

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	Программирование
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	5
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	9
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	10
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	10
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	11

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 23.08.2017 г. № 809)
ПС	

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Формирование первоначальных знаний, умений и навыков разработки программных средств на языках высокого уровня.

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	Лабораторные		
Семестр 1					
Экзамен	144	56	56	52	4
Семестр 2					
Экзамен	180	54	54	90	5
	324	110	110	142	9

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-6 Способен использовать педагогической деятельности основы знаний информационно-коммуникационных технологий	ИД-1.ОПК-6 Знать: изучаемые языки программирования, сетевые технологии, применение веб-технологий. Уметь: вести устную и письменную коммуникацию на изучаемом языке. Иметь навыки: практического опыта использования методики педагогической деятельности.

ОПК-2 Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности	ИД-1.ОПК-2 Знать: математические основы программирования и языков программирования, организации баз данных и компьютерного моделирования; математические методы оценки качества, надежности и эффективности программных продуктов; математические методы организации информационной безопасности при разработке и эксплуатации программных продуктов и программных. Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности. Иметь навыки: применения современных информационных технологий и программных средств, в том числе отечественного
---	---

Профессиональные компетенции (ПК)

Шифр и наименование компетенции	Индикаторы достижения компетенций
научно-исследовательский	
ПК-8 Способен проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности	ИД-1.ПК-8 Знать: основы научной работы, современные методы сбора и анализа полученного материала, способы аргументации; основные принципы защиты информации БД. Уметь: решать научные задачи в связи с поставленной целью и в соответствии с выбранной методикой. Иметь навыки: проведения научных исследований с использованием методов математического моделирования, а также решать задачи, связанные с выбором способов защиты информации БД.

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Наименование темы	Всего часов	Контактная работа (по уч.зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
			Часов				
Семестр 1		108					
Тема 1.	Введение в язык программирования высокого уровня	10		4		6	
Тема 2.	Компиляция и выполнение программы	10		4		6	
Тема 3.	Управляющие операторы	14		8		6	
Тема 4.	Массивы	14		8		6	
Тема 5.	Списки, строки и файлы	15		8		7	
Тема 6.	Основы тестирования	15		8		7	
Тема 7.	Рекурсия	15		8		7	
Тема 8.	Поиск и сортировка	15		8		7	
Семестр 2		144					
Тема 9.	Классы, структуры и объекты	19		8		11	
Тема 10.	Целостность данных	15		4		11	
Тема 11.	Наследование и интерфейсы	19		8		11	
Тема 12.	Обработка исключительных ситуаций	15		4		11	

Тема 13.	Коллекции и обобщения	19		8		11	
Тема 14.	Элементы функционального программирования	19		8		11	
Тема 15.	Оконные приложения	20		8		12	
Тема 16.	Многопоточное программирование	18		6		12	

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерии оценивания
Текущий контроль (Приложение 4)			
Темы 1-3	Тест (приложение 4)	Тест состоит из 10 вопросов	10 баллов
Темы 4-6	Контрольная работа (приложение 4)	Задания на программирование	10 баллов
Темы 7-8	Контрольная работа (приложение 4)	Задания на программирование	10 баллов
Промежуточный контроль (Приложение 5)			
2 семестр (Эк)	Экзаменационный билет (приложение 5)	2 теоретических вопроса, 1 практическое задание	100 баллов
1 семестр (Эк)	Экзаменационный билет (приложение 5)	2 теоретический вопрос, 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

7.2 Содержание практических занятий и лабораторных работ

<p>Тема 1. Введение в язык программирования высокого уровня</p> <p>Программа "Hello, world!". Числовые типы данных. Строки. Арифметические операции и var. <u>Методы переменные.</u></p>
<p>Тема 2. Компиляция и выполнение программы</p> <p>Ошибки на этапе компиляции. Ошибки на этапе выполнения. Отладка. Стилистические ошибки. <u>Рефакторинг кода. Константы и enum'ы. Выделение методов</u></p>
<p>Тема 3. Управляющие операторы</p> <p>Операторы сравнения и логический тип данных. Сравнение чисел с плавающей точкой. Полные и сокращенные операции. Операторы If и Else. Оператор Switch - Case. <u>Цикл While. Цикл For. Сравнение циклов.</u></p>
<p>Тема 4. Массивы</p> <p>Массивы и цикл foreach. Короткая форма записи. Типы ссылки и типы значения. Передача массива <u>в метод. Многомерные массивы. Массивы массивов.</u></p>
<p>Тема 5. Списки, строки и файлы</p> <p>Списки. Словари. Сравнение строк и массивов. Класс StringBuilder. Специальные символы и <u>форматированный вывод. Файлы и каталоги. Кодировка и работа с файлами.</u></p>
<p>Тема 6. Основы тестирования</p> <p>Введение в тестирование. Модульное тестирование. Покрытие тестами. <u>Функциональное тестирование.</u></p>
<p>Тема 7. Рекурсия</p> <p>Сложность алгоритмов. Масштаб роста функций. O-символика. Оценка сложности. Сложность алгоритмов с числами. Классы сложности. Рекурсия. Дерево рекурсии. Принцип "разделяй и властвуй". Перебор подмножеств. Перестановки. <u>Размещения.</u></p>
<p>Тема 8. Поиск и сортировка</p> <p>Линейный и бинарный поиск. Анализ линейного и бинарного поиска. Сравнение производительности. Создание графиков. Сортировка пузырьком. Сортировка слиянием. Быстрая сортировка. Сравнение алгоритмов <u>сортировки.</u></p>
<p>Тема 9. Классы, структуры и объекты</p> <p>Классы. Статическое и динамическое. Методы. Методы расширения. Статические классы. <u>Структуры. Инициализация полей структуры.</u></p>
<p>Тема 10. Целостность данных</p> <p>Целостность данных. Ключевое слово private. Свойства. Конструкторы. Поля readonly. <u>Статические конструкторы.</u></p>
<p>Тема 11. Наследование и интерфейсы</p> <p>Наследование. Иерархия наследования. Класс Array. Интерфейсы. Реализация IComparable и <u>IComparer. Полиморфизм. Виртуальные методы и переопределение методов.</u></p>
<p>Тема 12. Обработка исключительных ситуаций</p> <p>Использование блоков try и catch. Перехват исключений. Использование нескольких операторов catch. Вложенные блоки try. Генерирование исключений. Объект Exception. Ключевое слово finally. <u>Создание собственных классов исключений. Ключевые слова checked и unchecked.</u></p>

<p>Тема 13. Коллекции и обобщения</p> <p>Структуры данных "Стек", "Очередь" и "Дек". Универсальная очередь. Даункасты. Дженерик-классы. Стеки для анализа скобочных выражений. Стеки для вычислений. Очередь для скользящего среднего. Дженерики и сортировка массивов. Возврат из метода значения и ошибки. Ключевые слова ref и out. Дженерик-классы Tuple и Nullable.</p> <p>Интерфейсы IEnumerable и IEnumerator. Использование yield return.</p> <p>Индексация списков. Методы Contains и Equals. Перегрузка операторов. Хеширующие функции. Класс Dictionary. Метод GetHashCode</p>
<p>Тема 14. Элементы функционального программирования</p> <p>Делегаты. Дженерик-делегаты. Func и Action. Анонимные делегаты. Лямбда-выражения. Замыкание.</p> <p>Делегаты для диагностики кода, в разборе арифметических выражений. Лямбда-выражения в тестах. Язык LINO. Методы Where, Select, ToList. Другие конструкции LINO. События.</p>
<p>Тема 15. Оконные приложения</p> <p>Программирование GUI. Событийная модель. Использование делегатов и мультикаст-делегатов. Целостность событийной модели. События. Сокращенный синтаксис событий. Назначение protected virtual OnTick, sender, EventArgs.</p> <p>WindowsForms и WPF. Расположение элементов управления на форме. Дизайнер Windows Forms. Резиновый дизайн. TableLayoutPanel. Рисование. Повороты и переносы рисунка. Таймеры и анимация. Паттерн MVC.</p>
<p>Тема 16. Многопоточное программирование</p> <p>Потоки, домены и процессы. Класс Thread. BeginInvoke и EndInvoke. Parallel. Общие ресурсы и блокировки. Блокирование потока GUI. Асинхронные операции в GUI.</p>

7.3. Содержание самостоятельной работы

<p>Тема 1. Введение в язык программирования высокого уровня</p> <p>Повторение изученного материала. Изучение основной и дополнительной литературы по теме.</p>
<p>Тема 2. Компиляция и выполнение программы</p> <p>Решение задач "Angry Birds", "Бильярд", "Проценты".</p>
<p>Тема 3. Управляющие операторы</p> <p>Решение задач "Рубли", "Два прямоугольника", "Расстояние до отрезка".</p> <p>Работа над проектом по построению путей выхода из лабиринта.</p>
<p>Тема 4. Массивы</p> <p>Работа над проектом по построению гистограмм и тепловой карты.</p>
<p>Тема 5. Списки, строки и файлы</p> <p>Работа над проектом по анализу текстов.</p>
<p>Тема 6. Основы тестирования</p> <p>Работа над проектом по созданию парсера полей. Решение задач "Оттенки серого", "Медианный фильтр", "Пороговый фильтр", "Фильтр Собеля".</p>
<p>Тема 7. Рекурсия</p> <p>Решение задач "Перебор паролей 2", "Хождение по чекпоинтам".</p>
<p>Тема 8. Поиск и сортировка</p> <p>Решение задач "Левая граница", "Правая граница", "Автодополнение".</p> <p>Решение задач "Манипулятор", "Визуализация", "Поиск угла", "Решение манипулятора".</p>
<p>Тема 9. Классы, структуры и объекты</p> <p>Решение задач "Вектор", "Отрезок", "Нестатические методы", "256 оттенков серого".</p>
<p>Тема 10. Целостность данных</p> <p>Решение задач "ReadOnly Vector", "Счет из отеля", "Карманный гугл".</p>
<p>Тема 11. Наследование и интерфейсы</p> <p>Решение задач "Земля и Диггер", "Мешки и золото", "Монстры".</p>

Тема 12. Обработка исключительных ситуаций Решение задач на обработку исключительных ситуаций.
Тема 13. Коллекции и обобщения Решение задач "Limited Size Stack", "Отмена", "CVS". Решение задач "Экспоненциальное сглаживание", "Скользящее среднее", "Скользящий максимум". Решение задач "Readonly bytes" и "Ghosts".
Тема 14. Элементы функционального программирования Работа над проектом по созданию языка Brainfuck. Работа над проектом игры Rocket. Решение задач с использованием LINQ.
Тема 15. Оконные приложения Разработка графических приложений. Выполнение индивидуальной работы по созданию приложений с GUI. Разработка простой компьютерной игры.
Тема 16. Многопоточное программирование Работа над проектом по распараллеливанию алгоритма искусственного интеллекта.

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 предусмотрены.

7.4. Электронное портфолио обучающегося
Материалы не размещаются.

7.5. Методические рекомендации по выполнению контрольной работы
Не предусмотрены.

7.6 Методические рекомендации по выполнению курсовой работы
Не предусмотрены.

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. Немцова Т. И., Голова С. Ю., Терентьев А. И., Гагарина Л. Г.. Программирование на языке высокого уровня. Программирование на языке С++ [Электронный ресурс]: учебное пособие для студентов, обучающихся по направлениям подготовки 09.03.01 «Информатика и вычислительная техника», 09.03.03 «Прикладная информатика», 09.03.04 «Программная инженерия». - Москва: ФОРУМ: ИНФРА-М, 2019. - 512 с. – Режим доступа: <http://znanium.com/go.php?id=1000008>

2. Гуриков С. Р.. Введение в программирование на языке Visual С# [Электронный ресурс]: учебное пособие для студентов образовательных учреждений высшего образования, обучающихся по направлению подготовки 11.03.02 «Инфокоммуникационные технологии и системы связи» (квалификация (степень) «бакалавр»). - Москва: ФОРУМ: ИНФРА-М, 2019. - 447 с. – Режим доступа: <http://znanium.com/go.php?id=1017998znanium.com>

3. Плещев В. В., Шишков Е. И.. Основы программирования на языках С++ и С# с примерами и упражнениями [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2018. - 286 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf>

Дополнительная литература:

1. Корнеев В. И., Гагарина Л. Г., Корнеева М. В.. Программирование графики на С++. Теория и примеры [Электронный ресурс]: учебное пособие для студентов, обучающихся по направлению подготовки 09.04.04 "Программная инженерия" и группам направлений 11.03.04 "Электроника и наноэлектроника" и 11.03.02 "Инфокоммуникационные технологии и системы связи". - Москва: Форум: ИНФРА-М, 2018. - 517 с. – Режим доступа:

2. Кислицын Е. В., Шишков Е. И.. Разработка приложений на языке Java [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 86 с. – Режим доступа: <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf>

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.

Язык программирования Python. Python Software Foundation License (PSFL). Срок действия лицензии - без ограничения срока.

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

Программирование и разработка веб-приложений

<https://openedu.ru/course/ITMOUniversity/PWADEV/>

Программирование на С#

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

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 (первый семестр)

Раздел 1

1. Числовые типы данных.
2. Строковый тип данных. Основные методы работы со строками. Перевод числа в строку.
3. Арифметические операции в C#. Класс Math.
4. Методы: объявление метода, аргументы метода, возвращаемые значения. Перегрузка методов.
5. Глобальные и локальные переменные. Область видимости переменных.
6. Ошибки на этапе компиляции и выполнения. `NullReferenceException`. `IndexOutOfRangeException`. Блок `try-catch`.
7. Логический тип данных. Операторы сравнения.
8. Сравнение чисел с плавающей точкой.
9. Оператор `If – Else`.
10. Оператор `Switch – Case`.
11. Цикл `while`. Операторы `break` и `continue`.
12. Цикл `for`.
13. Массивы. Индексация массива. Цикл `foreach`.
14. Типы ссылки и типы значения. Стек вызовов. Куча.
15. Многомерные массивы. Массивы массивов.
16. Списки. Методы списка. Измерение времени работы программы.
17. Словари.
18. Сравнительный анализ строк и массивов. Методы `substring`, `ToUpper` и др. Неизменяемость строк. Паттерн `Immutable`.
19. Класс `StringBuilder`.
20. Специальные символы при работе со строками.
21. Форматированный вывод строк.
22. Файлы и каталоги.
23. Тестирование. Методы избавления от багов.
24. Библиотеки тестирования.
25. Модульные тесты.
26. Функциональное тестирование.

Раздел 2

1. Понятие алгоритма. Свойства алгоритма.
2. Временная и емкостная сложность алгоритма. Расчет сложности.
3. O-символика.
4. Оценка сложности. Сложность алгоритмов с числами.
5. Классы сложности.
6. Рекурсия. Дерево рекурсии.
7. Принцип «разделяй и властвуй».
8. Рекурсивный алгоритм перебора всех подмножеств.
9. Рекурсивный алгоритм поиска расстановок.
10. Рекурсивный алгоритм, перебирающий все возможные размещения.
11. Бинарный поиск.

12. Алгоритм линейного поиска.
13. Пузырьковая сортировка.
14. Сортировка слиянием.
15. Быстрая сортировка.

К экзамену (второй семестр)

1. Понятие класса. Объявление класса.
2. Статические и динамические поля.
3. Методы. Статические и динамические методы.
4. Методы расширения.
5. Статические классы.
6. Наследование: понятие, иерархия.
7. Класс Array.
8. Понятие интерфейса.
9. Интерфейс IComparable. Реализация интерфейса.
10. Интерфейс IComparer. Реализация интерфейса.
11. Полиморфизм. Перегрузка и переопределение методов. Виртуальные методы. Ключевые слова virtual и override.
12. Ключевое слово private. Свойства (get, set).
13. Конструкторы.
14. Поля readonly и статические конструкторы.
15. Структуры. Инициализация полей структуры.
16. Ключевые слова ref и out.
17. Структуры данных: стек, очередь, дек.
18. Реализация очереди на связных списках.
19. Понятие «апкаст» и «даункаст».
20. Дженерик-классы.
21. Использование стека для анализа скобочных выражений.
22. Использование очереди для нахождения скользящего среднего.
23. Использование дженерик-методов для создания универсальной сортировки.
24. Дженерик-классы Tuple и Nullable.
25. Интерфейсы IEnumerable и IEnumerator. Реализация интерфейсов.
26. Реализация метода GetEnumerator с помощью конструкции yield return.
27. Списки. Индексация в списках.
28. Реализация методов Contains и Equals.
29. Перегрузка операторов.
30. Хеш-функции. Виртуальный метод GetHashCode.
31. Алгоритм нахождения подстроки в строке с помощью хеш-функции.
32. Делегаты: понятие и свойства.
33. Дженерик-делегаты.
34. Конструкции Func и Action. Анонимные делегаты.
35. Лямбда-выражения.
36. Язык LINQ. Реализация метода Where.
37. Язык LINQ. Реализация методов Select и ToList.
38. Язык LINQ. Методы фильтрации и преобразования.
39. Язык LINQ. Функции агрегирования и группировки.
40. События. Определение и вызов событий. Обработчики событий.
41. Использование делегатов для обработки событий.
42. Мультикаст-делегаты.
43. Назначение protected virtual OnTick, sender и EventArgs.
44. WindowsForm и WPF
45. Элементы управления.
46. Реализация резинового дизайна. Менеджеры компоновки.
47. Рисование.
48. Таймеры и анимация.
49. Паттерн MVC.

50. Потоки, домены и процессы. Класс Thread.
51. Методы BeginInvoke и EndInvoke для управления потоками.
52. Класс Parallel для асинхронного выполнения функций. Общие ресурсы и lock.
53. Блокирование потока с графическим интерфейсом. Асинхронные операции в приложения с GUI.
54. Конструкции async и await. Класс BackgroundWorker.
55. Рефлексия типов.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену**Примерные практические задания к экзамену (зимний семестр)**

1. В трехмерном пространстве заданы две точки своими координатами. Определить, во сколько раз расстояние между ними отличается от модуля суммы произведений их соответствующих координат.
2. Вычислить длины сторон треугольника по заданным координатам его вершин в трехмерном пространстве.
3. Для параллелограмма с заданными сторонами и острым углом между ними определить его площадь, высоты и длины диагоналей.
4. Проверить, есть ли среди введенных трех целых чисел одинаковые.
5. Даны три точки A, B, C на числовой оси. Найти длины отрезков AC и BC и их сумму.
6. Решить линейное уравнение $Ax + B = 0$, заданное своими коэффициентами A и B. Составить 3 модульных теста, проверяющих правильность программы.
7. Заданы три целых положительных числа. Проверить, является ли третье из введенных чисел делителем какого-либо из первых двух. Составить 3 модульных теста, проверяющих правильность программы.
8. Дано целое число $N (> 0)$. Найти сумму $1 + 1/2 + 1/3 + \dots + 1/N$.
9. Найти значение факториала заданного числа N.
10. Найти все простые числа в диапазоне от a до b (вводятся с клавиатуры), или сообщить, что в указанном диапазоне таких чисел нет.
11. Число, сумма всех делителей которого (исключая его само) равна этому числу, называется совершенным. Например: $6=1+2+3$, $28=1+2+4+7+14$. Найти все совершенные числа в диапазоне от a до b (вводятся с клавиатуры).
12. Дано целое число $N (> 0)$. Найти сумму $1^2 + 2^2 + 3^2 + \dots + N^2$. Чтобы избежать целочисленного переполнения, вычислять слагаемые этой суммы с помощью вещественной переменной и выводить результат как вещественное число.
13. Дано целое число $N (> 0)$. Найти наименьшее целое положительное число K, квадрат которого превосходит N: $K^2 > N$. Функцию извлечения квадратного корня не использовать.
14. Задана последовательность целых чисел. Сформировать на основе нее массив, содержащий номера отрицательных элементов последовательности. Если таких элементов нет, – вывести сообщение об этом.
15. Задана последовательность действительных чисел. Подсчитать количество чисел последовательности, расположенных на отрезке, концами которого являются два последних элемента массива.
16. Задан одномерный массив вещественных чисел. Найти в нем максимальный и минимальный элементы, и вывести их с указанием их индексов.
17. Задана последовательность, содержащая натуральные числа. Найти сумму максимального элемента (среди четных чисел) и минимального элемента (среди нечетных чисел) заданной последовательности.
18. В двумерном массиве найти позиции всех максимальных элементов, если таких элементов несколько. Если максимальный элемент только один, - вывести его значение.
19. Даны матрица A размером $N \times N$ и вектор B размером N. Записать на главную диагональ элементы вектора, а в вектор – элементы главной диагонали.
20. В трехмерном пространстве заданы координаты N точек. Составить программу, определяющую пару точек, расстояние между которыми максимально.
21. Задан массив символов. Вывести на печать те из них, которые являются цифрами.
22. Задана строка текста. Определить, сколько раз встречается в ней заданный символ.

23. Во введенной строке символов найти пары, состоящие из одинаковых символов и заменить второй из них цифрой «1».
24. Составить программу, которая из заданной строки символов выводит буквы по алфавиту. Например: строка – «программирование», результат – «аавгеииммноопррр».
25. Задан текстовый файл, среди символов которого могут быть цифры. Подсчитать общую сумму всех цифр, содержащихся в файле.

Примерные практические задания к экзамену (летний семестр)

1. 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.
2. 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.
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.
4. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую все разработанные элементы класса.
5. 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.
6. 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.
7. 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 метод изменения времени на заданное количество часов, минут и секунд. Написать программу, демонстрирующую все разработанные элементы класса.
8. 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.
9. 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.
 10. 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.
 11. 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.
 12. 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.
 13. 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.
 14. 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.
 15. 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.
 16. 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.
 17. 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.
 18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.

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

Программирование

Тест

Вопрос 1. Укажите, какое из нижеследующих утверждений ложное.

- 1: Комментарии при выполнении программы вызывают печать компьютером на экране текста после символов `//`.
- 2: Если вывод осуществляется в `cout`, то `esc`-последовательность `\n` вызывает перемещение курсора к началу следующей строки на экране.
- 3: Все переменные должны быть объявлены до того, как они используются.
- 4: Все высказывания истинные

Вопрос 2. Укажите, какое из нижеследующих утверждений ложное.

- 1: Всем переменным, когда они объявляются, должен быть присвоен тип.
- 2: C++ рассматривает переменные `number` и `NUMbEr` как одинаковые.
- 3: Объявления в теле функции C++ могут появляться почти везде.
- 4: Все высказывания истинные

Вопрос 3. Укажите, какое из нижеследующих утверждений ложное.

- 1: Операция взятия по модулю (%) может применяться только к целым числам.
- 2: Все арифметические операции `*`, `/`, `%`, `+` и `-` имеют одинаковый уровень приоритета.
- 3: Пустые скобки, следующие за именем функции в прототипе, указывают, что функции для выполнения ее задачи не требуется никаких параметров.
- 4: Все высказывания истинные

Вопрос 4. Укажите, какое из нижеследующих утверждений ложное.

- 1: Тело каждой функции ограничивается левой и правой фигурными скобками (`{` и `}`).
- 2: Программа на C++, которая выводит три строки, должна содержать три оператора вывода, использующих `cout`.
- 3: Элементы данных или элемент-функции, объявленные со спецификатором доступа `private`, доступны для элемент-функций класса, в котором они объявлены.
- 4: Все высказывания истинные

Вопрос 5. Укажите, какое из нижеследующих утверждений истинное.

- 1: В соответствии с соглашением, имена функций начинаются с прописной буквы, и все последующие слова в имени также начинаются с прописной буквы.
- 2: Переменные, объявленные в теле некоторой элемент-функции, называются элементами данных и могут использоваться во всех элемент-функциях класса.
- 3: Для исполнения программы можно использовать любой файл исходного кода, содержащий `int main()`.
- 4: Указатель, объявленный как `void`, может быть разыменован.

Вопрос 6. Укажите, какое из нижеследующих утверждений истинное.

- 1: Типы аргументов в вызове функции должны быть согласованы с типами соответствующих параметров в списке параметров прототипа функции.
- 2: В операторе выбора `switch` должна быть метка `default`.
- 3: В операторе выбора `switch` в разделе `default` требуется оператор `break`.
- 4: Все высказывания истинные

Вопрос 7. Укажите, какое из нижеследующих утверждений истинное.

- 1: Выражение, содержащее операцию `||`, истинно, если истинны оба операнда этой операции.
- 2: Массив может хранить много различных типов данных.
- 3: Операция взятия адреса может быть применима только к константам, к выражениям, не дающим в качестве результата ссылки, и к переменным, объявленным с классом памяти `register`.
- 4: Все высказывания ложные

Вопрос 8. Укажите, какое из нижеследующих утверждений истинное.

- 1: Если количество начальных значений в списке инициализации меньше, чем количество элементов массива, оставшиеся элементы автоматически получают в качестве начальных значений последние значения из списка инициализации.
- 2: Если список инициализации содержит начальных значений больше, чем элементов массива, то это - ошибка.
- 3: Отдельный элемент массива, который передается функции и модифицируется в этой функции, будет содержать модифицированное значение после завершения выполнения вызываемой функции.
- 4: Все высказывания ложные

Вопрос 9. Укажите, какое из нижеследующих утверждений истинное.

- 1: Когда исполнение программы приостанавливается в контрольной точке, следующим оператором, который будет исполняться, является оператор после контрольной точки.
- 2: Когда значение переменной изменяется, в окнах Autos и Locals оно выделяется желтым цветом.
- 3: Во время отладки команда Step Out исполняет все оставшиеся операторы текущей функции и возвращает управление в точку, где была вызвана функция.
- 4: Все высказывания истинные

Вопрос 10. Укажите, какое из нижеследующих утверждений ложно.

- 1: Указатели разных типов нельзя присваивать друг другу без операции приведения типа.
- 2: Выражение $(x > y \ \&\& \ a < b)$ истинно, если $x > y$ или $a < b$.
- 3: Все высказывания ложные
- 4: Индексы массива обычно должны иметь тип float.

Вопрос 11. Выполнение каждой программы на C++ начинается с функции _____.

- 1: void
- 2: include
- 3: main
- 4: using

Вопрос 12. Каждый оператор заканчивается _____.

- 1: endl
- 2: return
- 3: точкой с запятой
- 4: управляющей последовательностью

Вопрос 13. Оператор _____ используется для принятия решений

- 1: cout
- 2: if
- 3: while
- 4: for

Вопрос 14. Оператор выбора _____ используется для выполнения одного действия, если его условие истинно, и другого действия, если условие ложно

- 1: If
- 2: If...then
- 3: If...then...else
- 4: If...else

Вопрос 15. Повторение набора инструкций заданное число раз называется _____ повторений

- 1: управляемым счетчиком
- 2: числом
- 3: заданием

Вопрос 16. Когда заранее не известно, сколько раз должна исполняться группа операторов, для прерывания повторения можно использовать _____ значение

- 1: контрольное
- 2: сигнальное
- 3: фиктивное
- 4: все ответы верны

Вопрос 17. Списки и таблицы значений хранятся в _____

- 1: массиве
- 2: указателе
- 3: списке
- 4: стандартной библиотеке

Вопрос 18. Элементы массива связаны тем, что они имеют одни и те же _____

- 1: имя
- 2: тип
- 3: значения
- 4: размерность

Вопрос 19. Число, используемое для обращения к отдельному элементу массива называется _____

- 1: тип
- 2: индекс
- 3: значение
- 4: позиция

Вопрос 20. Для объявления размера массива должна использоваться _____, потому что она делает программу более масштабируемой

- 1: переменная
- 2: константа
- 3: именованная константа
- 4: символ

Вопрос 21. Процесс упорядоченного размещения элементов в массиве называется _____

- 1: сортировка
- 2: поиск
- 3: проверка
- 4: изменение

Вопрос 22. Процесс определения значения ключа, содержащегося в массиве, называется _____

- 1: сортировка
- 2: поиск
- 3: проверка
- 4: изменение

Вопрос 23. Указатель - это переменная, которая содержит в качестве своего значения _____ другой переменной

- 1: индекс
- 2: адрес
- 3: код
- 4: ссылку

Вопрос 24. Для инициализации указателя можно использовать

- 1: null
- 2: 0
- 3: адрес
- 4: все перечисленное

Вопрос 25. Целое, которое может быть присвоено указателю, является

- 1: 0
- 2: положительным
- 3: отрицательным
- 4: все перечисленное

Вопрос 26. Что такое идентификаторы в языке Си++ ?

- 1: это последовательность знаков, начинающаяся с буквы или знака подчеркивания
- 2: это последовательность знаков
- 3: это последовательность знаков, начинающаяся с буквы
- 4: это последовательность знаков, начинающаяся со знака подчеркивания

Вопрос 27. В идентификаторах можно использовать

- 1: заглавные и строчные латинские буквы, цифры и знак подчеркивания
- 2: заглавные и строчные латинские буквы, цифры и знак подчеркивания
- 3: заглавные и строчные латинские буквы
- 4: заглавные и строчные латинские буквы, цифры
- 5: заглавные и строчные латинские буквы, знак подчеркивания

Вопрос 28. Что является выражениями в языке Си++ ?

- 1: это переменные, функции и константы объединенные знаками операций
- 2: это переменные объединенные знаками операций
- 3: это функции объединенные знаками операций
- 4: это константы объединенные знаками операций

Вопрос 29. Какая из операций не относится к операции сравнения?

- 1: (=)
- 2: (>)
- 3: (<)
- 4: (==)
- 5: (!=)

Вопрос 30. В результате успешной компиляции текста программы на С++ с каким расширением будет получен файл?

- 1: .obj
- 2: .exe
- 3: .cpp
- 4: .h

Вопрос 31. Если не будет указан базовый тип, то какой тип будет подразумеваться по умолчанию

- 1: int
- 2: char
- 3: short
- 4: long

Контрольная работа №1

Задания к контрольной работе №1

1. Вывести цифровое окно секундомера с кнопками **Сброс**, **Старт**, **Финиш**. Текущее время следует оперативно выводить в секундах и миллисекундах в отдельном поле через каждые 100 миллисекунд.
2. Вывести список из пяти различных рисунков, выбрав из списка нужный рисунок, вывести его.
3. Разработать программу-игру “Сапер”. Число мин и их расположение в таблице (7 строк на семь колонок) загадывается случайным образом. Если мина в соседней ячейке от текущей (по которой щелкнули мышкой), то выводится символ-предупреждение “!”. Если щелкнули по ячейке с миной, то выводится знак “X” и игра прекращается с сообщением: “Сапер убит”. Во всех остальных случаях выводится символ “-”. Для обезвреживания мины используется двойной щелчок по ячейке с миной. На экран выводится текущее число двойных щелчков. Выигрыш наступает, когда все мины обезврежены и число двойных щелчков минимально; на экран выводится сообщение “Мин нет”.
4. Составить столбиковую диаграмму для 10 случайных чисел в интервале 0-60.
5. Составить круговую диаграмму для 17 случайных чисел в интервале 0-34.
6. Разработать ActiveX-элемент кнопки **Выход**.
7. Изменить тип, размер и стиль шрифта в диалоге с выводом новых значений и образца.
8. Составить программу для тестирования знаний логических операций “И”, “Или”, “Нет”.
9. Составить форму с действующей блок-схемой нахождения максимального числа из трех вводимых пользователем чисел с кнопками **Начало**, **Далее** (переход на следующий шаг), **Выход**. Текущий символ блок-схемы выделяется красным цветом. Результаты выполнения шага помещают в текстовое поле.
10. Вывести бегущую строку с текстом, введенным в текстовое поле.
11. Реализовать игру “Крестики - нолики” между пользователем и компьютером (таблица 3x3).
12. Составить программу для тестирования знаний четырех любых простых законов физики. Числа случайные, целые. При неверном ответе выводить правильный ответ. За каждый неверный ответ оценка снижается на балл.
13. Составить программу компьютерного тестирования знаний. В текстовом редакторе создать файл с текстом. Каждое описание вопроса занимает несколько строк: текст вопроса, номер правильного ответа и строки с текстами вариантов ответа (не более семи), пустая строка - разделитель описаний вопросов. В форму последовательно выводятся вопрос с вариантами ответов, и тестируемый щелкает мышкой на предполагаемом правильным ответе. В конце выводится оценка по формуле: $5 \times \text{число верных ответов} / \text{число всех вопросов}$. На форме разместить кнопки **Начало**, **Выход**.
14. Игра “Поле чудес”. На форме представлены следующие элементы: поле для ввода загадываемого слова и кнопок его очистки, начала и выхода; поле с темой, к которой относится слово; поля для ввода буквы, с угаданными и неугаданными буквами; число попыток.
15. Составить программу “Будильник” с кнопками установки текущего времени, времени подачи звукового сигнала, отключения звука. Вывести на экран текущее время и время сигнала.
16. Для заданной целой матрицы случайных чисел (1-1000) размером 10×15 вывести красным цветом все ее седловые точки (п. 16.6, задача 8).

Контрольная работа №2

Задания к контрольной работе №2

1. Написать программу, которая записывает в файл последовательного доступа квадраты и кубы 30 первых натуральных чисел.

2. Написать программу, которая записывает в файл последовательного доступа информацию о студентах, проживающих в общежитии: фамилию, индекс группы, домашний адрес, номер комнаты в общежитии. Если в качестве фамилии студента вводится пустая строка, то выполнение программы прекращается.

3. Написать программу, в которой создается файл прямого доступа, содержащий сведения о числе деталей с заданным кодом. Код детали равен номеру записи. Сведения о числе сведены в целочисленный массив A (10).

4. Написать программу создания файла прямого доступа, содержащего информацию о личной библиотеке. Информация о книгах вводится с клавиатуры. Номером записи является шифр книги.

5. Написать программу создания файла «Телефонный справочник» объемом не более 100 абонентов, содержащего следующие сведения об абоненте: фамилия абонента, адрес, номер телефона.

6. Написать программу создания файла, содержащего сведения о сдаче сессии студентами специальности: индекс группы, фамилия студента, шифры предметов и оценки.

7. Результаты соревнований по шести видам спорта летней Олимпиады 2000 г. записаны в файл прямого доступа OLIMP. Написать программу, которая выполняет одну из следующих функций:

а) выдает данные в виде таблицы о всех призерах Олимпиады - золотая, серебряная, бронзовая медали по запрашиваемому виду спорта;

б) выдает список призеров страны.

8. Создать файл «ТОВАР», содержащий сведения о товарах, хранящихся на складе: код товара, наименование товара, количество единиц, стоимость одной единицы. Все записи в файле должны быть отсортированы в порядке возрастания товара.

9. Написать программу, которая на основании информации, содержащейся в файле «ТОВАР» (с полями: наименование товара, дата поступления, количество и единица измерения):

а) добавляет новые товары, поступившие на склад;

б) производит корректировку записи о товаре с кодом XX - изменение количества, стоимости единицы товара.

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

Программирование

Экзамен №1

Экзаменационный билет №1

1. Числовые типы данных.
2. Рекурсия. Дерево рекурсии.
3. Число, сумма всех делителей которого (исключая его само) равна этому числу, называется совершенным. Например: $6=1+2+3$, $28=1+2+4+7+14$. Найти все совершенные числа в диапазоне от a до b (вводятся с клавиатуры).

Экзаменационный билет №2

1. Строковый тип данных. Основные методы работы со строками. Перевод числа в строку.
2. Принцип «разделяй и властвуй».
3. В трехмерном пространстве заданы две точки своими координатами. Определить, во сколько раз расстояние между ними отличается от модуля суммы произведений их соответствующих координат.

Экзаменационный билет №3

1. Арифметические операции в C#. Класс Math.
2. Рекурсивный алгоритм перебора всех подмножеств.
3. Вычислить длины сторон треугольника по заданным координатам его вершин в трехмерном пространстве.

Экзаменационный билет №4

1. Методы: объявление метода, аргументы метода, возвращаемые значения. Перегрузка методов.
2. Рекурсивный алгоритм поиска расстановок.
3. Для параллелограмма с заданными сторонами и острым углом между ними определить его площадь, высоты и длины диагоналей.

Экзаменационный билет №5

1. Глобальные и локальные переменные. Область видимости переменных.
2. Рекурсивный алгоритм, перебирающий все возможные размещения.
3. Проверить, есть ли среди введенных трех целых чисел одинаковые.

Экзаменационный билет №6

1. Ошибки на этапе компиляции и выполнения. `NullReferenceException`. `IndexOutOfRangeException`. Блок `try—catch`.
2. Бинарный поиск.
3. Даны три точки A, B, C на числовой оси. Найти длины отрезков AC и BC и их сумму.

Экзаменационный билет №7

1. Логический тип данных. Операторы сравнения.
2. Алгоритм линейного поиска.
3. Решить линейное уравнение $Ax + B = 0$, заданное своими коэффициентами A и B. Составить 3 модульных теста, проверяющих правильность программы.

Экзаменационный билет №8

1. Сравнение чисел с плавающей точкой.
2. Пузырьковая сортировка.
3. Заданы три целых положительных числа. Проверить, является ли третье из введенных чисел делителем какого-либо из первых двух. Составить 3 модульных теста, проверяющих правильность программы.

Экзаменационный билет №9

1. Оператор `If – Else`.
2. Сортировка слиянием.
3. Дано целое число N (> 0). Найти сумму $1 + 1/2 + 1/3 + \dots + 1/N$.

Экзаменационный билет №10

1. Оператор `Switch – Case`.
2. Быстрая сортировка.
3. Найти значение факториала заданного числа N.

Экзаменационный билет №11

1. Цикл `while`. Операторы `break` и `continue`.
2. Понятие алгоритма. Свойства алгоритма.
3. Найти все простые числа в диапазоне от a до b (вводятся с клавиатуры), или сообщить, что в указанном диапазоне таких чисел нет.

Экзаменационный билет №12

1. Цикл for.
2. Временная и емкостная сложность алгоритма. Расчет сложности.
3. Число, сумма всех делителей которого (исключая его само) равна этому числу, называется совершенным. Например: $6=1+2+3$, $28=1+2+4+7+14$. Найти все совершенные числа в диапазоне от a до b (вводятся с клавиатуры).

Экзаменационный билет №13

1. Массивы. Индексация массива. Цикл foreach.
2. O-символика.
3. Дано целое число $N (> 0)$. Найти сумму $1^2 + 2^2 + 3^2 + \dots + N^2$. Чтобы избежать целочисленного переполнения, вычислять слагаемые этой суммы с помощью вещественной переменной и выводить результат как вещественное число.

Экзаменационный билет №14

1. Типы ссылки и типы значения. Стек вызовов. Куча.
2. Оценка сложности. Сложность алгоритмов с числами.
3. Дано целое число $N (> 0)$. Найти наименьшее целое положительное число K , квадрат которого превосходит N : $K^2 > N$. Функцию извлечения квадратного корня не использовать.

Экзаменационный билет №15

1. Многомерные массивы. Массивы массивов.
2. Классы сложности.
3. Задана последовательность целых чисел. Сформировать на основе нее массив, содержащий номера отрицательных элементов последовательности. Если таких элементов нет, – вывести сообщение об этом.

Экзаменационный билет №16

1. Списки. Методы списка. Измерение времени работы программы.
2. Рекурсия. Дерево рекурсии.
3. Задана последовательность действительных чисел. Подсчитать количество чисел последовательности, расположенных на отрезке, концами которого являются два последних элемента массива.

Экзаменационный билет №17

1. Словари.
2. Принцип «разделяй и властвуй».
3. Задан одномерный массив вещественных чисел. Найти в нем максимальный и минимальный элементы, и вывести их с указанием их индексов

Экзаменационный билет №18

1. Сравнительный анализ строк и массивов. Методы `substring`, `ToUpper` и др. Неизменяемость строк. Паттерн `Immutable`.
2. Рекурсивный алгоритм перебора всех подмножеств.
3. Задана последовательность, содержащая натуральные числа. Найти сумму максимального элемента (среди четных чисел) и минимального элемента (среди нечетных чисел) заданной последовательности.

Экзаменационный билет №19

1. Класс `StringBuilder`.
2. Рекурсивный алгоритм поиска расстановок.
3. В двумерном массиве найти позиции всех максимальных элементов, если таких элементов несколько. Если максимальный элемент только один, - вывести его значение.

Экзаменационный билет №20

1. Специальные символы при работе со строками.
2. Рекурсивный алгоритм, перебирающий все возможные размещения.
3. Даны матрица A размером $N \times N$ и вектор B размером N . Записать на главную диагональ элементы вектора, а в вектор – элементы главной диагонали.

Экзаменационный билет №21

1. Форматированный вывод строк.
2. Бинарный поиск.
3. В трехмерном пространстве заданы координаты N точек. Составить программу, определяющую пару точек, расстояние между которыми максимально.

Экзаменационный билет №22

1. Файлы и каталоги.
2. Алгоритм линейного поиска.

3. Задан массив символов. Вывести на печать те из них, которые являются цифрами.

Экзаменационный билет №23

1. Тестирование. Методы избавления от багов.
2. Пузырьковая сортировка.
3. Задана строка текста. Определить, сколько раз встречается в ней заданный символ.

Экзаменационный билет №24

1. Библиотеки тестирования.
2. Сортировка слиянием.
3. Во введенной строке символов найти пары, состоящие из одинаковых символов и заменить второй из них цифрой «1».

Экзаменационный билет №25

1. Модульные тесты.
2. Быстрая сортировка.
3. Составить программу, которая из заданной строки символов выводит буквы по алфавиту. Например: строка – «программирование», результат – «аавгеииммноопррр».

Экзаменационный билет №26

1. Функциональное тестирование.
2. Понятие алгоритма. Свойства алгоритма.
3. На складе медикаментов решили провести проверку лекарственных препаратов на истечение срока годности. Информация о препаратах заносится в запись, состоящую из следующих полей: название лекарства, производитель, дата выпуска, срок годности, стоимость. Организовать ввод сведений о лекарственных препаратах, а также сформировать список препаратов с истекающим сроком годности. Предусмотреть поиск по названию лекарственного препарата и по годам выпуска.

Экзаменационный билет №27

1. Цикл while. Операторы break и continue.
2. Временная и емкостная сложность алгоритма. Расчет сложности.
3. На складе медикаментов решили провести проверку лекарственных препаратов на истечение срока годности. Информация о препаратах заносится в запись, состоящую из следующих полей: название лекарства, производитель, дата выпуска, срок годности, стоимость. Организовать ввод сведений о

лекарственных препаратах, а также сформировать список препаратов с истекающим сроком годности. Предусмотреть поиск по названию лекарственного препарата и по годам выпуска.

Экзаменационный билет №28

1. Класс `StringBuilder`.
2. O-символика.
3. Задан текстовый файл, среди символов которого могут быть цифры. Подсчитать общую сумму всех цифр, содержащихся в файле.

Экзаменационный билет №29

1. Методы: объявление метода, аргументы метода, возвращаемые значения. Перегрузка методов.
2. Оценка сложности. Сложность алгоритмов с числами.
3. Записать в типизированный файл массив целых чисел, после чего считать данные из файла и найти максимальный элемент.

Экзаменационный билет №30

1. Массивы. Индексация массива. Цикл `foreach`.
2. Классы сложности.
3. Во введенной строке символов найти пары, состоящие из одинаковых символов и заменить второй из них цифрой «1»

Экзамен №2

Экзаменационный билет №1

1. Понятие класса. Объявление класса.
2. Использование стека для анализа скобочных выражений.
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 градуацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

Экзаменационный билет №2

1. Статические и динамические поля.
2. Использование очереди для нахождения скользящего среднего.
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 №3

1. Методы. Статические и динамические методы.
2. Использование дженерик-методов для создания универсальной сортировки..
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 №4

1. Методы расширения.
2. Дженерик-классы Tuple и Nullable.
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 №5

1. Статические классы.
2. Интерфейсы IEnumerable и IEnumerator. Реализация интерфейсов.
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 №6

1. Наследование: понятие, иерархия.
2. Реализация метода GetEnumerator с помощью конструкции yield return.
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 №7

1. Класс Array.
2. Списки. Индексация в списках.
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 №8

1. Понятие интерфейса.
2. Реализация методов Contains и Equals.
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 №9

1. Интерфейс IComparable. Реализация интерфейса.
2. Перегрузка операторов.
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 №10

1. Интерфейс IComparer. Реализация интерфейса.
2. Хеш-функции. Виртуальный метод GetHashCode.
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 №11

1. Полиморфизм. Перегрузка и переопределение методов. Виртуальные методы. Ключевые слова virtual и override.
2. Алгоритм нахождения подстроки в строке с помощью хеш-функции.
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 №12

1. Ключевое слово private. Свойства (get, set).
2. Делегаты: понятие и свойства.
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 №13

1. Конструкторы.
2. Дженерик-делегаты.
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 №14

1. Поля readonly и статические конструкторы.
2. Конструкции Func и Action. Анонимные делегаты.
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 №15

1. Структуры. Инициализация полей структуры.
2. Лямбда-выражения.
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 №16

1. Ключевые слова ref и out.
2. Язык LINQ. Реализация метода Where.
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 №17

1. Структуры данных: стек, очередь, дек.
2. Язык LINQ. Реализация методов Select и ToList.
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 №18

1. Реализация очереди на связных списках.
2. Язык LINQ. Методы фильтрации и преобразования.
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 №19

1. Понятие «апкаст» и «даункаст».
2. Язык LINQ. Функции агрегирования и группировки.
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 №20

1. Дженерик-классы.
2. События. Определение и вызов событий. Обработчики событий.
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.