и практических работ: количество рабочих станций – 12 (Intel Core i3-540).

Помещения для самостоятельной работы: ауд. 336а - учебная лаборатория для лабораторных, практических работ, курсового и дипломного проектирования: количество рабочих станций – 13 (Intel Core i7- 8700); читальные залы библиотеки: компьютеры со свободным доступом в сеть Интернет и Электронными библиотечными и информационно справочными системами.

### **8 Оценочные материалы для промежуточной аттестации обучающихся по дисциплине (модулю)**

Оценочные материалы (ОМ) для дисциплины (модуля) 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 в виде отдельного документа и входят в состав рабочей программы дисциплины (модуля) в виде приложения.

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

**ПРИЛОЖЕНИЕ А**  
**к рабочей программе**  
**Разработка приложений на С++**

**1. Организационно-методические данные дисциплины для очно-заочной или заочной форм обучения**

**1.1 Объемы различных форм учебной работы и виды контроля в соответствии с учебным планом**

Виды учебной работы	Всего часов	Семестр
	акад.	акад.
Общая трудоемкость дисциплины (модуля)	288	288
<b>Контактная работа, в т.ч. аудиторные занятия:</b>	21,9	21,9
Лекции	6	6
Практические занятия (ПЗ)	12	12
<i>в том числе в форме практической подготовки</i>	12	12
Семинары (С)	-	-
Лабораторные работы (ЛБ)	-	-
Консультации текущие	1,7	1,7
Проведение консультаций перед экзаменом	2	2
Виды аттестации (зачет, экзамен)	0,2	0,2
<b>Самостоятельная работа:</b>	259,3	259,3
Проработка материалов по конспекту лекций	110	110
Проработка материалов по учебникам, учебным пособиям	90	90
Курсовой проект (работа)		
Реферат		
Контрольная работа	59,3	59,3
<b>Подготовка к экзамену (контроль)</b>	6,8	6,8

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

по дисциплине  
**РАЗРАБОТКА ПРИЛОЖЕНИЙ НА C++**

## 1. Перечень компетенций с указанием этапов их формирования

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	ПКв-1	Способность применять современные методы и инструментальные средства прикладной информатики для автоматизации и информатизации решения прикладных задач различных классов и создания ИС	ИД1 <sub>ПКв-1</sub> Изучает современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС и сервисов, современные методы, средства, стандарты информатики для решения прикладных задач различных классов, логические методы и приемы системного научного анализа
			ИД2 <sub>ПКв-1</sub> Применяет современные методы и инструментальные средства для решения прикладных задач различных классов и создания информационных систем, логические методы и приемы системного научного анализа
2	ПКв-2	Способность проектировать информационные процессы и системы с использованием инновационных инструментальных средств	ИД1 <sub>ПКв-2</sub> – Анализирует современные способы и методы проектирования информационных процессов и систем с использованием инновационных инструментальных средств, современных интеллектуальных технологий; современное программное и аппаратное обеспечение ИС и автоматизированных систем; современные СУБД, принципы организации БД, их ведение и корректировку
			ИД2 <sub>ПКв-2</sub> – Использует методы, способы и средства проектирования информационных систем, процессов и баз данных на основе инновационных инструментальных средств создания, корректировки и обеспечения жизненного цикла ИС

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 <sub>ПКв-1</sub> Изучает современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС и сервисов, современные методы, средства, стандарты информатики для решения прикладных задач различных классов, логические методы и приемы системного научного анализа	Знает: современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС с использованием языка программирования С++
	Умеет: разрабатывать и проектировать приложения с использованием языка программирования С++
	Владеет: методикой решения прикладных задач различных классов с использованием языка программирования С++
ИД2 <sub>ПКв-1</sub> Применяет современные методы и инструментальные средства для решения прикладных задач различных классов и создания информационных систем, логические методы и приемы системного научного анализа	Знает: современные методы и инструментальные средства разработки с использованием языка программирования С++
	Умеет: решать прикладные задачи различных классов и создавать информационные системы с использованием языка программирования С++
	Владеет: методикой решения научных задач с использованием языка программирования С++
ИД1 <sub>ПКв-2</sub> – Анализирует современные способы и методы проектирования информационных процессов и систем с использованием инновационных инструментальных средств, современных интеллектуальных технологий; современное программное и аппаратное обеспечение ИС и автоматизированных систем; современные СУБД, принципы организации БД, их ведение и корректировку	Знает: современные способы и методы проектирования информационных процессов и систем с использованием языка С++.
	Умеет: проектировать информационные процессы и системы с использованием инновационных с использованием языка С++.
	Владеет: методологией проектирования информационных процессов и систем с использованием языка С++.

ИД2 <sub>ПКв-2</sub> – Использует методы, способы и средства проектирования информационных систем, процессов и баз данных на основе инновационных инструментальных средств создания, корректировки и обеспечения жизненного цикла ИС	Знает: особенности современных СУБД
	Умеет: создавать основные объекты БД
	Владеет: методологией проектирования современных БД

## 2 Паспорт оценочных материалов по дисциплине

№ п/п	Разделы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные средства		Технология оценивания (способ контроля)
			наименование	№№ заданий	
1	Структура программного обеспечения и этапы его разработки	ПКв-1	Вопросы к экзамену	1-11	Уровневая шкала
		ПКв-1	Тесты	1-6	Процентная шкала
2	Основы ООП в C++	ПКв-1	Вопросы к экзамену	12-19	Уровневая шкала
		ПКв-1	Тесты	7-12	Процентная шкала
3	Наследование в C++	ПКв-1	Вопросы к экзамену	20-26	Уровневая шкала
		ПКв-1	Тесты	13-15	Процентная шкала
4	Обработка исключений в C++	ПКв-2	Вопросы к экзамену	27-36	Уровневая шкала
		ПКв-2	Тесты	16-24	Процентная шкала
5	Потоки ввода-вывода в C++	ПКв-2	Вопросы к экзамену	37-44	Уровневая шкала
		ПКв-2	Тесты	25-33	Процентная шкала
6	Шаблоны в C++	ПКв-2	Вопросы к экзамену	45-50	Уровневая шкала
		ПКв-2	Тесты	34-35	Процентная шкала

**3 Оценочные материалы для промежуточной аттестации. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

Аттестация обучающегося по дисциплине проводится в форме тестирования и предусматривает возможность последующей сдачи экзамена.

Каждый вариант теста включает 10 контрольных заданий, из них:

- 4 контрольных задания на проверку знаний;
- 3 контрольных задания на проверку умений;
- 3 контрольных задания на проверку навыков.

Каждый билет включает в себя:

- 2 вопроса на проверку знаний;
- 1 задание на проверку умений и навыков.

### 3.1 Тесты (тестовые задания)

**3.1.1 ПКв-1** Способность применять современные методы и инструментальные средства прикладной информатики для автоматизации и информатизации решения прикладных задач различных классов и создания ИС

№ задания	Тестовое задание с вариантами ответов и правильными ответами
1	<p>Для чего предназначен оператор namespace:</p> <ol style="list-style-type: none"> <li>1) для использования классов, переменных и функций из других модулей программы без использования заголовочных файлов</li> <li>2) <b>для заключения в группу объявлений классов, переменных и функций в отдельный контекст со своим именем</b></li> <li>3) для заключения в группу объявлений классов, переменных и функций для использования только в текущем модуле</li> <li>4) все ответы правильные</li> </ol>
2	<p>В чем различие использования следующих выражений #include &lt;...&gt; и #include «...»:</p> <ol style="list-style-type: none"> <li>1) <b>различие заключается в методе поиска препроцессором включаемого файла</b></li> <li>2) в различии использования заголовочных и исходных файлов</li> <li>3) нет различий</li> <li>4) правильного ответа нет</li> </ol>
3	<p>Определите значение переменной a после выполнения фрагмента программы:</p> <pre>int a=2; if (a&lt;5) a=a+12; else a=a-7;</pre> <ol style="list-style-type: none"> <li>1) -5</li> <li>2) <b>14</b></li> <li>3) 5</li> <li>4) 12</li> </ol>
4	<p>Чем отличается цикл do ... while от цикла while?</p> <ol style="list-style-type: none"> <li>1) <b>Цикл do ... while выполняется как минимум один раз</b></li> <li>2) Цикл while выполняется до проверки условия</li> <li>3) Цикл do ... while проверяет условие до выполнения</li> <li>4) Ничем</li> </ol>
5	<p>Отметьте все правильные утверждения о массивах в языке C++.</p> <ol style="list-style-type: none"> <li>1) элементы массива могут быть разных типов</li> <li>2) <b>все элементы массива должны быть одного типа</b></li> <li>3) <b>элементы в памяти расположены рядом</b></li> <li>4) элементы могут быть расположены в памяти по одному</li> <li>5) <b>элементы всегда нумеруются с нуля</b></li> </ol>
6	<p>Какие из следующих объявлений структур имеют правильный синтаксис?</p> <ol style="list-style-type: none"> <li>1) <b><u>struct sA {char a[12]; sA* this_struct};</u></b></li> <li>2) <b><u>struct sA {int i; float f};</u></b></li> <li>3) struct sA {int j, sA this_struct};</li> <li>4) все</li> </ol>
7	<p>Что из себя представляет динамическое выделение памяти:</p> <ol style="list-style-type: none"> <li>1) <b><u>память под объект (переменную) может выделяться не сразу, а в процессе работы программы, освобождение памяти производится вручную</u></b></li> <li>2) память под объект (переменную) может выделяться не сразу, а в процессе работы программы, освобождение памяти производится автоматически после завершения программы</li> <li>3) память под объект (переменную) выделяется каждый раз при обращении к переменной</li> <li>4) Правильного ответа нет</li> </ol>
8	<p>Что не относится к базовым принципам ООП?</p> <ol style="list-style-type: none"> <li>1) <b><u>Преобразование</u></b></li> <li>2) Полиморфизм</li> <li>3) Инкапсуляция</li> <li>4) Наследование</li> </ol>
9	<p>Как в объектно-ориентированном программировании называется способность объекта скрывать свои данные и реализацию от других объектов системы?</p> <ol style="list-style-type: none"> <li>1) <b><u>инкапсуляция</u></b></li> <li>2) полиморфизм</li> <li>3) наследование</li> </ol>

№ задания	Тестовое задание с вариантами ответов и правильными ответами
	4) агрегация
10	Объект в объектно-ориентированном программировании характеризуется 1) <b><u>состоянием, именем и набором методов</u></b> 2) именем и типом данных 3) типом данных 4) набором методов
11	Какая разница между объектом и классом в объектно-ориентированном программировании: 1) <b><u>объект - это экземпляр класса</u></b> 2) класс может иметь много экземпляров, а объект один или ни одного 3) класс описывает категорию, к которой могут либо принадлежать, либо не принадлежать объекты данного класса 4) класс - это исходный код, а объект - скомпилированный и выполняемый код
12	Как в объектно-ориентированном программировании называют объект класса? 1) <b><u>экземпляром</u></b> 2) методом 3) полем 4) записью
13	Термин «наследование» обозначает, что 1) в производных классах присутствует часть состояния родительского класса 2) <b><u>производные классы содержат поля и методы родительского</u></b> 3) производные классы наследуют модификаторы доступа членов родительского класса 4) все ответы верны
14	Соотнесите понятия спецификаторов: а) private объявляет метод или свойство доступным только в том классе в котором он присутствует б) protected объявляет метод или свойство, которые не могут быть доступны из объекта, реализующего класс, но могут быть использованы в дочерних классах в) public объявляет классы и методы, которые могут быть доступны как внутри самого класса, так и в дочерних классах и в объектах, реализовавших класс <b><u>Ответ: а) – 1, б) – 2, в) -3.</u></b>
15	В каких отношениях может находиться один класс с другим: 1) наследования 2) включения 3) использования 4) <b><u>все варианты верны</u></b>

### 3.1.2 ПКв-2 Способность проектировать информационные процессы и системы с использованием инновационных инструментальных средств

№ задания	Тестовое задание с вариантами ответов и правильными ответами
16	Какие утверждения о работе с исключениями в языке C/C++ верны? 1) исключения используются для обработки нештатных ситуаций времени исполнения программы 2) если исключение не будет поймано, то будет вызвана системная функция terminate() 3) исключения используются для обработки нештатных ситуаций времени исполнения программы 4) <b><u>Все варианты верны</u></b>
17	Чем отличается обработка ошибок посредством прерываний от проверки кода возврата? 1) носителем информации о нештатной ситуации являются объекты заранее продуманной иерархии классов исключений 2) блок кода, обрабатывающий нештатную ситуацию при возбуждении исключения может быть размещён на любом вышестоящем по стеку уровне 3) игнорирование кода возврата не приводит к падению приложения 4) <b><u>все ответы верны</u></b>
18	Какие утверждения о блоках try и catch верны? 1) исключение, брошенное вне блока try функции может быть поймано только в блоках catch вызывающей функции 2) друг за другом могут следовать несколько блоков catch подряд

№ задания	Тестовое задание с вариантами ответов и правильными ответами
	<p>3) исключение, брошенное вне блока try функции может быть поймано только в блоках catch вызывающей функции</p> <p>4) <b><u>все ответы верны</u></b></p>
19	<pre>//===== start of sample.cpp ===== #include &lt;stdexcept&gt; void my_open1() {     throw int(8); }  void my_open2() try {     throw std::logic_error(""); } catch(std::runtime_error&amp; err) {}  void my_open3() try {     throw std::runtime_error("&amp;quot;runtime"); } catch(const std::runtime_error* err) {}  struct ERR_ {};  void my_open4() {     try     {         throw ERR_();     }     catch(...) {} }  int main() {     try {     }     catch(...) {         return 5;     }     return 0; }  //===== end of sample.cpp =====</pre> <p>Вызов каких из функций изнутри блока try в функции main() приведёт к завершению программы с результатом 5?</p> <p>1) <b><u>my_open1()</u></b>  2) <b><u>my_open2()</u></b>  3) my_open3()  4) my_open4()</p>
20	<pre>//===== start of sample.cpp ===== struct A {}; struct B {}; struct C {}; struct D {}; struct E {}; struct F {};</pre>

№ задания	Тестовое задание с вариантами ответов и правильными ответами
	<pre> int main() {     try     {         foo();     }     catch(const A&amp; a) {}     catch(static const B*) {}     catch(C c) {}     catch(volatile D* d) {}     catch(virtual E* e) {}     catch(F*&amp;) {}     catch(...)     {         return 5;     }     return 0; } //===== end of sample.cpp ===== </pre> <p>Обработчики исключений какого типа записаны неправильно?</p> <ol style="list-style-type: none"> <li>1) A</li> <li>2) <b>B</b></li> <li>3) C</li> <li>4) D</li> <li>5) <b>E</b></li> <li>6) <b>F</b></li> </ol>
21	<pre> //===== start of sample.cpp ===== struct A {}; struct B {}; struct C {}; struct D {}; struct E {}; struct F {};  void foo(int i) {     try     {         switch(i)         {             case 1: throw A();             case 2: throw B();             case 4: throw D();             default:                 throw F();         }     }     catch(A&amp;) { }     catch(B&amp;) { throw D(); }     catch(D&amp;) { throw C(); }     catch(...) { throw; }  int main(int argc, char* argv[]) {     try     {         foo(argc);     } } </pre>

№ задания	Тестовое задание с вариантами ответов и правильными ответами
	<pre> catch(const A&amp; a) {} catch(const B*) {} catch(C c) {} catch(E* e) {} catch(...) {     return 5; } return 0; } //===== end of sample.cpp ===== </pre> <p>При каких значениях argc программа вернёт значение 5?</p> <ol style="list-style-type: none"> <li>1) 1</li> <li>2) <u>2</u></li> <li>3) 4</li> <li>4) <b>6</b></li> </ol>
22	<p>Какие методы класса не должны бросать исключения наружу?</p> <ol style="list-style-type: none"> <li>1) конструкторы</li> <li>2) <b><u>деструкторы</u></b></li> <li>3) константные методы</li> <li>4) <b><u>безопасные методы</u></b></li> </ol>
23	<p>Какие утверждения про стандартные исключения верны?</p> <ol style="list-style-type: none"> <li>1) стандартная библиотека исключений содержит несколько независимых иерархий классов исключений</li> <li>2) иерархия исключений, используемая в разрабатываемом ПО должна быть основана на исключениях стандартной библиотеки</li> <li>3) <b><u>все классы исключений из стандартной библиотеки являются производными от std::exception</u></b></li> <li>4) <b><u>классы исключений из стандартной библиотеки являются безопасными</u></b></li> </ol>
24	<p>Почему лучше использовать исключения, унаследованные от стандартных исключений?</p> <ol style="list-style-type: none"> <li>1) <b><u>классы исключения из стандартной библиотеки используются объектами и функциями стандартной библиотеки</u></b></li> <li>2) <b><u>иерархию исключения стандартной библиотеки можно без больших усилий использовать в своих программах</u></b></li> <li>3) <b><u>стандартные исключения могут быть использованы и обработаны в любой распространяемой библиотеке классов</u></b></li> <li>4) только исключения, унаследованные от классов стандартной библиотеки, могут быть безопасны</li> </ol>
25	<p>Что понимается под потоком в языке C++</p> <ol style="list-style-type: none"> <li>1) обмен данными между программами</li> <li>2) обмен данными между компилятором и функцией main</li> <li>3) <b><u>механизм ввода-вывода</u></b></li> <li>4) все ответы верны</li> </ol>
26	<p>Что будет выведено в результате</p> <pre> double x = 12.4; cout &lt;&lt; setw(5) &lt;&lt; x &lt;&lt; setw(3)     &lt;&lt; setfill('*') &lt;&lt; "" &lt;&lt; endl; </pre> <ol style="list-style-type: none"> <li>1) <b><u>" 12.4***"</u></b></li> <li>2) "12.40***"</li> <li>3) "12.40"</li> <li>4) ".124e2**"</li> </ol>
27	<p>Какой из стандартных классов используется для вывода строк на терминал:</p> <ol style="list-style-type: none"> <li>1) stringstream</li> <li>2) <b><u>ostream</u></b></li> <li>3) istream</li> </ol>

№ задания	Тестовое задание с вариантами ответов и правильными ответами
	4) ifstream
28	<p>Для того чтобы выполнить чтение из файла с произвольной позиции, надо использовать объект класса</p> <ol style="list-style-type: none"> <li>1) stringstream</li> <li>2) ostream</li> <li>3) ofstream</li> <li>4) <b><u>ifstream</u></b></li> </ol>
29	<p>Результат работы программы:</p> <pre>#include &lt;iostream.h&gt; int main() { int ic; cout &lt;&lt; "Введите любую десятичную цифру:"; cin &gt;&gt; ic; switch (ic) { case 0: case 1: cout &lt;&lt; "один "; case 2: case 3: cout &lt;&lt; "три "; case 4: case 5: cout &lt;&lt; "пять "; case 6: case 7: cout &lt;&lt; "семь "; case 8: case 9: cout &lt;&lt; "девять "; break; default: cout &lt;&lt; "ERROR!!!"; } return 1; }</pre> <ol style="list-style-type: none"> <li>1) компилятор найдет ошибку</li> <li>2) если введена нечетная цифра, выводится ее название</li> <li>3) в любом случае выводится "ERROR!!!"</li> <li>4) <b><u>выводятся названия всех нечетных цифр больше или равных введенной</u></b></li> </ol>
30	<p>Какой класс используется для вывода данных во внутреннюю область памяти?</p> <ol style="list-style-type: none"> <li>1) ostream</li> <li>2) <b><u>stringstream</u></b></li> <li>3) cout</li> <li>4) strout</li> </ol>
31	<p>С помощью какого метода можно изменить текущую позицию в файле?</p> <ol style="list-style-type: none"> <li>1) put</li> <li>2) open</li> <li>3) <b><u>seekp</u></b></li> <li>4) все ответы верны</li> </ol>
32	<p>Что произойдет при выводе в файл, открытый с помощью ofstream("filename", ios::out ios::app ios::trunc)</p> <ol style="list-style-type: none"> <li>1) вывод будет производиться в двоичном виде</li> <li>2) <b><u>вывод будет производиться в конец файла</u></b></li> <li>3) можно читать из файла</li> <li>4) <b><u>если файл существовал, его содержимое сотрется</u></b></li> </ol>
33	<p>Какое из следующих утверждений верно для кода, представленного ниже?</p> <pre>FILE *fp; char ch; int i=1; fp = fopen("myfile.c", "r"); while((ch=getc(fp))!=EOF) { if(ch == '\n') i++;} fclose(fp);</pre>

№ задания	Тестовое задание с вариантами ответов и правильными ответами
	<b>1) Вычисляет количество строк в файле</b> 2) Вычисляет количество слов в файле 3) Вычисляет количество символов в файле 4) Вычисляет количество пустых строк в файле
34	Какие утверждения о параметрах шаблонов верны? <b>1) параметрами шаблона могут быть типы и константы</b> 2) параметры шаблона константы могут быть числовыми и строчными <b>3) параметры шаблона могут иметь значения по умолчанию</b> <b>4) задание значений параметров шаблона происходит при конкретизации шаблона</b>
35	Какие утверждения о шаблонах в языке C++ верны? <b>1) шаблонными могут быть не только классы или функции, но и структуры</b> 2) степень обобщённости класса зависит от количества параметров шаблона <b>3) обобщённое программирование реализуется в языке C++ через шаблоны</b> <b>4) шаблон класса позволяет автоматически создавать конкретный класс по обобщённому описанию класса</b>

### 3.3. Экзамен

**3.3.1 ПКв-1** Способность применять современные методы и инструментальные средства прикладной информатики для автоматизации и информатизации решения прикладных задач различных классов и создания ИС

#### Вопросы для экзамена

№	Текст вопроса
1.	Структурный подход в программировании. Понятие типа данных.
2.	Причины появления объектного подхода. Основные концепции объектно-ориентированного программирования.
3.	Событийное программирование. Проблемы, связанные с одновременным выполнением.
4.	Событийное программирование. Архитектура передачи сообщений в Windows.
5.	Событийное программирование. Архитектура ввода-вывода, независимая от внешних устройств.
6.	Событийное программирование. Связь и взаимодействие между процессами.
7.	Событийное программирование. Событие и сообщение. Виды событий. События от мыши и клавиатуры.
8.	Событийное программирование. Событийное управление.
9.	Событийное программирование. Способы генерации событий.
10.	Событийное программирование. Назначение сообщений.
11.	Событийное программирование. Программирование управления событиями.
12.	Основные концепции объектно-ориентированного программирования. Инкапсуляция свойств объекта
13.	Основные концепции объектно-ориентированного программирования. Наследование свойств.
14.	Основные концепции объектно-ориентированного программирования. Полиморфизм поведенческих свойств объектов.
15.	Основные концепции объектно-ориентированного программирования. Достоинства и недостатки ООП.
16.	Инкапсуляция, наследование, полиморфизм.
17.	Объектно-ориентированный подход к разработке программ.
18.	Объектно-ориентированные средства языка C++.
19.	Основные концепции объектно-ориентированного программирования. Объекты, классы.
20.	Наследование классов и производные классы.
21.	Конструкторы, деструкторы и наследование.
22.	Иерархия классов.

23.	Виртуальные функции.
24.	Абстрактные классы и чистые виртуальные функции.
25.	Множественное наследование.
26.	Виртуальные базовые классы.

**3.3.2 ПКв-2** Способность проектировать информационные процессы и системы с использованием инновационных инструментальных средств

#### Вопросы для экзамена

Номер вопроса (задачи, задания)	Текст вопроса
27.	Обработка ошибок в стандартном C
28.	Распознавание ситуаций.
29.	Использование assert, кодов возврата, сигналов, setjmp и longjmp.
30.	Использование set_new_handler().
31.	Объектно-ориентированная обработка исключений.
32.	Применение try, catch, throw.
33.	Раскрутка стека.
34.	Стандартные исключения в C++.
35.	Работа с конструкторами и исключениями.
36.	Функции terminate(), unexpected().
37.	Потоки в C++. Функции работы со строками и массивы символов в C++
38.	Потоки в C++. Строковые функции.
39.	Потоки в C++. Объекты ofstream и ifstream.
40.	Потоки в C++. Двоичные и текстовые файлы.
41.	Потоки в C++. Операции ввода-вывода для стандартных типов.
42.	Потоки в C++. Организация ввода-вывода для пользовательских типов.
43.	Потоки в C++. Работа с файлами и буферами в памяти.
44.	Потоки в C++. Средства форматирования вывода, манипуляторы.
45.	Шаблоны функций
46.	Шаблоны классов.
47.	Параметры шаблонов.
48.	Наследование и шаблоны.
49.	Примеры построения шаблонов.
50.	Итераторы.

#### 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания в ходе изучения дисциплины знаний, умений и навыков, характеризующих этапы формирования компетенций, регламентируются положениями:

- П ВГУИТ 2.4.03 Положение о курсовых экзаменах и зачетах;
- П ВГУИТ 4.1.02 Положение о рейтинговой оценке текущей успеваемости.

**5. 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 компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
<b>ПКв-1</b> Способность применять современные методы и инструментальные средства прикладной информатики для автоматизации и информатизации решения прикладных задач различных классов и создания ИС					
<p><b>Знать:</b></p> <ul style="list-style-type: none"> <li>- современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС с использованием языка программирования С++;</li> <li>- современные методы и инструментальные средства разработки с использованием языка программирования С++</li> </ul> <p><b>Уметь:</b></p> <ul style="list-style-type: none"> <li>- разрабатывать и проектировать приложения с использованием языка программирования С++;</li> <li>- решать прикладные задачи различных классов и создавать информационные системы с использованием языка программирования С++</li> </ul> <p><b>Владеть:</b></p> <ul style="list-style-type: none"> <li>- методикой решения научных задач с использованием языка программирования С++;</li> </ul>	Тесты (тестовые задания)	<p><b>Знает:</b></p> <ul style="list-style-type: none"> <li>- современные методы и инструментальные средства поддержки технологии проектирования и аудита ИС с использованием языка программирования С++;</li> <li>- современные методы и инструментальные средства разработки с использованием языка программирования С++</li> </ul> <p><b>Умеет:</b></p> <ul style="list-style-type: none"> <li>- разрабатывать и проектировать приложения с использованием</li> </ul>	Студент ответил на 0- 66 % вопросов теста	2 балла	Не освоена (недостаточный)
			Студент ответил на 66,1- 75 % вопросов теста	3 балла	Освоена (базовый)
			Студент правильно ответил на 75,1 - 85 % вопросов теста	4 балла	Освоена (повышенный)
			Студент правильно ответил на 85,1-100 % вопросов теста	5 баллов	Освоена (повышенный)
	Вопросы к экзамену		Студент не ответил на все вопросы, допустил более 3 ошибок	неудовлетв.	Не освоена (недостаточный)
			Студент ответил не на все вопросы, но в тех, на которые дал ответ не допустил ошибки	удовлетвор.	Освоена (базовый)
			Студент ответил на все вопросы и допустил более 1 ошибки, но менее 3 ошибок	хорошо	Освоена (повышенный)

Результаты обучения по этапам формирования компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
- методикой решения прикладных задач различных классов с использованием языка программирования С++		языка программирования С++; - решать прикладные задачи различных классов и создавать информационные системы с использованием языка программирования С++; <b>Владеет:</b> - методикой решения научных задач с использованием языка программирования С++; - методикой решения прикладных задач различных классов с использованием языка программирования С++	Студент ответил на все вопросы и допустил не более 1 ошибки в ответе	отлично	Освоена (повышенный)
<b>ПКв-2</b> Способность проектировать информационные процессы и системы с использованием инновационных инструментальных средств					
<b>Знать:</b> - современные способы и методы проектирования информационных процессов и систем	Тесты (тестовые задания)	<b>Знает:</b> - современные способы и методы проекти-	Студент ответил на 0- 66 % вопросов теста	2 балла	Не освоена (недостаточный)
			Студент ответил на 66,1- 75 % вопросов теста	3 балла	Освоена (базовый)

Результаты обучения по этапам формирования компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
<p>тем с использованием языка C++;</p> <p>- особенности современных СУБД</p> <p><b>Уметь:</b></p> <p>- проектировать информационные процессы и системы с использованием инновационных с использованием языка C++;</p> <p>- создавать основные объекты БД</p> <p><b>Владеть:</b></p> <p>- методологией проектирования информационных процессов и систем с использованием языка C++;</p> <p>- методологией проектирования современных БД</p>	Вопросы к экзамену	<p>рования информационных процессов и систем с использованием языка C++;</p> <p>- особенности современных СУБД</p> <p><b>Умеет:</b></p> <p>- проектировать информационные процессы и системы с использованием инновационных с использованием языка C++;</p> <p>- создавать основные объекты БД</p> <p><b>Владеет:</b></p> <p>- методологией проектирования информационных процессов и систем с использованием языка C++;</p> <p>- методологией проектирования современных БД</p>	Студент правильно ответил на 75,1 - 85 % вопросов теста	4 балла	Освоена (повышенный)
			Студент правильно ответил на 85,1-100 % вопросов теста	5 баллов	Освоена (повышенный)
			Студент не ответил на все вопросы, допустил более 3 ошибок	неудовлетв.	Не освоена (недостаточный)
			Студент ответил не на все вопросы, но в тех, на которые дал ответ не допустил ошибки	удовлетвор.	Освоена (базовый)
			Студент ответил на все вопросы и допустил более 1 ошибки, но менее 3 ошибок	хорошо	Освоена (повышенный)
			Студент ответил на все вопросы и допустил не более 1 ошибки в ответе	отлично	Освоена (повышенный)