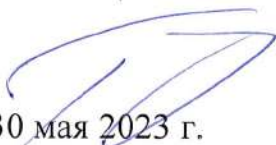




МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГУ»)

УТВЕРЖДАЮ

Директор Института цифровых
технологий и экономики



30 мая 2023 г.

Э.И. Беляев

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.13.06 Объектно-ориентированное программирование

Направление
подготовки

01.03.04 Прикладная математика

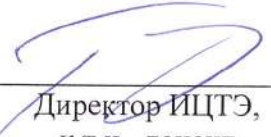
Квалификация

Бакалавр

г. Казань, 2023

Программу разработал(и):

Наименование кафедры	Должность, уч.степень, уч.звание	ФИО разработчика
Цифровые системы и модели	Доцент, к.т.н., доцент	Халидов А.А.

Согласование	Наименование подразделения	Дата	№ протокола	Подпись
Одобрена	Кафедра – разработчик ЦСМ	19.05.2023	5	 Зав. каф. ЦСМ к.ф.-м.н., доцент Смирнов Ю.Н.
Согласована	Выпускающая кафедра ЦСМ	19.05.2023	5	 Зав. каф. ЦСМ к.ф.-м.н., доцент Смирнов Ю.Н.
Согласована	Учебно- методический совет института ИЦТЭ	30.05.2023	7	 Директор ИЦТЭ, к.т.н., доцент Беляев Э.И.
Одобрена	Ученый совет института ИЦТЭ	30.05.2023	9	 Директор ИЦТЭ, к.т.н., доцент Беляев Э.И.

1. Цель, задачи и планируемые результаты обучения по дисциплине

Целью освоения дисциплины является изучение объектно-ориентированной технологии разработки программ.

Задачами дисциплины являются: формирование теоретических знаний и практических навыков разработки программ решения прикладных задач на компьютере с использованием объектно-ориентированной технологии.

Компетенции и индикаторы, формируемые у обучающихся:

Код и наименование компетенции	Код и наименование индикатора
ОПК-4: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-4.1: Способен разрабатывать алгоритмы для решения практических задач
	ОПК-4.2: Способен разрабатывать компьютерные программы, пригодные для практического применения

2. Место дисциплины в структуре ОП

Предшествующие дисциплины (модули), практики, НИР, др.:
Алгоритмизация и программирование.

Последующие дисциплины (модули), практики, НИР, др.:
Тестирование и оценка качества программного обеспечения, технологии разработки цифровых систем и моделей.

3. Структура и содержание дисциплины

3.1. Структура дисциплины

Для очной формы обучения

Вид учебной работы	Всего ЗЕ	Всего часов	Семестр
			3
ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ	6	216	216
КОНТАКТНАЯ РАБОТА*	-	89	89
АУДИТОРНАЯ РАБОТА	1,89	68	68
Лекции	0,94	34	34
Практические (семинарские) занятия	0	0	0
Лабораторные работы	0,94	34	34
САМОСТОЯТЕЛЬНАЯ РАБОТА ОБУЧАЮЩЕГОСЯ	4,11	148	148
Проработка учебного материала	3,11	112	112
Курсовой проект	0	0	0
Курсовая работа	0	0	0
Подготовка к промежуточной аттестации	1	36	36
Промежуточная аттестация:			-
			ЭКЗ

3.2. Содержание дисциплины, структурированное по разделам и видам занятий

Разделы дисциплины	Всего часов	Распределение трудоемкости по видам учебной работы				Формы и вид контроля	Индексы индикаторов формируемых компетенций
		лекции	лаб. раб.	пр. зан.	сам. раб.		
Раздел 1. Инкапсуляция.	42	8	8		26		ОПК-4.1, ЗУБ ОПК-4.2, ЗУБ
Раздел 2. Полиморфизм.	42	8	8		26	ТК1	ОПК-4.1, ЗУБ ОПК-4.2, ЗУБ
Раздел 3. Наследование.	42	8	8		26	ТК2	ОПК-4.1, ЗУБ ОПК-4.2, ЗУБ
Раздел 4. Интерфейсы. Делегаты. События.	54	10	10		34	ТК3	ОПК-4.1, ЗУБ ОПК-4.2, ЗУБ
Экзамен	36				36	ОМ	ОПК-4.1, ЗУБ ОПК-4.2, ЗУБ
Итого за 3 семестр	216	34	34		148		
ИТОГО	216	34	34		148		

3.3. Содержание дисциплины

Раздел 1. Инкапсуляция.

Тема 1.1. Классы.

Тема 1.2. Конструкторы.

Тема 1.3. Исключения

Тема 1.4. Статические члены класса.

Раздел 2. Полиморфизм.

Тема 2.1. Перегрузка методов.

Тема 2.2. Перегрузка конструкторов.

Тема 2.3. Перегрузка индексаторов.

Тема 2.4. Перегрузка операторов.

Раздел 3. Наследование.

Тема 3.1. Наследование классов.

Тема 3.2. Виртуальные методы.

Тема 3.3. Агрегация.

Тема 3.4. Абстрактный класс.

Раздел 4. Интерфейсы. Делегаты. События.

Тема 4.1. Интерфейсы.

Тема 4.2. Стандартные интерфейсы.

Тема 4.3. Делегаты.

Тема 4.4. Групповые делегаты.

Тема 4.5. События.

3.4. Тематический план практических занятий

Данный вид работы не предусмотрен учебным планом.

3.5. Тематический план лабораторных работ

Номер раздела дисциплины	Темы лабораторных работ	Трудоемкость, час.
1	Классы и объекты.	4
1	Методы. Конструкторы.	4
2	Перегрузка методов.	4
2	Перегрузка операторов.	4
3	Наследование.	4
3	Агрегация.	4
4	Интерфейсы.	4
4	События.	6
Всего		34

3.6. Курсовой проект /курсовая работа

Данный вид работы не предусмотрен учебным планом.

4. Оценивание результатов обучения

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

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции				
			Высокий	Средний	Ниже среднего	Низкий	
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54	
			Шкала оценивания				
			отлично	хорошо	удовлетворительно	неудовлетворительно	
			зачтено		не зачтено		
ОПК-4	ОПК-4.1	знать:					
		Порядок разработки алгоритмов для решения практических задач.	Твердо знает порядок разработок и алгоритмов.	Знает, но допускает неточности и при пояснении и деталей.	Слабо знает порядок разработок и алгоритмов.	Не знает порядок разработки алгоритмов.	

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции			
			Высокий	Средний	Ниже среднего	Низкий
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54
			Шкала оценивания			
			отлично	хорошо	удовлетворительно	неудовлетворительно
			зачтено			не зачтено
		уметь:				
		Разрабатывать алгоритмы для решения практических задач.	Свободно разрабатывает алгоритмы.	Наблюдается некоторая неуверенность	Умеет, но допускает ошибки в разработке.	Не умеет разрабатывать алгоритмы.
		владеть:				
		Техникой разработки алгоритмов.	Уверенно владеет техникой разработки	Владеет, но допускает неточности.	Слабо владеет разработкой алгоритмов.	Нет навыков разработки алгоритмов.
	ОПК-4.2	знать:				
		Методику разработки компьютерных программ, пригодных для практического применения.	Твердо знает методику разработки.	Знает, но допускает неточности и при пояснении деталей.	Слабо знает методику разработки.	Не знает методику разработки.
		уметь:				
		Разрабатывать компьютерные программы, пригодные для практического применения.	Свободно разрабатывает компьютерные программы.	Наблюдается некоторая неуверенность в разработке.	Умеет, но допускает ошибки в разработке.	Не умеет разрабатывать компьютерные программы.
		владеть:				
		Техникой разработки компьютерных программ.	Уверенно владеет техникой разработки компьютерных программ.	Владеет, но допускает неточности.	Слабо владеет разработкой компьютерных программ.	Нет навыков разработки компьютерных программ.

Оценочные материалы для проведения текущего контроля 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 разработчика.

5. Учебно-методическое и информационное обеспечение дисциплины

5.1. Учебно-методическое обеспечение

5.1.1. Основная литература

1. Теория и практика языков программирования : учебник / С. А. Орлов. - Санкт-Петербург : Питер, 2021. - 688 с. - URL: <https://ibooks.ru/bookshelf/377417>. - ISBN 978-5-496-00032-1. - Текст : электронный.

2. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие / Л. А. Залогова. - 3-е изд., стер. - Санкт-Петербург : Лань, 2021. - 192 с. - URL: <https://e.lanbook.com/book/176894>. - ISBN 978-5-8114-8481-2. - Текст : электронный.

3. Барков, И. А. Объектно-ориентированное программирование / И. А. Барков. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 700 с. — ISBN 978-5-507-47113-3. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/329549>.

5.1.2. Дополнительная литература

1. Вафин, Р. Р. Объектно-ориентированное программирование на C#.NET : учебно-методическое пособие / Р. Р. Вафин. — Казань : КНИТУ-КАИ, 2020. — 96 с. — ISBN 978-5-7579-2431-1. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/264920>.

2. Морозова, Ю. В. Практикум по объектно-ориентированному программированию : учебное пособие / Ю. В. Морозова. — Москва : ТУСУР, 2021. — 186 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/313631>.

3. Скворцова, Л. А. Объектно-ориентированное программирование на языке C++: Практикум : учебное пособие / Л. А. Скворцова, А. А. Бирюкова, К. В. Гусев. — Москва : РТУ МИРЭА, 2021. — 146 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/176540>.

БИБЛИОТЕКА
КГЭУ

5.2. Информационное обеспечение

5.2.1. Электронные и интернет-ресурсы

1. Полное руководство по языку программирования C# и платформе .NET. <https://metanit.com>.
2. Уроки по C# и платформе .NET. <https://professorweb.ru>.
3. Документация по .NET. <https://docs.microsoft.com/ru-ru/dotnet>.

5.2.2. Профессиональные базы данных / Информационно-справочные системы

1. Российская национальная библиотека, <http://nlr.ru/>;
2. Российская государственная библиотека, <http://www.rsl.ru/>;
3. Национальная электронная библиотека, <https://rusneb.ru/>;
4. Научная электронная библиотека, <http://www.elibrary.ru/>;
5. Единое окно доступа к образовательным ресурсам.
<http://window.edu.ru/>.

5.2.3. Лицензионное и свободно распространяемое программное обеспечение дисциплины

1. Visual Studio Community. Средство разработки ПО. Неискл. право.
2. .NET. Платформа для разработки ПО. Неискл. право.
3. SQL Server Express .Система управления базами данных. Неискл. право.
4. SQL Server Management Studio. Среда для доступа, настройки и администрирования СУБД. Неискл. право.
5. MySql Workbench 8.0 CE. Программа для визуального проектирования баз данных, интегрирующий проектирование, моделирование. Неискл.право
6. Windows 10. Пользовательская операционная система. Неискл. право
7. Office 365 ProPlus. Пакет программных продуктов содержащий в себе необходимые офисные программы. Неискл. право
8. 7-zip. Программа для архивирования файлов. Неискл.право
9. Git. Набор утилит осуществляющих отслеживание и фиксацию изменений в файлах. Неискл. право
10. LMS Moodle. ПО для эффективного онлайн-взаимодействия преподавателя и студента. Неискл. право

6. Материально-техническое обеспечение дисциплины

Наименование вида учебной работы	Наименование учебной аудитории, специализированной лаборатории	Перечень необходимого оборудования и технических средств обучения
Лекции	Учебная аудитория для проведения занятий лекционного типа	Специализированная учебная мебель, технические средства обучения, служащие для представления учебной информации большой аудитории (мультимедийный проектор, компьютер (ноутбук), экран),

		демонстрационное оборудование.
Лабораторные работы	Компьютерный класс с выходом в Интернет	Специализированная учебная мебель на 30 посадочных мест, 30 компьютеров, технические средства обучения (мультимедийный проектор, компьютер (ноутбук), экран), программное обеспечение.
	Компьютерный класс с выходом в Интернет В-600а	Специализированная учебная мебель на 30 посадочных мест, 30 компьютеров, технические средства обучения (мультимедийный проектор, компьютер (ноутбук), экран), видеокамеры, программное обеспечение
Самостоятельная работа	Компьютерный класс с выходом в Интернет В-600а	Специализированная учебная мебель на 30 посадочных мест, 30 компьютеров, технические средства обучения (мультимедийный проектор, компьютер (ноутбук), экран), видеокамеры, программное обеспечение
	Читальный зал библиотеки	Специализированная мебель, компьютерная техника с возможностью выхода в Интернет и обеспечением доступа в ЭИОС, экран, мультимедийный проектор, программное обеспечение

7. 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. 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.

8. 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 эстетической картины мира;

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

- 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.

Вносимые изменения и утверждения на новый учебный год

№ п/п	№ раздела внесения изменений	Дата внесения изменений	Содержание изменений	«Согласовано» Зав. каф. реализующей дисциплину	«Согласовано» председатель УМК института (факультета), в состав которого входит выпускающая
1	2	3	4	5	6
1					
2					
3					

*Приложение к рабочей
программе дисциплины*



КГУ

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КГУ»)**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
по дисциплине**

Б1.О.13.06 Объектно-ориентированное программирование

г. Казань, 2023

Оценочные материалы по дисциплине, предназначены для оценивания результатов обучения на соответствие индикаторам достижения компетенций.

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

1. Технологическая карта

Семестр 3.

[illegible]

2. Оценочные материалы текущего контроля и промежуточной аттестации

Шкала оценки результатов обучения по дисциплине:

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции				
			Высокий	Средний	Ниже среднего	Низкий	
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54	
			Шкала оценивания				
			отлично	хорошо	удовлетворительно	неудовлетворительно	
			зачтено		не зачтено		
ОПК-4	ОПК-4.1	знать:					
		Порядок разработки алгоритмов для решения практических задач.	Твердо знает порядок разработки и алгоритмов.	Знает, но допускает неточности и при пояснении деталей.	Слабо знает порядок разработки и алгоритмов.	Не знает порядок разработки алгоритмов.	
		уметь:					
		Разрабатывать алгоритмы для решения практических задач.	Свободно разрабатывает алгоритмы.	Наблюдается некоторая неуверенность в разработке.	Умеет, но допускает ошибки в разработке.	Не умеет разрабатывать алгоритмы.	
		владеть:					
		Техникой разработки алгоритмов.	Уверенно владеет техникой разработки и алгоритмов.	Владеет, но допускает неточности.	Слабо владеет разработкой и алгоритмов.	Нет навыков разработки алгоритмов.	

Код компетенции	Код индикатора компетенции	Запланированные результаты обучения по дисциплине	Уровень сформированности индикатора компетенции				
			Высокий	Средний	Ниже среднего	Низкий	
			от 85 до 100	от 70 до 84	от 55 до 69	от 0 до 54	
			Шкала оценивания				
			отлично	хорошо	удовлетворительно	неудовлетворительно	
			зачтено		не зачтено		
	ОПК-4.2	знать:					
		Методику разработки компьютерных программ, пригодных для практического применения.	Твердо знает методику разработки.	Знает, но допускает неточности при пояснении и деталей.	Слабо знает методику разработки.	Не знает методику разработки.	
		уметь:					
		Разрабатывать компьютерные программы, пригодные для практического применения.	Свободно разрабатывает компьютерные программы.	Наблюдается некоторая неуверенность в разработке.	Умеет, но допускает ошибки в разработке.	Не умеет разрабатывать компьютерные программы.	
		владеть:					
	Техникой разработки компьютерных программ.	Уверенно владеет техникой разработки и компьютерных программ.	Владеет, но допускает неточности.	Слабо владеет разработкой и компьютерных программ.	Нет навыков разработки компьютерных программ.		

Оценка **«отлично»** выставляется за выполнение (80-100)% тестовых заданий, самостоятельное выполнение лабораторных работ в полном объеме на занятии, правильные ответы на вопросы при защите отчетов о лабораторных работах.

Оценка **«хорошо»** выставляется за выполнение (60-79)% тестовых заданий, выполнение лабораторных работ отличается от требований на оценку «отлично» тем, что требуется некоторая помощь преподавателя и при ответах на вопросы допускаются ошибки.

Оценка **«удовлетворительно»** выставляется за выполнение (40-59)% тестовых заданий, выполнение лабораторных работ отличается от требований на оценку «хорошо» тем, что требуется периодическая помощь преподавателя,

при ответах на вопросы имеются ошибки, работа в ходе занятия выполнена не в полном объеме.

Оценка «**неудовлетворительно**» выставляется за выполнение менее 40% тестовых заданий, невыполнение лабораторной работы в ходе занятия.

3. 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	Описание оценочного средства
Отчет о лабораторной работе (ОЛР)	Выполнение лабораторной работы, обработка результатов испытаний, измерений, эксперимента. Оформление отчета, защита результатов лабораторной работы по отчету	Перечень заданий и вопросов для защиты лабораторной работы, перечень требований к отчету
Тест (Тест)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося	Комплект тестовых заданий

4. Перечень контрольных заданий или иные материалы, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций в процессе освоения дисциплины

Примеры заданий

Для текущего контроля ТК1:

Проверяемые компетенции: ОПК-4: ОПК-4.1, ОПК-4.2.

Тест

Вопрос	Варианты ответа
<i>Объектно-ориентированное программирование характеризуется</i>	<i>Наличием одной линейной программы</i>
	<i>Разделением программы на модули</i>
	<i>Все данные об объекте, его связи с другими объектами объединяются в одну структурную переменную</i>
	<i>Нет верного ответа</i>
<i>Метод это</i>	<i>Функция, получая в качестве обязательного параметра указатель на объект</i>
	<i>Структура, хранящая указатели this, parent, super</i>
	<i>Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии</i>
	<i>Определенный программистом абстрактный тип данных</i>
<i>Объект это</i>	<i>Функция, получая в качестве обязательного параметра указатель на объект</i>
	<i>Структура, хранящая указатели this, parent, super</i>
	<i>Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии</i>

	<i>Определенный программистом абстрактный тип данных</i>
<i>Класс это</i>	<i>Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии</i>
	<i>Определенный программистом абстрактный тип данных</i>
	<i>Переменная, описанная абстрактным типом данных</i>
	<i>Нет верного ответа</i>
<i>Когда данные являются не глобальными, доступными всей программе, а локальными доступными только малой части программы</i>	<i>При инкапсуляции</i>
	<i>При полиморфизме</i>
	<i>При объявлении метода</i>
	<i>Нет верного ответа</i>

Вопросы:

1. Основные принципы и понятия ООП.
2. Инкапсуляция: сущность, назначение, особенности реализации.
3. Классы: описание, доступ к членам класса, пример.
4. Методы: определение, общая форма записи, пример.
5. Конструкторы: определение, особенности, общая форма, пример.
6. Ключевое слово `this`.
7. Свойства: назначение, общая форма, пример.
8. Автоматически реализуемые свойства.
9. Типы значений и ссылочные типы.
10. Способы передачи параметров по значению, по ссылке.

Задания:

1. Объявить в программе закрытое поле – символьный массив.
2. Записать конструктор для создания объекта данного класса и инициализации массива символами из текстового файла.
3. Написать метод для вывода содержимого символьного массива на экран.
4. Написать метод для подсчета количества пробелов, а также гласных и согласных букв в символьном массиве.
5. Написать конструктор с одним параметром (`n`), создающий массив из `n` элементов и заполняющий его натуральным рядом чисел (`1, 2, ... n`).
6. Написать конструктор с одним параметром (имя файла с числами для заполнения массива), вычисляющий `n` - количество чисел в файле, создающий массив из `n` элементов и заполняющий его из файла с заданным именем.
7. Написать свойство, доступное только для чтения, для получения количества четных элементов массива.
8. Написать метод, выводящий содержимое массива на экран.
9. Написать метод, вычисляющий произведение элементов, расположенных после первого кратного 3 элемента.
10. Написать метод вывода полученных значений.

Для текущего контроля ТК2:

Проверяемые компетенции: ОПК-4: ОПК-4.1, ОПК-4.2.

Тест

Вопрос	Варианты ответа
<pre>class Base{ public virtual void Print() { Console.WriteLine("Hello "); }} class Derived : Base{ public override void Print() { Console.WriteLine("Bye "); }}class Program{ static void Main() { Base b = new Derived(); b.Print(); Derived d = new Derived(); d.Print(); Console.ReadKey(); }}</pre>	Что выведется в консоль? Вопрос открытого типа
<pre>public class Pattern { private static Pattern a; private Pattern() { } public static Pattern A { get { if (a == null) a = new Pattern(); return a; } } }</pre> Какому шаблону проектирования соответствует приведенный код?	Фабрика (Factory)
	Заместитель (Proxy)
	Одиночка (Singleton)
	Декоратор (Decorator)
Как называют переменную, объявленную внутри метода?	Статическая переменная
	Поле
	Свойство
	Локальная переменная
<pre>class Base { protected int Value { get; set; } = 0; } public class Derived : Base { private Derived() { Value = 1; } }</pre> Укажите причину, приводящую к ошибке компиляции.	К унаследованному свойству Value = 1; нужно обращаться через base
	Уровень доступа базового класса Base ниже, чем у производного
	Единственный конструктор Derived() должен быть публичным
	У автоматического свойства Value { get; set; } = 0; нельзя указывать инициализатор
<pre>public abstract class myClassBase { public virtual void Hello() { Console.WriteLine("Hello "); } } public class myClass : myClassBase { public override void Hello() { // base.Hello(); // 1 // this.Hello(); // 2 Console.WriteLine("World!"); } } class Program { static void Main() { myClassBase my = new myClass(); my.Hello(); Console.ReadKey(); } }</pre> Как получить в выводе консоли текст "Hello World!"?	Раскомментировать строку 1
	Раскомментировать строку 2
	Ничего не делать и так выводится "Hello World!"
	Невозможно, это не получится

Вопросы:

1. Обработка исключений.
2. Статические члены класса. Статический класс. Статический конструктор.
3. Полиморфизм: сущность, назначение, особенности реализации.
4. Перегрузка методов: сущность, особенности, пример.
5. Перегрузка конструкторов: сущность, назначение, пример.
6. Вызов перегруженного конструктора с помощью this. Последовательность выполнения конструкторов.
7. Перегрузка индексаторов.
8. Перегрузка двуместных операторов.

9. Перегрузка одноместных операторов.
10. Перегрузка операторов отношений.

Задания:

1. Написать метод для чтения из файла данных и запись их в массив, состоящий из десяти объектов.
2. Написать метод сортировки данных по возрастанию номера группы.
3. Написать метод вывода на экран фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0.
4. Случайным образом получить значения координат точки.
5. Создать объект родительского класса и применить к нему его методы.
6. Создать объект дочернего класса при первом повторении синего цвета, при втором повторении - красного цвета.
7. Применить к объекту дочернего класса унаследованный и переопределенный методы.
8. Конструктор, с двумя параметрами, присваивающий значения этим полям.
9. Написать метод для вывода координат точки на экран.
10. Написать метод для нахождения расстояния от данной точки до начала координат.

Для текущего контроля ТКЗ:

Проверяемые компетенции: ОПК-4: ОПК-4.1, ОПК-4.2.

Тест

<i>Вопрос</i>	<i>Варианты ответа</i>
<i>Как называется один из фундаментальных механизмов объектно-ориентированного программирования, позволяющий одному классу расширять функциональность другого, заимствуя при этом поля и методы?</i>	<i>Вопрос открытого типа</i>
<i>Как называется один из фундаментальных механизмов объектно-ориентированного программирования, позволяющий классу ограничивать доступ к своим членам?</i>	<i>Вопрос открытого типа</i>
<i>Как называется один из фундаментальных механизмов объектно-ориентированного программирования, позволяющий элементам с одинаковой спецификацией иметь различную реализацию?</i>	<i>Вопрос открытого типа</i>
<i>Как называется специализированный метод класса, предназначенный для уничтожения существующего экземпляра</i>	<i>Вопрос открытого типа</i>
<i>Какое ключевое слово используется для указания класса-родителя?</i>	<i>Вопрос открытого типа</i>

Вопросы:

1. Наследование: сущность, назначение, особенности реализации.
2. Доступ к членам базового класса из класса-наследника.
3. Виртуальные методы и их переопределение.
4. Переопределение свойств.
5. Абстрактный класс.

6. Основные понятия интерфейса. Множественная реализация интерфейсов.
7. Делегаты. Основные понятия.
8. Использование методов экземпляра в качестве делегатов.
9. Анонимные методы.
10. События.

Задания:

1. Объявить в программе закрытое поле – символьный массив.
2. Записать конструктор для создания объекта данного класса и инициализации массива символами из текстового файла.
3. Написать метод для вывода содержимого символьного массива на экран.
4. Написать метод для подсчета количества пробелов, а также гласных и согласных букв в символьном массиве.
5. Написать конструктор с одним параметром (n), создающий массив из n элементов и заполняющий его натуральным рядом чисел (1, 2, ... n).
6. Написать конструктор с одним параметром (имя файла с числами для заполнения массива), вычисляющий n - количество чисел в файле, создающий массив из n элементов и заполняющий его из файла с заданным именем.
7. Написать свойство, доступное только для чтения, для получения количества четных элементов массива.
8. Написать метод, выводящий содержимое массива на экран.
9. Написать метод, вычисляющий произведение элементов, расположенных после первого кратного 3 элемента.
10. Написать метод вывода полученных значений.

Для промежуточной аттестации:

Задание № 1

1. Классы: описание, доступ к членам класса, пример.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (n), создающий массив из n элементов и заполняющий его натуральным рядом чисел (1, 2, ... n), знак числа задается случайным образом. Свойство, доступное только для чтения, для получения количества четных элементов массива. Метод, выводящий содержимое массива в файл. Метод, вычисляющий произведение элементов массива, расположенных между первым и третьим положительными элементами.

Задание № 2

1. Типы значений и ссылочные типы. Способы передачи параметров в метод.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (имя файла с числами для заполнения массива), вычисляющий n – количество чисел в файле, создающий массив из n элементов и заполняющий его из файла с заданным именем. Свойство, доступное только для чтения, для получения суммы четных элементов массива.

Метод, выводящий содержимое массива на экран. Метод, вычисляющий сумму модулей элементов, расположенных после максимального элемента.

Задание № 3

1. Методы: определение, общая форма записи, пример.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (строка string с числами, числа в строке разделяются ровно одним пробелом), вычисляющий n – количество чисел в строке, создающий массив из n элементов и заполняющий его числами из строки. Свойство, доступное только для чтения, для получения количества нечетных элементов массива. Метод, выводящий содержимое массива на экран. Метод, вычисляющий произведение элементов, расположенных до последнего кратного трем элемента.

Задание № 4

1. Использование ключевых слов `ref` и `out` с параметрами.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (n), создающий массив из n элементов и заполняющий его однозначными случайными числами (от -9 до 9). Свойство, доступное только для чтения, для получения количества нечетных элементов массива. Метод, выводящий содержимое массива в файл. Метод, вычисляющий сумму элементов массива, расположенных между первым и четвертым положительными элементами.

Задание № 5

1. Конструкторы: определение, особенности, общая форма, пример. Ключевое слово `this`.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (имя файла с числами для заполнения массива), вычисляющий n – количество чисел в файле, создающий массив из n элементов и заполняющий его из файла с заданным именем. Свойство, доступное только для чтения, для получения суммы четных элементов массива. Метод, выводящий содержимое массива на экран. Метод, вычисляющий сумму модулей элементов, расположенных до (левее) максимального по модулю элемента.

Задание № 6

1. Статические члены класса. Статический конструктор. Статический класс.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с двумя параметрами (n и x), создающий массив из n элементов и заполняющий его случайными числами из интервала $(-x; x)$. Свойство, доступное только для чтения, для получения суммы нечетных элементов массива. Метод, выводящий содержимое массива на экран. Метод, вычисляющий произведение элементов, расположенных после первого кратного трем элемента.

Задание № 7

1. Свойства: назначение, общая форма, пример. Автоматически реализуемые свойства.
2. Составить программу, демонстрирующую все элементы класса. Члены

класса: Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются ровно одним пробелом), вычисляющий n – количество чисел в строке, создающий массив из n элементов и заполняющий его числами из строки. Свойство, доступное только для чтения, для получения суммы нечетных элементов массива. Метод, выводящий содержимое массива в файл. Метод, вычисляющий сумму модулей элементов, расположенных до (левее) максимального по модулю элемента.

Задание № 8

1. Индексаторы, назначение, общая форма, пример. Перегрузка индексаторов.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (n), создающий массив из n элементов и заполняющий его натуральным рядом чисел ($1, 2, \dots, n$), знак числа задается случайным образом. Свойство, доступное только для чтения, для получения суммы четных элементов массива. Метод, выводящий содержимое массива на экран. Метод, вычисляющий сумму модулей элементов, расположенных после максимального элемента.

Задание № 9

1. Конструкторы и наследование: сущность, пример. Порядок вызова конструкторов при наследовании.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с одним параметром (строка `string` с числами, числа в строке разделяются ровно одним пробелом), вычисляющий n – количество чисел в строке, создающий массив из n элементов и заполняющий его числами из строки. Свойство, доступное только для чтения, для получения количества положительных элементов массива. Метод, выводящий содержимое массива в файл. Метод, вычисляющий сумму элементов, расположенных до последнего кратного трем элемента.

Задание № 10

1. Перегрузка методов: сущность, особенности, пример.
2. Составить программу, демонстрирующую все элементы класса. Члены класса: Конструктор с тремя параметрами (a, b, x), заполняющий элементы массива по заданным начальному значению (a), конечному значению (b) и шагу (x). То есть такими числами: $a, a + x, \dots, b$, знак числа задается случайным образом. Свойство, доступное только для чтения, для получения количества положительных элементов массива. Метод, выводящий содержимое массива на экран. Метод, вычисляющий сумму элементов массива, расположенных между первым и последним положительными элементами.