

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Уральский государственный экономический университет»

Одобрена

на заседании кафедры

10.01.2020 г.

протокол № 6

Зав. кафедрой Сурнина Н.М.

Утверждена

Советом по учебно-методическим вопросам
и качеству образования

15 января 2020 г.

протокол № 5

Председатель



Карх Д.А.

(подпись)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины	Программная инженерия
Направление подготовки	02.03.03 Математическое обеспечение и администрирование информационных систем
Профиль	Разработка и администрирование информационных систем
Форма обучения	очная
Год набора	2020

Разработана:
Доцент, к.э.н.
Кислицын Е.В.

Екатеринбург
2020 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
3. ОБЪЕМ ДИСЦИПЛИНЫ	3
4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП	3
5. ТЕМАТИЧЕСКИЙ ПЛАН	4
6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ	4
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	8
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	8
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	9
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	10

ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы бакалавриата, разработанной в соответствии с ФГОС ВО

ФГОС ВО	Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата) (приказ Минобрнауки России от 23.08.2017г. №809)
ПС	

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

изучение современных инженерных принципов (методов) создания надежного, качественного программного обеспечения, удовлетворяющего предъявляемым к нему требованиям; формирование у студентов умений и навыков применения методов программной

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к базовой части учебного плана.

3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточный контроль	Часов					З.е.
	Всего за семестр	Контактная работа (по уч.зан.)			Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лекции	Лабораторные		
Семестр 5						
Экзамен	144	42	14	28	66	4

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

Общепрофессиональные компетенции (ОПК)

Шифр и наименование компетенции	Индикаторы достижения компетенций
ОПК-3 Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ИД-1.ОПК-3 Знать: основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов. Уметь: решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности. Иметь навыки: иметь практические навыки разработки современного программного обеспечения с учетом требований информационной безопасности.

Профессиональные компетенции (ПК)

Шифр и наименование компетенции	Индикаторы достижения компетенций
производственно-технологический	

ПК-1 Способен применять современные информационные технологии при проектировании и реализации программного обеспечения для решения задач в различных предметных областях	ИД-1.ПК-1 Знать: современные технологии проектирования и производства программного продукта. Уметь: использовать подобные технологии при создании программных продуктов, анализировать исходные данные и применять методы моделирования информационных процессов. Иметь навыки: разработки моделей и информационных процессов; проектирования структур и баз данных.
--	--

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Наименование темы	Всего часов	Контактная работа (по уч.зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
			Часов				
Семестр 5		108					
Тема 1.	Жизненный цикл программного продукта	12	2			10	
Тема 2.	Анализ требований и определение спецификаций программного обеспечения.	18	2	4		12	
Тема 3.	Проектирование программного обеспечения.	24	4	6		14	
Тема 4.	Конструирование программного обеспечения.	18	2	6		10	
Тема 5.	Тестирование, сопровождение и оценка качества программного обеспечения.	18	2	6		10	
Тема 6.	Индивидуальная и коллективная разработка программного обеспечения.	18	2	6		10	

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Темы 1-2	Тест (приложение 4)	Тест содержит 29 вопросов	10 баллов
Темы 3-4	Практическая работа (приложение 4)	Содержит практические задания	10 баллов
Темы 5-6	Практическая работа (приложение 4)	Содержит практические задания	10 баллов
Промежуточный контроль (Приложение 5)			
5 семестр (Эк)	Экзаменационный билет (приложение 5)	Билет содержит 2 теоретических вопроса и 1 практическое задание	100 баллов

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показатель оценки освоения ОПОП формируется на основе объединения текущей и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущая аттестация. Используется 100-балльная система оценивания. Оценка работы студента в течении семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущей аттестации, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончанию дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончанию формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

Показатель оценки	По 5-балльной системе	Характеристика показателя
100% - 85%	отлично	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне
84% - 70%	хорошо	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответа и т.д.)
69% - 50%	удовлетворительно	обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно.
49 % и менее	неудовлетворительно	обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для решения
100% - 50%	зачтено	характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно»
49 % и менее	не зачтено	характеристика показателя соответствует «неудовлетворительно»

7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7.1. Содержание лекций

Тема 1. Жизненный цикл программного продукта
Особенности создания программного продукта. Жизненный цикл программы

<p>Тема 2. Анализ требований и определение спецификаций программного обеспечения. Определение требований к программным продуктам. Выбор архитектуры программного обеспечения. Структура и формат данных. Статические, полустатические и динамические структуры. Модульное программирование. Анализ требований и определение спецификаций при структурном подходе. Анализ требований и определение спецификаций при объектном подходе</p>
<p>Тема 3. Проектирование программного обеспечения. Проектирование программного обеспечения при структурном подходе. Проектирование программного обеспечения при объектном подходе. Экстремальное программирование</p>
<p>Тема 4. Конструирование программного обеспечения. Конструирование ПО.</p>
<p>Тема 5. Тестирование, сопровождение и оценка качества программного обеспечения. Тестирование ПО. Сопровождение ПО. Оценка качества ПО.</p>
<p>Тема 6. Индивидуальная и коллективная разработка программного обеспечения. Инструментальные средства разработки программ. Технологии программирования. Защита программных продуктов. Пакеты прикладных программ. Системы контроля версий.</p>

7.2 Содержание практических занятий и лабораторных работ

<p>Тема 2. Анализ требований и определение спецификаций программного обеспечения. Использование структурно-функциональных диаграмм. Использование Use-Case диаграмм.</p>
<p>Тема 3. Проектирование программного обеспечения. Проектирование программного обеспечения средствами UML.</p>
<p>Тема 4. Конструирование программного обеспечения. Конструирование программного обеспечения в инструментальной среде.</p>
<p>Тема 5. Тестирование, сопровождение и оценка качества программного обеспечения. Основные методики тестирования ПО.</p>
<p>Тема 6. Индивидуальная и коллективная разработка программного обеспечения. Изучение систем контроля версий.</p>

7.3. Содержание самостоятельной работы

<p>Тема 1. Жизненный цикл программного продукта Изучение литературы и интернет-источников по теме. Выполнение практических работ.</p>
<p>Тема 2. Анализ требований и определение спецификаций программного обеспечения. Изучение литературы и интернет-источников по теме. Выполнение практических работ.</p>
<p>Тема 3. Проектирование программного обеспечения. Изучение литературы и интернет-источников по теме. Выполнение практических работ.</p>
<p>Тема 4. Конструирование программного обеспечения. Изучение литературы и интернет-источников по теме. Выполнение практических работ.</p>
<p>Тема 5. Тестирование, сопровождение и оценка качества программного обеспечения. Изучение литературы и интернет-источников по теме. Выполнение практических работ.</p>
<p>Тема 6. Индивидуальная и коллективная разработка программного обеспечения. Изучение литературы и интернет-источников по теме. Выполнение практических работ.</p>

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену
Приложение 2.

7.3.3. Перечень курсовых работ
Не предусмотрено.

7.4. Электронное портфолио обучающегося
Материалы не размещаются.

7.5. Методические рекомендации по выполнению контрольной работы
Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы
Не предусмотрено.

8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

По заявлению студента

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Сайт библиотеки УрГЭУ

<http://lib.usue.ru/>

Основная литература:

1. Антипов В. А., Бубнов А. А., Пылькин А. Н., Столчнев В. К.. Введение в программную инженерию:учебник для студентов вузов, обучающихся по направлению подготовки 2.09.03.04 "Программная инженерия" (квалификация "бакалавр"). - Москва: КУРС: ИНФРА-М, 2018. - 336 с.

2. Гагарина Л. Г., Кокорева Е. В., Сидорова-Виснадул Б. Д., Гагарина Л. Г.. Технология разработки программного обеспечения:учебное пособие для студентов вузов, обучающихся по направлениям подготовки 09.04.01 и 09.03.03 «Информатика и вычислительная техника». - Москва: ФОРУМ: ИНФРА-М, 2019. - 400 с.

Дополнительная литература:

1. Федорова Г. Н.. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие (09.02.05 Приклад. информатика (по отр.), ПМ "Разработка, адаптация и внедрение програм. обеспечения отр. направленности) для профессиональных образовательных организаций. - Москва: КУРС: ИНФРА-М, 2016. - 336 с.
2. Виноградова Е. Ю.. Паттерны программирования: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 27 с.
3. Плещев В. В., Шишков Е. И.. Основы программирования на языках С++ и С# с примерами и упражнениями: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2018. - 286 с.

**10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ
ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ
СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ
ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ**

Перечень лицензионное программное обеспечение:

Astra Linux Common Edition. Договор № 1 от 13 июня 2018, акт от 17 декабря 2018. Срок действия лицензии - без ограничения срока.

Libre Office. Лицензия GNU LGPL. Срок действия лицензии - без ограничения срока.

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

MySQL Community Server. Стандартная общественная лицензия GNU (GPL). Срок действия лицензии - без ограничения срока.

Microsoft Visio. Акт предоставления прав № Tr020776 от 07.04.2017. Срок действия лицензии - 07.02.2020.

Microsoft Project. Акт предоставления прав № Tr020776 от 07.04.2017. Срок действия лицензии - 07.02.2020.

Система контроля версий Git. Лицензия GNU GPL v2 and GNU LGPL v2.1. Срок действия лицензии - без ограничения срока.

Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:

Формализация моделирования

<https://openedu.ru/course/spbstu/FOMO/>

Технологии программирования

<https://openedu.ru/course/urfu/PRGRMM/>

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду УрГЭУ.

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия обеспечивающие тематические иллюстрации

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

К экзамену

1. Основные понятия и цели управления сложными проектами.
 2. Классы программ. Архитектура программных средств.
 3. Особенности планирования и управления разработкой.
 4. Уровни стандартизации. Основные стандарты.
 5. Каскадная модель, каскадная модель с промежуточным контролем и спиральная модели жизненного цикла программных средств.
 6. Стандарты жизненного цикла программных средств.
 7. Принципы разработки человеко-машинного интерфейса.
 8. Планирование процессов разработки программных средств.
 9. Методы достижения качества.
 10. Сертификация и аттестация программных средств.
 11. Классификация стандартов качества.
 12. Основные показатели качества.
 13. Планирование и управление обеспечением качества.
 14. Конфигурационное управление версиями.
 15. Тестирование программ. Цели, методы и этапы тестирования.
 16. Особенности проектирования информационных систем.
 17. Методология быстрой разработки приложений (RAD).
 18. Сущность структурного подхода к проектированию информационных систем.
 19. Моделирование потоков данных (процессов).
 20. Функциональное моделирование. Состав функциональной модели. Иерархия диаграмм.
 21. Моделирование данных. Основные понятия.
 22. Методология IDEF1.
 23. Функциональная модель IDEF0. Принципы построения модели. Работы. Стрелки. Нумерация работ и диаграмм. Диаграммы дерева узлов и экспозиций (FEO).
 24. Общая характеристика и классификация CASE-средств.
 25. Интеллектуализация вычислительных систем.
 26. Рынок программных продуктов.
 27. Классификация стандартов.
 28. Хранилище данных. Типы и методы их построения.
 29. Методы установки систем защиты программных средств.
 30. Методы защиты программных средств.
 31. Принципы функционирования систем защиты программных средств.
 32. Показатели оценки систем защиты программных средств.
- 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену

Примерные практические задания к экзамену

1. Создание функциональной модели (IDEF0) в BPWin.
2. Оценка модели в BPWin
3. Создание модели DFD в BPWin.
4. Создание модели данных в ERWin
5. Создание отчета в RPTwin
6. Разработка моделей в Rational Rose

Федеральное государственное бюджетное образовательное учреждение высшего образования

УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНЫ

на заседании кафедры информационных
технологий и статистики

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ

ТЕКУЩЕГО КОНТРОЛЯ

по дисциплине

Программная инженерия

Тест

1. Вопрос: Объектно-ориентированные технологии имеют следующие преимущества:

Ответ 1: Уменьшенную связность между модулями

Ответ 2: Повышенное качество кода

Ответ 3: Высокий уровень абстракции

Ответ 4: Все из вышеперечисленного

2. Вопрос: В Объектно-ориентированной Технологии слово "UML" означает

Ответ 1: Unified Module Language

Ответ 2: Unified Modeling Language

Ответ 3: Universal Module Leveling

Ответ 4: Universal Module Language

3. Вопрос: Объект имеет

Ответ 1: Поведение

Ответ 2: Состояние

Ответ 3: Атрибуты

Ответ 4: Все из вышеперечисленного

4. Вопрос: Какое из следующих утверждений верно?

Ответ 1: Объект - это экземпляр класса

Ответ 2: Класс - это абстрактное определение для множества объектов

Ответ 3: Объект может быть более чем в одном классе

Ответ 4: Объект имеет линию жизни

Ответ 5: Все из вышеперечисленного

5. Вопрос: Состояние объекта определяется

Ответ 1: Значением всех его атрибутов

Ответ 2: Его связями с другими объектами

Ответ 3: Его поведением в любой данный момент времени

Ответ 4: Операциями, которые он может выполнять

Ответ 5: Ответы 2 и 4

Ответ 6: Ответы 1 и 2

Ответ 7: Ответы 1, 3 и 4

Ответ 8: Ответы 1, 2 и 3

6. Вопрос: Структура класса

Ответ 1: Представлена в коде

Ответ 2: Представлена атрибутами и связями

Ответ 3: Представлена операциями

Ответ 4: Представлена взаимодействиями объектов

Ответ 5: Ответы 1 и 2

Ответ 6: Ответы 2 и 3

7. Вопрос: Какое из следующих утверждений правильное:

Ответ 1: Классы на диаграммах классов могут быть сгруппированы в пакеты, чтобы показать общую организацию модели (архитектуру системы)

Ответ 2: На диаграмме объектов наименования экземпляров должны быть выделены наклонным шрифтом

Ответ 3: Если пакет В зависит от пакета А, то любые изменения в А вызывают изменения в пакете В.

Ответ 4: Диаграммы объектов и диаграммы классов взаимозаменяемы

8. Вопрос: Какое из следующих утверждений о кратности роли в ассоциации является неверным?

Ответ 1: "1" - именно один и только один

Ответ 2: "0..n" - любое вещественное число, включая ноль

Ответ 3: "0..1" - ноль или один

Ответ 4: "3..7" - 3 или 7

Ответ 5: "3, 7" - 3 или 7

9. Вопрос: На диаграмме требований (Use Case diagram) есть три основных элемента, это:

Ответ 1: Объекты, действия и связи

Ответ 2: Субъекты (акторы), сообщения и связи

Ответ 3: Объекты, требования и связи

Ответ 4: Субъекты (акторы), требования и связи

10. Вопрос: Какое утверждение описывает отношение наследования наилучшим образом?

Ответ 1: Отношение "имеет"

Ответ 2: Отношение "является"

Ответ 3: Отношение "выполняет"

Ответ 4: Отношение "использует"

Ответ 5: Отношение "является элементом"

11. Вопрос: Какое утверждение описывает отношение агрегации наилучшим образом?

Ответ 1: Отношение "имеет"

Ответ 2: Отношение "является"

Ответ 3: Отношение "выполняет"

Ответ 4: Отношение "использует"

Ответ 5: Отношение "является частью"

12. Вопрос: Операция в классе «нечто» вызывает операцию в классе «что-то». Других отношений между двумя классами нет. Каким типом отношений является данный?

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

13. Вопрос: Машина имеет 4 колеса. Какой тип отношений имеют класс "Машина" и класс "Колесо"?

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

14. Вопрос: Студент посещает несколько аудиторий. Аудитория может вмещать несколько студентов. Какой тип отношений существует между студентом и аудиторией?

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

15. Вопрос: Какое из следующих утверждений правильное:

Ответ 1: Классы на диаграммах классов могут быть сгруппированы в пакеты, чтобы показать общую организацию модели (архитектуру системы)

Ответ 2: На диаграмме объектов наименования экземпляров должны быть выделены наклонным шрифтом

Ответ 3: Если пакет В зависит от пакета А, то любые изменения в А вызывают изменения в пакете В.

Ответ 4: Диаграммы объектов и диаграммы классов взаимозаменяемы

17. Вопрос: Выберите отношение, которое может иметь кратность, отличную от “1”.

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

Ответ 6: Все из вышеперечисленного

Ответ 7: Ответы 1, 3 и 4

Ответ 8: Ответы 1 и 2

Ответ 9: Ответы 3, 4 и 5

18. Вопрос: Какое из следующих утверждений является неверным?

Ответ 1: Класс может иметь отношение к самому себе.

Ответ 2: Объект может иметь отношение к другим объектам того же класса.

Ответ 3: Класс может иметь только одно отношение к другому классу.

Ответ 4: Отношение может существовать с мощностью “0”.

Ответ 5: Класс может существовать без каких-либо отношений к другим объектам.

19. Вопрос: Какой тип отношений к самому себе может иметь класс?

Ответ 1: Ассоциация

Ответ 2: Агрегация

Ответ 3: Наследование

Ответ 4: Реализация

Ответ 5: Зависимость

Ответ 6: Ответы 1 и 2

Ответ 7: Ответы 3 и 5

Ответ 8: Ответы 4 и 5

Ответ 9: Ничего из вышеперечисленного

20. Вопрос: Что такое “актор” в анализе варианта использования?

Ответ 1: Любая сущность, которая взаимодействует с системой.

Ответ 2: Термин для бизнес-объектов, которые присутствуют в системе.

Ответ 3: То, что реагирует на внешние воздействия системы.

Ответ 4: В анализе варианта использования этого термина нет.

21. Вопрос: Диаграмма Последовательности содержит

Ответ 1: Объекты

Ответ 2: Сообщения

Ответ 3: Видимость объекта

Ответ 4: Временные ограничения

Ответ 6: Все из вышеперечисленного

Ответ 7: Все из вышеперечисленного за исключением 3

Ответ 9: Все из вышеперечисленного за исключением 1

22. Вопрос: Диаграмма Состояний содержит

- Ответ 1: Состояния класса
- Ответ 2: Переходы между состояниями
- Ответ 3: Действия, выполняемые классом
- Ответ 4: События, которые являются действиями в классе
- Ответ 5: Все из вышеперечисленного

23. Вопрос: Диаграмма Состояний может иметь только одно (один)

- Ответ 1: Конечное состояние
- Ответ 2: Начальное состояние
- Ответ 3: Действие в состоянии
- Ответ 4: Переход из состояния
- Ответ 5: Все из вышеперечисленного

24. Вопрос: Какое из следующих утверждений является неверным?

- Ответ1: В какой-либо момент времени объект может быть более чем в одном состоянии
- Ответ2: Автомат может иметь множество конечных состояний
- Ответ3: Автомат может запоминать, какое подсостояние было последним состоянием
- Ответ4: Переход может срабатывать условно

25. Вопрос: Какое из следующих утверждений о диаграмме состояний является правильным:

- Ответ 1: Все действия на диаграмме состояний связаны с переходами
- Ответ 2: Событие может явиться причиной того, что объект останется в том же состоянии, предшествующем событию
- Ответ 3: Если объект выходит из состояния, он не может вернуться в это состояние.
- Ответ 4: Два различных перехода из одного состояния могут перекрывать друг друга (они могут быть вызваны одним и тем же событием)

26. Что означает аббревиатура RUP?

- Ответ 1: Rational Unified Process
- Ответ 2: Rational Universal Process
- Ответ 3: Rational Unified Program

28. Для чего нужен словарь модели

- Ответ1: Чтобы ограничивать количество используемых терминов
- Ответ2: Чтобы формировать терминологию предметной области
- Ответ3: Чтобы добавлять новые термины

29. Названия прецедентов должны начинаться:

- Ответ1: с глагола или существительного
- Ответ2: с глагола
- Ответ3: с прилагательного
- Ответ4: с глагола или прилагательного

Практическая работа №1

Вариант 1

1. Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, вычисления площади, периметра и точки пересечения медиан.

Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (по автору, по году издания или категории), добавления книг в библиотеку, удаления книг из нее, доступа к книге по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «поезд», содержащий следующие закрытые поля:

- название пункта назначения;
- номер поезда (может содержать буквы и цифры);
- время отправления.

Предусмотреть свойства для получения состояния объекта.

Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности:

- вывод информации о поезде по номеру с помощью индекса;
- вывод информации о поездах, отправляющихся после введенного с клавиатуры времени;
- перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов;
- вывод информации о поездах, отправляющихся в заданный пункт назначения.

Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

Вариант 2

1. Описать класс, реализующий шестнадцатеричный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, — и

свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс, представляющий круг. Предусмотреть методы для создания объектов, вычисления площади круга, длины окружности и проверки попадания заданной точки внутрь круга. Описать свойства для получения состояния объекта.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по фамилии и доступа к записи по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «поезд», содержащий следующие закрытые поля:

- название пункта назначения;
- номер поезда (может содержать буквы и цифры);
- время отправления.

Предусмотреть свойства для получения состояния объекта.

Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности:

- вывод информации о поезде по номеру с помощью индекса;
- вывод информации о поездах, отправляющихся после введенного с клавиатуры времени;
- перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов;
- вывод информации о поездах, отправляющихся в заданный пункт назначения.

Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

Вариант 3

1. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение.

Написать программу, демонстрирующую все разработанные элементы класса

2. Описать класс для работы со строкой, позволяющей хранить только двоичное число и выполнять с ним арифметические операции. Предусмотреть инициализацию с проверкой допустимости значений. В случае недопустимых значений выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и удаления записей, сортировки по разным полям, доступа к записи по номеру.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «товар», содержащий следующие закрытые поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

Предусмотреть свойства для получения состояния объекта.

Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности:

- вывод информации о товаре по номеру с помощью индекса;
- вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение;
- сортировку товаров по названию магазина, по наименованию и по цене;
- перегруженную операцию сложения товаров, выполняющую сложение их цен.

Написать программу, демонстрирующую все разработанные элементы классов.

Вариант 4

1. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса и проверки допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс дробей — рациональных чисел, являющихся отношением двух целых чисел.

Предусмотреть методы сложения, вычитания, умножения и деления дробей.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «автостоянка» для хранения сведений об автомобилях. Для каждого автомобиля записываются госномер, цвет, фамилия владельца и признак присутствия на стоянке. Обеспечить возможность поиска автомобиля по разным критериям, вывода списка присутствующих и отсутствующих на стоянке автомобилей, доступа к имеющимся сведениям по номеру места.

Написать программу, демонстрирующую все разработанные элементы класса.

4. Описать класс «товар», содержащий следующие закрытые поля:

- название товара;
- название магазина, в котором продается товар;
- стоимость товара в рублях.

Предусмотреть свойства для получения состояния объекта.

Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности:

- вывод информации о товаре по номеру с помощью индекса;

- вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение;
- сортировку товаров по названию магазина, по наименованию и по цене;
- перегруженную операцию сложения товаров, выполняющую сложение их цен.

Написать программу, демонстрирующую все разработанные элементы классов.

Вариант 5

1. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

Написать программу, демонстрирующую все разработанные элементы класса.

2. Описать класс «файл», содержащий сведения об имени, дате создания и длине файла. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать метод добавления информации в конец файла и свойства для получения состояния файла.

Написать программу, демонстрирующую все разработанные элементы класса.

3. Описать класс «колода карт», включающий закрытый массив элементов класса «карта». В карте хранятся масть и номер. Обеспечить возможность вывода карты по номеру, вывода всех карт, перемешивания колоды и выдачи всех карт из колоды поодиночке и по 6 штук в случайном порядке.

Написать программу, демонстрирующую все разработанные элементы классов.

4. Описать класс «самолет», содержащий следующие закрытые поля:

- название пункта назначения;
- шестизначный номер рейса;
- время отправления.

Предусмотреть свойства для получения состояния объекта.

Описать класс «аэропорт», содержащий закрытый массив самолетов. Обеспечить

следующие возможности:

- вывод информации о самолете по номеру рейса с помощью индекса;
- вывод информации о самолетах, отправляющихся в течение часа после введенного с клавиатуры времени;
- вывод информации о самолетах, отправляющихся в заданный пункт назначения;
- перегруженную операцию сравнения, выполняющую сравнение времени отправления двух самолетов.

Информация должна быть отсортирована по времени отправления.

Написать программу, демонстрирующую все разработанные элементы классов.

Практическая работа №2

Вариант 1

Написать Windows-приложение, которое выполняет анимацию изображения.

Создать меню с командами **Show picture**, **Choose**, **Animate**, **Stop**, **Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **Show picture** в центре экрана рисуется объект, состоящий из нескольких графических примитивов.

При выборе команды **Choose** открывается диалоговое окно, содержащее:

- поле типа **TextBox** с меткой **Speed** для ввода скорости движения объекта;
- группу **Direction** из двух переключателей (**Up-Down**, **Left-Right**) типа **RadioButton** для выбора направления движения;
- кнопку типа **Button**.

По команде **Animate** объект начинает перемещаться в выбранном направлении до края окна и обратно с заданной скоростью, по команде **Stop** — прекращает движение.

Вариант 2

Написать Windows-приложение, которое по заданным в файле исходным данным строит график или столбиковую диаграмму.

Создать меню с командами **Input data**, **Choose**, **Line**, **Bar**, **Quit**.

Команды **Line** и **Bar** недоступны. Команда **Quit** завершает работу приложения.

При выборе команды **Input data** из файла читаются исходные данные (файл сформировать самостоятельно).

По команде **Choose** открывается диалоговое окно, содержащее:

- список для выбора цвета графика типа **ListBox**;
- группу из двух переключателей (**Line**, **Bar**) типа **RadioButton**;
- кнопку типа **Button**.

Обеспечить возможность ввода цвета и выбора режима: построение графика (**Line**) или столбиковой диаграммы (**Bar**). После указания параметров становится доступной соответствующая команда меню.

По команде **Line** или **Bar** в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Вариант 3

Написать Windows-приложение, которое строит графики четырех заданных функций.

Создать меню с командами **Chart**, **Build**, **Clear**, **About**, **Quit**.

Команда **Quit** завершает работу приложения. При выборе команды **About** открывается окно с информацией о разработчике.

Команда **Chart** открывает диалоговое окно, содержащее:

- список для выбора цвета графика типа **ListBox**;
- список для выбора типа графика типа **ListBox**, содержащий четыре пункта: **sin(x)**, **sin(x+π/4)**, **cos(x)**, **cos(x-π/4)**;
- кнопку типа **Button**.

Обеспечить возможность выбора цвета и вида графика. После щелчка на кнопке **OK** в главном окне приложения строится график выбранной функции на интервале от $-\pi/2$ до $+\pi/2$. Окно должно содержать заголовок графика, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Команда **Clear** очищает окно.

Вариант 4

Написать Windows-приложение — графическую иллюстрацию сортировки методом выбора.

Создать меню с командами **File**, **Animate**, **About**, **Exit**.

Команда **Animate** недоступна. Команда **Exit** завершает работу приложения. Команда **About** открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда **File**) использовать объект класса `OpenFileDialog`.

Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).

После чтения данных становится доступной команда **Animate**.

При выборе команды **Animate** в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.

Вариант 5

Написать Windows-приложение — графическую иллюстрацию сортировки методом пузырька.

Создать меню с командами **File**, **Animate**, **About**, **Exit**.

Команда **Animate** недоступна. Команда **Exit** завершает работу приложения. Команда **About** открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда **File**) использовать объект класса `OpenFileDialog`.

Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).

После чтения данных становится доступной команда **Animate**.

При выборе команды **Animate** в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.

Федеральное государственное бюджетное образовательное учреждение высшего образования

УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДЕНЫ

на заседании кафедры информационных
технологий и статистики

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ

ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

по дисциплине

Программная инженерия

Экзаменационный билет №1

1. Основные принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, наследование.
2. Интерфейсы.
3. Разработать Windows-приложение «Конвертер», которое пересчитывает цену из долларов в рубли. Обеспечить безопасный ввод: пользователь может ввести в поля редактирования только числа и одну запятую.

Экзаменационный билет №2

1. Среда .NET Framework и ее две составляющие.
2. Структуры и перечисления.
3. Разработать Windows-приложение «Автозаправка», которое вычисляет количество литров бензина, которое можно купить на заданную сумму. Выбор бензина реализовать с помощью выплывающего списка. В поле «Сумма» можно ввести только числовые данные.

Экзаменационный билет №3

1. Обычные типы данных в языке C#. Способы форматирования вывода.
2. Обработка исключений. Блоки try – catch – finally.
3. Разработать Windows-приложение «Кафе», которое рассчитывает сумму заказа, исходя из выбранных пунктов меню. Выбор блюд реализовать с помощью элементов CheckBox, а выбор альтернативных продуктов – с помощью элементов RadioButton.

Экзаменационный билет №4

1. Литералы. Символьные escape-последовательности.
2. Генерирование исключений. Оператор throw.
3. Разработать Windows-приложение «Инженерный калькулятор». Ввод данных обеспечить как по нажатию на кнопки формы, так и по нажатию на числа на клавиатуре. Помимо стандартных операций, реализовать операции возведения в степень и вычисления квадратного корня.

Экзаменационный билет №5

1. Понятие «переменная». Области видимости и время жизни переменных.
2. Ключевые слова checked и unchecked.
3. Разработать Windows-приложение «Таймер», используя компоненты Timer и NumericUpDown. Программа должна позволять задать время (минуты и секунды) и по нажатию кнопки Пуск начинать отсчет заданного времени.

Экзаменационный билет №6

1. Арифметические, логические операторы и операторы сравнения в C#.
2. Потоки ввода/вывода в C#.
3. Разработать Windows-приложение, строящее диаграмму или график по заданным в файле данным. Создать меню с командами Input Data, Choose, Line, Bar, Quit. Команды Line и Bar недоступны. При выборе команды Input Data из файла читаются исходные данные. По команде Choose открывается диалоговое окно, содержащее: список для выбора цвета графика типа TListBox; группу из двух переключателей (Line, Bar) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода цвета и выбора режима: график или диаграмма. После указания параметров становится доступной соответствующая команда меню. По команде Line или Bar в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Экзаменационный билет №7

1. Операторы присваивания. Понятие «выражение». Правила автоматического преобразования типов в выражениях.
2. Работа с файлами в C#.
3. Описать класс для работы с одномерным массивом целых чисел. Обеспечить следующие возможности:
 1. задание произвольных целых границ;
 2. Обращение к отдельному элементу массива с контролем выхода за пределы массива;
 3. Выполнение операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов;
 4. Выполнение операций умножения и деления всех элементов массива на скаляр;
 5. Вывод на экран элемента массива по заданному индексу и всего массива.Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №8

1. Условный оператор и его модификации. Оператор множественного условия.
2. Делегаты.
3. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №9

1. Циклы в C#. Циклы с предусловием, постусловием и параметром.
2. События.

3. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №10

1. Операторы break, continue, goto.

2. Пространства имен.

3. Составить описание класса для вектора, заданного координатами его концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №11

1. Понятия «класс» и «объект». Общий синтаксис и определения класса. Переменные ссылочного типа.

2. Препроцессор. Директивы препроцессора.

3. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №12

1. Методы.

2. Атрибуты.

3. Составить описание класса для представления времени. Предусмотреть возможности установки времени и изменения его отдельных полей (час, минута, секунда) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Создать методы изменения времени на заданное количество часов, минут и секунд. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №13

1. Конструкторы. Оператор new.

2. Работа с формами в C#. Свойства окна. Toolbox.

3. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, вычисления площади, периметра и точки пересечения медиан. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №14

1. Деструкторы. Ключевое слово `this`.
2. Многопоточное программирование в C#.
3. Описать класс, представляющий круг. Предусмотреть методы для создания объектов, вычисления площади круга, длины окружности и проверки попадания заданной точки внутрь круга. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №15

1. Одномерные и многомерные массивы в C#.
2. Язык разметки XML.
3. Описать класс дробей — рациональных чисел, являющихся отношением двух целых чисел. Предусмотреть методы сложения, вычитания, умножения и деления дробей. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №16

1. Цикл `foreach`.
2. Основные элементы управления в Visual Studio.
3. Описать класс «комната», содержащий сведения о метраже, высоте потолков количестве окон. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать методы вычисления площади и объема комнаты и свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №17

1. Операции со строками в C#.
2. Работа с базами данных в Visual Studio.
3. Описать класс, представляющий квадратное уравнение вида $ax^2 + vx + c = 0$. Описать метод, вычисляющий решение этого уравнения и выбрасывающий исключение в случае отсутствия корней. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №18

1. Побитовые операторы в C#.
2. Технология ADO.NET.

3. Описать класс «процессор», содержащий сведения о марке, тактовой частоте, объеме кэша и стоимости. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать свойства для получения состояния объекта. Описать класс «материнская плата», включающий класс «процессор» и объем установленной оперативной памяти. Предусмотреть инициализацию с проверкой допустимости значений поля объема памяти. В случае недопустимых значений поля выбрасывается исключение. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы классов.

Экзаменационный билет №19

1. Модификаторы доступа в C#.
2. Элементы управления Visual Studio для работы с печатью и изображениями
3. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по фамилии и доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №20

1. Передача объектов методу. Модификаторы ref и out.
2. Работа с графическими объектами в C#.
3. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и удаления записей, сортировки по разным полям, доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.

Экзаменационный билет №21

1. Использование переменного количества аргументов в методах. Возвращение объектов.
2. Основные принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, наследование.
3. Описать класс «колода карт», включающий закрытый массив элементов класса «карта». В карте хранятся масть и номер. Обеспечить возможность вывода карты по номеру, вывода всех карт, перемешивания колоды и выдачи всех карт из колоды поодиночке и по 6 штук в случайном порядке. Написать программу, демонстрирующую все разработанные элементы классов.

Экзаменационный билет №22

1. Перегрузка методов и конструкторов.
2. Среда .NET Framework и ее две составляющие.
3. Описать класс «поезд», содержащий следующие закрытые поля: название пункта назначения; номер поезда (может содержать буквы и цифры); время отправления. Предусмотреть свойства для получения

состояния объекта. Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности: вывод информации о поезде по номеру с помощью индекса; вывод информации о поездах, отправляющихся после введенного с клавиатуры времени; перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов; вывод информации о поездах, отправляющихся в заданный пункт назначения. Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

Экзаменационный билет №23

1. Метод Main().
2. Обычные типы данных в языке C#. Способы форматирования вывода.
3. Описать класс «товар», содержащий следующие закрытые поля: название товара; название магазина, в котором продается товар; стоимость товара в рублях. Предусмотреть свойства для получения состояния объекта. Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности: вывод информации о товаре по номеру с помощью индекса; вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение; сортировку товаров по названию магазина, по наименованию и по цене; перегруженную операцию сложения товаров, выполняющую сложение их цен. Написать программу, демонстрирующую все разработанные элементы классов.

Экзаменационный билет №24

1. Рекурсия. Ключевое слово static.
2. Литералы. Символьные escape-последовательности.
3. Описать класс «самолет», содержащий следующие закрытые поля: название пункта назначения; шестизначный номер рейса; время отправления. Предусмотреть свойства для получения состояния объекта. Описать класс «аэропорт», содержащий закрытый массив самолетов. Обеспечить следующие возможности: вывод информации о самолете по номеру рейса с помощью индекса; вывод информации о самолетах, отправляющихся в течение часа после введенного с клавиатуры времени; вывод информации о самолетах, отправляющихся в заданный пункт назначения; перегруженную операцию сравнения, выполняющую сравнение времени отправления двух самолетов. Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

Экзаменационный билет №25

1. Перегрузка операторов. Основные положения и ограничения при перегрузке операторов.
2. Понятие «переменная». Области видимости и время жизни переменных.
3. Описать класс «запись», содержащий следующие закрытые поля: фамилия, имя; номер телефона; дата рождения (массив из трех чисел). Предусмотреть свойства для получения состояния объекта. Описать класс «записная книжка», содержащий закрытый массив записей. Обеспечить следующие возможности: вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение; поиск людей, день рождения которых сегодня или в заданный день; поиск людей, день рождения которых будет на следующей неделе; поиск людей, номер телефона

которых начинается на три заданных цифры. Написать программу, демонстрирующую все разработанные элементы классов.

Экзаменационный билет №26

1. Индексаторы и свойства. Ограничения в использовании свойств.
2. Арифметические, логические операторы и операторы сравнения в C#.
3. Описать базовый класс «Строка». Обязательные поля класса: поле для хранения символов строки; значение типа word для хранения длины строки в байтах. Реализовать обязательные методы следующего назначения: конструктор без параметров; конструктор, принимающий в качестве параметра строковый литерал; конструктор, принимающий в качестве параметра символ; метод получения длины строки; метод очистки строки. Описать производный от «Строка» класс «Комплексное число». Строки данного класса состоят из двух полей, разделенных символом *i*. Первое поле задает значение действительной части числа, второе – значение мнимой. Каждое из полей может содержать только символы десятичных цифр и символы + и -, задающие знак числа. Символы + и – могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Для класса «Комплексное_число» определить следующие методы: проверка на равенство; сложение чисел; умножение чисел.

Экзаменационный билет №27

1. Наследование. Модификатор protected. Ключевое слово super.
2. Операторы присваивания. Понятие «выражение». Правила автоматического преобразования типов в выражениях.
3. Создать меню с командами Begin, Work, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее: два поля ввода; группу из двух переключателей (First letter, All letter) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода предложения и выбора режима его преобразования: либо начинать с прописной буквы каждое слово (First letter), либо перевести все буквы в верхний регистр (All letter). При выборе команды Work открывается диалоговое окно с результатами преобразования.

Экзаменационный билет №28

1. Виртуальные методы и переопределение.
2. Условный оператор и его модификации. Оператор множественного условия.
3. Написать Windows-приложение – графическую иллюстрацию сортировки методом выбора. Создать меню с командами File, Animate, About, Exit. Команда Animate недоступна. Команда Exit завершает работу приложения. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда File) использовать объект класса OpenFileDialog. Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа). После чтения данных становится доступной команда Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.

Экзаменационный билет №29

1. Абстрактные классы.
2. Циклы в C#. Циклы с предусловием, постусловием и параметром.
3. Создать меню с командами Begin, Work, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее: два поля ввода; группу из двух переключателей (First letter, All letter) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода предложения и выбора режима его преобразования: либо начинать с прописной буквы каждое слово (First letter), либо перевести все буквы в верхний регистр (All letter). При выборе команды Work открывается диалоговое окно с результатами преобразования.

Экзаменационный билет №30

1. Класс Object. Упаковка и распаковка.
2. Операторы break, continue, goto.
3. Написать Windows-приложение – графическую иллюстрацию сортировки методом выбора. Создать меню с командами File, Animate, About, Exit. Команда Animate недоступна. Команда Exit завершает работу приложения. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда File) использовать объект класса OpenFileDialog. Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа). После чтения данных становится доступной команда Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.