



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное  
учреждение высшего образования

КГЭУ

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

АКТУАЛИЗИРОВАНО  
Решением Ученого совета ИЦТЭ КГЭУ  
Протокол №7 от 19.03.2024

УТВЕРЖДАЮ  
Директор института  
Цифровых технологий и экономики  
\_\_\_\_\_ Торкунова Ю.В.

«24» июня 2021 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Платформы разработки бизнес-приложений и языки программирования

Направление подготовки 09.03.03 Прикладная информатика

Направленность (профиль) Прикладная информатика в экономике и анализ данных

Квалификация

бакалавр

г. Казань, 2021

Рабочая программа дисциплины разработана в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Программу разработала:

Доцент, к.т.н. \_\_\_\_\_ Тазиева Р.Ф.

Рабочая программа рассмотрена и одобрена на заседании кафедры Информатика и информационно-управляющие системы, протокол № 9 от 07.06.2021

Зав. кафедрой \_\_\_\_\_ Торкунова Ю.В.

Программа рассмотрена и одобрена на заседании выпускающей кафедры Информатика и информационно-управляющие системы, протокол № 9 от 07.06.2021

Зав. кафедрой \_\_\_\_\_ Торкунова Ю.В.

Программа одобрена на заседании методического совета института Цифровых технологий и экономики, протокол № 10 от 22.06.2021

Зам. директора института Цифровых технологий и экономики \_\_\_\_\_ Косулин В.В.

Программа принята решением Ученого совета института Цифровых технологий и экономики протокол № 10 от 22.06.2021

Согласовано:

Руководитель ОПОП \_\_\_\_\_ \_Смбаева Г. Р.\_

## 1. Цель, задачи и планируемые результаты обучения по дисциплине

Целью изучения дисциплины «Платформы разработки бизнес-приложений и язык программирования» является теоретическая и практическая подготовка студентов в области разработки программного обеспечения с использованием объектно-ориентированного языка программирования C# и платформы .Net Core. 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 навыков работы на платформе .Net Core.
- приобретение студентами знаний о сущности объектно-ориентированного подхода в программировании;
- ознакомление с технологиями создания новых пользовательских типов данных на языке программирования C#;
- 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)
ПК-2 Способен проводить тестирование компонентов программного обеспечения ИС	ПК-2.1 Применяет современные методики тестирования разрабатываемых информационных систем	<i>Знать:</i> инструменты и методы тестирования нефункциональных и функциональных характеристик ИС <i>Уметь:</i> верифицировать код ИС и баз данных структуры баз данных ИС относительно дизайна ИС <i>Владеть:</i> навыками выполнения работ по тестированию компонентов ИС, автоматизирующих задачи организационного управления и бизнес-процессы
ПК-2 Способен проводить тестирование компонентов программного обеспечения ИС	ПК-2.2 Использует технологические и функциональные стандарты, современные модели и методы оценки качества и надежности при проектировании, конструировании и отладке программных средств	<i>Знать:</i> технологические и функциональные стандарты, современные модели и методы оценки качества и надежности на всех этапах разработки ИС. <i>Уметь:</i> проводить оценку качества и надежности при проектировании, конструировании и отладке программных средств с учётом технологических и функциональных

		стандартов, современных моделей и методов их оценки. <i>Владеть:</i> навыками ручного и автоматизированного тестирования разрабатываемых информационных систем.
--	--	---

## 2. Место дисциплины в структуре ОПОП

Дисциплина Платформы разработки бизнес-приложений и языки программирования относится к части, формируемой участниками образовательных отношений учебного плана по направлению подготовки 09.03.03 Прикладная информатика.

Код компетенции	Предшествующие дисциплины (модули), практики, НИР, др.	Последующие дисциплины (модули), практики, НИР, др.
УК-1	Выполнение и защита выпускной квалификационной работы	
УК-2	Выполнение и защита выпускной квалификационной работы	
УК-3	Выполнение и защита выпускной квалификационной работы	
УК-4	Выполнение и защита выпускной квалификационной работы	
УК-5	Выполнение и защита выпускной квалификационной работы	
УК-6	Выполнение и защита выпускной квалификационной работы	
УК-7	Выполнение и защита выпускной квалификационной работы	
УК-8	Выполнение и защита выпускной квалификационной работы	
ОПК-1	Выполнение и защита выпускной квалификационной работы	
ОПК-2	Выполнение и защита выпускной квалификационной работы	
ОПК-3	Выполнение и защита выпускной квалификационной работы	
ОПК-4	Выполнение и защита выпускной квалификационной работы	
ОПК-5	Выполнение и защита выпускной квалификационной работы	
ОПК-6	Выполнение и защита выпускной квалификационной работы	

ОПК-7	Выполнение и защита выпускной квалификационной работы	
ОПК-8	Выполнение и защита выпускной квалификационной работы	
ОПК-9	Выполнение и защита выпускной квалификационной работы	
ПК-1		Разработка и использование мобильных приложений
ПК-1	Выполнение и защита выпускной квалификационной работы	
ПК-2	Выполнение и защита выпускной квалификационной работы	
ПК-3	Выполнение и защита выпускной квалификационной работы	
ПК-4	Выполнение и защита выпускной квалификационной работы	

Для освоения дисциплины обучающийся должен:

**Знать:** методы и технологии программирования, синтаксис и основные конструкции изучаемого языка программирования, 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 3 зачетных(ые) единиц(ы) (ЗЕ), всего 108 часов, из которых 43 часов составляет контактная работа обучающегося с преподавателем (занятия лекционного типа 32 час., занятия семинарского типа (практические, семинарские занятия, лабораторные работы и т.п.) 8 час., групповые и индивидуальные консультации 2 час., прием экзамена (КПА), зачета с оценкой - 1 час., самостоятельная работа обучающегося 30 час, контроль самостоятельной работы (КСР) - 0 час. Практическая подготовка по виду профессиональной деятельности составляет 4 часа.

Вид учебной работы	Всего часов	Семестр
		8
<b>ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ</b>	108	108
<b>КОНТАКТНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ С ПРЕПОДАВАТЕЛЕМ,</b> в том числе:	43	43
Лекционные занятия (Лек)	32	32
Практические занятия (Пр)	8	8
Консультации (Конс)	2	2
Контактные часы во время аттестации (КПА)	1	1
<b>САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ (СРС):</b>	30	30
Подготовка к промежуточной аттестации в форме: (экзамен)	35	35
<b>ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ</b>	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							Итого	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						
<b>Раздел 1. Структурный подход к программированию</b>														
1. Базовые алгоритмические конструкции следование, ветвление и циклы.	8	4	2						6	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -З1, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	ПР		

2. Структуры данных : массивы, строки, списки.	8	4	2							6	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -З1, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр		
3. Работа с потоками файловой системой.	8	4	2							6	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -З1, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр		
Раздел 2. Объектно-ориентированный подход к программированию															
4. Основные принципы объектно-ориентированной модели. Статические классы. Разработка библиотеки классов.	8	4	2			8				14	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -З1, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр		15
5. Организация иерархии классов.	8	4				8				12	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -З1, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр		15

6. Интерфейсы. Стандартные интерфейсы.	8	4				4				8	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -31, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр		7
7. Делегаты. События.	8	4				8				12	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -31, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр		16
8. Класс Linq.	8	4				2				6	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -31, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Пр Тест		7
<b>Раздел 3. Промежуточная аттестация</b>															
Подготовка промежуточной аттестации	к					2	2			1					
Промежуточная аттестация (Э)	8								35	38	ПК-2.1 -У1, ПК-2.1 -В1, ПК-2.2 -31, ПК-2.2 -У1, ПК-2.2 -В1	Л1.1, Л1.2, Л1.3, Л1.4, Л2.1, Л2.2	Э		40
<b>ИТОГО</b>		32	8		2	30			35	1	108				100



### 3.3. Тематический план лекционных занятий

Номер раздела дисциплины	Темы лекционных занятий	Трудоемкость, час.
1	Программирование на Visual Studio, ЯП C#. .Net - открытая среда разработки. Лексические основы языка C#. Типы данных. Операторы языка C#.	4
2	Программирование на Visual Studio, ЯП C#. Массивы. Строки. Списки.	4
3	Программирование на ЯП C#. BinaryReader, BinaryWriter, FileStream, StreamReader, StreamWriter, File, Directory.	4
4	Основные элементы класса: поля, методы, конструкторы, свойства. Виды параметров методов. Модель "делегирования - включения".	4
5	Организация иерархий классов. Раннее и позднее связывание. Виртуальные методы. Абстрактные и бесплодные классы. Виды взаимоотношений между классами.	4
6	Описание и использование интерфейсов. Применение стандартных интерфейсов .NET для сравнения, перебора, сортировки и клонирования объектов.	4
7	Делегаты. Обеспечения связи между объектами по типу «источник — наблюдатель».Создания универсальных методов (поддержки механизма обратных вызовов). Анонимные методы. События. Лямбда-выражения.	4
8	Linq to Objects. Сортировка, фильтрация, группировка, агрегатные операции.	4
	Всего	32

### 3.4. Тематический план практических занятий

Номер раздела дисциплины	Темы практических занятий	Трудоемкость, час.
1	Класс математических функций Math. Следование. Условный оператор. Условная тернарная операция.	1
2	Циклы. Оператор обработки исключений.	1
3	Одномерные, двумерные, ступенчатые массивы.	1
4	Работа со структурой данных список.	1
5	Работа со строками. Сохранение и чтение данных из текстового файла.	2
6	Разработка пользовательского типа данных. Методы. Переопределение операций.	2
	Всего	8

### 3.5. Тематический план лабораторных работ

Данный вид работы не предусмотрен учебным планом

### 3.6. Самостоятельная работа студента

Номер раздела дисциплины	Вид СРС	Содержание СРС	Трудоемкость, час.
1	Создание библиотеки классов со статическими методами.	Изучение конспектов лекций и составление отчета по заданию.	4
2	Индексаторы.	Изучение конспектов лекций и составление отчета по заданию.	4
3	Создание иерархии пользовательских типов данных. Виртуальные методы.	Изучение конспектов лекций и составление отчета по заданию.	4
4	Создание приложения согласно паттерну "делегирование-включение".	Изучение конспектов лекций и составление отчета по заданию.	4
5	Организация клонирования (копирования) объектов, сравнения и перебора на основе применения стандартных интерфейсов IComparable, IComparer, IEnumerable, IEnumerator, ICloneable.	Изучение конспектов лекций и составление отчета по заданию.	4
6	Делегаты. Применение метода функциональной параметризации для реализации метода половинного деления отрезка с целью решения трансцендентных уравнений.	Изучение конспектов лекций и составление отчета по заданию.	4
7	Событийное программирование. События. Задача на регистрацию нарушений скоростного режима. Моделирование игры в ping-pong	Изучение конспектов лекций и составление отчета по заданию.	4

8	Применение операторов класса Linq для организации обработки данных пользовательских классов.	Изучение конспектов лекций и составление отчета по заданию.	2
Всего			30

#### 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 и написание курсовой работы.

При реализации дисциплины «Платформы разработки бизнес-приложений и языки программирования» по образовательной программе «Прикладная информатика в экономике и анализ данных» направления подготовки бакалавров 09.03.03 «Прикладная информатика» применяются электронное обучение и дистанционные образовательные технологии.

В образовательном процессе используются:

- дистанционные курсы (ДК), размещенные на площадке LMS Moodle, URL: <http://lms.kgeu.ru/>;

- электронные образовательные ресурсы (ЭОР), размещенные в личных кабинетах студентов Электронного университета КГЭУ, URL: <http://e.kgeu.ru/>

#### 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 практических 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.

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	Продemonстрированы все основные умения, решены все основные задачи с отдельными несущественными недочетами, выполнены все
----------------	--	--	---	---

			объеме, но некоторые с недочетами	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	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов
Характеристика сформированности компетенции (индикатора достижения компетенции)	Компетенция в полной мере не сформирована. Имеющихся знаний, умений, навыков недостаточно для решения практических (профессиональных) задач	Сформированность компетенции соответствует минимальным требованиям. Имеющихся знаний, умений, навыков в целом достаточно для решения практических (профессиональных) задач, но требуется дополнительная практика по большинству практических задач	Сформированность компетенции в целом соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в целом достаточно для решения стандартных практических (профессиональных) задач	Сформированность компетенции полностью соответствует требованиям. Имеющихся знаний, умений, навыков и мотивации в полной мере достаточно для 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 компетенции	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-2	ПК-	Знать				

2.1	инструменты и методы тестирования нефункциональных и функциональных характеристик ИС	Знает инструменты и методы тестирования нефункциональных и функциональных характеристик ИС.	Знает базовые инструменты и методы тестирования нефункциональных и функциональных характеристик ИС.	Знает некоторые инструменты и методы тестирования нефункциональных и функциональных характеристик ИС.	Уровень знаний ниже минимального уровня требований.
	Уметь				
	верифицировать код ИС и баз данных структуры баз данных ИС относительно дизайна ИС	Умеет верифицировать код ИС и баз данных структуры баз данных ИС относительно дизайна ИС.	Умеет верифицировать код ИС и баз данных структуры баз данных ИС относительно дизайна ИС. Допускает ряд незначительных ошибок.	Умеет верифицировать код ИС и баз данных структуры баз данных ИС относительно дизайна ИС. Задания выполняет не в полном объеме.	Допускает грубые ошибки при верификации кода ИС.
	Владеть				
	навыками выполнения работ по тестированию компонентов ИС, автоматизирующих задачи организационного управления и бизнес-процессы	Демонстрирует хорошие навыки выполнения работ по тестированию компонентов ИС, автоматизирующих задачи организационного управления и бизнес-процессы.	Демонстрирует базовые навыки выполнения работ по тестированию компонентов ИС, автоматизирующих задачи организационного управления и бизнес-процессы.	Демонстрирует некоторые навыки выполнения работ по тестированию компонентов ИС, автоматизирующих задачи организационного управления и бизнес-процессы.	Допускает грубые ошибки при выполнении работ по тестированию компонентов ИС, автоматизирующих задачи организационного управления и бизнес-процессы.
ПК-	Знать				

		технологические и функциональные стандарты, современные модели и методы оценки качества и надежности на всех этапах разработки ИС	Знает технологические и функциональные стандарты, современные модели и методы оценки качества и надежности на всех этапах разработки ИС.	Знает базовые технологические и функциональные стандарты, современные модели и методы оценки качества и надежности на всех этапах разработки ИС.	Знает некоторые технологические и функциональные стандарты, современные модели и методы оценки качества и надежности на всех этапах разработки ИС.	Уровень знаний ниже минимального уровня требований.
		Уметь				
	2.2	проводить оценку качества и надежности при проектировании, конструировании и отладке программных средств с учётом технологических и функциональных стандартов, современных моделей и методов их оценки	Уметь проводить оценку качества и надежности при проектировании, конструировании и отладке программных средств с учётом технологических и функциональных стандартов, современных моделей и методов их оценки.	Уметь проводить оценку качества и надежности при проектировании, конструировании и отладке программных средств с учётом технологических и функциональных стандартов, современных моделей и методов их оценки. Допускает ряд незначительных ошибок.	Уметь проводить оценку качества и надежности при проектировании, конструировании и отладке программных средств с учётом технологических и функциональных стандартов, современных моделей и методов их оценки. Допускает ряд незначительных ошибок. Задания выполняет не в полном объеме.	Допускает грубые ошибки при оценке качества и надежности при проектировании, конструировании и отладке программных средств с учётом технологических и функциональных стандартов, современных моделей и методов их оценки.
		Владеть				



		навыками ручного и автоматизированного тестирования разрабатываемых информационных систем	Демонстрирует хорошие навыки ручного и автоматизированного тестирования разрабатываемых информационных систем .	Демонстрирует базовые навыки ручного и автоматизированного тестирования разрабатываемых информационных систем .	Демонстрирует некоторые навыки ручного и автоматизированного тестирования разрабатываемых информационных систем .	Допускает грубые ошибки при проведении ручного и автоматизированного тестирования разрабатываемых информационных систем .
--	--	---	---	---	---	---

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	Симонова Е. В.	Структуры данных в С#: линейные и нелинейные динамические структуры	учебное пособие	СПб.: Лань	2018	<a href="https://e.lanbook.com/book/110938">https://e.lanbook.com/book/110938</a>	1
2	Тюкачев Н. А., Хлебостроев В. Г.	С#. Основы программирования	учебное пособие	СПб.: Лань	2018	<a href="https://e.lanbook.com/book/104962">https://e.lanbook.com/book/104962</a>	1
3	Гвоздева Т. В., Баллод Б. А.	Проектирование информационных систем. Стандартизация	учебное пособие	СПб.: Лань	2019	<a href="https://e.lanbook.com/book/115515">https://e.lanbook.com/book/115515</a>	1
4	Остроух А. В., Суркова Н. Е.	Проектирование информационных систем	монография	СПб.: Лань	2019	<a href="https://e.lanbook.com/book/118650">https://e.lanbook.com/book/118650</a>	1

## 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	Павловская Т. А.	С#. Программирование на языке высокого уровня	учебник для вузов	СПб.: Питер	2012	<a href="https://ibooks.ru/reading.php?productid=28473">https://ibooks.ru/reading.php?productid=28473</a>	1
2	Стилмен Э., Грин Д.	Изучаем С#		СПб.: Питер	2014	<a href="https://ibooks.ru/reading.php?productid=341182">https://ibooks.ru/reading.php?productid=341182</a>	1

## 6.2. Информационное обеспечение

### 6.2.1. Электронные и интернет-ресурсы

№ п/п	Наименование электронных и интернет-ресурсов	Ссылка
1	ИНТУИТ. Национальный Открытый Университет.	<a href="https://www.intuit.ru/">https://www.intuit.ru/</a>
2	Документация по С#.	<a href="https://docs.microsoft.com/ru-ru/dotnet/csharp/">https://docs.microsoft.com/ru-ru/dotnet/csharp/</a>
3	Сайт о программировании.	<a href="https://metanit.com/">https://metanit.com/</a>
4	Электронный университет КГЭУ - виртуальная образовательная среда.	<a href="https://lms.kgeu.ru/course/view.php?id=2663">https://lms.kgeu.ru/course/view.php?id=2663</a>

### 6.2.2. Профессиональные базы данных

№ п/п	Наименование профессиональных баз данных	Адрес	Режим доступа
1	Официальный сайт Министерства науки и высшего образования РФ	<a href="https://www.minobrnauki.gov.ru/">https://www.minobrnauki.gov.ru/</a>	<a href="https://www.minobrnauki.gov.ru/">https://www.minobrnauki.gov.ru/</a>
2	Общероссийский математический портал	<a href="http://www.mathnet.ru/">http://www.mathnet.ru/</a>	<a href="http://www.mathnet.ru/">http://www.mathnet.ru/</a>
3	Мировая цифровая библиотека	В <a href="http://wdl.org">http://wdl.org</a>	В <a href="http://wdl.org">http://wdl.org</a>
4	eLIBRARY.RU	<a href="http://www.elibrary.ru">www.elibrary.ru</a>	<a href="http://www.elibrary.ru">www.elibrary.ru</a>

### 6.2.3. Информационно-справочные системы

№ п/п		Адрес	Режим доступа
1	Научная электронная библиотека	<a href="http://elibrary.ru">http://elibrary.ru</a>	<a href="http://elibrary.ru">http://elibrary.ru</a>
2	Российская государственная библиотека	<a href="http://www.rsl.ru">http://www.rsl.ru</a>	<a href="http://www.rsl.ru">http://www.rsl.ru</a>
3	Международная реферативная база данных	<a href="http://www.zbmath.org">http://www.zbmath.org</a>	<a href="http://www.zbmath.org">http://www.zbmath.org</a>
4	Образовательный портал	<a href="http://www.uceba.com">http://www.uceba.com</a>	<a href="http://www.uceba.com">http://www.uceba.com</a>

6.2.4. Лицензионное и свободно распространяемое программное обеспечение дисциплины

№ п/п	Наименование программного обеспечения	Описание	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, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
3	Visual Studio 2019 Community	Средство для разработки ПО	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
4	.NET Framework	Платформа для разработки ПО	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
5	.NET Core	Платформа для разработки ПО	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
6	MySQL WorkBench 8.0 CE	Система управления базами данных	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.
7	Microsoft SQL Server Tools 18	Среда для доступа, настройки и администрирования СУБД	Свободная лицензия, тип (вид) лицензии - неискл. право, срок действия лицензии - бессрочно.

**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	персональный компьютер (26 шт.), интерактивная доска, мультимедийный проектор
2	Практические занятия	Учебная аудитория	персональный компьютер (26 шт.), интерактивная доска, мультимедийный проектор

## **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.

### Структура дисциплины по заочной форме обучения

Вид учебной работы	Всего часов	Курс
		5
<b>ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ</b>	108	108
КОНТАКТНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ С ПРЕПОДАВАТЕЛЕМ, в том числе:	25	25
Лекционные занятия (Лек)	12	12
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):	75	75
Подготовка к промежуточной аттестации в форме: (экзамен)	8	8
<b>ФОРМА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ</b>	Эк	Эк

## Лист регистрации изменений

Дополнения и изменения в рабочей программе дисциплины на 20\_\_ /20\_\_  
учебный год

В программу вносятся следующие изменения:

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Программа одобрена на заседании кафедры –разработчика «\_\_» \_\_\_\_\_ 20\_\_ г.,  
протокол № \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_ Торкунова Ю.В.

Программа одобрена методическим советом института \_\_\_\_\_  
«\_\_» \_\_\_\_\_ 20\_\_ г., протокол № \_\_\_\_\_

Зам. директора по УМР \_\_\_\_\_ / \_\_\_\_\_ /

*Подпись, дата*

Согласовано:

Руководитель ОПОП \_\_\_\_\_ / \_\_\_\_\_ /

*Подпись, дата*



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.

«24» июня 2021 г.

## ОЦЕНОЧНЫЕ МАТЕРИАЛЫ по дисциплине

Платформы разработки бизнес-приложений и языки программирования

Направление подготовки 09.03.03 Прикладная информатика

Направленность(профиль) Прикладная информатика в экономике и анализ данных

Квалификация

бакалавр

г. Казань, 2021



Оценочные материалы по дисциплине «Платформы разработки бизнес-приложений и языки программирования» - комплект контрольно-измерительных материалов, предназначенных для оценивания результатов обучения на соответствие индикаторам достижения компетенции:

ПК-2 Способен проводить тестирование компонентов программного обеспечения ИС

ПК-2.1 Применяет современные методики тестирования разрабатываемых информационных систем

ПК-2.2 Использует технологические и функциональные стандарты, современные модели и методы оценки качества и надежности при проектировании, конструировании и отладке программных средств

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: тест, практическая работа.

Промежуточная аттестация имеет целью определить уровень достижения запланированных результатов обучения по дисциплине за 8 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 8

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								
4	Создание библиотеки классов со статическими методами.	Пр	ПК-2.1, ПК-2.2	менее 4	5 - 5	6 - 6	6 - 7	
4	Индексаторы.	Пр	ПК-2.1, ПК-2.2	менее 4	5 - 5	6 - 6	7 - 8	
5	Создание иерархии пользовательских типов данных. Виртуальные методы.	Пр	ПК-2.1, ПК-2.2	менее 4	5 - 5	6 - 6	6 - 7	
5	Создание приложения согласно паттерну "делегирование-включение".	Пр	ПК-2.1, ПК-2.2	менее 6	6 - 6	7 - 7	8 - 8	

6	Организация клонирования (копирования) объектов, сравнения и перебора на основе применения стандартных интерфейсов IComparable, IComparer, IEnumerable, IEnumerator, ICloneable.	Пр	ПК-2.1, ПК-2.2	менее 4	5 - 5	6 - 6	7 - 7
7	Делегаты. Применение метода функциональной параметризации для реализации метода половинного деления отрезка с целью решения трансцендентных уравнений.	Пр	ПК-2.1, ПК-2.2	менее 4	5 - 5	6 - 6	7 - 8
7	Событийное программирование. События. Задача на регистрацию нарушений скоростного режима. Моделирование игры в ping-pong	Пр	ПК-2.1, ПК-2.2	менее 4	5 - 5	6 - 6	7 - 8
8	Применение операторов класса Linq для организации обработки данных пользовательских классов.	Пр	ПК-2.1, ПК-2.2	менее 6	6 - 6	6 - 6	7 - 7
	Промежуточная аттестация	Эк	ПК-2.1, ПК-2.2	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
Тест (Тест)	Тест из вопросов различной сложности	Тест из 40 вопросов
Практическая работа (Пр)	Практическая работа выполняется в соответствии с выбранной темой курсовой работы. Заключается в построении UML диаграмм. Отчет по практической работе оформляется отдельно каждым студентом.	Задание к практической работе
Экзамен (Эк)	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	Практическая работа 1. Класс математических функций Math. Следование. Условный оператор. Условная тернарная операция.
Представление и содержание оценочных материалов	<b>Задание 1.</b> С клавиатуры введено натуральное число, выражающее расстояние в миллиметрах. Вывести это расстояние на экран, выделив метры, сантиметры и миллиметры. <b>Задание 2.</b> Составить программу вычисления значений функций $z1$ и $z2$ , для которых значение аргумента введете с клавиатуры. Округлить полученный результат до тысячных и вывести на экран. Задание выполнить по вариантам.

$$1. \quad z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$$

$$z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right).$$

$$2. \quad z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha;$$

$$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right).$$

$$3. \quad z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha};$$

$$z_2 = 2 \sin \alpha.$$

$$4. \quad z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha};$$

$$z_2 = \operatorname{tg} 3\alpha.$$

$$5. \quad z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha;$$

$$z_2 = \cos^2 \alpha + \cos^4 \alpha.$$

$$6. \quad z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha;$$

$$z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5}{2} \alpha \cdot \cos 4\alpha$$

$$7. \quad z_1 = \cos^2\left(\frac{3}{8}\pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{\alpha}{4}\right);$$

$$z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}.$$

$$8. \quad z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1;$$

$$z_2 = \sin(y+x) \cdot \sin(y-x).$$

$$9. \quad z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2;$$

$$z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta).$$

$$10. \quad z_1 = \left( \sin\left(\frac{\pi}{2} + 3\alpha\right) \right) / (1 - \sin(3\alpha - \pi));$$

$$z_2 = \operatorname{ctg}\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right).$$

### **Задание 1 (тернарная условная операция)**

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

Варианты заданий

$$1. \quad y = \begin{cases} ax^2 + b \lg |2x|, & \sqrt{a+b} < x \\ \sqrt{a + \sin 2x}, & \sqrt{a+b} \geq x \end{cases} \quad 2.$$

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

$$3. \quad y = \begin{cases} e^{\sin x} + b, & 3b^2 > a \\ e^{-x} + a \lg x, & 3b^2 \leq a \end{cases} \quad 4.$$

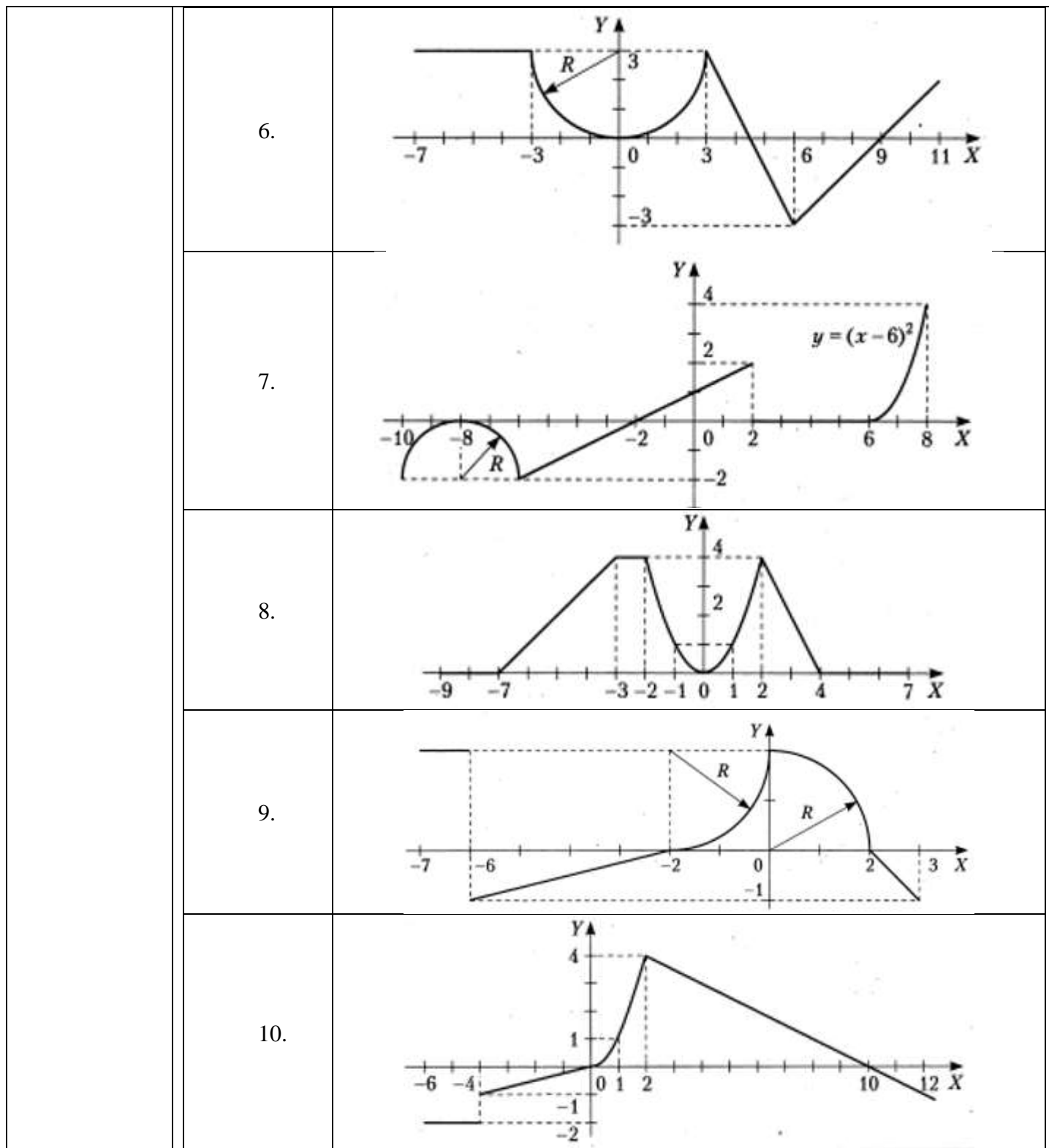
$$y = \begin{cases} a^{\frac{x}{2}} - e^{\cos x}, & |a^2 - b^2| > b \\ \operatorname{tg} 4x - a, & |a^2 - b^2| \leq b \end{cases}$$

	<p>5. <math display="block">y = \begin{cases} \ln  ax - b  - e^{\operatorname{tg} x}, &amp;  a - b^2  &gt; b \\ \operatorname{tg} 4x - a, &amp;  a - b^2  \leq b \end{cases}</math> 6.</p> <p><math display="block">y = \begin{cases} 2 \ln x - e^{\frac{ax-b}{10}}, &amp; 3a - b &gt; 0 \\ \arcsin\left(\frac{x}{10}\right), &amp; 3a - b \leq 0 \end{cases}</math></p> <p>7. <math display="block">y = \begin{cases} e^{-2x} + \sqrt[4]{a^4 + x}, &amp; ax &lt; b \\ \sin x - b^2, &amp; ax \geq b \end{cases}</math> 8.</p> <p><math display="block">y = \begin{cases} be^{-3x} + \sin^2 a, &amp; ab &lt; 2x \\ \frac{x^3}{a} - \frac{x}{b}, &amp; ab \geq 2x \end{cases}</math></p> <p>9. <math display="block">y = \begin{cases} ax^2 + \operatorname{tg} 4x, &amp; \sqrt{a+3b} &lt; 3x \\ \sqrt{a + \sin  3x }, &amp; \sqrt{a+3b} \geq 3x \end{cases}</math></p> <p>10. <math display="block">y = \begin{cases} \ln x^2 - \arcsin\left(\frac{x}{10}\right), &amp; 3a &gt; 2b \\ \operatorname{arctg}(2x - 0,6) + 2 \ln x, &amp; 3a \leq 2b \end{cases}</math></p> <p><b>Задание 2(условный оператор if)</b> Составить программу, которая запрашивает четырехзначное число и определяет, равна ли сумма двух первых его цифр сумме двух его последних цифр.</p> <p><b>Задание 3 (оператор выбора switch)</b> 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. <u>Справка:</u> 1996 г. — год крысы — начало очередного цикла.</p>
Критерии оценки и шкала оценивания в баллах	Максимальная оценка – 1 б. Минимальная оценка – 0,6 б.
<b>Наименование оценочного средства</b>	<b>Практическая работа 2.</b> Циклы. Оператор обработки исключений.
Представление и содержание оценочных материалов	<p><b>Задание 1. (оператор цикла while)</b> Вычислить и вывести на экран в виде таблицы значения функции, заданной графически, на интервале от <math>x_{\text{нач}}</math> до <math>x_{\text{кон}}</math> с шагом <math>dx</math>. Интервал и шаг задать таким образом, чтобы проверить все ветви программы. Таблицу снабдить шапкой.</p> <p>Примечание: Уравнение прямой на плоскости:</p> $y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$ <p>Уравнение окружности:</p>

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

$x_0, y_0$ - координаты центра окружности.

№ варианта	Задание
1.	
2.	
3.	
4.	
5.	



Критерии оценки и шкала оценивания в баллах

Максимальная оценка – 2 б.  
Минимальная оценка – 1,2 б.

Наименование оценочного средства

**Практическая работа 3.** Одномерные, двумерные, ступенчатые массивы.

Представление и содержание оценочных материалов

**Задание 1. Одномерные массивы.**

1.	<p>В одномерном массиве, состоящем из <math>n</math> целочисленных элементов, вычислить: Произведение элементов массива с четными номерами</p>
----	--

	<p>Сумму элементов массива, расположенных между первым и последним нулевыми элементами. Вывести массив на экран.</p>
2.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить: Сумму элементов массива с нечетными номерами Сумму элементов массива, расположенных между первым и последним отрицательными элементами. Вывести массив на экран.</p>
3.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить: Максимальный отрицательный элемент массива. Сумму элементов массива, расположенных до последнего положительного элемента массива. Вывести массив на экран.</p>
4.	<p>В одномерном массиве, состоящем из <math>n</math> целочисленных элементов, вычислить: Номер минимального положительного элемента массива. Среднее арифметическое элементов массива, расположенных между первым и последним отрицательными элементами. Вывести массив на экран.</p>
5.	<p>В одномерном массиве, состоящем из <math>n</math> целочисленных элементов, вычислить: Максимальный по модулю элемент массива. Среднее арифметическое элементов массива, расположенных между первым и вторым отрицательными элементами. Вывести массив на экран.</p>
6.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить: Количество элементов массива, лежащих в диапазоне от <math>A</math> до <math>B</math>. Сумму элементов массива, расположенных после максимального элемента. Вывести массив на экран.</p>
7.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить: Количество положительных элементов массива. Сумму элементов массива, расположенных до минимального элемента. Вывести массив на экран.</p>
8.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить: Количество элементов массива, меньших <math>C</math>. Сумму целых частей массива, расположенных после последнего отрицательного элемента. Вывести массив на экран.</p>
9.	<p>В одномерном массиве, состоящем из <math>n</math> целочисленных элементов, вычислить: Количество отрицательных элементов массива с четными номерами. Сумму элементов массива, расположенных до последнего нулевого элемента. Вывести массив на экран.</p>
10.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить: Количество элементов массива равных нулю. Сумму отрицательных нечетных элементов массива. 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 $8 \times 8$ 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 двумерный массив на экран. Поменять местами 1 и 2 строки матрицы. Вывести на экран полученную матрицу.
10.	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.

**Задание 3. Ступенчатый массив**

Задан ступенчатый массив следующей размерности:

- 0 строка 5 элементов,
- 1 строка 3 элемента,
- 2 строка 8 элементов,
- 3 строка 4 элемента,
- 4 строка 6 элементов.

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

Критерии  
оценки и

Максимальная оценка – 2 б.  
Минимальная оценка – 1,2 б.

шкала оценивания в баллах	
<b>Наименование оценочного средства</b>	<b>Практическая работа 4. Работа со списками.</b>
Представление и содержание оценочных материалов	<p><b>РАБОТА СО СПИСКАМИ</b></p> <ol style="list-style-type: none"> <li>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.</li> <li>2. Объявить и инициализировать второй список, содержащий 3 целочисленных элемента. Вывести на экран элементы второго списка.</li> <li>3. Вставить второй список в первый начиная с третьей позиции. Вывести на экран элементы первого списка.</li> <li>4. Вывести на экран количество элементов в первом списке.</li> <li>5. Вывести максимальный элемент первого списка.</li> <li>6. Вывести минимальный элемент первого списка.</li> <li>7. Скопировать элементы второго списка в массив. Вывести массив на экран.</li> <li>8. Удалить второй элемент во втором списке. Ввести элементы второго списка на экран.</li> </ol>
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 1 б.  Минимальная оценка – 0,6 б.</p>
<b>Наименование оценочного средства</b>	<b>Практическая работа 5. Работа со строками. Сохранение и чтение данных из текстового файла.</b>
Представление и содержание оценочных материалов	<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"> <li>— «язык программирования С#» - «С# программирования язык»</li> <li>— «язык программирования С#»- «#С яинавориммаргорп кызя»</li> </ul> <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 проводится замена каждой английской буквы на букву, следующую за ней К-ю по счету в алфавите (алфавит считается циклическим, то есть после буквой</p>

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><b>Вариант 1.</b> 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><b>Вариант 2.</b> Написать программу, которая считывает из текстового файла фамилию врачей, специальность. Вывести в другой текстовый файл специальности и подсчитанное количество врачей данной специальности.</p>
<p><b>Вариант 3.</b> Написать программу, которая считывает из текстового файла фамилию, дисциплину и оценку ученика. Вывести в другой текстовый файл дисциплины и средние баллы.</p>	<p><b>Вариант 4.</b> 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><b>Вариант 5.</b> 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><b>Вариант 6.</b> Написать программу, которая считывает из текстового файла фамилию автора, название произведения, жанр. Вывести в другой текстовый файл количество книг по определенным жанрам.</p>
<p><b>Вариант 7.</b> Написать программу, которая считывает из текстового файла названия фильмов,</p>	<p><b>Вариант 8.</b> Написать программу, которая считывает из текстового файла название</p>

	режиссера и год выпуска. Вывести в другой текстовый файл данные по фильмам отсортированные в порядке возрастания года выпуска.	спортивных клубов, вид спорта, адрес. Вывести в другой текстовый файл сгруппированные по виду спорта названия спортивных клубов и их адреса.																				
	<p><b>Вариант 9.</b> 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><b>Вариант 10.</b> 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>																				
Критерии оценки и шкала оценивания в баллах	Максимальная оценка – 4 б. Минимальная оценка – 2,4 б.																					
<b>Наименование оценочного средства</b>	<b>Практическая работа 6.</b> Разработка пользовательского типа данных. Методы. Переопределение операций.																					
Представление и содержание оценочных материалов	<p>Классы. Конструкторы. Методы. Свойства. Операторы.</p> <ol style="list-style-type: none"> <li>1. Определить пользовательский класс в соответствии с вариантом задания.</li> <li>2. Определить в классе следующие конструкторы: без параметров, с параметрами.</li> <li>3. Определить в классе функции для просмотра, изменения полей данных, сохранения данных об экземплярах класса в текстовый файл.</li> <li>4. В методе Main основной программы создать два экземпляра пользовательского класса и продемонстрировать работу конструкторов и методов.</li> <li>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.</li> <li>6. Перегрузить операции сложения («+») и сравнения («&lt;», «&gt;»).</li> </ol> <table border="1" data-bbox="316 1496 1501 2130"> <thead> <tr> <th>Вариант</th> <th>Класс</th> <th>Названия полей</th> <th>Свойства</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Книга</td> <td>Название Автор Количество страниц</td> <td>Название Автор Количество страниц (&gt;0)</td> </tr> <tr> <td>2</td> <td>Товар</td> <td>Наименование Шифр Количество</td> <td>Наименование Шифр Количество(&gt;0)</td> </tr> <tr> <td>3</td> <td>Бакалавр</td> <td>Фамилия Специальность Курс</td> <td>Фамилия Специальность Курс (от 1 до 4)</td> </tr> <tr> <td>4</td> <td>Автомобиль</td> <td>Марка Мощность Стоимость</td> <td>Марка Мощность (&gt;0) Стоимость(&gt;0)</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 Мощность Стоимость	Марка Мощность (>0) Стоимость(>0)
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 Мощность Стоимость	Марка 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	Максимальная оценка – 4 б. Минимальная оценка – 2,4 б.											
<b>Наименование оценочного средства</b>	<b>Практическая работа 7.</b> 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> <ul style="list-style-type: none"> <li>A. Заполнение массива случайными целыми числами в указанном диапазоне. Заполнение массива случайными вещественными числами в диапазоне [0,1].</li> <li>B. Вычисления суммы элементов массива (возвращает число).</li> <li>C. Вычисление произведения элементов массива (возвращает число).</li> <li>D. Возврат максимального элемента массива и его координат.</li> <li>E. Вывод массива на экран (возвращает строку соответствующего формата).</li> </ul> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Объявить</th> <th style="width: 50%;">Выполнить</th> </tr> </thead> <tbody> <tr> <td>1 массив - целочисленный</td> <td>A-E</td> </tr> <tr> <td>1 массив - вещественный</td> <td>A-E</td> </tr> <tr> <td>1 строковый массив</td> <td>D (максимальное по длине строку), E</td> </tr> </tbody> </table>				Объявить	Выполнить	1 массив - целочисленный	A-E	1 массив - вещественный	A-E	1 строковый массив	D (максимальное по длине строку), E
Объявить	Выполнить											
1 массив - целочисленный	A-E											
1 массив - вещественный	A-E											
1 строковый массив	D (максимальное по длине строку), E											
Критерии оценки и шкала оценивания в баллах	Максимальная оценка – 3 б. Минимальная оценка – 1,8 б.											
<b>Наименование оценочного средства</b>	<b>Практическая работа 8.</b> Индексаторы											
Представление и содержание оценочных материалов	<p>Описать класс множество (Set), содержащий следующие элементы:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%;">Поля</td> <td>1. Elements (целочисленный одномерный массив для хранения элементов множества). 2. Count (количество элементов в множестве).</td> </tr> <tr> <td>Конструктор</td> <td>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>Методы</td> <td>1. Fill () – присвоение элементам множества вводимых с клавиатуры значений.</td> </tr> </table>				Поля	1. Elements (целочисленный одномерный массив для хранения элементов множества). 2. Count (количество элементов в множестве).	Конструктор	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	1. Fill () – присвоение элементам множества вводимых с клавиатуры значений.		
Поля	1. Elements (целочисленный одномерный массив для хранения элементов множества). 2. Count (количество элементов в множестве).											
Конструктор	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	1. Fill () – присвоение элементам множества вводимых с клавиатуры значений.											

		<ol style="list-style-type: none"> <li>2. <code>IndexOf (int Value)</code> – возвращает индекс искомого в множестве значения или -1, если значение не было найдено.</li> <li>3. <code>ShowSet ()</code> – вывод элементов множества на экран.</li> <li>4. <code>Add (int NewElement)</code> – 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>)</li> </ol>	
	Перегрузка операций	<ol style="list-style-type: none"> <li>1. <code>++</code> – увеличение значений элементов множества на 1 <code>public static Set operator ++(Set set1)</code></li> <li>2. <code>+</code> – объединение множеств <code>public static Set operator +(Set set1, Set set2)</code></li> <li>3. <code>*</code> – пересечение множеств <code>public static Set operator *(Set set1, Set set2)</code></li> <li>4. <code>/</code> – разность множеств <code>public static Set operator /(Set set1, Set set2)</code></li> <li>5. <code>&lt;</code> и <code>&gt;</code> – сравнение количества элементов множеств <code>public static bool operator &lt;(Set set1, Set set2)</code> <code>public static bool operator &gt;(Set set1, Set set2)</code></li> </ol>	
	Индексатор	Для обращения к отдельным значениям поля <code>Elements</code> , являющегося одномерным массивом хранящим элементы множества.	
Критерии оценки и шкала оценивания в баллах	Максимальная оценка – 4 б. Минимальная оценка – 2,4 б.		
Наименование оценочного средства	<b>Самостоятельная работа 9.</b> Создание иерархии пользовательских типов данных. Виртуальные методы.		
Представление и содержание оценочных материалов	<p>Иерархия классов Примечание: выполнить лабораторную работу на основе имеющейся лаб.раб. № 7.</p> <ol style="list-style-type: none"> <li>7. Создать базовый класс согласно варианту задания.</li> <li>8. Добавить в базовый класс два конструктора: по умолчанию и с одним параметром.</li> <li>9. Добавить в базовый класс по одному общему в производных классах полю, свойству и методу, которые будут наследоваться.</li> <li>10. Определить иерархию классов. Для этого связать производные классы (классы наследники) с базовым классом отношением наследования (по вариантам).</li> <li>11. Добавить в производные классы по одному дополнительному полю и свойству (не считая наследуемых от базового класса полей и свойств).</li> <li>12. Добавить в производные класса конструкторы (по умолчанию и с параметрами).</li> <li>13. В производных классах переопределить наследуемый метод.</li> <li>14. Создать массив из четырех элементов (тип массива соответствует базовому классу), в который будут помещены все создаваемые экземпляры классов.</li> <li>15. Продемонстрировать работу методов и конструкторов.</li> </ol>		
	<b>Вариант</b>	<b>Базовый класс</b>	<b>Производные классы</b>

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 в баллах	Максимальная оценка – 3 б. Минимальная оценка – 1,8 б.
---	---

<b>Наименование оценочного средства</b>	<b>Самостоятельная работа 10.</b> 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"> <li>• добавления в коллекцию новых объектов,</li> <li>• изменения свойств и полей имеющихся объектов по номеру объекта в коллекции;</li> <li>• удаление из коллекции объектов по его номеру.</li> <li>• вывода информации, об имеющихся в коллекции объектах.</li> </ul> <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"> <li>- добавить в коллекцию объект (производного класса).</li> <li>- удалить объект из коллекции.</li> <li>- вывести сведения по имеющимся в коллекции объектам.</li> <li>- внести изменения в характеристики объекта коллекции.</li> <li>- выход из программы.</li> </ul>
---	---

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	Максимальная оценка – 5 б. Минимальная оценка – 3 б.				
<b>Наименование оценочного средства</b>	<b>Самостоятельная работа 11.</b> Организация клонирования (копирования) объектов, сравнения и перебора на основе применения стандартных интерфейсов IComparable, IComparer, IEnumerable, IEnumerator, ICloneable.				
Представление и содержание оценочных материалов	Реализовать интерфейсы ICloneable, IComparable, IComparer, IEnumerable. Продемонстрировать реализацию поверхностного и глубокого копирования объектов, вывода отсортированного списка объектов класса (с применением методов интерфейсов IComparable и IComparer), применения foreach.				
Критерии оценки и шкала оценивания в баллах	Максимальная оценка – 2 б. Минимальная оценка – 1,2 б.				
<b>Наименование оценочного средства</b>	<b>Самостоятельная работа 12.</b> Делегаты. Применение метода функциональной параметризации для реализации метода половинного деления отрезка с целью решения трансцендентных уравнений.				
Представление и содержание оценочных материалов	<p>1. На основе принципа функциональной параметризации создать универсальный метод решения уравнений <math>f(x) = 0</math>, 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>				
Критерии оценки и	Максимальная оценка – 4 б. Минимальная оценка – 2,4 б.				



шкала оценивания в баллах	
<b>Наименование оценочного средства</b>	<b>Самостоятельная работа 13.</b> Событийное программирование. События. Задача на регистрацию нарушений скоростного режима. Моделирование игры в ping-pong.
Представление и содержание оценочных материалов	<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> <p>1) Ping received Pong.  2) Pong received Ping.  3) Ping received Pong.  4) Pong received Ping.  5) Ping received Pong.</p> <p>По окончании игры выводится счет и имя игрока одержавшего победу.</p>
Критерии оценки и шкала оценивания в баллах	<p>Максимальная оценка – 4 б.  Минимальная оценка – 2,4 б.</p>

#### 4. Оценочные материалы промежуточной аттестации

Наименование оценочного средства	Тест
Представление и содержание оценочных материалов	<p>Вопрос 1  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 {     static void Main(string[] args)     { </pre>

```
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, intval)
```

```
{  
    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)
```

Вопрос 17

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");
```

Вопрос 18

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

Верно ли утверждение: "Величины ссылочного типа равны, если они ссылаются на одни и те же данные"?

Верно

Неверно

Вопрос 37

Верно ли утверждение: "Если в конструкторе производного класса явный вызов конструктора базового класса отсутствует, автоматически вызывается конструктор базового класса без параметров"?

Верно

Неверно

	<p>Вопрос 38 Верно ли утверждение: "Объекту базового класса можно присвоить объект производного класса"?</p> <p>Верно Неверно</p> <p>Вопрос 39 Верно ли утверждение: «Операция as выполняет преобразование к заданному типу, а если это невозможно, формирует результат null»?</p> <p>Верно Неверно</p> <p>Вопрос 40 Укажите название пользовательского типа данных, предназначенного только для порождения потомков?</p>
Критерии оценки и шкала оценивания в баллах	<p>При оценке тестовых заданий учитываются следующие критерии: 1 правильный ответ = 0,5 Баллы <b>Максимальное количество баллов - 20</b></p>
<b>Наименование оценочного средства</b>	<b>Экзаменационные вопросы</b>
Представление и содержание оценочных материалов	<ol style="list-style-type: none"> <li>1. Принципы ООП. Достоинства и недостатки ООП.</li> <li>2. Основные понятия ООП: класс, объект. Классы: описание, доступ к полям.</li> <li>3. Методы: определение, общая форма записи.</li> <li>4. Конструкторы: определение, общая форма, особенности.</li> <li>5. Инкапсуляция. Управление доступом к членам класса: спецификаторы доступа.</li> <li>6. Передача параметров в метод по значению.</li> <li>7. Передача параметров в метод по ссылке.</li> <li>8. Полиморфизм. Перегрузка методов: сущность, особенности реализации.</li> <li>9. Перегрузка конструкторов: сущность, особенности реализации.</li> <li>10. Свойства: назначение, общая форма.</li> <li>11. Наследование: сущность, описание производного класса.</li> <li>12. Операции класса: перегрузка унарных, бинарных операций. Операции преобразования типа.</li> <li>13. Наследование. Класс Object. Абстрактные и бесплодные классы.</li> <li>14. Интерфейсы. Применение стандартных интерфейсов .NET для сравнения, перебора, сортировки и клонирования объектов.</li> <li>15. Делегаты и события. Функциональная параметризация.</li> </ol>
Критерии оценки и шкала оценивания в баллах	<p>При оценке ответов на экзаменационный вопрос учитываются следующие критерии: Отвечает на основной вопрос правильно +2 дополнительно – 20 баллов. Отвечает на основной вопрос правильно +1 дополнительно – 15 баллов. Отвечает на основной вопрос правильно – 10 баллов. Отвечает не полно на основной вопрос правильно – 5 баллов. Не отвечает ни на 1 вопрос – 0 баллов. <b>Максимальное количество баллов - 20</b></p>