



КГЭУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования

«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГЭУ»)

«УТВЕРЖДАЮ»

Директор института Цифровых технологий и
экономики

 Торкунова Ю.В.

«26» октября 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование на языке C#

Направление подготовки 09.03.01 Информатика и вычислительная техника

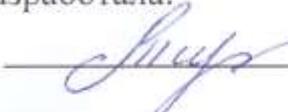
Направленность Технологии разработки программного обеспечения

Квалификация бакалавр

г. Казань, 2020

Рабочая программа дисциплины разработана в соответствии с ФГОС ВО по направлению подготовки 09.03.01 Информатика и вычислительная техника (уровень бакалавриата) (приказ Минобрнауки России от 19.09.2017 г. № 929)

Программу разработала:

доцент, к.т.н.  Тагиева Рамиля Фаридовна

Рабочая программа рассмотрена и одобрена на заседании кафедры Информатика и информационно-управляющие системы, протокол №24 от 26.10.2020

Зав. кафедрой  Торкунова Ю.В.

Программа рассмотрена и одобрена на заседании выпускающей кафедры Информатика и информационно-управляющие системы, протокол № 24 от 26.10.2020

Зав. кафедрой  Торкунова Ю.В.

Программа одобрена на заседании учебно-методического совета института цифровых технологий и экономики, протокол № 2 от 26.10.2020

Зам. директора института ЦТЭ 

В. В. Косулин

Программа принята решением Ученого совета института Цифровых технологий и экономики, протокол № 2 от 26.10.2020

1. Цель, задачи и планируемые результаты обучения по дисциплине

Целью изучения дисциплины «Объектно-ориентированное программирование на языке С#» является теоретическая и практическая подготовка студентов в области разработки программного обеспечения с использованием объектно-ориентированной модели. Знания, полученные в результате освоения дисциплины, помогут при разработке системных программных компонентов современных информационных и расчетных программ, в проектировании и реализации системных компонентов операционных систем в такой степени, чтобы студенты могли самостоятельно выбирать средства реализации, находить необходимые программные и технологические решения для практически важных системных и предметно-ориентированных задач.

Основные задачи изучения дисциплины:

- приобретение студентами знаний о сущности объектно-ориентированного подхода в программировании;
- ознакомление с технологиями создания новых пользовательских типов данных на языке программирования С#;
- приобретение практических навыков по использованию средств переопределения операций, обработки исключений, механизма наследования классов и интерфейсов, класса LINQ, технологии Entity Framework для разработки приложений.

Компетенции, формируемые у обучающихся, запланированные результаты обучения по дисциплине, соотнесенные с индикаторами достижения компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Запланированные результаты обучения по дисциплине (знать, уметь, владеть)
Профессиональные компетенции (ПК)		
ПК-1 Способен к проектированию и разработке программного обеспечения с применением современных технологий	ПК-1.1 Проектирует и согласовывает с заинтересованными сторонами архитектуру программного обеспечения	<i>Знать:</i> -основные принципы объектно-ориентированного программирования. -основные принципы и этапы проектирования программного обеспечения -основные алгоритмические конструкции и операторы языка С#. <i>Уметь:</i> -составлять программный код на основе объектно-ориентированное модели. -проводить анализ предметной области и составлять UML-диаграммы для проектирования архитектуры программного обеспечения. <i>Владеть:</i> -навыками проектирования и разработки программного обеспечения на основе объектно-ориентированной модели. -навыками проектирования линейных и нелинейных структур данных.

<p>ПК-1 Способен к проектированию и разработке программного обеспечения с применением современных технологий</p>	<p>ПК-1.3 Разрабатывает программные интерфейсы</p>	<p><i>Знать:</i> -основы работы с технологией Entity Framework, выступающей в качестве программного модуля для реализации методов и средств проектирования баз данных. -методы и средства проектирования программных интерфейсов с использованием визуального и событийного подхода к программированию на основе языка программирования С#. -типовые решения задачи, библиотеки программных модулей, шаблоны (паттерны) проектирования, классы объектов, библиотеки классов, используемые при разработке программного обеспечения</p> <p><i>Уметь:</i> -применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов на основе объектно-ориентированной модели, технологии Entity Framework, событийного, визуального подхода к программированию на языке С#. -использовать существующие типовые решения и шаблоны (паттерны) проектирования программного обеспечения на языке С#.</p> <p><i>Владеть:</i> -навыками проектирования и разработки программных интерфейсов. -методами проектирования баз данных на основе технологии Entity Framework.</p>
--	--	---

2. Место дисциплины в структуре ОПОП

Дисциплина Объектно-ориентированное программирование на языке С# относится к части, формируемой участниками образовательных отношений учебного плана по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Код компетенции	Предшествующие дисциплины (модули), практики, НИР, др.	Последующие дисциплины (модули), практики, НИР, др.
УК-1		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
УК-2		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ

УК-3		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
УК-4		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
УК-5		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
УК-6		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
УК-7		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
УК-8		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-1		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-2		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-2	Информационные технологии Базы данных	
ОПК-3		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-4		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-5		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-6		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-7		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ

ОПК-8		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ОПК-8	Базы данных Алгоритмизация и программирование	
ОПК-9		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ПК-1		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ
ПК-2		Выполнение и защита выпускной квалификационной работы ГОСУДАРСТВЕННАЯ ИТОГОВАЯ АТТЕСТАЦИЯ

Для освоения дисциплины обучающийся должен:

Знать: базовые алгоритмы обработки данных, корректные постановки классических задач; аналитические и технологические решения в области программного обеспечения (системного, прикладного и инструментального) и компьютерной обработки информации.

Уметь: разрабатывать алгоритмы, реализовывать алгоритмы на языке программирования высокого уровня, описывать основные структуры данных, реализовывать методы анализа и обработки данных, работать в средах программирования; умеет ориентироваться в информационном потоке, использовать рациональные способы получения, преобразования, систематизации и хранения информации, актуализировать ее в необходимых ситуациях интеллектуально-познавательной деятельности, структурировать информацию; диагностировать работоспособность вычислительной системы и устранять неполадки.

Владеть: методами и технологиями разработки алгоритмов, описания структур данных и других базовых представлений данных, программирования на языке высокого уровня, навыками работы в некоторой среде программирования.

3. Структура и содержание дисциплины

3.1. Структура дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц (ЗЕ), всего 216 часов, из которых 85 часов составляет контактная работа обучающегося с преподавателем (занятия лекционного типа 16 час., занятия семинарского типа (практические, семинарские занятия, лабораторные работы и т.п.) 64 час., групповые и индивидуальные консультации 2 час., прием экзамена (КПА), зачета с оценкой - 1 час., самостоятельная работа обучающегося 96 час. Практическая подготовка по виду профессиональной деятельности составляет 8 часов.

Вид учебной работы	Всего часов	Семестр
		3
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	216	216
КОНТАКТНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ С ПРЕПОДАВАТЕЛЕМ, в том числе:	85	85
Лекционные занятия (Лек)	16	16
Лабораторные занятия (Лаб)	32	32
Практические занятия (Пр)	32	32
Контроль самостоятельной работы и иная контактная работа (КСР)*	2	2
Консультации (Конс)	2	2
Контактные часы во время аттестации (КПА)	1	1
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ (СРС), в том числе:	96	96
Подготовка к промежуточной аттестации в форме: (экзамен)	35	35
ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	Эк	Эк

3.2. Содержание дисциплины, структурированное по разделам и видам занятий

Разделы дисциплины	Семестр	Распределение трудоемкости (в часах) по видам учебной работы, включая СРС							Формируемые результаты обучения (знания, умения, навыки)	Литература	Формы текущего контроля успеваемости	Формы промежуточной аттестации	Максимальное количество баллов по балльно - рейтинговой системе
		Занятия лекционного типа	Занятия практического / семинарского типа	Лабораторные работы	Групповые консультации	Самостоятельная работа студента, в т.ч.	Контроль самостоятельной работы (КСР)	подготовка к промежуточной аттестации					
Раздел 1. Структурный подход к программированию													

1. Базовые алгоритмические конструкции следование, ветвление и циклы.	3	1		3						4	ПК-1.1-31, ПК-1.1-33, ПК-1.3-33, ПК-1.3-У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		1
2. Структуры данных : массивы, строки, списки.	3	1		3						4	ПК-1.1-31, ПК-1.1-33, ПК-1.1-В2, ПК-1.3-33, ПК-1.3-У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		3
3. Работа с потоками файловой системой.	3	1		2		5				8	ПК-1.1-31, ПК-1.1-33, ПК-1.3-33, ПК-1.3-У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР КНР		6
Раздел 2. Объектно-ориентированный подход к программированию															
4. Основные принципы объектно-ориентированной модели.	3	2		2						4	ПК-1.1-31, ПК-1.1-33, ПК-1.1-У1, ПК-1.1-В1, ПК-1.1-В2, ПК-1.3-33, ПК-1.3-У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		3

5. Статические классы. Разработка библиотеки классов.	3			4					4	ПК-1.1 -У1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В1, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		3
6. Организация иерархии классов	3	2		2					4	ПК-1.1 -У1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В1, ПК-1.3 -33, ПК-1.3 -У1, ПК-1.1 -В2	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		3
7. Модель "делегирования включения"	3			3					3	ПК-1.1 -У1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В1, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У1, ПК-1.3 -У2	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		4

8. Интерфейсы. Стандартные интерфейсы.	3	1		1						2	ПК-1.1 -У1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В1, ПК-1.3 -33, ПК-1.3 -У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		2
9. Делегаты.	3	2		2						4	ПК-1.1 -У1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В1, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У1	Л1.5, Л1.4, Л2.3, Л2.4	ЛР		4
10. События.	3			2		5				7	ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -У1, ПК-1.1 -В1, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У1, ПК-1.3 -У2	Л1.5, Л1.4, Л2.3, Л2.4, Л2.1	ЛР КНР		8

11. Класс Linq.	3	2		2						4	ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У1, ПК-1.3 -У2	Л2.3	ЛР		3
-----------------	---	---	--	---	--	--	--	--	--	---	---	------	----	--	---

Раздел 3. Технология Entity Framework.

12. Способы взаимодействия с базой данных.	3	2				5					7	ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.3 -31, ПК-1.3 -32, ПК-1.3 -В1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.3 -33, ПК-1.3 -У1, ПК-1.3 -В2	Л2.2			
--	---	---	--	--	--	---	--	--	--	--	---	---	------	--	--	--

13. CRUD- операции.	3			2						2	ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.3 -31, ПК-1.3 -32, ПК-1.3 -В1, ПК-1.3 -У1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У2, ПК-1.3 -В2	Л2.2	ЛР		3
------------------------	---	--	--	---	--	--	--	--	--	---	---	------	----	--	---

14. Связь «один ко многим», "многие-ко-многим", "один-к -одному".	3	2		4	5					11	ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.3 -31, ПК-1.3 -32, ПК-1.3 -В1, ПК-1.3 -У1, ПК-1.1 -31, ПК-1.1 -32, ПК-1.1 -33, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У2, ПК-1.3 -В2	Л2.2	ЛР		5
Раздел 4. UML															
15. Стадия исследования предметной области и проектирования.	3		16		30					46	ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.3 -32, ПК-1.3 -В1, ПК-1.1 -31, ПК-1.1 -32, ПК-1.3 -У1	Л1.1, Л1.2, Л1.3	ЛР		6

<p>16. Стадия разработки программного продукта и тестирования.</p>	3	16	46						56	<p>ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.3 -31, ПК-1.3 -32, ПК-1.3 -В1, ПК-1.1 -31, ПК-1.1 -32, ПК-1.1 -33, ПК-1.1 -В2, ПК-1.3 -33, ПК-1.3 -У1, ПК-1.3 -У2, ПК-1.3 -В2</p>	<p>Л1.1, Л1.2, Л1.3, Л2.2, Л1.5, Л1.4, Л2.3, Л2.4</p>	<p>ПР</p>	7
--	---	----	----	--	--	--	--	--	----	---	---	-----------	---

Промежуточная аттестация	3					2	35	1	38	ПК-1.1 -У1, ПК-1.1 -У2, ПК-1.1 -В1, ПК-1.3 -31, ПК-1.3 -32, ПК-1.3 -У1, ПК-1.3 -В1, ПК-1.1 -31, ПК-1.1 -33, ПК-1.1 -В2, ПК-1.3 -33	Л1.5, Л1.4, Л2.3, Л2.4	Эк	40
ИТОГО		16	32	32		96	2	35	1	216			100

3.3. Тематический план лекционных занятий

Номер раздела дисциплины	Темы лекционных занятий	Трудоемкость, час.
1	Программирование на Visual Studio, ЯП С#. .Net - открытая среда разработки. Лексические основы языка С#. Типы данных. Операторы языка С#.	1
2	Программирование на Visual Studio, ЯП С#. Массивы. Строки. Списки.	1
3	Программирование на ЯП С#. BinaryReader, BinaryWriter, FileStream, StreamReader, StreamWriter, File, Directory.	1
4	Основные элементы класса: поля, методы, конструкторы, свойства. Виды параметров методов.	2
5	Организация иерархий классов. Раннее и позднее связывание. Виртуальные методы. Абстрактные и бесплодные классы. Виды взаимоотношений между классами.	2
6	Описание и использование интерфейсов. Применение стандартных интерфейсов .NET для сравнения, перебора, сортировки и клонирования объектов.	1
7	Делегаты. Обеспечения связи между объектами по типу «источник — наблюдатель».Создания универсальных методов (поддержки механизма обратных вызовов). Анонимные методы. События. Лямбда-выражения.	2
8	Linq to Objects. Сортировка, фильтрация, группировка, агрегатные операции.	2

9	Entity Data Model. Code first. Операции Create, Read, Update, Delete.	2
10	Связь «один ко многим». Связь "один-к-одному". Связь "многие-ко-многим". Eager loading("жадная загрузка"), explicit loading("явная загрузка"), lazy loading("ленивая загрузка").	2
Всего		16

3.4. Тематический план практических занятий

Номер раздела дисциплины	Темы практических занятий	Трудоемкость, час.
1	Разработка UML диаграмм: диаграмма прецедентов, диаграмма последовательностей, диаграмма действий, диаграмма классов, диаграмма состояний.	16
2	Разработка программного продукта на основе UML диаграмм, полученных на стадии проектирования, объектно-ориентированной модели и технологии Entity Framework.	16
Всего		32

3.5. Тематический план лабораторных работ

Номер раздела дисциплины	Темы лабораторных работ	Трудоемкость, час.
1	Класс математических функций Math. Следование. Условный оператор. Условная тернарная операция.	2
2	Циклы. Оператор обработки исключений.	1
3	Одномерные, двумерные, ступенчатые массивы.	2
4	Работа со списками.	1
5	Работа со строками. Сохранение и чтение данных из текстового файла.	2
6	Разработка пользовательского типа данных. Методы. Переопределение операций.	2
7	Создание библиотеки классов со статическими методами.	2
8	Индексаторы.	2
9	Создание иерархии пользовательских типов данных. Виртуальные методы.	2
10	Создание приложения согласно паттерну "делегирование-включение".	3
11	Организация клонирования (копирования) объектов, сравнения и перебора на основе применения стандартных интерфейсов IComparable, IComparer, IEnumerable, IEnumerator, ICloneable.	1
12	Делегаты. Применение метода функциональной параметризации для реализации метода половинного деления отрезка с целью решения трансцендентных уравнений.	2
13	Событийное программирование. События. Задача на регистрацию нарушений скоростного режима. Моделирование игры в ping-pong.	2
14	Применение операторов класса Linq для организации обработки данных пользовательских классов.	2

15	Разработка консольного приложения согласно варианту задания на основе применения технологии Entity Framework и подхода Code First.	2
16	Визуальное программирование. Создание визуального приложения на основе технологии Entity Framework. Создание связи между данными "один-ко-многим".	4
Всего		32

3.6. Самостоятельная работа студента

Номер раздела дисциплины	Вид СРС	Содержание СРС	Трудоемкость, час.
1	Подготовка к контрольной работе по структурному подходу программирования.	Повторение пройденного материала. Изучение конспектов лекций и программных кодов выполненных лабораторных работ.	5
2	Подготовка к контрольной работе по объектно-ориентированному программированию.	Повторение пройденного материала. Изучение конспектов лекций и программных кодов выполненных лабораторных работ.	5
3	Изучение технологии разработки программного обеспечения и информационных систем на основе технологии Entity Framework. Database first. Model first.	Краткий конспект по предложенной теме.	5
4	LINQ to Entities. Введение в LINQ to Entities. Выборка и проекция из базы данных. Сортировка. Соединение таблиц. Операции с множествами. Агрегатные операции.	Краткий конспект по предложенной теме.	5

5	Изучение бесплатного программного продукта UML Star применяемого на стадии проектирования информационных систем, разрабатываемых на основе объектно-ориентированной модели. Создание UML диаграмм : диаграмма прецедентов, диаграмма последовательностей, диаграмма действий, диаграмма классов, диаграмма состояний.	Отчет с диаграммами UML согласно варианту задания.	30
6	Разработка программного продукта согласно выбранной теме.	Полнофункциональный, рабочий программный продукт.	46
Всего			96

4. Образовательные технологии

По основным формам организации образовательного процесса: чтение лекций, проведение лабораторных занятий, практических работ, организация самостоятельной образовательной деятельности, организация и проведение консультаций, проведение экзаменов. Все виды занятий проводятся с использованием технических средств обучения, презентаций. По основным видам и формам деятельности преподавателей: тестирования, общения преподавателя со студентами, организации групповой работы, организации самостоятельной деятельности. В качестве основных форм самостоятельной работы студентов предполагается аналитическая обработка текста (аннотирование, конспектирование); работа со справочной литературой; выполнение индивидуальных заданий; работа в электронной среде LMS Moodle. Также используются дистанционные образовательные технологии, реализуемые в электронной форме через сеть Интернет с применением площадки LMS Moodle, ссылка на курс <https://lms.kgeu.ru/course/view.php?id=2663>, а также электронные образовательные ресурсы (ЭОР), размещенные в личных кабинетах студентов Электронного университета КГЭУ.

5. Оценивание результатов обучения

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля успеваемости, проводимого по балльно-рейтинговой системе (БРС), и промежуточной аттестации.

Текущий контроль успеваемости осуществляется в течение семестра, включает: индивидуальный и групповой опрос (устный или письменный), защиты лабораторных работ; проведение компьютерного тестирования.

Итоговой оценкой результатов освоения дисциплины является оценка, выставленная во время промежуточной аттестации обучающегося (экзамен) с учетом результатов текущего контроля успеваемости. На экзамен выносятся теоретические и практические задания, проработанные в течение семестра на учебных занятиях и в процессе самостоятельной работы обучающихся. Экзаменационные билеты содержат 2 теоретических задания.

Обобщенные критерии и шкала оценивания уровня сформированности компетенции (индикатора достижения компетенции) по итогам освоения дисциплины:

Планируемые результаты обучения	Обобщенные критерии и шкала оценивания результатов обучения			
	неудовлетворительно	удовлетворительно	хорошо	отлично
	не зачтено	зачтено		
Полнота знаний	Уровень знаний ниже минимальных требований, имеют место грубые ошибки	Минимально допустимый уровень знаний, имеет место много негрубых ошибок	Уровень знаний в объеме, соответствующем программе, имеет место несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки, без ошибок

Наличие умений	При решении стандартных задач не продемонстрированы основные умения, имеют место грубые ошибки	Продemonстрированы основные умения, решены типовые задачи с негрубыми ошибками, выполнены все задания, но не в полном объеме	Продemonстрированы все основные умения, решены все основные задачи с негрубыми ошибками, выполнены все задания в полном объеме, но некоторые с недочетами	Продemonстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме
Наличие навыков (владение опытом)	При решении стандартных задач не продемонстрированы базовые навыки, имеют место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов
Характеристика сформированности компетенции (индикатора)	Компетенция в полной мере не сформирована. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных)	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практи-	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для
Достижения компетенции)	задач	ческих (профессиональных) задач, но требуется дополнительная практика по большинству практических задач	стандартных практических (профессиональных) задач	решения сложных практических (профессиональных) задач
Уровень сформированности компетенции (индикатора достижения компетенции)	Низкий	Ниже среднего	Средний	Высокий

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора достижения	Запланированные результаты обучения по дисциплине	Уровень сформированности компетенции (индикатора достижения компетенции)			
			Высокий	Средний	Ниже среднего	Низкий
Шкала оценивания						

			отлично	хорошо	удовлетворительно	неудовлетворительно	
			зачтено				не зачтено
ПК-1	ПК-1.1	Знать					
		Основные принципы объектно-ориентированного программирования.	Знает полностью все принципы ООП.	Знает практически все основные принципы ООП.	Знает некоторые принципы ООП.	Уровень знаний ниже минимального уровня требований.	
		Основные принципы и этапы проектирования программного обеспечения	Знает полностью все принципы и этапы проектирования информационных систем.	Знает практически все принципы и этапы проектирования информационных систем.	Знает некоторые принципы и этапы проектирования информационных систем.	Уровень знаний ниже минимального уровня требований.	
		Основные алгоритмические конструкции и операторы языка C#.	Знает синтаксис операторов и основных алгоритмических конструкций языка C#.	Знает практически весь синтаксис операторов и основных алгоритмических конструкций языка C#.	Знает синтаксис некоторых операторов и основных алгоритмических конструкций языка C#.	Уровень знаний ниже минимального уровня требований.	
		Уметь					
		Составлять программный код на основе объектно-ориентированной модели.	Демонстрирует умение составлять программы на основе принципов инкапсуляции, наследования и полиморфизма не допуская ошибок.	Демонстрирует умение составлять программы на основе принципов инкапсуляции, наследования и полиморфизма, допускает ряд незначительных ошибок.	Демонстрирует умение составлять программы на основе принципов инкапсуляции, наследования и полиморфизма. Задания выполняет не в полном объеме.	Допускает грубые ошибки при составлении программ на основе принципов инкапсуляции, наследования и полиморфизма.	

		Проводить анализ предметной области и составлять UML-диаграммы для проектирования архитектуры программного обеспечения.	Демонстрирует умение проводить анализ предметной области, составлять UML-диаграммы, проектировать архитектуру программного обеспечения.	Демонстрирует умение проводить анализ предметной области, составлять UML-диаграммы, проектировать архитектуру программного обеспечения. Допускает ряд незначительных ошибок.	Демонстрирует умение проводить анализ предметной области, составлять UML-диаграммы, проектировать архитектуру программного обеспечения. Задания выполняет не в полном объеме.	Допускает грубые ошибки при проведении анализа предметной области, составлении UML-диаграмм, проектировании и архитектуры программного обеспечения.
Владеть						
		навыками проектирования и разработки программного обеспечения на основе объектно-ориентированной модели.	Демонстрирует хорошие навыки проектирования и разработки программного обеспечения на основе объектно-ориентированной модели.	Демонстрирует базовые навыки проектирования и разработки программного обеспечения на основе объектно-ориентированной модели.	Демонстрирует минимальные навыки проектирования и разработки программного обеспечения на основе объектно-ориентированной модели.	Допускает грубые ошибки на стадиях проектирования и разработки программного обеспечения на основе объектно-ориентированной модели.
		навыками проектирования линейных и нелинейных структур данных.	Демонстрирует хорошие навыки проектирования линейных и нелинейных структур данных.	Демонстрирует базовые навыки проектирования линейных и нелинейных структур данных.	Демонстрирует минимальные навыки проектирования линейных и нелинейных структур данных.	Допускает грубые ошибки при проектировании линейных и нелинейных структур данных.

ПК-1	ПК-1.3	Знать				
		<p>основы работы с технологией Entity FrameWork, выступающей в качестве программного модуля для реализации методов и средств проектирования баз данных.</p>	<p>Знает три подхода разработки: Database first, Model first, Code first.</p>	<p>Знает два подхода разработки: Model first, Code first.</p>	<p>Знает один подход разработки Code first.</p>	<p>Уровень знания ниже минимального уровня требований.</p>
		<p>методы и средства проектирования программных интерфейсов с использованием визуального и событийного подхода к программированию на основе языка программирования C#.</p>	<p>Знает полностью все методы и средства проектирования программных интерфейсов с использованием визуального и событийного подхода к программированию.</p>	<p>Знает практически все основные методы и средства проектирования программных интерфейсов с использованием визуального и событийного подхода к программированию.</p>	<p>Знает некоторые методы и средства проектирования программных интерфейсов с использованием визуального и событийного подхода к программированию.</p>	<p>Уровень знаний ниже минимального уровня требований.</p>
		<p> типовые решения задачи, библиотеки программных модулей, шаблоны (паттерны) проектирования, классы объектов, библиотеки классов используемые при разработке программного обеспечения</p>	<p>Знает типовые решения задач, библиотеки программных модулей, шаблоны (паттерны) проектирования, классы объектов, библиотеки классов используемые при разработке программного обеспечения.</p>	<p>Знает базовые типовые решения задач, библиотеки программных модулей, шаблоны (паттерны) проектирования, классы объектов, библиотеки классов используемые при разработке программного обеспечения.</p>	<p>Знает некоторые типовые решения задач, библиотеки программных модулей, шаблоны (паттерны) проектирования, классы объектов, библиотеки классов используемые при разработке программного обеспечения.</p>	<p>Уровень знаний ниже минимального уровня требований.</p>
Уметь						

		<p>применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов на основе объектно-ориентированной модели, технологии Entity Framework, событийного, визуального подхода к программированию на языке C#.</p>	<p>Демонстрирует отличные умения применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов на основе объектно-ориентированной модели, технологии Entity Framework, событийного, визуального подхода к программированию на языке C#</p>	<p>Демонстрирует хорошие умения применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов на основе объектно-ориентированной модели, технологии Entity Framework, событийного, визуального подхода к программированию на языке C#</p>	<p>Демонстрирует минимальные умения применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов на основе объектно-ориентированной модели, технологии Entity Framework, событийного, визуального подхода к программированию на языке C#</p>	<p>Допускает грубые ошибки при применении методов и средств проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов на основе объектно-ориентированной модели, технологии Entity Framework, событийного, визуального подхода к программированию на языке C#</p>
		<p>использовать существующие типовые решения и шаблоны (паттерны) проектирования программного обеспечения на языке C#.</p>	<p>Демонстрирует отличные умения при использовании существующих типовых решений и шаблонов (паттернов) проектирования программного обеспечения на языке C#.</p>	<p>Демонстрирует хорошие умения при использовании существующих типовых решений и шаблонов (паттернов) проектирования программного обеспечения на языке C#.</p>	<p>Демонстрирует минимальные умения при использовании существующих типовых решений и шаблонов (паттернов) проектирования программного обеспечения на языке C#.</p>	<p>Допускает грубые ошибки при использовании существующих типовых решений и шаблонов (паттернов) проектирования программного обеспечения на языке C#.</p>
	Владеть					
		<p>навыками проектирования и разработки программных интерфейсов.</p>	<p>Демонстрирует хорошие навыки проектирования и разработки программных интерфейсов.</p>	<p>Демонстрирует базовые навыки проектирования и разработки программных интерфейсов.</p>	<p>Демонстрирует минимальные навыки проектирования и разработки программных интерфейсов.</p>	<p>Допускает грубые ошибки при проектировании и разработке программных интерфейсов.</p>

		методами проектирования баз данных на основе технологии Entity Framework.	Демонстрирует отличные навыки применения методов проектирования баз данных на основе технологии Entity Framework.	Демонстрирует базовые навыки применения методов проектирования баз данных на основе технологии Entity Framework.	Демонстрирует минимальные навыки применения методов проектирования баз данных на основе технологии Entity Framework.	Допускает грубые ошибки при применении методов проектирования баз данных на основе технологии Entity Framework.
--	--	---	---	--	--	---

Оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации приведены в Приложении к рабочей программе дисциплины. Полный комплект заданий и материалов, необходимых для оценивания результатов обучения по дисциплине, хранится на кафедре-разработчике в бумажном и электронном виде.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Учебно-методическое обеспечение

Основная литература

№ п/п	Автор(ы)	Наименование	Вид издания (учебник, учебное пособие, др.)	Место издания, издательство	Год издания	Адрес электронного ресурса	Кол-во экземпляров в библиотеке КГЭУ
1	Тюкачев Н. А., Хлебостроев В. Г.	С#. Алгоритмы и структуры данных	учебное пособие	СПб.: Лань	2017	https://e.lanbook.com/book/94748	
2	Тюкачев Н. А., Хлебостроев В. Г.	С#. Основы программирования	учебное пособие	СПб.: Лань	2017	https://e.lanbook.com/book/94749	
3	Рочев К. В.	Информационные технологии. Анализ и проектирование информационных систем	учебное пособие	СПб.: Лань	2019	https://e.lanbook.com/book/122181	
4	Гвоздева Т. В., Баллод Б. А.	Проектирование информационных систем. Стандартизация	учебное пособие	СПб.: Лань	2019	https://e.lanbook.com/book/115515	

5	Остроух А. В., Суркова Н. Е.	Проектирование информационных систем	монография	СПб.: Лань	2019	https://e.lanbook.com/book/118650	
---	------------------------------	--------------------------------------	------------	------------	------	---	--

Дополнительная литература

№ п/п	Автор(ы)	Наименование	Вид издания (учебник, учебное пособие, др.)	Место издания, издательство	Год издания	Адрес электронного ресурса	Кол-во экземпляров в библиотеке КГЭУ
1	Симонова Е. В.	Структуры данных в С#: линейные и нелинейные динамические структуры	учебное пособие	СПб.: Лань	2018	https://e.lanbook.com/book/110938	
2	Павловская Т. А.	С#. Программирование на языке высокого уровня	учебник для вузов	СПб.: Питер	2012	https://ibooks.ru/reading.php?productid=28473	
3	Малыхина М. П.	Базы данных : основы, проектирование, использование	учебное пособие	СПб.: БХВ-Петербург	2007		20

6.2. Информационное обеспечение

6.2.1. Электронные и интернет-ресурсы

№ п/п	Наименование электронных и интернет-ресурсов	Ссылка
1	Научная электронная библиотека	http://elibrary.ru
2	Российская государственная библиотека	http://www.rsl.ru
3	Международная реферативная база данных научных изданий zbMATH	http://www.zbmath.org
4	Международная реферативная база данных научных изданий Springerlink	http://link.springer.com
5	Образовательный портал	http://www.ucheba.com

6.2.2. Профессиональные базы данных

№ п/п		Адрес	Режим доступа
1	Справочная правовая система	http://consultant.ru	

6.2.3. Информационно-справочные системы

№ п/п	Наименование информационно-справочных систем	Адрес	Режим доступа
1	Научная электронная библиотека	http://elibrary.ru	http://www.consultant.ru/
2	Российская государственная библиотека	http://www.rsl.ru	
3	Международная реферативная база данных научных изданий zbMATH	http://www.zbmath.org	

6.2.4. Лицензионное и свободно распространяемое программное обеспечение дисциплины

№ п/п	Наименование программного обеспечения	Способ распространения (лицензионное/свободно)	Реквизиты подтверждающих документов
1	Windows 7 Профессиональная (Pro)	Пользовательская операционная система	ЗАО "СофтЛайнТрейд" №2011.25486 от 28.11.2011 Неискл. право. Бессрочно
2	LMS Moodle	ПО для эффективного онлайн-взаимодействия преподавателя и студента	Свободная лицензия Неискл. право. Бессрочно
3	Браузер Chrome	Система поиска информации в сети интернет	Свободная лицензия Неискл. право. Бессрочно
4	Visual Studio Community	Инструмент создания Web приложений	Свободная лицензия, тип(вид) лицензий - неискл. Право, срок -
5	Office Standard 2007 Russian OLP NL AcademicEdition+:	Офисные приложения	договор №21/2010 от 04.05.2010, лицензиар - ЗАО «Софт Лайн

7. Материально-техническое обеспечение дисциплины

№ п/п	Вид учебной работы	Наименование специальных помещений и помещений для СРС	Оснащенность специальных помещений и помещений для СРС
1	Лекционные занятия	Учебная аудитория для проведения занятий лекционного типа	<p>Оснащение: доска аудиторная, акустическая система, проектор, усилитель-микшер для систем громкой связи, экран, микрофон, миникомпьютер, монитор</p> <p>Программное обеспечение : 1. Windows 7 Профессиональная (Pro): договор №2011.25486 №2011.25486 от 28.11.2011, лицензиар – ЗАО «Софт Лайн Трейд», тип (вид) лицензии – неискл. право, срок действия лицензии - бессрочно. 2. Office Standard 2007 Russian OLP NL AcademicEdition+: договор</p>

			<p>№21/2010 от 04.05.2010, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. 3. Браузер Chrome, свободная лицензия, тип (вид) лицензии – неискл.право, срок действия лицензии - бессрочно. 4. LMS Moodle, свободная лицензия, тип (вид) лицензии – неискл.право, срок действия лицензии - бессрочно</p>
2	Практические занятия	Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	<p>Оснащение: персональный компьютер (26 шт.), интерактивная доска, мультимедийный проектор.</p> <p>Программное обеспечение:</p> <p>1. Windows 7 Профессиональная (Pro): договор №2011.25486 от 28.11.2011 , лицензиар – ЗАО «Софт Лайн Трейд», тип (вид) лицензии – неискл. право, срок действия лицензии - бессрочно. 2. Office Standard 2007 Russian OLP NL AcademicEdition+: договор №21/2010 от 04.05.2010, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно 3. Браузер Chrome, LMS Moodle. Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. ПО в свободном доступе: 4. Visual Studio Community . Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. ПО в свободном доступе:.</p>
3	Лабораторные работы	Учебная аудитория для проведения занятий семинарского типа, групповых и индивидуальных консультаций, лабораторных работ, текущего контроля и промежуточной аттестации	<p>Оснащение: персональный компьютер (26 шт.), интерактивная доска, мультимедийный проектор.</p> <p>Программное обеспечение:</p> <p>1. Windows 7 Профессиональная (Pro): договор №2011.25486 от 28.11.2011 , лицензиар – ЗАО «Софт Лайн Трейд», тип (вид) лицензии –</p>

			<p>неискл. право, срок действия лицензии - бессрочно.</p> <p>2. Office Standard 2007 Russian OLP NL AcademicEdition+: договор №21/2010 от 04.05.2010, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно</p> <p>3. Браузер Chrome, LMS Moodle. Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. ПО в свободном доступе:</p> <p>4. Visual Studio Community . Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. ПО в свободном доступе:.</p>
4	Самостоятельная работа обучающегося	Компьютерный класс с выходом в Интернет В-600а	<p>Оснащение: моноблок (30 шт.), проектор, экран</p> <p>Программное обеспечение: Windows 10: договор № Tr096148 от 29.09.2020, лицензиар - ООО "Софтлайн трейд", тип (вид) лицензии - неискл. право, срок действия лицензии - до 14.09.2021.</p> <p>Office Standard 2007 Russian OLP NL AcademicEdition+: договор №21/2010 от 04.05.2010, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии – бессрочно;</p> <p>Браузер Chrome, свободная лицензия, тип (вид) лицензии – неискл.право, срок действия лицензии – бессрочно; LMS Moodle, свободная лицензия, тип (вид) лицензии – неискл.право, срок действия лицензии - бессрочно</p>
5	Промежуточная аттестация	Учебная аудитория для проведения промежуточной аттестации	<p>Оснащение: доска аудиторная, персональный компьютер (25 шт.), проектор</p> <p>Программное обеспечение:</p> <p>1. Windows 7 Профессиональная (Pro): договор №2011.25486 от 28.11.2011 , лицензиар – ЗАО «Софт Лайн Трейд», тип (вид) лицензии –</p>

			<p>неискл. право, срок действия лицензии - бессрочно.</p> <p>2. Office Standard 2007 Russian OLP NL AcademicEdition+: договор №21/2010 от 04.05.2010, лицензиар - ЗАО «Софт Лайн Трейд», тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно</p> <p>3. Браузер Chrome, LMS Moodle. Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. ПО в свободном доступе:</p> <p>4. Visual Studio Community . Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно. ПО в свободном доступе:.</p>
--	--	--	--

8. Особенности организации образовательной деятельности для лиц с ограниченными возможностями здоровья и инвалидов

Лица с ограниченными возможностями здоровья (ОВЗ) и инвалиды имеют возможность беспрепятственно перемещаться из одного учебно-лабораторного корпуса в другой, подняться на все этажи учебно-лабораторных корпусов, заниматься в учебных и иных помещениях с учетом особенностей психофизического развития и состояния здоровья.

Для обучения лиц с ОВЗ и инвалидов, имеющих нарушения опорно-двигательного аппарата, обеспечены условия беспрепятственного доступа во все учебные помещения. Информация о специальных условиях, созданных для обучающихся с ОВЗ и инвалидов, размещена на сайте университета [www//kgeu.ru](http://kgeu.ru). Имеется возможность оказания технической помощи ассистентом, а также услуг сурдопереводчиков и тифлосурдопереводчиков.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушенным слухом справочного, учебного материала по дисциплине обеспечиваются следующие условия:

- для лучшей ориентации в аудитории, применяются сигналы оповещения о начале и конце занятия (слово «звонок» пишется на доске);
- внимание слабослышащего обучающегося привлекается педагогом жестом (на плечо кладется рука, осуществляется нерезкое похлопывание);
- разговаривая с обучающимся, педагогический работник смотрит на него, говорит ясно, короткими предложениями, обеспечивая возможность чтения по губам.

Компенсация затруднений речевого и интеллектуального развития слабослышащих обучающихся проводится путем:

- использования схем, диаграмм, рисунков, компьютерных презентаций с гиперссылками, комментирующими отдельные компоненты изображения;
- регулярного применения упражнений на графическое выделение существенных признаков предметов и явлений;
- обеспечения возможности для обучающегося получить адресную консультацию по электронной почте по мере необходимости.

Для адаптации к восприятию лицами с ОВЗ и инвалидами с нарушениями зрения справочного, учебного, просветительского материала, предусмотренного образовательной программой по выбранному направлению подготовки, обеспечиваются следующие условия:

- ведется адаптация официального сайта в сети Интернет с учетом особых потребностей инвалидов по зрению, обеспечивается наличие крупношрифтовой справочной информации о расписании учебных занятий;
- педагогический работник, его собеседник (при необходимости), присутствующие на занятии, представляются обучающимся, при этом каждый раз называется тот, к кому педагогический работник обращается;
- действия, жесты, перемещения педагогического работника коротко и ясно комментируются;
- печатная информация предоставляется крупным шрифтом (от 18 пунктов), тотально озвучивается;
- обеспечивается необходимый уровень освещенности помещений;- предоставляется возможность использовать компьютеры во время занятий и право записи объяснений на диктофон (по желанию обучающихся).

Форма проведения текущей и промежуточной аттестации для обучающихся с ОВЗ и инвалидов определяется педагогическим работником в соответствии с учебным планом.

При необходимости обучающемуся с ОВЗ, инвалиду с учетом их индивидуальных психофизических особенностей дается возможность пройти промежуточную аттестацию устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п., либо предоставляется дополнительное время для подготовки ответа.

Раздел 9. Методические рекомендации для преподавателей по организации воспитательной работы с обучающимися

Методическое обеспечение процесса воспитания обучающихся выступает одним из определяющих факторов высокого качества образования. Преподаватель вуза, демонстрируя высокий профессионализм, эрудицию, четкую гражданскую позицию, самодисциплину, творческий подход в решении профессиональных задач, в ходе образовательного процесса способствует формированию гармоничной личности.

При реализации дисциплины преподаватель может использовать следующие методы воспитательной работы:

- методы формирования сознания личности (беседа, диспут, внушение, инструктаж, контроль, объяснение, пример, самоконтроль, рассказ, совет, убеждение и др.);

- методы организации деятельности и формирования опыта поведения (задание, общественное мнение, педагогическое требование, поручение, приучение, создание воспитывающих ситуаций, тренинг, упражнение, и др.);

- методы мотивации деятельности и поведения (одобрение, поощрение социальной активности, порицание, создание ситуаций успеха, создание ситуаций для эмоционально-нравственных переживаний, соревнование и др.)

При реализации дисциплины преподаватель должен учитывать следующие направления воспитательной деятельности:

Гражданское и патриотическое воспитание:

формирование у обучающихся целостного мировоззрения, российской идентичности, уважения к своей семье, обществу, государству, принятым в семье и обществе духовно-нравственным и социокультурным ценностям, к национальному, культурному и историческому наследию, формирование стремления к его сохранению и развитию;

формирование у обучающихся активной гражданской позиции, основанной на традиционных культурных, духовных и нравственных ценностях российского общества, для повышения способности ответственно реализовывать свои конституционные права и обязанности;

развитие правовой и политической культуры обучающихся, расширение конструктивного участия в принятии решений, затрагивающих их права и интересы, в том числе в различных формах самоорганизации, самоуправления, общественно-значимой деятельности;

формирование мотивов, нравственных и смысловых установок личности, позволяющих противостоять экстремизму, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам, межэтнической и межконфессиональной нетерпимости, другим негативным социальным явлениям.

Духовно-нравственное воспитание:

воспитание чувства достоинства, чести и честности, совестливости, уважения к родителям, учителям, людям старшего поколения;

формирование принципов коллективизма и солидарности, духа милосердия и сострадания, привычки заботиться о людях, находящихся в трудной жизненной ситуации;

формирование солидарности и чувства социальной ответственности по отношению к людям с ограниченными возможностями здоровья, преодоление психологических барьеров по отношению к людям с ограниченными возможностями;

формирование эмоционально насыщенного и духовно возвышенного отношения к миру, способности и умения передавать другим свой эстетический опыт.

Культурно-просветительское воспитание:

- формирование уважения к культурным ценностям родного города, края, страны;

- формирование эстетической картины мира;

- повышение познавательной активности обучающихся.

Научно-образовательное воспитание:

- формирование у обучающихся научного мировоззрения;

- формирование умения получать знания;

- формирование навыков анализа и синтеза информации, в том числе в профессиональной области.

Структура дисциплины по заочной форме обучения

Вид учебной работы	Всего часов	Курс
		3
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	216	216
КОНТАКТНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ С ПРЕПОДАВАТЕЛЕМ, в том числе:	21	21
Лекционные занятия (Лек)	4	4
Лабораторные занятия (Лаб)	4	4
Практические занятия (Пр)	8	8
Контроль самостоятельной работы и иная контактная работа (КСР)*	4	4
Контактные часы во время аттестации (КПА)	1	1
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ (СРС):	187	187
Подготовка к промежуточной аттестации в форме: (экзамен)	8	8
ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	Эк	Эк

Лист внесения изменений

Дополнения и изменения в рабочей программе дисциплины на 2021 /2022
учебный год

В программу вносятся следующие изменения:

1. РПД дополнена разделом 9 «Методические рекомендации для преподавателей по организации воспитательной работы с обучающимися» (стр.32-33).

Программа одобрена на заседании кафедры –разработчика « 17 » 06 2021г.,
протокол № 9 Зав. кафедрой Ю.В.Торкунова

Программа одобрена методическим советом института _____ ИЦТЭ
« 22 » 06 2021г., протокол № 10

Зам. директора по УМР _____  В.В. Косулин

Согласовано:

Руководитель ОПОП _____  С.М. Куценко

*Приложение к рабочей программе
дисциплины*



КГЭУ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное
учреждение высшего образования**

**«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГЭУ»)**

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Объектно-ориентированное программирование на языке C#

Направление подготовки 09.03.01 Информатика и вычислительная техника

Направленность (профиль) Технологии разработки программного обеспечения

Квалификация бакалавр

Форма обучения очная

Рецензия

на оценочные материалы для проведения промежуточной аттестации обучающихся по дисциплине «Объектно-ориентированное программирование на языке С#»

Содержание ОМ соответствует требованиям федерального государственного стандарта высшего образования по направлению подготовки 09.03.01 Информатика и вычислительная техника и учебному плану.

ОМ соответствует требованиям, предъявляемым к структуре, содержанию ОМ по дисциплине, а именно:

1 Перечень формируемых компетенций, которыми должен овладеть обучающийся в результате освоения дисциплины, соответствует ФГОС ВО и профстандарту, будущей профессиональной деятельности выпускника.

2 Показатели и критерии оценивания компетенций, а также шкалы оценивания обеспечивают возможность проведения всесторонней оценки результаты обучения, уровней сформированности компетенций.

3 Контрольные задания и иные материалы оценки результатов освоения разработаны на основе принципов оценивания: валидности, определённости, однозначности, надёжности, а также соответствуют требованиям к составу и взаимосвязи оценочных средств, полноте по количественному составу оценочных средств и позволяют объективно оценить результаты обучения, уровни сформированности компетенций.

4 Методические материалы ОМ содержат чётко сформулированные рекомендации по проведению процедуры оценивания результатов обучения и сформированности компетенций.

2. Направленность ОМ по дисциплине соответствует целям ОПОП ВО по направлению 09.03.01 «Информатика и вычислительная техника», профстандартам.

3. Объём ОМ соответствует учебному плану подготовки.

4. Качество ОМ в целом обеспечивают объективность и достоверность результатов при проведении оценивания с различными целями.

Заключение. На основании проведенной экспертизы можно сделать заключение, что ОМ по дисциплине соответствует требованиям ФГОС ВО, профессионального стандарта, современным требованиям рынка труда и рекомендуются для использования в учебном процессе.

Следует отметить, что созданы условия для максимального приближения системы оценки и контроля компетенций обучающихся к условиям их будущей профессиональной деятельности.

Рассмотрено на заседании учебно-методического совета ИЦТЭ «26» октября 2020 г., протокол № 2

Председатель УМС ИЦТЭ
Рецензент



Ю.В.Торкунова

инженер поддержки программно-аппаратных комплексов ООО «ДжиДиСи Сервисез», к.ф.-м.н.



А.С.Петрова

Оценочные материалы по дисциплине «Объектно-ориентированное программирование на языке C#» - комплект контрольно-измерительных материалов, предназначенных для оценивания результатов обучения на соответствие индикаторам достижения компетенции:

ПК-1 Способен к проектированию и разработке программного обеспечения с применением современных технологий

Оценивание результатов обучения по дисциплине осуществляется в рамках текущего контроля успеваемости, проводимого по балльно-рейтинговой системе (БРС), и промежуточной аттестации.

Текущий контроль успеваемости обеспечивает оценивание процесса обучения по дисциплине. При текущем контроле успеваемости используются следующие оценочные средства: контрольная работа, лабораторная работа, тест, практическая работа.

Промежуточная аттестация имеет целью определить уровень достижения запланированных результатов обучения по дисциплине за 3 семестр. Форма промежуточной аттестации экзамен.

Оценочные материалы включают задания для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, разработанные в соответствии с рабочей программой дисциплины.

1. Технологическая карта

Семестр 3

Номер раздела/ темы дисциплины	Вид СРС	Наименование оценочного средства	Код индикатора достижения компетенций	Уровень освоения дисциплины, баллы			
				неудов-но	удов-но	хорошо	отлично
				не зачтено	зачтено		
				низкий	ниже среднего	средний	высокий
Текущий контроль успеваемости							

1	Изучение теоретического материала. Подготовка отчета по выполнению лабораторной	ЛР	ПК-1.1, ПК-1.3	менее 3	3-3	3-4	4-5
1	Контрольная работа по объектно-ориентированному программированию	КР	ПК-1.1, ПК-1.3	Менее 3	3-4	3-4	4-5
2	Изучение теоретического материала. Подготовка отчета по выполнению лабораторной	ЛР	ПК-1.1, ПК-1.3	Менее 8	9-16	17-20	21-26
2	Контрольная работа по объектно-ориентированному программированию	КР	ПК-1.1, ПК-1.3	менее 2	3-3	3-4	4-5
3	Изучение теоретического материала. Подготовка отчета по выполнению лабораторной	ЛР	ПК-1.1, ПК-1.3	Менее 3	3-4	4-5	5-6
4	Изучение теоретического материала. Подготовка отчета по выполнению лабораторной	ПР	ПК-1.1, ПК-1.3	Менее 8	9-10	10-12	12-13

Промежуточная аттестация	Экз	ПК-1, ПК-1	менее 25	25-29	30-34	35-40
Всего баллов			0 - 54	55-69	70-84	85-100

2. Перечень оценочных средств

Краткая характеристика оценочных средств, используемых при текущем контроле успеваемости и промежуточной аттестации обучающегося по дисциплине:

Наименование оценочного средства	Краткая характеристика оценочного средства	Оценочные материалы
Контрольная работа (КнР)	Одна из форм проверки и оценки усвоенных знаний, получения информации о характере познавательной деятельности, уровне самостоятельности и активности студента в учебном процессе, об эффективности методов, форм и способов учебной деятельности.	Текст задания контрольной работы, 30 вариантов
Лабораторная работа (ЛР)	Лабораторная работы выполняется согласно заданию. Отчет по лабораторной работе оформляется индивидуально каждым студентом.	Задания к лабораторным работам
Тест (Тест)	Тест из 40 вопросов.	Тест из 40 вопросов различной сложности.
Практическая работа (ПР)	Практическая работа выполняется в соответствие с выбранной темой. Заключается в построении UML диаграмм. Отчет по практической работе оформляется отдельно каждым студентом.	Задание к практической работе
Экзамен (Экз)	Экзамен проводится в тестовой форме и включает в себя ответ на экзаменационные вопросы.	Перечень экзаменационных вопросов

3. Оценочные материалы текущего контроля успеваемости обучающихся

Наименование оценочного средства	Лабораторная работа 1. Класс математических функций Math. Следование. Условный оператор. Условная тернарная операция.
Представление и содержание оценочных материалов	<p>Задание 1.</p> <p>С клавиатуры введено натуральное число, выражающее расстояние в миллиметрах. Вывести это расстояние на экран, выделив метры, сантиметры и миллиметры.</p> <p>Задание 2.</p>

Составить программу вычисления значений функций z_1 и z_2 , для которых значение аргумента введите с клавиатуры. Округлить полученный результат до тысячных и вывести на экран. Задание выполнить по вариантам.

- | | |
|--|--|
| 1. $z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$ | $z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right).$ |
| 2. $z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$ | $z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right).$ |
| 3. $z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha};$ | $z_2 = 2 \sin \alpha.$ |
| 4. $z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha};$ | $z_2 = \operatorname{tg} 3\alpha.$ |
| 5. $z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha;$ | $z_2 = \cos^2 \alpha + \cos^4 \alpha.$ |
| 6. $z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha;$ | $z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2}\alpha \cdot \cos 4\alpha.$ |
| 7. $z_1 = \cos^2\left(\frac{3}{8}\pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{\alpha}{4}\right);$ | $z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}.$ |
| 8. $z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1;$ | $z_2 = \sin(y + x) \cdot \sin(y - x).$ |
| 9. $z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2;$ | $z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta).$ |
| 10. $z_1 = \left(\sin\left(\frac{\pi}{2} + 3\alpha\right)\right) / (1 - \sin(3\alpha - \pi));$ | $z_2 = \operatorname{ctg}\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right).$ |

Задание 1 (тернарная условная операция)

Составить программу для вычисления значения функции y при заданных с клавиатуры значениях a , b и x .

Варианты заданий

- | | |
|--|---|
| 1. $y = \begin{cases} ax^2 + b \lg 2x , & \sqrt{a+b} < x \\ \sqrt{a + \sin 2x}, & \sqrt{a+b} \geq x \end{cases}$ | 2. $y = \begin{cases} \ln x^2 - e^{\frac{x}{3}}, & 3a > b \\ \operatorname{arctg}(2x - 0,6), & 3a \leq b \end{cases}$ |
| 3. $y = \begin{cases} e^{\sin x} + b, & 3b^2 > a \\ e^{-x} + a \lg x, & 3b^2 \leq a \end{cases}$ | 4. $y = \begin{cases} \frac{x}{a^2} - e^{\cos x}, & a^2 - b^2 > b \\ \operatorname{tg} 4x - a, & a^2 - b^2 \leq b \end{cases}$ |
| 5. $y = \begin{cases} \ln ax - b - e^{\operatorname{tg} x}, & a - b^2 > b \\ \operatorname{tg} 4x - a, & a - b^2 \leq b \end{cases}$ | 6. $y = \begin{cases} \frac{ax - b}{2 \ln x - e^{10}}, & 3a - b > 0 \\ \operatorname{arcsin}\left(\frac{x}{10}\right), & 3a - b \leq 0 \end{cases}$ |

$$7. y = \begin{cases} e^{-2x} + \sqrt[4]{a^4 + x}, & ax < b \\ \sin x - b^2, & ax \geq b \end{cases}$$

$$8. y = \begin{cases} be^{-3x} + \sin^2 a, & ab < 2x \\ \frac{x^3}{a} - \frac{x}{b}, & ab \geq 2x \end{cases}$$

$$9. y = \begin{cases} ax^2 + \operatorname{tg} 4x, & \sqrt{a+3b} < 3x \\ \sqrt{a + \sin|3x|}, & \sqrt{a+3b} \geq 3x \end{cases}$$

$$10. y = \begin{cases} \ln x^2 - \arcsin\left(\frac{x}{10}\right), & 3a > 2b \\ \operatorname{arctg}(2x - 0,6) + 2 \ln x, & 3a \leq 2b \end{cases}$$

Задание 2 (условный оператор if)

Составить программу, которая запрашивает четырехзначное число и определяет, равна ли сумма двух первых его цифр сумме двух его последних цифр.

Задание 3 (оператор выбора switch)

В старояпонском календаре был принят двенадцатилетний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, петуха, собаки и свиньи. Написать программу, которая позволяет ввести номер года и печатает его название по старояпонскому календарю. Справка: 1996 г. — год крысы — начало очередного цикла.

Критерии оценки и шкала оценивания в баллах

Максимальная оценка – 1 б.
Минимальная оценка – 0,6 б.

Наименование оценочного средства

Лабораторная работа 2. Циклы. Оператор обработки исключений.

Представление и содержание оценочных

Задание 1. (оператор цикла while) Вычислить и вывести на экран в виде таблицы значения функции, заданной графически, на интервале от $x_{\text{нач}}$ до $x_{\text{кон}}$ с шагом dx . Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить шапкой.

Примечание:

Уравнение прямой на плоскости:

материал
ЛОВ

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

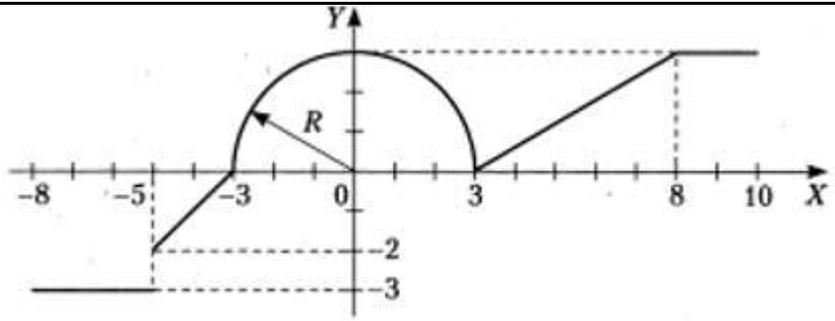
Уравнение окружности:

$$(x - x_0)^2 + (y - y_0)^2 = R^2,$$

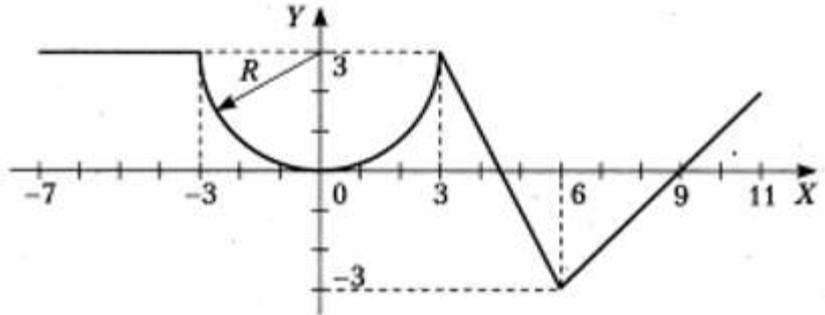
x_0, y_0 - координаты центра окружности.

№ варианта	Задание
1.	
2.	
3.	
4.	

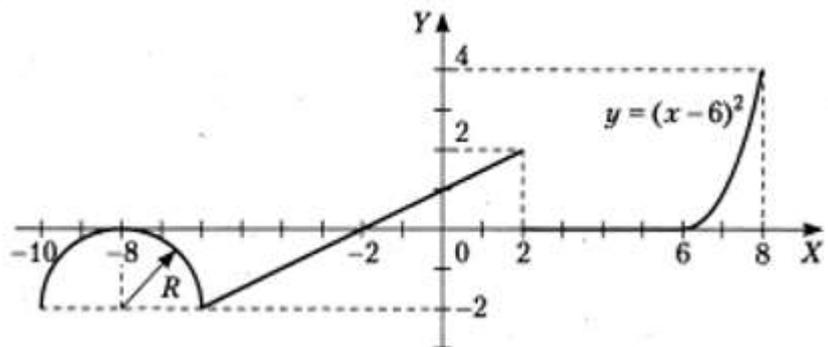
5.



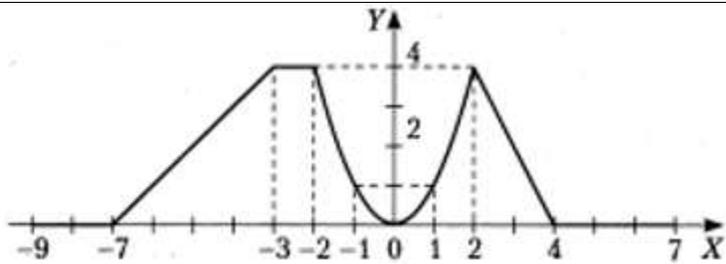
6.



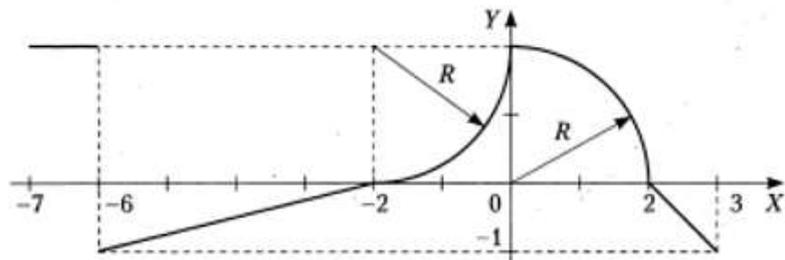
7.



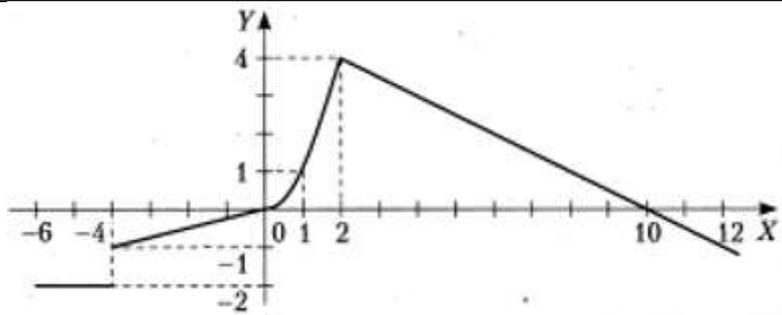
8.



9.



10.



<p>Критерии оценки и шкала оценивания в баллах</p>	<p>Максимальная оценка – 2 б. Минимальная оценка – 1,2 б.</p>								
<p>Наименование оценочного средства</p>	<p>Лабораторная работа 3. Одномерные, двумерные, ступенчатые массивы.</p>								
<p>Представление и содержание оценочных материалов</p>	<p>Задание 1. Одномерные массивы.</p> <table border="1" data-bbox="231 862 1412 2094"> <tr> <td data-bbox="231 862 399 1220"> <p>1.</p> </td> <td data-bbox="399 862 1412 1220"> <p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Произведение элементов массива с четными номерами</p> <p>Сумму элементов массива, расположенных между первым и последним нулевыми элементами.</p> <p>Вывести массив на экран.</p> </td> </tr> <tr> <td data-bbox="231 1220 399 1579"> <p>2.</p> </td> <td data-bbox="399 1220 1412 1579"> <p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Сумму элементов массива с нечетными номерами</p> <p>Сумму элементов массива, расположенных между первым и последним отрицательными элементами.</p> <p>Вывести массив на экран.</p> </td> </tr> <tr> <td data-bbox="231 1579 399 1937"> <p>3.</p> </td> <td data-bbox="399 1579 1412 1937"> <p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Максимальный отрицательный элемент массива.</p> <p>Сумму элементов массива, расположенных до последнего положительного элемента массива.</p> <p>Вывести массив на экран.</p> </td> </tr> <tr> <td data-bbox="231 1937 399 2094"> <p>4.</p> </td> <td data-bbox="399 1937 1412 2094"> <p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Номер минимального положительного элемента массива.</p> </td> </tr> </table>	<p>1.</p>	<p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Произведение элементов массива с четными номерами</p> <p>Сумму элементов массива, расположенных между первым и последним нулевыми элементами.</p> <p>Вывести массив на экран.</p>	<p>2.</p>	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Сумму элементов массива с нечетными номерами</p> <p>Сумму элементов массива, расположенных между первым и последним отрицательными элементами.</p> <p>Вывести массив на экран.</p>	<p>3.</p>	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Максимальный отрицательный элемент массива.</p> <p>Сумму элементов массива, расположенных до последнего положительного элемента массива.</p> <p>Вывести массив на экран.</p>	<p>4.</p>	<p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Номер минимального положительного элемента массива.</p>
<p>1.</p>	<p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Произведение элементов массива с четными номерами</p> <p>Сумму элементов массива, расположенных между первым и последним нулевыми элементами.</p> <p>Вывести массив на экран.</p>								
<p>2.</p>	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Сумму элементов массива с нечетными номерами</p> <p>Сумму элементов массива, расположенных между первым и последним отрицательными элементами.</p> <p>Вывести массив на экран.</p>								
<p>3.</p>	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Максимальный отрицательный элемент массива.</p> <p>Сумму элементов массива, расположенных до последнего положительного элемента массива.</p> <p>Вывести массив на экран.</p>								
<p>4.</p>	<p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Номер минимального положительного элемента массива.</p>								

	<p>Среднее арифметическое элементов массива, расположенных между первым и последним отрицательными элементами.</p> <p>Вывести массив на экран.</p>
5.	<p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Максимальный по модулю элемент массива.</p> <p>Среднее арифметическое элементов массива, расположенных между первым и вторым отрицательными элементами.</p> <p>Вывести массив на экран.</p>
6.	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Количество элементов массива, лежащих в диапазоне от A до B.</p> <p>Сумму элементов массива, расположенных после максимального элемента.</p> <p>Вывести массив на экран.</p>
7.	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Количество положительных элементов массива.</p> <p>Сумму элементов массива, расположенных до минимального элемента.</p> <p>Вывести массив на экран.</p>
8.	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p> <p>Количество элементов массива, меньших C.</p> <p>Сумму целых частей массива, расположенных после последнего отрицательного элемента.</p> <p>Вывести массив на экран.</p>
9.	<p>В одномерном массиве, состоящем из n целочисленных элементов, вычислить:</p> <p>Количество отрицательных элементов массива с четными номерами.</p> <p>Сумму элементов массива, расположенных до последнего нулевого элемента.</p> <p>Вывести массив на экран.</p>
10.	<p>В одномерном массиве, состоящем из n вещественных элементов, вычислить:</p>

<p>Количество элементов массива равных нулю.</p> <p>Сумму отрицательных нечетных элементов массива.</p> <p>Вывести массив на экран.</p>

Задание 2. Двумерные массивы.

1.	Дана целочисленная прямоугольная матрица. Определить количество строк, не содержащих ни одного нулевого элемента. Вывести двумерный массив на экран. Поменять местами 1 и 2 строки матрицы. Вывести на экран полученную матрицу.
2.	Дана целочисленная квадратная матрица. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов. Вывести двумерный массив на экран. Поменять местами 1 и 2 столбцы матрицы. Вывести на экран полученную матрицу.
3.	Для заданной матрицы размером 8x8 найти такие k, при которых k-я строка матрицы совпадает с k-м столбцом. Вывести двумерный массив на экран. Поменять местами первую и последнюю строки матрицы. Вывести на экран полученную матрицу.
4.	Дана целочисленная прямоугольная матрица. Определить количество строк, среднее арифметическое которых меньше заданной величины. Вывести двумерный массив на экран. Поменять местами первый и последний столбцы матрицы. Вывести на экран полученную матрицу.
5.	Дана целочисленная квадратная матрица. Определить номера строк, состоящие только из четных элементов. Вывести двумерный массив на экран. Поменять местами 1 и 3 строки матрицы. Вывести на экран полученную матрицу.
6.	Дана целочисленная квадратная матрица. Определить номера строк, состоящие только из четных элементов. Вывести двумерный массив на экран. Поменять местами 1 и 3 столбцы матрицы. Вывести на экран полученную матрицу.
7.	Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного отрицательного элемента. Вывести двумерный массив на экран. Поменять местами 2 последние строки матрицы. Вывести на экран полученную матрицу.
8.	Дана целочисленная квадратная матрица. Определить сумму элементов в тех столбцах, которые не содержат отрицательных элементов. Вывести двумерный массив на экран. Поменять местами 2 последних столбца матрицы. Вывести на экран полученную матрицу.
9.	Дана целочисленная квадратная матрица. Определить максимальный элемент, расположенный на главной диагонали матрицы. Вывести

		<p>двумерный массив на экран. Поменять местами 1 и 2 строки матрицы. Вывести на экран полученную матрицу.</p>
	<p>10.</p>	<p>Дана целочисленная квадратная матрица. Определить номер строки, в которой расположена самая длинная возрастающая последовательность элементов. Вывести двумерный массив на экран. Поменять местами 1 и 2 столбцы матрицы. Вывести на экран полученную матрицу.</p> <p>Задание 3. Ступенчатый массив</p> <p>Задан ступенчатый массив следующей размерности:</p> <p>0 строка 5 элементов, 1 строка 3 элемента, 2 строка 8 элементов, 3 строка 4 элемента, 4 строка 6 элементов.</p> <ol style="list-style-type: none"> 1. Заполнить его случайными числами в диапазоне от -500 до 500. 2. Вычислить сумму элементов в каждой строке (вывести ее на экран). 3. Вывести ступенчатый массив на экран
<p>Критерии оценки и шкала оценивания в баллах</p>		<p>Максимальная оценка – 2 б. Минимальная оценка – 1,2 б.</p>
<p>Наименование оценочного средства</p>		<p>Лабораторная работа 4. Работа со списками.</p>
<p>Представление и содерж</p>		<p>РАБОТА СО СПИСКАМИ</p>

<p>ание оценоч ных материа лов</p>	<ol style="list-style-type: none"> 1. Объявить и инициализировать первый список, содержащий 5 целочисленных элементов. Вывести на экран элементы списка. Добавить в конец списка один элемент. Вывести на экран элементы списка. 2. Объявить и инициализировать второй список, содержащий 3 целочисленных элемента. Вывести на экран элементы второго списка. 3. Вставить второй список в первый начиная с третьей позиции. Вывести на экран элементы первого списка. 4. Вывести на экран количество элементов в первом списке. 5. Вывести максимальный элемент первого списка. 6. Вывести минимальный элемент первого списка. 7. Скопировать элементы второго списка в массив. Вывести массив на экран. 8. Удалить второй элемент во втором списке. Ввести элементы второго списка на экран.
<p>Критер ии оценки и шкала оценив ания в баллах</p>	<p>Максимальная оценка – 1 б. Минимальная оценка – 0,6 б.</p>
<p>Наименование оценочного средства</p>	<p>Лабораторная работа 5. Работа со строками. Сохранение и чтение данных из текстового файла.</p>
<p>Представление и содержание оценочных материалов</p>	<p>Задание 1.</p> <p>Написать программу, меняющую в символьной строке порядок слов на зеркальный и выводящую строку в обратном порядке.</p> <p>Пример:</p> <ul style="list-style-type: none"> — «язык программирования С#» - «С# программирования язык» — «язык программирования С#»- «#С яинавориммаргорп кызья» <p>Задание 2.</p> <p>На вход программе подаётся английский текст, заканчивающийся точкой (символ «точка» во входных данных единственный). Текст зашифровать следующим образом: сначала определяется количество букв в самом длинном слове, обозначив полученное число К (словом называется непрерывная последовательность английских букв, слова друг от друга отделяются пробелами и запятыми, длина слова не превышает 20 символов). Затем</p>

проводится замена каждой английской буквы на букву, следующую за ней K-ю по счету в алфавите (алфавит считается циклическим, то есть после буквой Z стоит буква A), оставив другие символы неизменными. Строчные буквы при этом остались строчными, а прописные - прописными. Требуется написать программу, которая будет выводить на экран зашифрованный текст и число K.

Задание 3.

1.	Написать программу, которая считывает английский текст из файла и выводит на экран предложения, начинающиеся с гласных букв.
2.	Написать программу, которая считывает текст из файла и выводит на экран предложения, не содержащие запятых.
3.	Написать программу, которая считывает текст из файла и выводит его на экран, указав в конце каждого предложения количество слов в нем.
4.	Написать программу, которая считывает текст из файла и выводит на экран предложения, содержащие цифры.
5.	Написать программу, которая считывает английский текст из файла и выводит на экран слова текста, начинающиеся и оканчивающиеся на гласные буквы.
6.	Написать программу, которая считывает текст из файла и выводит его на экран, заменив цифры от 0 до 9 словами «ноль»,..., «девять».
7.	Написать программу, которая считывает текст из файла и выводит на экран сначала вопросительные, затем повествовательные предложения.
8.	Написать программу, которая считывает текст из файла, находит самое длинное слово и определяет, сколько раз оно встретилось в тексте.
9.	Написать программу, которая считывает текст из файл и выводит на экран сначала предложения, начинающиеся с однобуквенных слов, а затем все остальные.
10.	Написать программу, которая считывает текст из файл и выводит на экран предложения, содержащие максимальное количество знаков пунктуации.

Задание 4.

<p>Вариант 1.</p> <p>Написать программу, которая считывает из текстового файла фамилию, имя, отчество, возраст. Определить есть ли в списке однофамильцы и выводит в другой текстовый файл их имена и отчество. Рассчитать максимальный возраст, и вывести в текстовый файл.</p>	<p>Вариант 2.</p> <p>Написать программу, которая считывает из текстового файла фамилию врачей, специальность. Вывести в другой текстовый файл специальности и подсчитанное количество врачей данной специальности.</p>
---	---

	<p>Вариант 3.</p> <p>Написать программу, которая считывает из текстового файла фамилию, дисциплину и оценку ученика. Вывести в другой текстовый файл дисциплины и средние баллы.</p>	<p>Вариант 4.</p> <p>Написать программу, которая считывает из текстового файла фамилию спортсменов, имя, место. Вывести в другой текстовый файл отсортированные по фамилии в алфавитном порядке данные о спортсменах.</p>
	<p>Вариант 5.</p> <p>Написать программу, которая считывает из текстового файла названия торговых фирм, специфику товара (продуктовый, обувной и т.д.), адрес. Вывести в другой текстовый файл сгруппированные по специфике товара наименования и адреса магазинов.</p>	<p>Вариант 6.</p> <p>Написать программу, которая считывает из текстового файла фамилию автора, название произведения, жанр. Вывести в другой текстовый файл количество книг по определенным жанрам.</p>
	<p>Вариант 7.</p> <p>Написать программу, которая считывает из текстового файла названия фильмов, режиссера и год выпуска. Вывести в другой текстовый файл данные по фильмам отсортированные в порядке возрастания года выпуска.</p>	<p>Вариант 8.</p> <p>Написать программу, которая считывает из текстового файла название спортивных клубов, вид спорта, адрес. Вывести в другой текстовый файл сгруппированные по виду спорта названия спортивных клубов и их адреса.</p>
	<p>Вариант 9.</p> <p>Написать программу, которая считывает из текстового файла фамилию учителей, дисциплины, стаж работы. Вывести в другой текстовый файл сгруппированные по дисциплинам фамилии учителей. Определить преподавателя с наибольшим стажем и вывести в текстовый файл.</p>	<p>Вариант 10.</p> <p>Написать программу, которая считывает из текстового файла названия вузов, профиль (технический, технологический, гуманитарный и др.). Вывести в другой текстовый файл профиль и количество вузов данного профиля.</p>
<p>Критерии оценки и шкала оценивания в баллах</p>	<p>Максимальная оценка – 4 б.</p> <p>Минимальная оценка – 2,4 б.</p>	
<p>Наименование</p>	<p>Контрольная работа по структурному подходу к программированию</p>	

е оценоч ного средств а		
Предст авление и содерж ание оценоч ных материа лов	Варианты к контрольной работе	
	1.	<p>Задание 1. Двумерный массив размерностью 5×5. Заполнить случайными числами из интервала [1,20]. Исходный массив вывести на экран. Вычислить произведение элементов кратных 6. Переставить местами 1 и 5 строки. Полученный массив и произведение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая преобразует строку Nick со значением «It is warm and sunny now.» в строку со значением «It is Spring, the weather is warm and sunny.»</p> <p>Использовать вспомогательные строки Mike1 со значением «Spring,», Mike2 со значением «the weather» и стандартные строковые функции.</p>
	2.	<p>Задание 1. Двумерный массив размерностью 4×4. Заполнить случайными числами из интервала [-10,10]. Исходный массив вывести на экран. Вычислить количество отрицательных нечетных элементов. Переставить местами 4 и 2 столбцы. Полученный массив и количество вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры последовательность символов произвольной длины, а на экран выводит длину последовательности и преобразованную строку, в которой все буквы исходной строки будут заглавными.</p>
	3.	<p>Задание 1. Двумерный массив размерностью 5×4. Заполнить случайными числами из интервала [-10,10]. Вычислить сумму элементов по столбцам. Исходный массив и сумму элементов вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой предварительно удалены все цифры.</p>
	4.	<p>Задание 1. Двумерный массив размерностью 3×3. Заполнить случайными числами из интервала [-10,10]. Вычислить максимальный нечетный элемент и его расположение. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой предварительно удалены первые символы из каждого слова строки.</p>

5.	<p>Задание 1. Двумерный массив размерностью 3×4. Заполнить случайными числами из интервала $[-50, 50]$. Вычислить количество четных элементов по столбцам. Исходный массив и сумму элементов вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой предварительно удалено каждое второе слово.</p>
6.	<p>Задание 1. Двумерный массив размерностью 7×3. Заполнить случайными числами из интервала $[-10, 15]$. Исходный массив вывести на экран. Вычислить количество элементов кратных 2 или 3. Переставить местами 1 и 3 столбцы. Полученный массив и количество вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой предварительно удалены все четырехбуквенные слова.</p>
7.	<p>Задание 1. Двумерный массив размерностью 5×3. Заполнить случайными числами из интервала $[1, 15]$. Исходный массив вывести на экран. Вычислить среднее арифметическое элементов массива кратных 3. Переставить местами 2 и 5 строки. Полученный массив и среднее арифметическое вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры последовательность символов произвольной длины и выводит на экран преобразованную строку, в которой, каждый второй символ преобразован в заглавный.</p>
8.	<p>Задание 1. Двумерный массив размерностью 4×3. Заполнить случайными числами из интервала $[-10, 10]$. Исходный массив вывести на экран. Вычислить среднее арифметическое отрицательных четных чисел. Переставить местами 2 и 3 столбцы. Полученный массив и среднее арифметическое вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и находит в ней рифмы для заданного с клавиатуры слова (рифма-совпадение трех последних символов).</p>
9.	<p>Задание 1. Двумерный массив размерностью 5×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить среднее арифметическое четных элементов по строкам. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и находит сумму всех цифр присутствующих в тексте.</p>

10.	<p>Задание 1. Двумерный массив размерностью 3×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить минимальный четный элемент его расположение. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры 2 строки произвольной длины. Составить третью, включив в нее только те слова, которые есть и в первой, и во второй строке.</p>
11.	<p>Задание 1. Двумерный массив размерностью 3×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить среднее арифметическое отрицательных элементов массива. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу исправления ошибочного набора текста вида «после символов '.', '!', '?' должен стоять пробел».</p>
12.	<p>Задание 1.</p> <p>Двумерный массив размерностью 5×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить количество отрицательных элементов по строкам. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2.</p> <p>Составить программу, которая принимает с клавиатуры строку произвольной длины. Найти максимальную цифру среди цифр, образованных входящими в текст цифрами.</p>
13.	<p>Задание 1. Двумерный массив размерностью 4×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить среднее геометрическое элементов кратных 2. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2. Вводится строка вида «$4+54$», «$456+32$». Получить строку вида «$4+54=59$», «$456+32=488$» (числа могут быть произвольными). Преобразованную строку вывести на экран.</p>
14.	<p>Задание 1. Двумерный массив размерностью 6×4. Заполнить случайными числами из интервала $[-10, 10]$. Исходный массив вывести на экран. Вычислить сумму элементов кратных 2 и не кратных 4. Переставить местами 4 и 6 строки. Полученный массив и сумму вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины. Преобразовать строку таким образом, чтобы в начале каждого слова был введенный заранее символ. Преобразованную строку вывести на экран.</p>

	15.	<p>Задание 1. Двумерный массив размерностью 3×6. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить среднее арифметическое положительных элементов массива. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая преобразует строку Vika со значением «Soon the Summer will come!» в строку со значением «Soon the warm Summer will come to us!».</p> <p>Использовать вспомогательную строку Nika1 со значением «warm», Nika2 со значением «tous», и стандартные строковые функции.</p>
	16.	<p>Задание 1. Двумерный массив размерностью 6×6. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить минимальный элемент, расположенный на диагонали. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины. Удалить из строки все слова, начинающиеся с гласной буквы. Преобразованную строку вывести на экран.</p>
	17.	<p>Задание 1. Двумерный массив размерностью 5×6. Заполнить случайными числами из интервала $[-15, 15]$. Вычислить количество элементов кратных 3 и 4. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины. Зеркально отразить каждое слово строки, не меняя при этом порядок следования слов в предложении.</p>
	18.	<p>Задание 1. Двумерный массив размерностью 5×4. Заполнить случайными числами из интервала $[-15, 15]$. Исходный массив вывести на экран. Вычислить сумму элементов диагонали. Переставить местами 1 и 3 столбцы. Полученный массив и сумму вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины. Определить количество слов, которые содержат хотя бы одну букву 'd'.</p>
	19.	<p>Задание 1. Двумерный массив размерностью 5×6. Заполнить случайными числами из интервала $[-20, 10]$. Вычислить значение и позицию минимального элемента, расположенного во второй строке. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит её на экран предварительно удалив из неё все знаки препинания.</p>

	20.	<p>Задание 1. Двумерный массив размерностью 3×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить минимальный элемент 3 столбца и его координаты. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой предварительно удалены последние символы каждого слова строки.</p>
	21.	<p>Задание 1. Двумерный массив размерностью 3×3. Элементы массива принимают значение $a^2 + b^2$, где a, b - случайные числа из интервала $[1, 6]$. Вычислить сумму и количество нечетных элементов. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой второе слово повторено после третьего.</p>
	22.	<p>Задание 1. Двумерный массив размерностью 4×3. Заполнить случайными числами из интервала $[-20, 30]$. Исходный массив вывести на экран. Вычислить количество отрицательных элементов из интервала $[-10, -3]$. Переставить местами 1 и 3 столбцы. Полученный массив и количество вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и выводит на экран преобразованную строку, в которой максимальное по длине слово переставлено в начало.</p>
	23.	<p>Задание 1. Двумерный массив размерностью 7×4. Заполнить случайными числами из интервала $[-10, 10]$. Исходный массив вывести на экран. Вычислить сумму четных отрицательных элементов. Переставить местами 1 и 4 столбцы. Полученный массив и сумму вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры последовательность символов произвольной длины и выводит ее на экран, преобразуя каждый третий символ в заглавный.</p>
	24.	<p>Задание 1. Двумерный массив размерностью 5×4. Заполнить случайными числами из интервала $[-10, 10]$. Вычислить среднее арифметическое нечетных элементов по столбцам. Исходный массив и полученные значения вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и находит все повторяющиеся слова.</p>

25.	<p>Задание 1. Двумерный массив размерностью 3×3. Заполнить случайными числами из интервала $[10,30]$. Вычислить количество элементов оканчивающихся на 3. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и удваивает вхождение некоторой буквы в тексте. Буква задается пользователем.</p>
26.	<p>Задание 1. Двумерный массив размерностью 3×3. Заполнить случайными числами из интервала $[1,10]$. Исходный массив вывести на экран. Вычислить произведение элементов кратных 3. Переставить местами 2 и 3 строки. Полученный массив и произведение вывести в текстовый файл.</p> <p>Задание 2. Дана строка, содержащая минимум две буквы 'z'. Изменить ее следующим образом: символы строки, расположенные между первой и последней буквой 'z', переставить в обратном порядке.</p>
27.	<p>Задание 1. Двумерный массив размерностью 5×6. Заполнить случайными числами из интервала $[-20,10]$. Вычислить минимальный элемент, расположенный во второй строке. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Дана непустая последовательность непустых слов из латинских букв; соседние слова отделены друг от друга запятой, за последним словом – точка. Вывести текст, составленный из последних символов всех слов текста.</p>
28.	<p>Задание 1. Двумерный массив размерностью 3×4. Заполнить случайными числами из интервала $[10,30]$. Вычислить количество элементов, оканчивающихся на 2 или 5. Исходный массив и полученное значение вывести в текстовый файл.</p> <p>Задание 2. Дана непустая последовательность непустых слов из латинских букв; соседние слова отделены друг от друга запятой, за последним словом – точка. Определить количество слов, которые содержат ровно три буквы 'e'.</p>
29.	<p>Задание 1. Двумерный массив размерностью 3×4. Заполнить случайными числами из интервала $[-50,50]$. Вычислить сумму отрицательных элементов по строкам. Исходный массив и сумму элементов вывести в текстовый файл.</p> <p>Задание 2. Составить программу, которая принимает с клавиатуры строку произвольной длины и удваивает вхождение каждого слова в тексте.</p>

	30.	<p>Задание 1. Двумерный массив размерностью 6×3. Заполнить случайными числами из интервала [-30,20]. Исходный массив вывести на экран. Вычислить максимальный по модулю элемент. Переставить местами 4 и 5 строки. Полученный массив и максимальный элемент вывести в текстовый файл.</p> <p>Задание 2. Вводится строка вида «4*30», «5*60». Получить строку вида «4*30=120», «5*60=300».</p>
--	-----	---

Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 5 б.</p> <p>Минимальная оценка - 3 б.</p> <p>Контрольная не сдана- < 3 б.</p>
---	---

Наименование оценочного средства	<p>Лабораторная работа 6. Разработка пользовательского типа данных. Методы. Переопределение операций.</p>
---	--

Представление и содержание оценочных материалов	<p>Лабораторная работа 1. Классы. Конструкторы. Методы. Свойства. Операторы.</p> <ol style="list-style-type: none"> 1. Определить пользовательский класс в соответствии с вариантом задания. 2. Определить в классе следующие конструкторы: без параметров, с параметрами. 3. Определить в классе функции для просмотра, изменения полей данных, сохранения данных об экземплярах класса в текстовый файл. 4. В методе Main основной программы создать два экземпляра пользовательского класса и продемонстрировать работу конструкторов и методов. 5. Сделать поля класса закрытыми, регулировать доступ к изменению и чтению их значений с помощью свойств. Вложить дополнительную логику при осуществлении присваивания значений свойствам и полям класса. 6. Перегрузить операции сложения («+») и сравнения («<», «>»). 																				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Вариант</th> <th style="width: 15%;">Класс</th> <th style="width: 25%;">Названия полей</th> <th style="width: 45%;">Свойства</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">Книга</td> <td style="text-align: center;">Название Автор Количество страниц</td> <td style="text-align: center;">Название Автор Количество страниц (>0)</td> </tr> <tr> <td style="text-align: center;">2</td> <td style="text-align: center;">Товар</td> <td style="text-align: center;">Наименование Шифр Количество</td> <td style="text-align: center;">Наименование Шифр Количество(>0)</td> </tr> <tr> <td style="text-align: center;">3</td> <td style="text-align: center;">Бакалавр</td> <td style="text-align: center;">Фамилия Специальность Курс</td> <td style="text-align: center;">Фамилия Специальность Курс (от 1 до 4)</td> </tr> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">Автомобиль</td> <td style="text-align: center;">Марка</td> <td style="text-align: center;">Марка</td> </tr> </tbody> </table>	Вариант	Класс	Названия полей	Свойства	1	Книга	Название Автор Количество страниц	Название Автор Количество страниц (>0)	2	Товар	Наименование Шифр Количество	Наименование Шифр Количество(>0)	3	Бакалавр	Фамилия Специальность Курс	Фамилия Специальность Курс (от 1 до 4)	4	Автомобиль	Марка	Марка
Вариант	Класс	Названия полей	Свойства																		
1	Книга	Название Автор Количество страниц	Название Автор Количество страниц (>0)																		
2	Товар	Наименование Шифр Количество	Наименование Шифр Количество(>0)																		
3	Бакалавр	Фамилия Специальность Курс	Фамилия Специальность Курс (от 1 до 4)																		
4	Автомобиль	Марка	Марка																		

			Мощность Стоимость	Мощность (>0) Стоимость(>0)								
	5	Животное	Наименование Класс Средний вес	Наименование Класс Средний вес(>0)								
	6	Корабль	Наименование Водоизмещение Тип	Наименование Водоизмещение (>0) Тип								
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 4 б.</p> <p>Минимальная оценка – 2,4 б.</p>											
Наименование оценочного средства	Лабораторная работа 7. Создание библиотеки классов со статическими методами.											
Представление и содержание оценочных материалов	<p>. Работа со статическими методами. Перегрузка методов.</p> <p>Описать библиотеку содержащую статический класс, элементами которого являются статические методы для работы с одномерными и двумерными массивами:</p> <p>А. Заполнение массива случайными целыми числами в указанном диапазоне. Заполнение массива случайными вещественными числами в диапазоне [0,1].</p> <p>В. Вычисления суммы элементов массива (возвращает число).</p> <p>С. Вычисление произведения элементов массива (возвращает число).</p> <p>Д. Возврат максимального элемента массива и его координат.</p> <p>Е. Вывод массива на экран (возвращает строку соответствующего формата).</p> <table border="1" data-bbox="231 1579 1500 1859"> <thead> <tr> <th>Объявить</th> <th>Выполнить</th> </tr> </thead> <tbody> <tr> <td>1 массив - целочисленный</td> <td>А-Е</td> </tr> <tr> <td>1 массив - вещественный</td> <td>А-Е</td> </tr> <tr> <td>1 строковый массив</td> <td>Д (максимальное по длине строку), Е</td> </tr> </tbody> </table>				Объявить	Выполнить	1 массив - целочисленный	А-Е	1 массив - вещественный	А-Е	1 строковый массив	Д (максимальное по длине строку), Е
Объявить	Выполнить											
1 массив - целочисленный	А-Е											
1 массив - вещественный	А-Е											
1 строковый массив	Д (максимальное по длине строку), Е											
Критерии оценки	<p>Максимальная оценка – 3 б.</p> <p>Минимальная оценка – 1,8 б.</p>											

и шкала оценив ания в баллах											
Наименование оценочного средства	Лабораторная работа 8. Индексаторы										
Представление и содержание оценочных материалов	<p>Описать класс множество (Set), содержащий следующие элементы:</p> <table border="1" data-bbox="231 734 1444 1957"> <tr> <td data-bbox="231 734 491 846">Поля</td> <td data-bbox="491 734 1444 846"> <ol style="list-style-type: none"> 1. Elements (целочисленный одномерный массив для хранения элементов множества). 2. Count (количество элементов в множестве). </td> </tr> <tr> <td data-bbox="231 846 491 1070">Конструктор</td> <td data-bbox="491 846 1444 1070"> <ol style="list-style-type: none"> 1. Без параметров. Запрашивает ввести с клавиатуры количество элементов множества и вызывает метод заполнения значений элементов множества (Fill) . 2. С одним параметром – одномерный целочисленный массив. Элементы множества принимают значения элементов целочисленного массива. </td> </tr> <tr> <td data-bbox="231 1070 491 1397">Методы</td> <td data-bbox="491 1070 1444 1397"> <ol style="list-style-type: none"> 1. Fill () – присвоение элементам множества вводимых с клавиатуры значений. 2. IndexOf (int Value) – возвращает индекс искомого в множестве значения или -1, если значение не было найдено. 3. ShowSet () – вывод элементов множества на экран. 4. Add (int NewElement) – добавление в множество нового элемента (перед добавлением элемента следует увеличить размерность массива элементов на единицу, т.е. <code>Array.Resize(ref Elements, Elements.Length + 1)</code>) </td> </tr> <tr> <td data-bbox="231 1397 491 1805">Перегрузка операций</td> <td data-bbox="491 1397 1444 1805"> <ol style="list-style-type: none"> 1. ++ – увеличение значений элементов множества на 1 <code>public static Set operator ++(Set set1)</code> 2. + – объединение множеств <code>public static Set operator +(Set set1, Set set2)</code> 3. * – пересечение множеств <code>public static Set operator *(Set set1, Set set2)</code> 4. / – разность множеств <code>public static Set operator /(Set set1, Set set2)</code> 5. < и > – сравнение количества элементов множеств <code>public static bool operator <(Set set1, Set set2)</code> <code>public static bool operator >(Set set1, Set set2)</code> </td> </tr> <tr> <td data-bbox="231 1805 491 1957">Индексатор</td> <td data-bbox="491 1805 1444 1957">Для обращения к отдельным значениям поля Elements, являющегося одномерным массивом хранящим элементы множества.</td> </tr> </table>	Поля	<ol style="list-style-type: none"> 1. Elements (целочисленный одномерный массив для хранения элементов множества). 2. Count (количество элементов в множестве). 	Конструктор	<ol style="list-style-type: none"> 1. Без параметров. Запрашивает ввести с клавиатуры количество элементов множества и вызывает метод заполнения значений элементов множества (Fill) . 2. С одним параметром – одномерный целочисленный массив. Элементы множества принимают значения элементов целочисленного массива. 	Методы	<ol style="list-style-type: none"> 1. Fill () – присвоение элементам множества вводимых с клавиатуры значений. 2. IndexOf (int Value) – возвращает индекс искомого в множестве значения или -1, если значение не было найдено. 3. ShowSet () – вывод элементов множества на экран. 4. Add (int NewElement) – добавление в множество нового элемента (перед добавлением элемента следует увеличить размерность массива элементов на единицу, т.е. <code>Array.Resize(ref Elements, Elements.Length + 1)</code>) 	Перегрузка операций	<ol style="list-style-type: none"> 1. ++ – увеличение значений элементов множества на 1 <code>public static Set operator ++(Set set1)</code> 2. + – объединение множеств <code>public static Set operator +(Set set1, Set set2)</code> 3. * – пересечение множеств <code>public static Set operator *(Set set1, Set set2)</code> 4. / – разность множеств <code>public static Set operator /(Set set1, Set set2)</code> 5. < и > – сравнение количества элементов множеств <code>public static bool operator <(Set set1, Set set2)</code> <code>public static bool operator >(Set set1, Set set2)</code> 	Индексатор	Для обращения к отдельным значениям поля Elements, являющегося одномерным массивом хранящим элементы множества.
Поля	<ol style="list-style-type: none"> 1. Elements (целочисленный одномерный массив для хранения элементов множества). 2. Count (количество элементов в множестве). 										
Конструктор	<ol style="list-style-type: none"> 1. Без параметров. Запрашивает ввести с клавиатуры количество элементов множества и вызывает метод заполнения значений элементов множества (Fill) . 2. С одним параметром – одномерный целочисленный массив. Элементы множества принимают значения элементов целочисленного массива. 										
Методы	<ol style="list-style-type: none"> 1. Fill () – присвоение элементам множества вводимых с клавиатуры значений. 2. IndexOf (int Value) – возвращает индекс искомого в множестве значения или -1, если значение не было найдено. 3. ShowSet () – вывод элементов множества на экран. 4. Add (int NewElement) – добавление в множество нового элемента (перед добавлением элемента следует увеличить размерность массива элементов на единицу, т.е. <code>Array.Resize(ref Elements, Elements.Length + 1)</code>) 										
Перегрузка операций	<ol style="list-style-type: none"> 1. ++ – увеличение значений элементов множества на 1 <code>public static Set operator ++(Set set1)</code> 2. + – объединение множеств <code>public static Set operator +(Set set1, Set set2)</code> 3. * – пересечение множеств <code>public static Set operator *(Set set1, Set set2)</code> 4. / – разность множеств <code>public static Set operator /(Set set1, Set set2)</code> 5. < и > – сравнение количества элементов множеств <code>public static bool operator <(Set set1, Set set2)</code> <code>public static bool operator >(Set set1, Set set2)</code> 										
Индексатор	Для обращения к отдельным значениям поля Elements, являющегося одномерным массивом хранящим элементы множества.										

Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 4 б.</p> <p>Минимальная оценка – 2,4 б.</p>																					
Наименование оценочного средства	<p>Лабораторная работа 9. Создание иерархии пользовательских типов данных. Виртуальные методы.</p>																					
Представление и содержание оценочных материалов	<p>Лабораторная работа 3. Иерархия классов</p> <p>Примечание: выполнить лабораторную работу на основе имеющейся лаб.раб. № 7.</p> <ol style="list-style-type: none"> 7. Создать базовый класс согласно варианту задания. 8. Добавить в базовый класс два конструктора: по умолчанию и с одним параметром. 9. Добавить в базовый класс по одному общему в производных классах полю, свойству и методу, которые будут наследоваться. 10. Определить иерархию классов. Для этого связать производные классы (классы наследники) с базовым классом отношением наследования (по вариантам). 11. Добавить в производные классы по одному дополнительному полю и свойству (не считая наследуемых от базового класса полей и свойств). 12. Добавить в производные класса конструкторы (по умолчанию и с параметрами). 13. В производных классах переопределить наследуемый метод. 14. Создать массив из четырех элементов (тип массива соответствует базовому классу), в который будут помещены все создаваемые экземпляры классов. 15. Продемонстрировать работу методов и конструкторов. <table border="1" data-bbox="229 1473 1485 1816"> <thead> <tr> <th>Вариант</th> <th>Базовый класс</th> <th>Производные классы</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Печатное издание</td> <td>журнал, книга, газета</td> </tr> <tr> <td>2</td> <td>Товар</td> <td>продуктовые товары, бытовая химия, напитки</td> </tr> <tr> <td>3</td> <td>Учащийся</td> <td>Бакалавр, Магистр, Аспирант</td> </tr> <tr> <td>4</td> <td>транспортное средство</td> <td>автомобиль, поезд, пароход</td> </tr> <tr> <td>5</td> <td>Животное</td> <td>млекопитающие, парнокопытные, птицы</td> </tr> <tr> <td>6</td> <td>Судно</td> <td>корабль, пароход, катер</td> </tr> </tbody> </table>	Вариант	Базовый класс	Производные классы	1	Печатное издание	журнал, книга, газета	2	Товар	продуктовые товары, бытовая химия, напитки	3	Учащийся	Бакалавр, Магистр, Аспирант	4	транспортное средство	автомобиль, поезд, пароход	5	Животное	млекопитающие, парнокопытные, птицы	6	Судно	корабль, пароход, катер
Вариант	Базовый класс	Производные классы																				
1	Печатное издание	журнал, книга, газета																				
2	Товар	продуктовые товары, бытовая химия, напитки																				
3	Учащийся	Бакалавр, Магистр, Аспирант																				
4	транспортное средство	автомобиль, поезд, пароход																				
5	Животное	млекопитающие, парнокопытные, птицы																				
6	Судно	корабль, пароход, катер																				
Критерии оценки и шкала оценивания	<p>Максимальная оценка – 3 б.</p> <p>Минимальная оценка – 1,8 б.</p>																					

В баллах																								
Наименование оценочного средства	Лабораторная работа 10. Создание приложения согласно паттерну "делегирование-включение".																							
Представление и содержание оценочных материалов	<p>Абстрактные классы. «Включение-делегирование».</p> <p>Определить базовый класс как абстрактный.</p> <p>Давить в абстрактный класс дополнительный абстрактный метод для изменения значений свойств и/или полей класса.</p> <p>В производных классах переопределить наследуемый метод изменения свойств и полей класса.</p> <p>Создать класс, содержащий коллекции производных от базового абстрактного класса объектов.</p> <p>Описать в классе, построенном по принципу «включение-делегирование», следующие методы:</p> <ul style="list-style-type: none"> • добавления в коллекцию новых объектов, • изменения свойств и полей имеющихся объектов по номеру объекта в коллекции; • удаление из коллекции объектов по его номеру. • вывода информации, об имеющихся в коллекции объектах. <p>Продемонстрировать работу класса, построенного по принципу «включение-делегирование». Для этого в методе Main вывести меню для выбора одного из возможных действий:</p> <ul style="list-style-type: none"> - добавить в коллекцию объект (производного класса). - удалить объект из коллекции. - вывести сведения по имеющимся в коллекции объектам. - внести изменения в характеристики объекта коллекции. - выход из программы. 																							
	<table border="1" data-bbox="229 1823 1506 2159"> <thead> <tr> <th data-bbox="229 1823 405 1935">Вариант</th> <th data-bbox="405 1823 718 1935">Класс, содержащий коллекцию объектов</th> <th data-bbox="718 1823 1031 1935">Базовый абстрактный класс</th> <th data-bbox="1031 1823 1506 1935">Производные классы</th> </tr> </thead> <tbody> <tr> <td data-bbox="229 1935 405 1975">1</td> <td data-bbox="405 1935 718 1975">Книжный магазин</td> <td data-bbox="718 1935 1031 1975">печатное издание</td> <td data-bbox="1031 1935 1506 1975">журнал, книга, газета</td> </tr> <tr> <td data-bbox="229 1975 405 2047">2</td> <td data-bbox="405 1975 718 2047">Магазин</td> <td data-bbox="718 1975 1031 2047">Товар</td> <td data-bbox="1031 1975 1506 2047">Продуктовые товары, бытовая химия, напитки</td> </tr> <tr> <td data-bbox="229 2047 405 2087">3</td> <td data-bbox="405 2047 718 2087">Университет</td> <td data-bbox="718 2047 1031 2087">Учащийся</td> <td data-bbox="1031 2047 1506 2087">Бакалавр, Магистр, Аспирант</td> </tr> <tr> <td data-bbox="229 2087 405 2159">4</td> <td data-bbox="405 2087 718 2159">Транспортная компания</td> <td data-bbox="718 2087 1031 2159">транспортное средство</td> <td data-bbox="1031 2087 1506 2159">автомобиль, поезд, пароход</td> </tr> </tbody> </table>	Вариант	Класс, содержащий коллекцию объектов	Базовый абстрактный класс	Производные классы	1	Книжный магазин	печатное издание	журнал, книга, газета	2	Магазин	Товар	Продуктовые товары, бытовая химия, напитки	3	Университет	Учащийся	Бакалавр, Магистр, Аспирант	4	Транспортная компания	транспортное средство	автомобиль, поезд, пароход			
Вариант	Класс, содержащий коллекцию объектов	Базовый абстрактный класс	Производные классы																					
1	Книжный магазин	печатное издание	журнал, книга, газета																					
2	Магазин	Товар	Продуктовые товары, бытовая химия, напитки																					
3	Университет	Учащийся	Бакалавр, Магистр, Аспирант																					
4	Транспортная компания	транспортное средство	автомобиль, поезд, пароход																					

	5	Зоопарк	Животное	млекопитающие, парнокопытные, птицы
	6	Порт	Судно	корабль, пароход, катер
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 5 б.</p> <p>Минимальная оценка – 3 б.</p>			
Наименование оценочного средства	<p>Лабораторная работа 11. Организация клонирования (копирования) объектов, сравнения и перебора на основе применения стандартных интерфейсов <code>Comparable</code>, <code>Comparer</code>, <code>Enumerable</code>, <code>IEnumerator</code>, <code>IClonable</code>.</p>			
Представление и содержание оценочных материалов	<p>Реализовать интерфейсы <code>IClonable</code>, <code>Comparable</code>, <code>Comparer</code>, <code>Enumerable</code>. Продемонстрировать реализацию поверхностного и глубокого копирования объектов, вывода отсортированного списка объектов класса (с применением методов интерфейсов <code>Comparable</code> и <code>Comparer</code>), применения <code>foreach</code>.</p>			
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 2 б.</p> <p>Минимальная оценка – 1,2 б.</p>			
Наименование оценочного	<p>Лабораторная работа 12. Делегаты. Применение метода функциональной параметризации для реализации метода половинного деления отрезка с целью решения трансцендентных уравнений.</p>			

средств а	
Предст авление и содерж ание оценоч ных материа лов	<p>1. На основе принципа функциональной параметризации создать универсальный метод решения уравнений $f(x) = 0$, в основе которого заложен численный метод деления отрезка пополам. С помощью делегата задается функция левой части уравнения (применить лямбда-выражение).</p> <p>2. Создать с помощью делегатов метод сортировки списка объектов класса "Товар" (алгоритм сортировки может быть выбран произвольно, например, метод «пузырька»). С помощью делегатов обеспечить возможность проведения сортировки товаров по наименованию (тип string), по цене (тип double), по количеству на складе (тип double), по сроку годности (тип DateTime). Делегат должен передавать информацию о функции сравнения двух объектов класса "Товар".</p>
Критер ии оценки и шкала оценив ания в баллах	<p>Максимальная оценка – 4 б.</p> <p>Минимальная оценка – 2,4 б.</p>
Наиме новани е оценоч ного средств а	Лабораторная работа 13. Событийное программирование. События. Задача на регистрацию нарушений скоростного режима. Моделирование игры в ping-pong.
Предст авление и содерж ание оценоч ных материа лов	<p>1. Добавить 3 класса автомобиль, камера и ГИБДД. У класса автомобиль объявлены свойства марка, гос. знак, ФИО владельца. Камера через определенный промежуток времени фиксирует скорость автомобиля. Если скорость автомобиля превышает разрешенную скорость, тогда срабатывает событие отправки сообщения в ГИБДД. ГИБДД выписывает штраф за превышение скорости с сохранением его в текстовый файл.</p> <p>2. Напишите игру "пинг-понг", в котором два класса (Ping и Pong) уведомляют друг друга о том, был ли отбит мяч (реализовать с использованием датчика случайных чисел). Пара объектов классов Ping и Pong «играют» между собой и выводить на консоль соответствующие сообщения</p> <ol style="list-style-type: none"> 1) Ping received Pong. 2) Pong received Ping. 3) Ping received Pong. 4) Pong received Ping. 5) Ping received Pong.

	По окончании игры выводится счет и имя игрока одержавшего победу.		
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 4 б.</p> <p>Минимальная оценка – 2,4 б.</p>		
Наименование оценочного средства	Контрольная работа – «Объектно-ориентированное программирование»		
Представление и содержание оценочных материалов	<p>Вариант 1.</p> <p>Класс: Погода</p> <p>Поля: Город, Дата</p> <p>Свойства:</p> <p>Температура (-78 до 50 °С), Влажность (от 0 до 100%)</p> <p>Методы:</p> <p>Пересчет температуры из Цельсия в Фаренгейты.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие:</p> <p>При понижении температуры ниже нуля, выдает сообщение о заморозках.</p> <p>Сортировка: По температуре</p>	<p>Вариант 2.</p> <p>Класс: Ежедневник</p> <p>Поля: Наименование, Дата</p> <p>Свойства:</p> <p>Время начала (00:00 до 24:00)</p> <p>Продолжительность («+» значение)</p> <p>Методы:</p> <p>Вывод информации о статусе события (предстоит или оно уже прошло).</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Напоминание о количестве оставшихся дней до события</p> <p>Сортировка: По дате</p>	<p>Вариант 3.</p> <p>Класс: Турист</p> <p>Поля: Имя, Направление</p> <p>Свойства:</p> <p>Продолжительность («+» значение)</p> <p>Цена за день пребывания («+» значение)</p> <p>Методы:</p> <p>Расчет стоимости тура (запрашивает количество дней, если продолжительность не указана)</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Продажа тура</p> <p>Сортировка: По направлению</p>

	<p>Вариант 4.</p> <p>Класс: Трапеция</p> <p>Поля: Периметр, Площадь</p> <p>Свойства:</p> <p>Длины оснований («+» значение)</p> <p>Высота («+» значение)</p> <p>Методы: Расчет площади трапеции.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Вывод информации при изменении свойств трапеции</p> <p>Сортировка: По периметру</p>	<p>Вариант 5.</p> <p>Класс: Преподаватель</p> <p>Поля: Фамилия, Кафедра</p> <p>Свойства:</p> <p>Табельный номер («+» значение)</p> <p>Год поступления на работу (>=1965)</p> <p>Методы: Расчет стажа работы</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Выплата премии в случае 10-летнего стажа работы</p> <p>Сортировка: По кафедре</p>	<p>Вариант 6.</p> <p>Класс: Водитель</p> <p>Поля: Фамилия, Номер водительского удостоверения</p> <p>Свойства:</p> <p>Автомобиль (содержит 6 символов)</p> <p>Показания одометра («+» значение)</p> <p>Методы:</p> <p>По вводимым показаниям одометра, рассчитывает количество пройденных км и записывает новые значения одометра.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: В случае неправильно введенных показаний одометра выдает соответствующее сообщение</p> <p>Сортировка: По фамилии</p>

	<p>Вариант 7.</p> <p>Класс: Грузоперевозчик</p> <p>Поля: Автомобиль, Пункт назначения</p> <p>Свойства:</p> <p>Расстояние («+» значение)</p> <p>Цена за км («+» значение)</p> <p>Методы: Расчет стоимости перевозки.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Предоставление скидки в случае расстояния более 800 км и соответствующий пересчет суммы</p> <p>Сортировка: По расстоянию</p>	<p>Вариант 8.</p> <p>Класс: Билетная касса (билет)</p> <p>Поля: Наименование спектакля, Дата</p> <p>Свойства:</p> <p>Количество билетов («+» значение).</p> <p>Количество купленных билетов («+» значение)</p> <p>Методы:</p> <p>Рассчитать количество оставшихся билетов.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Покупка билета</p> <p>Сортировка: По количеству оставшихся билетов</p>	<p>Вариант 9.</p> <p>Класс: Картина</p> <p>Поля: Название, Художник</p> <p>Свойства:</p> <p>Год выпуска («+» значение)</p> <p>Стоимость («+» значение)</p> <p>Методы: Аукцион.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Продажа картины на торгах, в случае, если клиент предложит сумму большую стоимости картины</p> <p>Сортировка: По художнику</p>
	<p>Вариант 10.</p> <p>Класс: Банковская карта</p> <p>Поля: Номер счета. Фамилия</p> <p>Свойства:</p> <p>Количество денежных средств («+» значение)</p> <p>Процентная ставка (не более 15%)</p> <p>Методы:</p> <p>Расчет суммы денежных средств незастрахованных государством.</p> <p>Вывод информации</p>	<p>Вариант 11.</p> <p>Класс: Фильм</p> <p>Поля: Наименование, Год выпуска</p> <p>Свойства:</p> <p>Методы:</p> <p>Бюджет («+» значение)</p> <p>Кассовые сборы («+» значение)</p> <p>Переопределить операцию: >, < Методы:</p> <p>Увеличение кассовых сборов</p> <p>Вывод информации</p>	<p>Вариант 12.</p> <p>Класс: Квитанция</p> <p>Поля: Платежный код, Пенни</p> <p>Свойства:</p> <p>Крайний срок оплаты (номер месяца)</p> <p>Сумма платежа («+» значение)</p> <p>Методы: Оплата услуги</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Начисление пенни в случае если текущий месяц, в котором</p>

<p>Переопределить операцию: +</p> <p>Событие: Начисление процентов первого числа каждого месяца (проверяет текущую дату)</p> <p>Сортировка: По процентной ставке</p>	<p>Событие: Вывод информации об успехе или кассовом провале фильма, если кассовые сборы не превышают бюджет.</p> <p>Сортировка: По кассовым сборам</p>	<p>производится оплата, не соответствует месяцу крайнего срока(платеж просрочен).</p> <p>Сортировка: По сумме платежа</p>
<p>Вариант 13.</p> <p>Класс: Продукт</p> <p>Поля: Название, Калорийность</p> <p>Свойства:</p> <p>Белки, Жиры Углеводы («+» значение)</p> <p>Методы:</p> <p>Расчет калорийности в зависимости от веса продукции.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Вывод оповещения о высококалорийных продуктах (450 ккал и более)</p> <p>Сортировка: По содержанию углеводов</p>	<p>Вариант 14.</p> <p>Класс: Штрафы</p> <p>Поля: Наименование, Статус (оплачен или нет)</p> <p>Свойства:</p> <p>Сумма штрафа («+» значение)</p> <p>Скидка («+» значение)</p> <p>Методы:</p> <p>Расчет суммы штрафа с учетом скидки.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Изменение статуса платежа в результате оплаты</p> <p>Сортировка: По величине скидки</p>	<p>Вариант 15.</p> <p>Класс: Треугольник</p> <p>Поля: Периметр, Площадь</p> <p>Свойства:</p> <p>Длины сторон («+» значение)</p> <p>Методы:</p> <p>Определение вида треугольника (равнобедренный, равносторонний).</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Вывод информации при изменении свойств треугольника</p> <p>Сортировка: По периметру</p>
<p>Вариант 16.</p> <p>Класс: Процессор</p> <p>Поля: Модель, Количество ядер</p> <p>Свойства:</p> <p>Частота («+» значение)</p>	<p>Вариант 17.</p> <p>Класс: Стройматериал</p> <p>Поля: Наименование, Номер склада</p> <p>Свойства: Количество на складе</p>	<p>Вариант 18.</p> <p>Класс: Дипломы</p> <p>Поля: Тема, Количество страниц,</p> <p>Свойства:</p> <p>Статус (принят / переделать), %</p>

<p>Средняя загрузка (0 до 100 %)</p> <p>Методы: Расчет количества проводимых операций за введенное время с учетом средней нагрузки.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Вывод информации при изменении тактовой частоты процессора.</p> <p>Сортировка: По количеству ядер</p>	<p>(«+» значение) Цена («+» значение)</p> <p>Методы: Расчет стоимости с учетом количества и цены материала.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Покупка материала (в случае отсутствия на складе выдает соответствующее сообщение, иначе уменьшает количество товара на складе)</p> <p>Сортировка: По номеру склада</p>	<p>оригинальности (от 0 до 100 %)</p> <p>Методы:</p> <p>Расчет объема материала, который необходимо переделать в соответствии с % оригинальности.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Присвоение статуса «принят», если оригинальность теста выше 70 %</p> <p>Сортировка: По % оригинальности</p>
<p>Вариант 19.</p> <p>Класс: Страховой полис</p> <p>Поля: Автомобиль, Страхователь</p> <p>Свойства:</p> <p>Стоимость страховки</p> <p>(«+» значение), Базовый тариф</p> <p>(«+» значение)</p> <p>Методы: Расчет стоимости страховки (Стоимость полиса = Базовый тариф</p> <p>* Территориальный коэф.*Коэф. возрастного стажа*Коэф. аварийности).</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Оформление страхового полиса</p>	<p>Вариант 20.</p> <p>Класс: Гостиничный номер</p> <p>Поля: Номер, Дата брони</p> <p>Свойства: Цена («+» значение)</p> <p>Количество дней</p> <p>(«+» значение)</p> <p>Методы: Расчет стоимости гостиничного номера.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Бронирование и отмена брони (установление даты)</p> <p>Сортировка: По дате</p>	<p>Вариант 21.</p> <p>Класс: Производство</p> <p>Поля: Название, Директор</p> <p>Свойства: Производимая продукция (строка с названиями производимых товаров отделенных символом /)</p> <p>Сумма основных средств («+» значение)</p> <p>Методы:</p> <p>Поиск производимой продукции по наименованию.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Обновление ассортимента производимых товаров</p>

	<p>Сортировка: По стоимости страхового полиса</p>		<p>Сортировка: По сумме основных средств</p>
	<p>Вариант 22.</p> <p>Класс: Организация</p> <p>Поля: Наименование, Директор</p> <p>Свойства:</p> <p>Уставной капитал (более 10 000 руб.)</p> <p>Средства на счету («+» значение)</p> <p>Методы: Снятие денежных средств.</p> <p>Вывод информации</p> <p>Переопределить операцию: Вывод информации о статусе операции снятия денег</p> <p>Событие: >, <</p> <p>Сортировка: По значению уставного капитала</p>	<p>Вариант 23.</p> <p>Класс: Озеро</p> <p>Поля: Название, Уровень воды</p> <p>Свойства:</p> <p>Площадь («+» значение)</p> <p>Максимальная глубина («+» значение)</p> <p>Методы: Изменение величины уровня воды.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: При изменении уровня воды в озере подать сообщении об его увеличении или уменьшении.</p> <p>Сортировка: По площади</p>	<p>Вариант 24.</p> <p>Класс: Параллелограмм</p> <p>Поля: Периметр</p> <p>Свойства:</p> <p>Длины противоположных сторон («+» значение)</p> <p>Высота («+» значение)</p> <p>Методы:</p> <p>Расчет периметра параллелограмма.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Вывод информации при изменении свойств параллелограмма</p> <p>Сортировка: По площади</p>

<p>Вариант 25.</p> <p>Класс: Участники олимпиады</p> <p>Поля: Фамилия, Дисциплина</p> <p>Свойства: Количество баллов набранных на первом этапе (от 0 до 40)</p> <p>Количество баллов набранных на втором этапе (от 0 до 60)</p> <p>Методы: Расчет среднего балла за два этапа.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Вывод информации об абсолютных лидерах</p> <p>Сортировка: По среднему баллу</p>	<p>Вариант 26.</p> <p>Класс: Такси</p> <p>Поля: Номер авто, Пункт подачи</p> <p>Свойства:</p> <p>Пункт назначения (не превышает 10 символов)</p> <p>Километраж («+» значение)</p> <p>Методы:</p> <p>Расчет стоимости перевозки.</p> <p>Вывод информации</p> <p>Переопределить операцию: +</p> <p>Событие: Пересчет суммы платежа в случае предъявления дисконтной карты</p> <p>Сортировка: По номеру авто</p>	<p>Вариант 27.</p> <p>Класс: Погода</p> <p>Поля: Город, Дата</p> <p>Свойства:</p> <p>Температура (-78 до 50 °С), Влажность (от 0 до 100%)</p> <p>Методы:</p> <p>Пересчет температуры из Цельсия в Фаренгейты.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие:</p> <p>При понижении температуры ниже нуля, выдает сообщение о заморозках.</p> <p>Сортировка: По температуре</p>
<p>Вариант 28.</p> <p>Класс: Кавдрат</p> <p>Поля: Периметр, Площадь</p> <p>Свойства:</p> <p>Длины оснований («+» значение)</p> <p>Высота («+» значение)</p> <p>Методы: Расчет площади.</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: Вывод информации при изменении свойств трапеции</p>	<p>Вариант 29.</p> <p>Класс: Преподаватель</p> <p>Поля: Фамилия, Кафедра</p> <p>Свойства:</p> <p>Табельный номер («+» значение)</p> <p>Год поступления на работу (>=1965)</p> <p>Методы: Расчет стажа работы</p> <p>Вывод информации</p> <p>Переопределить операцию: >, <</p>	<p>Вариант 30.</p> <p>Класс: Водитель</p> <p>Поля: Фамилия, Номер водительского удостоверения</p> <p>Свойства:</p> <p>Автомобиль (содержит 6 символов)</p> <p>Показания одометра («+» значение)</p> <p>Методы:</p> <p>По вводимым показаниям одометра, рассчитывает количество пройденных км и записывает новые значения одометра.</p>

	<p>Сортировка: По периметру</p>	<p>Событие: Выплата премии в случае 10-летнего стажа работы</p> <p>Сортировка: По кафедре</p>	<p>Вывод информации</p> <p>Переопределить операцию: >, <</p> <p>Событие: В случае неправильно введенных показаний одометра выдает соответствующее сообщение</p> <p>Сортировка: По фамилии</p>
<p>Критерии оценки и шкала оценивания в баллах</p>	<p>Максимальная оценка – 5 б.</p> <p>Минимальная оценка - 3 б.</p> <p>Контрольная не сдана- < 3 б.</p>		
<p>Наименование оценочного средства</p>	<p>Лабораторная работа 14. Применение операторов класса Linq для организации обработки данных пользовательских классов.</p>		
<p>Представление и содержание оценочных материалов</p>	<p>Создать класс.</p> <p>Из текстового файла заполнить список, элементами которого являются экземпляры пользовательского класса (минимум 15 элементов в списке).</p> <p>На основе применения операторов класса LINQ выполнить следующие действия:</p> <ul style="list-style-type: none"> • Фильтрация согласно варианту задания. • Проекцию согласно варианту задания. • Сортировку согласно варианту задания. • Группировку согласно варианту задания. • Группировку с подсчетом количества элементов в каждой группе согласно варианту задания. • Агрегатную операцию согласно варианту задания. 		

- Извлечение согласно варианту задания.
- Проверку на соответствие элементов списка условию согласно варианту задания.
- Объединения двух последовательностей (придумать вторую самостоятельно).

Класс и свойства:	Фильтрация	Проекция Новый тип со свойствами:	Сортировка по	Группировка по	Агрегатные функции	Skip, Take, SkipWhile, TakeWhile	All и Any
Личность: фамилия, имя, возраст.	Совершеннолетних с фамилией начиная с гласной буквы	фамилия, имя, совершеннолетний/несовершеннолетний	фамилия и возраст	фамилия	Число совершеннолетних	Извлечь из списка половину элементов	Имеются ли несовершеннолетние
Врач: фамилия, специализация, категория	Врачи офтальмологов и неврологов, 1 категории	Фамилия врача, опытный/новичок (стаж меньше 2 лет)	Фамилия и категория	специализация	Количество врачей хирургов	Извлечь из списка вторую половину элементов	Имеется хотя бы один хирург
Ученик: фамилия, дисциплина, оценка (цифра)	Ученики, которые получили по математике положительные оценки	Фамилия, дисциплина, оценка (прописью)	Фамилия и оценка	дисциплина	Средний балл по информатике	Извлечь центральные 3 элемента из списка	Все ли ученики получили положительные оценки
Спортсмен: фамилия, вид спорта, место	Спортсмены занявшие призовые места по плаванию и фехтованию	Фамилии спортсменов в призеров (1-3 место). Медаль (золото/серебро/бронза)	Фамилия и месту	Вид спорта	Количество спортсменов с фамилией начинающихся с	Извлечь из списка все элементы с момента появления	Есть ли среди спортсменов те, кто занял 1 место

						гласной буквы	спорте мена занявш его 1 места.	
Торговая фирма: наименование, специфика товара (продуктовый, обувной и т.д.), адрес.	Торговые фирмы занимающиеся продажей обувной продукции и или одежды.	Только обувные магазины: Наименование Адрес	Адресу и названию торговой фирмы	Специфике товара	Число торговых фирм продающих их продукты питания.	Извлечь из списка элементы, пока они соответствуют магазину одежды или обуви)	Имеются ли торговые фирмы продающие ювелирные изделия.	
Писатель: Фамилия, название произведения, жанр	Повести и рассказы с указанием писателя	Названия произведений в порядке от Я до А	Фамилии и названию произведения	жанру	Самое длинное название произведения	Извлечь из списка 2/3 части элементов	Все ли произведения в списке написаны Толстым Л.Н.	
Фильм: наименование, режиссера, год выпуска	Советские фильмы Леонида Гайдая или Сергея Бондарчука	Наименование Сколько лет в прокате	Наименованию и году выпуска	режиссеру	Количество фильмов вышедших в прокат в текущем году.	Извлечь из списка элементы начиная с 1990 года выпуска (примечание: отсортировать по году выпуска)	Все ли фильмы вышли в прокат в текущем году	

	<p>Учитель : фамилия , дисциплина, стаж работы</p>	<p>Учителей физики или химии</p>	<p>Фамилия Год поступления на работу</p>	<p>Фамилии и стажу работы</p>	<p>дисциплине</p>	<p>Стаж самого опытного преподавателя физики</p>	<p>Извлечь из списка все элементы начиная с первого появления учителя с максимальным стажем работы)</p>	<p>Имеются ли преподаватели обучающие теорию вероятности.</p>
	<p>Институт Наименование ВУЗа, профиль (технический, технологический, гуманитарный и др.) адрес</p>	<p>Технические или технологические ВУЗЫ Казани</p>	<p>Названия Адреса Элементы отсортировать в алфавитном порядке по адресу.</p>	<p>Наименование и адресу</p>	<p>профилю</p>	<p>Количество гуманитарных ВУЗов в Москве</p>	<p>Извлечь из списка 1/5 часть с начала.</p>	<p>Все ли ВУЗы из списка технические.</p>
<p>Критерии оценки и шкала оценивания в баллах</p>	<p>Максимальная оценка – 3 б. Минимальная оценка – 1,8 б.</p>							

Наименование оценочного средства	Лабораторная работа 15. Разработка консольного приложения согласно варианту задания на основе применения технологии Entity Framework и подхода Code First.								
Представление и содержание оценочных материалов	На основе технологии Entity Framework и подхода Code First разработать консольное приложение и связать его с базой данных согласно варианту задания. Продемонстрировать операции сохранения, извлечения, удаления и изменения данных в БД.								
	<table border="1" style="width: 100%; text-align: center;"> <tr> <td>Класс и свойства:</td> </tr> <tr> <td>Личность: фамилия, имя, возраст.</td> </tr> <tr> <td>Врач: фамилия, специализация, категория</td> </tr> <tr> <td>Ученик: фамилия, дисциплина оценка (цифра)</td> </tr> <tr> <td>Спортсмен: фамилия, вид спорта, место</td> </tr> <tr> <td>Торговая фирма: наименование, специфика товара (продуктовый, обувной и т.д.), адрес.</td> </tr> <tr> <td>Писатель: Фамилия, название произведение, жанр</td> </tr> <tr> <td>Фильм: наименование, режиссера,</td> </tr> </table>	Класс и свойства:	Личность: фамилия, имя, возраст.	Врач: фамилия, специализация, категория	Ученик: фамилия, дисциплина оценка (цифра)	Спортсмен: фамилия, вид спорта, место	Торговая фирма: наименование, специфика товара (продуктовый, обувной и т.д.), адрес.	Писатель: Фамилия, название произведение, жанр	Фильм: наименование, режиссера,
Класс и свойства:									
Личность: фамилия, имя, возраст.									
Врач: фамилия, специализация, категория									
Ученик: фамилия, дисциплина оценка (цифра)									
Спортсмен: фамилия, вид спорта, место									
Торговая фирма: наименование, специфика товара (продуктовый, обувной и т.д.), адрес.									
Писатель: Фамилия, название произведение, жанр									
Фильм: наименование, режиссера,									

	<p style="text-align: center;">год выпуска</p> <hr/> <p style="text-align: center;">Учитель: фамилия, дисциплина, стаж работы</p> <hr/> <p style="text-align: center;">Институт Наименование ВУЗа, профиль (технический, технологический, гуманитарный и др.) адрес</p>
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 3 б.</p> <p>Минимальная оценка – 1,8 б.</p>
Наименование оценочного средства	<p style="text-align: center;">Лабораторная работа 16. Визуальное программирование. Создание визуального приложения на основе технологии Entity FrameWork. Создание связи между данными "один-ко-многим".</p>
Представление и содержание оценочных материалов	<p>Разработать визуальное приложение, содержащее минимум 2 таблицы (вторую придумать самостоятельно). Продемонстрировать работу операторов LINQ to Entity.</p> <p>На основе применения операторов класса LINQ выполнить следующие действия:</p> <ul style="list-style-type: none"> • Фильтрация согласно варианту задания. • Проекцию согласно варианту задания. • Сортировку согласно варианту задания. • Группировку согласно варианту задания. • Группировку с подсчетом количества элементов в каждой группе согласно варианту задания. • Агрегатную операцию согласно варианту задания. • Извлечение согласно варианту задания. • Проверку на соответствие элементов списка условию согласно варианту задания. • Объединения двух последовательностей (придумать вторую самостоятельно).

Класс и свойств а:	Фильтра ция	Проекция Новый тип со свойствами:	Сорти ровка по	Групп ировка по	Агрегат ные функции	Skip, Take, SkipW hile, TakeW hile	All и Any
Личност ь: фамилия , имя, возраст.	Соверше ннолетни х личносте й с фамилией начинаю щихся с гласно буквы	фамилия, имя, совершенно летний/ несовершен нолетний	фамили и и возраст у	фамили и	Число совершен нолетних	Извлеч ь из списка полови ну элемен тов	Имеются ли несоверше ннолетние
Врач: фамилия , специал изация, категори я	Врачи офтальмо логов и невролог ов, 1 категории	Фамилия врача, опытный/но вичок (стаж меньше 2 лет)	Фамил ии и категор ии	специа лизаци и	Количест во врачей хирургов	Извлеч ь из списка вторую полови ну элемен тов	Имеется хотя бы один хирург
Ученик: фамилия , дисципл ина оценка (цифра)	Ученики, которые получили по математи ке положите льные оценки	Фамилия, дисциплина , оценка (прописью)	Фамил ии и оценке	дисцип лине	Средний балл по информа тике	Извлеч ь центра льные 3 элемен та из списка	Все ли ученики получили положител ьные оценки
Спортс мен: фамилия , вид спорта, место	Спортсме ны занявшие призовые места по плаванию и фехтован ию	Фамилии спортсмено в призеров (1-3 место). Медаль (золото/сере бро/бронза)	Фамил ии и месту	Вид спорта	Количест во спортсме нов с фамилие й начинаю щих с гласной буквы	Извлеч ь из списка все элемен ты с момент а появле ния спортс мена занявш	Есть ли среди спортсмен ов те, кто занял 1 место

							его 1 места.	
Торговая фирма: наименование, специфика товара (продуктовый, обувной и т.д.), адрес.	Торговые фирмы занимающиеся продажей обувной продукции и или одежды.	Только обувные магазины: Наименование Адрес	Адресу и названию торговой фирмы	Специфике товара	Число торговых фирм продающих их продукты питания.	Извлечь из списка элементы, пока они соответствуют магазину одежды или обуви)	Имеются ли торговые фирмы продающие ювелирные изделия.	
Писатель: Фамилия, название произведения, жанр	Повести и рассказы с указанием писателя	Названия произведений в порядке от Я до А	Фамилии и названия произведения	жанру	Самое длинное название произведения	Извлечь из списка 2/3 части элементов	Все ли произведения в списке написаны Толстым Л.Н.	
Фильм: наименование, режиссера, год выпуска	Советские фильмы Леонида Гайдая или Сергея Бондарчука	Наименование Сколько лет в прокате	Наименованию и году выпуска	режиссеру	Количество фильмов вышедших в прокат в текущем году.	Извлечь из списка элементы начиная с 1990 года выпуска (примечание: отсортировать по году выпуска)	Все ли фильмы вышли в прокат в текущем году	
Учитель:	Учителей физики	Фамилия	Фамилии и	дисциплине	Стаж самого опытного	Извлечь из списка	Имеются ли преподава	

	<p>фамилия , дисциплина, стаж работы</p>	<p>или химии</p>	<p>Год поступления на работу</p>	<p>стажу работы</p>		<p>преподавателя физики</p>	<p>все элементы начина я с первог о появле ния учител я с максим альным стажем работы)</p>	<p>тели обучающи м теории вероятнос ти.</p>
	<p>Институ т Наимено вание ВУЗа, профиль (техниче ский, технолог ический, гуманит арный и др.) адрес</p>	<p>Техничес кие или технолог ические ВУЗЫ Казани</p>	<p>Названия Адреса Элементы отсортирова ть в алфавитном порядке по адресу.</p>	<p>Наимен ованию и адресу</p>	<p>профил ю</p>	<p>Количест во гуманита рных ВУЗов в Москве</p>	<p>Извлеч ь из списка 1/5 часть с начала.</p>	<p>Все ли ВУЗы из списка техническ ие.</p>
<p>Критер ии оценки и шкала оценив ания в баллах</p>	<p>Максимальная оценка – 5 б. Минимальная оценка – 3 б.</p>							
<p>Наиме новани е</p>	<p>Практическая работа 1 и 2</p>							

<p>оценочного средства</p>	
<p>Представление и содержание оценочных материалов</p>	<p>В соответствие с выбранной темой построить следующие UML-диаграммы: диаграмма прецедентов, поток событий, диаграмма последовательностей, диаграмма деятельности, диаграмма классов, диаграмма состояний.</p> <p>В соответствие с формулированными требованиями к функциональным возможностям программного обеспечения, алгоритмам последовательности действий, диаграмме классов разработать приложение, автоматизирующее задачи организационного управления, бизнес-процессы согласно выбранной теме. Применить изученные подходы структурного, объектно-ориентированного, событийного и визуального программирования. Готовый программный продукт должен хранить данные в структурированном виде (т.е. обязательно подключение к СУБД MS SQL Server). Провести верификацию и тестирование готового программного продукта.</p> <p>Перечь тем практической работы:</p> <ol style="list-style-type: none"> 1. Проектирование системы интернет-бронирования гостиницы. 2. Проектирование системы реализации готовой продукции. 3. Проектирование системы интернет-заказов товаров магазина электроники. 4. Проектирование системы предоставления и запроса вакансий для бюро по трудоустройству. 5. Проектирование системы электронной записи клиентов нотариальной конторы. 6. Проектирование системы интернет-заказов у поставщиков автозапчастей. 7. Проектирование системы записи и учета прохождения курсов повышения квалификации. 8. Проектирование электронной системы учета оценок студентов 9. Проектирование электронной системы распределения нагрузки преподавателей. 10. Проектирование информационной системы страховой компании. 11. Проектирование системы контроля сроков и обслуживания клиентов ломбарда. 12. Проектирование электронной системы записи на прием пациентов частной клиники. 13. Проектирование системы учета кадров на предприятии. 14. Проектирование электронной системы заказа книг в библиотеке. 15. Проектирование театральной интернет-кассы. 16. Проектирование системы бронирования для проката автомобилей. 17. Проектирование системы учета рекламы в эфире телеканала. 18. Проектирование системы электронного расписания работы телеканала. 19. Проектирование системы интернет-заказов ювелирной мастерской. 20. Проектирование интернет-магазина одежды. 21. Проектирование электронной системы сдачи в аренду торговых площадей. 22. Проектирование системы продажи и бронирования билетов кинотеатра через интернет. 23. Проектирование интернет-афиши и справки кинотеатра. 24. Проектирование системы учета технического обслуживания станков. 25. Проектирование информационной системы турфирмы. 26. Проектирование системы покупки и бронирования билетов на поезд. 27. Проектирование информационной системы компании грузоперевозок. 28. Проектирование системы учета телефонных разговоров сотрудников. 29. Проектирование интернет-системы подачи заявок на оформление кредита. 30. Проектирование интернет-кабинета клиента банка. 31. Проектирование информационной системы агентства недвижимости. 32. Проектирование интернет-системы записи и учета скидок клиентов салона красоты. 33. Проектирование системы регистрации и контроля сообщений участников интернет-форума. 34. Проектирование системы доставки товаров из магазина. 35. Проектирование интернет-системы заказа и доставки пиццы. 36. Проектирование информационной системы детского сада. 37. Проектирование системы курсов дистанционного обучения. 38. Проектирование системы футбольных ставок. 39.

	<p>Проектирование системы бронирования столиков и заказа блюд меню ресторана по интернету. 40. Проектирование системы обслуживания клиентов частной почтовой службы. 41. Проектирование системы учета сбыта продукции сельскохозяйственного предприятия. 42. Проектирование системы маркетинга предприятия. 43. Проектирование информационной системы компании прямых продаж косметики. 44. Проектирование каталога и системы заказов легковых автомобилей по интернету. 45. Проектирование системы гарантийного обслуживания электротоваров.</p>
<p>Критерии оценки и шкала оценивания в баллах</p>	<p>Максимальное количество баллов за защиту практической работы – 13.</p> <p>Минимальное количество баллов за защиту практической работы – 7,8.</p>

4. Оценочные материалы промежуточного тестирования

Наименование оценочного средства	Тест
<p>Представление и содержание оценочных материалов</p>	<p>Вопрос 1</p> <p>Что будет выведено на консоль в результате выполнения следующей программы?</p> <pre> class Person { private int age = 15; public int Age { get { return age; } set { } } } class Program { </pre>

```
static void Main(string[] args)
{
    Person tom = new Person();
    tom.Age = 25;
    Console.WriteLine(tom.Age);

    Console.ReadKey();
}
}
```

ОТВЕТЫ:

15

25

0

Программа не скомпилируется

Вопрос 2

Корректна ли следующая перегрузка методов?

```
static void IncrementVal(ref int val)
{
    val++;
    Console.WriteLine(val);
}
```

```
static void IncrementVal(int val)
{
    val++;
    Console.WriteLine(val);
}
```

Да

нет

Вопрос 3

Какое ключевое слово применяется для переопределения виртуальных методов и свойств?

static

override

virtual

abstract

Вопрос 4

Что неправильно в определении следующих интерфейсов?

```
public abstract interface IAction
```

```
{  
    void Move();  
}
```

```
internal interface IRunAction : IAction
```

```
{  
    void Run();  
}
```

Интерфейсы не могут иметь модификатор доступа internal

Интерфейсы не могут иметь модификатор abstract

Интерфейс IRunAction должен определить метод Move, так как базовый интерфейс IAction является абстрактным

От абстрактного интерфейса нельзя наследоваться

Вопрос 5

Интерфейсы

Какие интерфейсы применяют для сортировки наборов сложных объектов:

IComparable

IComparer

ICloneable

IEnumerator

Вопрос 6

Какой спецификатор необходимо использовать, чтобы заменить элемент базового класса новым элементом?

new

private

override

private

Вопрос 7

Какие члены класса могут быть определены как абстрактные?

поля

события

свойства

методы

индексаторы

Вопрос 8

В чем различие между ключевыми словами "ref" и "out"?

Параметр с ключевым словом ref может быть не инициализирован, а параметр с ключевым словом out обязательно должен быть инициализирован до вызова метода, который использует эти параметры.

Параметр с ключевым словом out может быть не инициализирован, а параметр с ключевым словом ref обязательно должен быть инициализирован до вызова метода, который использует эти параметры.

Нет различий.

Ключевым словом ref может использоваться только со значимыми типами (valuetypes), а out может использоваться как со значимыми, так и с ссылочными типами.

Вопрос 9

В чем отличия интерфейса от абстрактного класса?

В C# множественное наследование возможно только за счет реализации интерфейсов.

Интерфейс не может содержать полей и обычных методов.

Абстрактные классы не наследуются.

При наследовании от интерфейса необходимо переопределять все методы, в то время как при использовании абстрактных классов такой необходимости нет.

Вопрос 10

Какой уровень доступа имеют поля класса, если модификатор доступа не указан:

protected

internal

protected internal

private

public

Вопрос 11

Какой уровень доступа имеют методы класса, если модификатор доступа не указан:

protected

internal

protected internal

private

public

Вопрос 12

Какой уровень доступа имеют методы интерфейса, если модификатор доступа не указан:

protected

internal

protected internal

private

public

Вопрос 13

Какой спецификатор применяют для создания бесплодных классов:

protected

internal

abstract

private

sealed

Вопрос 14

В классе Counter:

Class Counter

```
{
```

```
    Public int Number{ get; set; }
```

```
    // определение оператора сложения
```

```
}
```

определен оператор сложения, который позволяет складывать объект Counter с объектом типа int следующим образом:

```
Counter counter = new Counter { Number = 45 };
```

```
int x = counter + 6;
```

```
Console.WriteLine(x); // 51
```

Какой именно оператор сложения определен в классе Counter?

```
public int operator + (Counter counter, int val)
```

```
{
```

```
    return counter.Number + val;
```

```
}
```

```
public static int operator + (Counter counter, intval)
```

```
{
```

```
    return counter.Number + val;
```

```
}
```

```
public static int operator + (intval, Counter counter)
```

```
{
```

```
    return counter.Number + val;
```

```
}
```

```
public static Counter operator + (Counter counter, intval)
```

```
{
```

```
    return new Counter {Number = counter.Number + val};
```

```
}
```

```
public static Counter operator + (intval, Counter counter)
```

```
{
```

```
    return new Counter {Number = counter.Number + val};
```

```
}
```

Вопрос 15

Инкапсулированная абстракция с четко определенным интерфейсом называется

Вопрос 16

Дан следующий класс:

```
class Person
```

```
{
```

```
    public string name = "Sam";
```

```
    public int age;
```

```
    public Person(string name, int age)
```

```
{
```

```
this.name = name;
```

```
this.age = age;
```

```
}
```

```
}
```

Какое значение поле name будет иметь при выполнении следующего кода?

```
Person tom = new Person("Tom", 34)
```

Вопрос17

Дан следующий класс:

```
class Person
```

```
{
```

```
    public string name = "Ben";
```

```
    public int age = 18;
```

```
    public string email = "ben@gmail.com";
```

```
    public Person(string name)
```

```
    {
```

```
        this.name = name;
```

```
    }
```

```
    public Person(string name, int age) : this(name)
```

```
    {
```

```
        this.age = age;
```

```
    }
```

```
    public Person(string name, int age, string email) : this("Bob", age)
```

```
    {
```

```
        this.email = email;
```

```
    }
```

```
}
```

Какое значения будет иметь поле name после выполнения следующего кода?

```
Person person = new Person("Tom", 31, "tom@gmail.com");
```

Вопрос18

Даны следующие классы:

```
class Person
```

```
{
```

```
    string name;
```

```
    int age;
```

```
    public Person()
```

```
    {
```

```
    }
```

```
    public Person(string name) : this(name, 18)
```

```
    {
```

```
    }
```

```
    public Person(string name, int age)
```

```
    {
```

```
        this.name = name;
```

```
        this.age = age;
```

```
    }
```

```
}
```

```
class Employee : Person
```

```
{
```

```
    string company;
```

```
    public Employee()
```

```
    {
```

```
    }
```

```
    public Employee(string name, int age, string company): base(name,  
age)
```

```
{
    this.company = company;
}

public Employee(string name, string company) : base(name)
{
    this.company = company;
}
}
```

Укажите название базового класса.

Вопрос 19

Дана программа:

```
class Auto // легковой автомобиль
{
    public int Seats { get; set; } // количество сидений
    public Auto(int seats)
    {
        Seats = seats;
    }
}

class Truck : Auto // грузовой автомобиль
{
    public decimal Capacity { get; set; } // грузоподъемность
    public Truck(int seats, decimal capacity)
    {
        Seats = seats;
        Capacity = capacity;
    }
}

class Program
```

```
{  
  
    static void Main(string[] args)  
  
    {  
  
        Truck truck = new Truck(2, 1.1m);  
  
        Console.WriteLine($"Грузовик с грузоподъемностью  
{truck.Capacity} тонн");  
  
        Console.ReadKey();  
  
    }  
}
```

Укажите тип свойства объявленного в классе потомке.

Вопрос 20

Что выведет на консоль следующая программа?

```
class Auto // легковой автомобиль  
  
{  
  
    public int Seats { get; set; } // количество сидений  
  
    public Auto()  
  
    {  
  
        Console.WriteLine("Auto has been created");  
  
    }  
  
    public Auto(int seats)  
  
    {  
  
        Seats = seats;  
  
    }  
}  
  
class Truck : Auto // грузовой автомобиль  
  
{  
  
    public decimal Capacity { get; set; } // грузоподъемность  
  
    public Truck(decimal capacity)  
  
    {
```

```
        Seats = 2;

        Capacity = capacity;

        Console.WriteLine("Truck has been created");

    }
}

class Program
{
    static void Main(string[] args)
    {
        Truck truck = new Truck(1.1m);

        Console.WriteLine($"Truck with capacity {truck.Capacity}");

        Console.ReadKey();

    }
}
```

Укажите вторую строчку, которая будет выведена на консоль.

Вопрос 21

Что будет выведено на консоль в результате выполнения следующей программы?

```
class Person
{
    public string Name { get; set; }

    public virtual void Display()
    {
        Console.WriteLine($"Person {Name}");
    }
}

class Employee : Person
{
    public string Company { get; set; }
```

```

public override void Display()
{
    Console.WriteLine($"Employee {Name}");
}
}

class Program
{
    static void Main(string[] args)
    {
        Person person = new Employee { Name = "Sam", Company
= "Microsoft" };
        person.Display();

        Console.ReadKey();
    }
}

```

Вопрос 22

Что будет выведено на консоль в результате выполнения следующей программы?

```

class Person
{
    public string Name { get; set; }
    public virtual void Display()
    {
        Console.WriteLine($"Person {Name}");
    }
}

class Employee : Person
{
    public string Company { get; set; }
}

```

```
public override void Display()
{
    Console.WriteLine($"Employee {Name}");
}
}

class Manager : Employee
{
    public override void Display()
    {
        Console.WriteLine($"Manager {Name}");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Person person = new Manager { Name = "Bob", Company
= "Microsoft" };
        Employee employee = (Employee)person;
        employee.Display();

        Console.ReadKey();
    }
}
```

Вопрос 23

Есть следующий делегат:

```
delegate int Operation (int val);
```

Сколько методов соответствуют данному делегату (укажите число):

```
static void Method1(int x)
```

```
{  
    Console.WriteLine(x * x);  
}  
static int Method2(ref int x)  
{  
    return x * x;  
}  
static int Method3(int x, int y)  
{  
    return x * y;  
}  
static int Method4(int x, int y = 7)  
{  
    return x * y;  
}  
static int Method5(out int x)  
{  
    x = 7;  
    return x * x;  
}
```

Вопрос 24

Что будет выведено на консоль в результате выполнения следующей программы:

```
class Program  
{  
    delegate int Operation(int x, int y);  
  
    static void Main(string[] args)
```

```
{  
  
    Operation del = Add;  
  
    del += Multiply;  
  
  
    int result = del(6, 5);  
  
    Console.WriteLine(result);  
  
    Console.Read();  
  
}  
  
private static int Add(int x, int y) { return x + y; }  
private static int Multiply(int x, int y) { return x * y; }  
}
```

Вопрос 25

Переменные, описанные в классе, называются _____ класса.

Вопрос 26

Функциональный элемент класса, реализующий вычисления или другие действия, называется _____.

Вопрос 27

При инициализации объекта вызывается элемент класса, который называется _____.

Вопрос 28

Класс, предназначенный для хранения ссылок на методы называется _____.

Вопрос 29

Элемент класса, позволяющий ему посылать другим объектам (наблюдателям) уведомления об изменении своего состояния называется _____.

Вопрос 30

Для исключения из программы повторяющихся фрагментов кода; упрощения модификации программы; упрощения создания новых программ, на основе существующих применяют _____.

Вопрос 31

Скрытие деталей реализации объекта, представление его в укрупненном виде и использование через интерфейс называется _____.

Вопрос 32

Понятие полиморфизма связывают с механизмом реализации _____ методов.

Вопрос 33

Использование нескольких методов с одним и тем же именем, но различными типами параметров называется _____ методов.

Вопрос 34

Верно ли утверждение : "Конструктор, вызываемый без параметров, называется конструктором по умолчанию"?

Верно

Неверно

Вопрос 35

Верно ли утверждение: "Статический метод вызывается через имя экземпляра класса"?

Верно

Неверно

Вопрос 36

	<p>Верно ли утверждение: "Величины ссылочного типа равны, если они ссылаются на одни и те же данные"?</p> <p>Верно</p> <p>Неверно</p> <p>Вопрос 37</p> <p>Верно ли утверждение: "Если в конструкторе производного класса явный вызов конструктора базового класса отсутствует, автоматически вызывается конструктор базового класса без параметров"?</p> <p>Верно</p> <p>Неверно</p> <p>Вопрос 38</p> <p>Верно ли утверждение: "Объекту базового класса можно присвоить объект производного класса"?</p> <p>Верно</p> <p>Неверно</p> <p>Вопрос 39</p> <p>Верно ли утверждение: «Операция as выполняет преобразование к заданному типу, а если это невозможно, формирует результат null»?</p> <p>Верно</p> <p>Неверно</p> <p>Вопрос 40</p> <p>Укажите название пользовательского типа данных, предназначенного только для порождения потомков?</p>
<p>Критерии оценки и шкала оценивания в баллах</p>	<p>При оценке тестовых заданий учитываются следующие критерии:</p> <p>1 правильный ответ = 0,5 Баллы</p> <p>Максимальное количество баллов - 20</p>
<p>Наименование оценочного средства</p>	<p>Экзаменационные вопросы</p>

<p>Представление и содержание оценочных материалов</p>	<ol style="list-style-type: none"> 1. Принципы ООП. Достоинства и недостатки ООП. 2. Основные понятия ООП: класс, объект. Классы: описание, доступ к полям. 3. Методы: определение, общая форма записи. 4. Конструкторы: определение, общая форма, особенности. 5. Инкапсуляция. Управление доступом к членам класса: спецификаторы доступа. 6. Передача параметров в метод по значению. 7. Передача параметров в метод по ссылке. 8. Полиморфизм. Перегрузка методов: сущность, особенности реализации. 9. Перегрузка конструкторов: сущность, особенности реализации. 10. Свойства: назначение, общая форма. 11. Наследование: сущность, описание производного класса. 12. Операции класса: перегрузка унарных, бинарных операций. Операции преобразования типа. 13. Наследование. Класс Object. Абстрактные и бесплодные классы. 14. Интерфейсы. Применение стандартных интерфейсов .NET для сравнения, перебора, сортировки и клонирования объектов. 15. Делегаты и события. Функциональная параметризация.
<p>Критерии оценки и шкала оценивания в баллах</p>	<p>При оценке ответов на экзаменационный вопрос учитываются следующие критерии:</p> <p>Отвечает на основной вопрос правильно +2 дополнительно – 20 баллов.</p> <p>Отвечает на основной вопрос правильно +1 дополнительно – 15 баллов.</p> <p>Отвечает на основной вопрос правильно – 10 баллов.</p> <p>Отвечает не полно на основной вопрос правильно – 5 баллов.</p> <p>Не отвечает ни на 1 вопрос – 0 баллов.</p> <p>Максимальное количество баллов - 20</p>