

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Гаранин Максим Александрович
Должность: Ректор
Дата подписания: 08.10.2025 09:02:47
Уникальный программный ключ:
7708e3a47e66a8ee02711b298d7c78bd1e40bf88

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПРИВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ СООБЩЕНИЯ»

Объектно-ориентированное программирование рабочая программа дисциплины (модуля)

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) Проектирование АСОИУ на транспорте

Квалификация **бакалавр**

Форма обучения **очная**

Общая трудоемкость **7 ЗЕТ**

Виды контроля в семестрах:

экзамены 3

зачеты 2

курсовые работы 3

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	2 (1.2)		3 (2.1)		Итого	
	уп	рп	уп	рп	уп	рп
Неделя	16,5		16,3			
Вид занятий	уп	рп	уп	рп	уп	рп
Лекции	16	16	16	16	32	32
Лабораторные	16	16	16	16	32	32
Практические			16	16	16	16
Конт. ч. на аттест.			1,5	1,5	1,5	1,5
Конт. ч. на аттест. в период ЭС	0,25	0,25	2,35	2,35	2,6	2,6
В том числе в форме практ.подготовки	16	16	66	66	82	82
Итого ауд.	32	32	48	48	80	80
Контактная работа	32,25	32,25	51,85	51,85	84,1	84,1
Сам. работа	31	31	103,5	103,5	134,5	134,5
Часы на контроль	8,75	8,75	24,65	24,65	33,4	33,4
Итого	72	72	180	180	252	252

Программу составил(и):

к.т.н., доцент, Зав. кафедрой, Авсиевич А.В.

Рабочая программа дисциплины

Объектно-ориентированное программирование

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

составлена на основании учебного плана: 09.03.01-25-4-ИВТб.plm.plx

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) Проектирование АСОИУ на транспорте

Рабочая программа одобрена на заседании кафедры

Цифровые технологии

Зав. кафедрой к.э.н., доцент Ефимова Т.Б.

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
1.1	формирование компетенций для осуществления задач профессиональной деятельности в области объектно-ориентированного программирования, обучении студентов принципам, используемым при разработке и реализации иерархии классов объектов, современным визуальным средствам разработки и создания программ с использованием взаимодействующих объектов.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.03

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ПК-1 Способен разрабатывать требования и проектировать программное обеспечение	
ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня	
ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня	
06.028. Профессиональный стандарт "СИСТЕМНЫЙ ПРОГРАММИСТ", утверждённый приказом Министерства труда и социальной защиты Российской Федерации от 29 сентября 2020 г. N 678н (зарегистрирован Министерством юстиции Российской Федерации 26 ноября 2020 г., регистрационный N 60582)	
ПК-1. А. Разработка компонентов системных программных продуктов	
А/04.6 Создание инструментальных средств программирования	

В результате освоения дисциплины (модуля) обучающийся должен

3.1 Знать:	
3.1.1	объектно-ориентированное программирование;
3.1.2	общие принципы конструирования программ с использованием объектно-ориентированной парадигмы.
3.2 Уметь:	
3.2.1	разрабатывать и реализовывать классы объектов;
3.2.2	использовать визуальную среду программирования.
3.3 Владеть:	
3.3.1	современными средствами разработки объектно-ориентированных систем;
3.3.2	объектной декомпозицией и проектированием.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Примечание
	Раздел 1. Лекции			
1.1	Объектно-ориентированный подход. Основные принципы объектно-ориентированного программирования. /Лек/	2	2	
1.2	Дружественные функции и перегрузка операций. Перегрузка унарных и бинарных операторов, операторов преобразования типов, операторов присваивания и индексирования. /Лек/	2	4	
1.3	Приведение типов на этапе выполнения программы. Информация о типе данных. Пространство имен. /Лек/	2	4	
1.4	Объектно-ориентированное проектирование. Классы. Описания протокола класса. Классы и методы в языке C++. Ключевое слово this. Функции типа inline. Управление доступом к элементам классов. Функции-члены класса. Конструкторы и деструкторы /Лек/	2	6	
1.5	Виртуальные функции и полиморфизм. /Лек/	3	4	
1.6	Абстрактные классы и интерфейсы. Производные классы: множественное наследование. Конфликты имен. Порядок вызова конструкторов. Виртуальные базовые классы. Множественное наследование /Лек/	3	4	
1.7	Шаблоны и контейнеры. Шаблоны функций. Шаблоны классов. /Лек/	3	4	
1.8	Обработка исключительных ситуаций. Спецификация исключений. Иерархия классов - исключений. Перехват исключительных ситуаций.	3	4	
	Раздел 2. Лабораторные работы			
2.1	Создать структуру классов /Лаб/	2	2	Практическая подготовка
2.2	Композиция классов /Лаб/	2	2	Практическая подготовка

2.3	Наследование. /Лаб/	2	4	Практическая подготовка
2.4	Перегрузка операций /Лаб/	2	4	Практическая подготовка
2.5	Параметризованные классы (шаблоны классов). Перегруженные функции и функции-шаблоны. /Лаб/	2	4	Практическая подготовка
2.6	Использование классов и алгоритмов библиотеки STL. /Лаб/	3	4	Практическая подготовка
2.7	Разработка классов с учетом обработки исключительных ситуаций /Лаб/	3	6	Практическая подготовка
2.8	Транзакции /Лаб/	3	6	Практическая подготовка
Раздел 3. Практические работы				
3.1	Программирование вычислительных процессов с использованием рекурсивных алгоритмов /Пр/	3	4	Практическая подготовка
3.2	Объектно-ориентированная декомпозиция прикладной задачи /Пр/	3	6	Практическая подготовка
3.3	Разработка программ средствами языка Си++ в объектной парадигме /Пр/	3	6	Практическая подготовка
Раздел 4. Самостоятельная работа				
4.1	Абстракция, инкапсуляция, наследование и полиморфизм. Дополнения к С. Прототипы функций. Операции расширения области видимости. /Ср/	2	2	
4.2	Изучение теоретического материала /Ср/	2	5	
4.3	Массивы объектов. Конструктор копирования. Статические члены и статические функции. Константные объекты, члены классов и функции-члены. /Ср/	3	5,5	
4.4	Правила доступа для классов и объектов. Правила доступа для друзей классов и производных классов. /Ср/	3	6	
4.5	Позднее связывание. Виртуальные деструкторы. Замещение и уточнение. Таблицы виртуальных методов. Виртуальные деструкторы. /Ср/	3	6	
4.6	Конфликты имен. Порядок вызова конструкторов. Виртуальные базовые классы. Множественное наследование /Ср/	3	11	
4.7	Выполнение курсовой работы /Ср/	3	35	Практическая подготовка
4.8	Подготовка к лекциям /Ср/	2	8	
4.9	Подготовка к лекциям /Ср/	3	8	
4.10	Подготовка к лабораторным занятиям /Ср/	2	16	
4.11	Подготовка к лабораторным занятиям /Ср/	3	16	
4.12	Подготовка к практическим занятиям /Ср/	3	16	
Раздел 5. Контактные часы на аттестацию				
5.1	Зачет /КЭ/	2	0,25	
5.2	Курсовая работа /КА/	3	1,5	
5.3	Экзамен /КЭ/	3	2,35	
5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ				
<p>Оценочные материалы для проведения промежуточной аттестации обучающихся приведены в приложении к рабочей программе дисциплины.</p> <p>Формы и виды текущего контроля по дисциплине (модулю), виды заданий, критерии их оценивания, распределение баллов по видам текущего контроля разрабатываются преподавателем дисциплины с учетом ее специфики и доводятся до сведения обучающихся на первом учебном занятии.</p> <p>Текущий контроль успеваемости осуществляется преподавателем дисциплины (модуля) в рамках контактной работы и самостоятельной работы обучающихся. Для фиксирования результатов текущего контроля может использоваться ЭИОС.</p>				

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
6.1. Рекомендуемая литература				
6.1.1. Основная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Скворцова Л. А.	Объектно-ориентированное программирование на языке C++: Учебное пособие	Москва: МИРЭА, 2020	https://e.lanbook.com/book/163862?category=1557
Л1.2	Огнева М. В., Кудрина Е. В.	Программирование на языке C++: практический курс: учебное пособие для вузов	Москва: Юрайт, 2021	https://urait.ru/bcode/473054
6.1.2. Дополнительная литература				
	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Подбельский В. В.	Программирование. Базовый курс C#: учебник для вузов	Москва: Юрайт, 2020	https://urait.ru/bcode/450868
6.2 Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине (модулю)				
6.2.1 Перечень лицензионного и свободно распространяемого программного обеспечения				
6.2.1.1	Microsoft Windows			
6.2.2 Перечень профессиональных баз данных и информационных справочных систем				
6.2.2.1	База книг и публикаций Электронной библиотеки "Наука и Техника"- http://www.n-t.ru			
6.2.2.2	Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки- https://github.com/			
6.2.2.3	Портал для разработчиков электронной техники: http://www.espec.ws/			
6.2.2.4	База данных «Библиотека программиста» https://proglib.io/			
6.2.2.5	Консультант плюс			
6.2.2.6	Информационная система ГАРАНТ			
7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
7.1	Учебные аудитории для проведения занятий лекционного типа, укомплектованные специализированной мебелью и техническими средствами обучения: мультимедийное оборудование для предоставления учебной информации большой аудитории и/или звукоусиливающее оборудование (стационарное или переносное).			
7.2	Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, укомплектованные специализированной мебелью и техническими средствами обучения: мультимедийное оборудование и/или звукоусиливающее оборудование (стационарное или переносное)			
7.3	Помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета.			
7.4	Помещения для хранения и профилактического обслуживания учебного оборудования			
7.5	Помещения для выполнения курсовых работ укомплектованы специализированной мебелью и техническими средствами обучения (стационарными или переносными).			
7.6	Учебные аудитории для проведения лабораторных работ укомплектованы специализированной мебелью и техническими средствами обучения: ноутбуки или компьютеры, подключенные к локальной сети ПривГУПС.			

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Объектно-ориентированное программирование

(наименование дисциплины (модуля))

Направление подготовки / специальность

09.03.01 Информатика и вычислительная техника

(код и наименование)

Направленность (профиль) / специализация

«Проектирование АСОИУ на транспорте»

(наименование)

Содержание

1. Пояснительная записка.
2. Типовые контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций.
3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации.

1. Пояснительная записка

Цель промежуточной аттестации – оценивание промежуточных и окончательных результатов обучения по дисциплине, обеспечивающих достижение планируемых результатов освоения образовательной программы.

Формы промежуточной аттестации:

Очная форма обучения:

- зачет 2 семестр
- курсовая работа 3 семестр
- экзамен 3 семестр.

Перечень компетенций, формируемых в процессе освоения дисциплины

Код и наименование компетенции	Код индикатора достижения компетенции
ПК-1: Способен разрабатывать требования и проектировать программное обеспечение	ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня

Результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине	Оценочные материалы (семестр)
ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня	Обучающийся знает: объектно-ориентированное программирование	Вопросы (№1–№5)
	Обучающийся умеет: разрабатывать и реализовывать классы объектов	Задания (№1–№3)
	Обучающийся владеет: современными средствами разработки объектно-ориентированных систем	Задания (№1–№3)
ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня	Обучающийся знает: общие принципы конструирования программ с использованием объектно-ориентированной парадигмы	Вопросы (№6–№10)
	Обучающийся умеет: использовать визуальную среду программирования	Задания (№4–№6)
	Обучающийся владеет: объектной декомпозицией и проектированием;	Задания (№4–№6)

Промежуточная аттестация (зачет) проводится в одной из следующих форм:

- 1) собеседование
- 2) выполнение заданий в ЭИОС университета.

Промежуточная аттестация (курсовая работа) проводится в одной из следующих форм:

- 1) собеседование
- 2) выполнение заданий в ЭИОС университета.

Промежуточная аттестация (Экзамен) проводится в одной из следующих форм

- 1) ответ на билет, состоящий из теоретических вопросов и практических заданий
- 2) выполнение заданий в ЭИОС университета.

2. Типовые¹ контрольные задания или иные материалы для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций

¹ Приводятся типовые вопросы и задания. Оценочные средства, предназначенные для проведения аттестационного мероприятия, хранятся на кафедре в достаточном для проведения оценочных процедур количестве вариантов. Оценочные средства подлежат актуализации с учетом развития науки, образования, культуры, экономики, техники, технологий и

2.1 Типовые вопросы (тестовые задания) для оценки знаниевого образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня	Обучающийся знает: объектно-ориентированное программирование
<p><i>Вопросы:</i></p> <ol style="list-style-type: none"> 1. Какие требования должны учитываться при разработке компонентных объектов? <ul style="list-style-type: none"> • четкая специализация компонента с точки зрения конкретной решаемой задачи • возможность взаимодействия с другими компонентами • возможность использования компонентов в инструментах быстрой разработки приложений • соблюдение правил используемой при разработке компонентной модели 2. Какие утверждения справедливы относительно механизма сериализации? <ul style="list-style-type: none"> • сериализация позволяет сохранять объекты в файлах в виде потока байтов • сериализация применима для объектов любых классов, которые реализуют интерфейс Serializable • сериализация позволяет сохранять и восстанавливать объектную структуру любой сложности • сериализация основана на использовании механизма позднего связывания 3. Какие преимущества дает использование компонентного подхода при разработке приложений? <ul style="list-style-type: none"> • сокращение объема кода, написанного «вручную» • уменьшение сроков разработки приложений • получение более быстрого программного кода • возможность максимально полного использования аппаратных особенностей процессоров 4. Какие шаги включает в себя процесс разработки оконных приложений с помощью стандартных компонентов? <ul style="list-style-type: none"> • размещение необходимых компонентов на форме • установка свойств размещенных на форме компонентов • написание необходимых обработчиков событий • разработка используемых компонентных классов 5. Какие утверждения справедливы относительно понятия «компонент»? <ul style="list-style-type: none"> • компонент — это объект специального вида • компоненты являются экземплярами компонентных классов • компоненты являются основой средств быстрой разработки приложений • компоненты используются только для создания оконного пользовательского интерфейса 	
ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня	Обучающийся знает: общие принципы конструирования программ с использованием объектно-ориентированной парадигмы
<p><i>Вопросы:</i></p> <ol style="list-style-type: none"> 6. Какие утверждения относительно методов-конструкторов являются правильными? <ul style="list-style-type: none"> • конструктор выполняет инициализацию свойств объекта • конструктор вызывается раньше всех остальных методов • конструктор отвечает за создание объекта при выполнении программы • конструктор отвечает за освобождение памяти, выделенной объекту 7. Какие типы свойств может содержать объект? <ul style="list-style-type: none"> • основные стандартные базовые типы • структурированные свойства-массивы • объектные свойства • программные свойства 8. Какие типы методов обычно содержат классы? <ul style="list-style-type: none"> • конструкторы • методы доступа к свойствам • методы, реализующие функциональность объектов класса • деструкторы 9. Какие утверждения относительно понятия «Программный объект» являются правильными? <ul style="list-style-type: none"> • с каждым объектом связываются данные и программный код 	

- программный объект является моделью исходного объекта
- объект существует при выполнении программы, занимая часть оперативной памяти
- объекты используются для описания только физических сущностей
- 10. Какие утверждения относительно методов-деструкторов являются правильными?
 - деструктор отвечает за освобождение памяти, выделенной объекту
 - деструкторы реализованы не во всех объектных языках
 - в языках Java и C# вместо деструкторов используется механизм сборки мусора
 - деструктор отсутствуют в языке C++

2.2 Типовые задания для оценки навыкового образовательного результата

Проверяемый образовательный результат:

Код и наименование индикатора достижения компетенции	Образовательный результат
ПК-1.3 Разрабатывает программный код на языках программирования высокого уровня	Обучающийся умеет: разрабатывать и реализовывать классы объектов Обучающийся владеет: современными средствами разработки объектно-ориентированных систем
<i>Пример задания:</i> <ol style="list-style-type: none"> 1. Декомпозировать прикладную задачу под объектно-ориентированный подход 2. Разработать программу средствами языка Си++ в объектной парадигме 3. Разработать программу используя классы и алгоритмы библиотеки STL. 	
ПК-1.4 Осуществляет отладку программ, написанных на языке высокого уровня	Обучающийся умеет: использовать визуальную среду программирования Обучающийся владеет: объектной декомпозицией и проектированием;
<i>Пример задания</i> <ol style="list-style-type: none"> 4. Программирование вычислительных процессов с использованием рекурсивных алгоритмов 5. Разработать классы с учетом обработки исключительных ситуаций 6. Разработать параметризованные классы (шаблоны классов). Перегруженные функции и функции-шаблоны. 	

2.3. Перечень вопросов для подготовки обучающихся к промежуточной аттестации

Вопросы к зачету:

1. Общая характеристика и основные принципы Объектно-ориентированного программирования.
2. Определение класса в C++. Понятие объектов.
3. Функциональные компоненты класса. Типы функциональных компонент.
4. Конструкторы.
5. Деструкторы.
6. Перегрузка функциональных элементов и операций.
7. Константные объекты и функции. Ключевое слово this.
8. Наследование классов. Конструкторы и деструкторы при наследовании.
9. Виртуальные функции. Понятие полиморфизма и позднего связывания.
10. Неоднозначность при множественном наследовании. Виртуальный базовый класс.
11. Абстрактный класс и чистые виртуальные функции.
12. Дружественные функции и классы.
13. Ссылки.
14. Поточковые классы. Операции извлечения и помещения данных.
15. Форматирование данных. Флаги и форматирующие методы.
16. Манипуляторы.
17. Методы обмена с потоками.
18. Файловые потоки.
19. Ошибки потоков.
20. Обработка исключительных ситуаций. Понятие исключений. Общий механизм обработки исключений.
21. Синтаксис исключений. Перехват исключений.
22. Списки исключений функций. Исключения в конструкторах и деструкторах.

23. Стандартные исключения. Иерархии исключений.
24. Шаблоны функций.

Вопросы к экзамену:

1. Общая характеристика и основные принципы Объектно-ориентированного программирования.
2. Определение класса в C++. Понятие объектов.
3. Функциональные компоненты класса. Типы функциональных компонент.
4. Конструкторы.
5. Деструкторы.
6. Перегрузка функциональных элементов и операций.
7. Константные объекты и функции. Ключевое слово `this`.
8. Наследование классов. Конструкторы и деструкторы при наследовании.
9. Виртуальные функции. Понятие полиморфизма и позднего связывания.
10. Неоднозначность при множественном наследовании. Виртуальный базовый класс.
11. Абстрактный класс и чистые виртуальные функции.
12. Дружественные функции и классы.
13. Классы потоков языка C++. Операции извлечения и помещения данных.
14. Способы форматирования при работе с классами потоков. Форматирующие функции.
15. Флаги форматирования.
16. Манипуляторы.
17. Ошибки потоков.
18. Файловый ввод-вывод с применением потоков. Открытие файла.
19. Ввод-вывод в файлы с использованием потоков.
20. Строко - ориентированный ввод-вывод.
21. Неформатированный ввод-вывод.
22. Пространство имен.
23. Обработка исключительных ситуаций. Понятие исключений. Общий механизм обработки исключений.
24. Синтаксис исключений. Перехват исключений.
25. Списки исключений функций. Исключения в конструкторах и деструкторах.
26. Стандартные исключения. Иерархии исключений.
27. Шаблоны функций. Использование шаблонов функций.
28. Специализированная функция шаблона. Перегрузка шаблонов.
29. Определение и использование шаблонов классов.
30. Использование в шаблонных классах аргументов по умолчанию. Специализация шаблонов классов. Достоинства и недостатки шаблонов.
31. Класс `string`. Конструкторы. Преобразование строк.
32. Класс `string`. Поиск подстрок. Сравнение частей строк.
33. Класс `string`. Получение количества элементов. Изменение размера строки.
34. Библиотека шаблонов STL. Назначение STL и состав библиотеки.
35. Понятие контейнера и их типы. Итератора. Поля контейнеров.
36. Итераторы. Их типы. Методы для просмотра контейнеров с помощью итераторов.
37. Последовательные контейнеры. Общие операции для работы с контейнерами.
38. Контейнерный класс `vector`. Его конструкторы. Операции присваивания и копирования. Доступ к элементам вектора.
39. Контейнерный класс `vector`. Его методы `capacity`, `reserve`, `resize`. Методы для изменения объектов класса `vector`.
40. Векторы логических значений.
41. Двухсторонние очереди (`deque`). Конструкторы. Операции, которые реализованы в `deque`. Методы добавления и выборки.
42. Списки `list`. Понятие и реализация. Доступ к элементам. Занесение в начало и конец.
43. Методы изменения объектов списка. Сцепление списков. Удаление элементов. Сортировка.
44. Адаптеры стеки и очереди.

2.4. Перечень примерных тем курсовых работ

1. Разработка программы управления ИТ-проектами: задача подбора персонала.
2. Разработка программы управления ИТ-проектами: задача составления расписания выполнения проекта.
3. Разработка программы управления ИТ-проектами: задача прогнозирования себестоимости ИТ-проекта
4. Разработка программного обеспечения учебного тренажера по булевым функциям
5. Разработка математических методов и программ оптимизации проектирования сетей передачи данных
6. Разработка программной системы стеганографического встраивания информации в цифровое изображение
7. Разработка программной системы стеганографического встраивания информации в цифровое аудио.
8. Разработка и программная реализация атак на методы встраивания информации, реализованные в существующем стеганографическом программном обеспечении.
9. Разработка программной системы встраивания цифрового водяного знака в цифровое изображение
10. Создание программы на выбранную тематику в среде MS Access.
11. Создание программы на выбранную тематику в среде Microsoft SQL Server.
12. Создание программы на выбранную тематику с использованием базы данных MySQL с доступом через Web-интерфейс.
13. Разработка программного комплекса для обработки данных на выбранную тематику.
14. Разработка методики и программного комплекса для хранения и обработки данных расчётных систем.
15. Программное обеспечение системы резервирования билетов
16. Разработка параллельных программ для многопроцессорных систем
17. Создание программы (ИС), поддерживающей реестр юридических лиц
18. Разработка программы автоматизации работы деканата ВУЗа
19. Разработка программы автоматизации учета изделий на предприятии
20. Разработка программы автоматизации автоматизированного рабочего места оператора библиотеки
21. Разработка программы автоматизации предприятий автосервиса
22. Разработка программы автоматизации учета кадров на предприятии
23. Разработка программы автоматизации учета занятости аудиторий в ВУЗе
24. Разработка программы автоматизации паспортно-визовой службы
25. Разработка программы автоматизации формирования модели бюджета
26. Разработка программы автоматизации проектной организации
27. Разработка программы автоматизации строительной организации
28. Разработка программы автоматизации гостиничного комплекса
29. Разработка программы автоматизации аптеки
30. Разработка программы автоматизации туристической фирмы

3. Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Критерии формирования оценок по ответам на вопросы, выполнению тестовых заданий

- оценка «отлично» выставляется обучающемуся, если количество правильных ответов на вопросы составляет 100 – 90% от общего объёма заданных вопросов;
- оценка «хорошо» выставляется обучающемуся, если количество правильных ответов на вопросы – 89 – 76% от общего объёма заданных вопросов;
- оценка «удовлетворительно» выставляется обучающемуся, если количество правильных ответов на тестовые вопросы – 75–60 % от общего объёма заданных вопросов;
- оценка «неудовлетворительно» выставляется обучающемуся, если количество правильных ответов – менее 60% от общего объёма заданных вопросов.

Критерии формирования оценок по результатам выполнения заданий

«Отлично/зачтено» – ставится за работу, выполненную полностью без ошибок и недочетов.

«Хорошо/зачтено» – ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно/зачтено» – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и двух недочетов.

«Неудовлетворительно/не зачтено» – ставится за работу, если число ошибок и недочетов превысило норму для оценки «удовлетворительно» или правильно выполнено менее 2/3 всей работы.

Виды ошибок:

- *грубые ошибки: незнание основных понятий, правил, норм; незнание приемов решения задач; ошибки, показывающие неправильное понимание условия предложенного задания.*

- *негрубые ошибки: неточности формулировок, определений; нерациональный выбор хода решения.*

- *недочеты: нерациональные приемы выполнения задания; отдельные погрешности в формулировке выводов; небрежное выполнение задания.*

Критерии формирования оценок по зачету

«Зачтено» - обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности.

«Не зачтено» - выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.

Критерии формирования оценок по написанию и защите курсовой работы

«Отлично» (5 баллов) – получают обучающиеся, оформившие курсовую работу (курсовой проект) в соответствии с предъявляемыми требованиями, в которой отражены все необходимые результаты проведенного анализа, сделаны обобщающие выводы и предложены рекомендации в соответствии с тематикой курсовой работы, а также грамотно и исчерпывающе ответившие на все встречные вопросы преподавателя.

«Хорошо» (4 балла) – получают обучающиеся, оформившие курсовую работу (курсовой проект) в соответствии с предъявляемыми требованиями, в которой отражены все необходимые результаты проведенного анализа, сделаны обобщающие выводы и предложены рекомендации в соответствии с тематикой курсовой работы. При этом при ответах на вопросы преподавателя обучающийся допустил не более двух ошибок.

«Удовлетворительно» (3 балла) – получают обучающиеся, оформившие курсовую работу (курсовой проект) в соответствии с предъявляемыми требованиями. При этом при ответах на вопросы преподавателя обучающийся допустил более трёх ошибок.

«Неудовлетворительно» (0 баллов) – ставится за курсовую работу, если число ошибок и недочетов превысило удовлетворительный уровень компетенции

Критерии формирования оценок по экзамену

«Отлично» (5 баллов) – обучающийся демонстрирует знание всех разделов изучаемой дисциплины: содержание базовых понятий и фундаментальных проблем; умение излагать программный материал с демонстрацией конкретных примеров. Свободное владение материалом

должно характеризоваться логической ясностью и четким видением путей применения полученных знаний в практической деятельности, умением связать материал с другими отраслями знания.

«Хорошо» (4 балла) – обучающийся демонстрирует знания всех разделов изучаемой дисциплины: содержание базовых понятий и фундаментальных проблем; приобрел необходимые умения и навыки, освоил вопросы практического применения полученных знаний, не допустил фактических ошибок при ответе, достаточно последовательно и логично излагает теоретический материал, допуская лишь незначительные нарушения последовательности изложения и некоторые неточности. Таким образом данная оценка выставляется за правильный, но недостаточно полный ответ.

«Удовлетворительно» (3 балла) – обучающийся демонстрирует знание основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. Однако знание основных проблем курса не подкрепляется конкретными практическими примерами, не полностью раскрыта сущность вопросов, ответ недостаточно логичен и не всегда последователен, допущены ошибки и неточности.

«Неудовлетворительно» (0 баллов) – выставляется в том случае, когда обучающийся демонстрирует фрагментарные знания основных разделов программы изучаемого курса: его базовых понятий и фундаментальных проблем. У экзаменуемого слабо выражена способность к самостоятельному аналитическому мышлению, имеются затруднения в изложении материала, отсутствуют необходимые умения и навыки, допущены грубые ошибки и незнание терминологии, отказ отвечать на дополнительные вопросы, знание которых необходимо для получения положительной оценки.