



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО  
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

**(Институт технологий (филиал) ДГТУ в г. Волгодонске)**



**УТВЕРЖДАЮ**  
И.о. директора  
Н.М. Сидоркина  
«22» апреля 2024 г.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
(ОЦЕНОЧНЫЕ СРЕДСТВА)**  
для проведения текущего контроля и промежуточной аттестации  
по дисциплине  
**«Объектно-ориентированное программирование»**  
для обучающихся по направлению подготовки  
*09.03.02 Информационные системы и технологии*  
программа бакалавриата «Информационные системы»  
2024 года набора

## Лист согласования

Оценочные материалы (оценочные средства) по дисциплине \_\_\_\_\_ Объектно-ориентированное программирование  
(наименование)

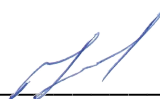
составлены в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности)

09.03.02 Информационные системы и технологии,  
(код направления (специальности), наименование)

Рассмотрены и одобрены на заседании кафедры «ТСиИТ» протокол № 9 от « 22» апреля 2024 г.

Разработчики оценочных материалов (оценочных средств)

Доцент

  
\_\_\_\_\_ Н.В. Кочковая  
подпись

Заведующий кафедрой

  
\_\_\_\_\_ Н.В. Кочковая  
подпись

—  
**Согласовано:**

Директор НПЦ  
«Микроэлектроника»

  
\_\_\_\_\_ С.Л. Бондаренко  
подпись

Начальник отдела ПО  
ООО «Топаз-сервис»

  
\_\_\_\_\_ Д.В. Чубукин  
подпись

## 1 1 Содержание

1 Паспорт оценочных материалов (оценочных средств)	4
1.1 Перечень компетенций, формируемых дисциплиной (модулем), с указанием этапов их формирования в процессе освоения ОПОП	4
1.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования	8
1.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, описание шкал оценивания	8
2 Контрольные задания (демоверсии) для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы	15
2.1 Задания для оценивания результатов обучения в виде знаний	15
2.2 Задания для оценивания результатов в виде владений и умений	17
2.3 Типовые экзаменационные материалы	20

## **1 Паспорт оценочных материалов (оценочных средств)**

Оценочные материалы (оценочные средства) прилагаются к рабочей программе дисциплины и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения обучающимися установленных результатов обучения.

Оценочные материалы (оценочные средства) используются при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся.

### **1.1 Перечень компетенций, формируемых дисциплиной, с указанием этапов их формирования в процессе освоения ОПОП**

Перечень компетенций, формируемых в процессе изучения дисциплины:

ПК-2: Способность проводить тестирование компонентов программного обеспечения ИС.

ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение.

Конечными результатами освоения дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение всего семестра по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (табл. 1).

Таблица 1 – Формирование компетенций в процессе изучения дисциплины

Код компетенции	Уровень	Дескрипторы компетенции(результаты обучения, показатели достижения результата обучения, которые обучающийся может продемонстрировать)	Вид учебных занятий, работы <sup>1</sup> , формы и методы обучения, способствующие формированию и развитию компетенции <sup>2</sup>	Контролируемые разделы и темы дисциплины <sup>3</sup>	Оценочные материалы (оценочные средства), используемые для оценки уровня сформированности компетенции	Критерии оценивания компетенций <sup>4</sup>
ПК-2	Знает виды угроз ИС и методы обеспечения информационной безопасности; средства проектирования и особенности администрирования БД в локальных и глобальных сетях ; средства, используемые в тестировании компонентов программного обеспечения ИС	Классификацию информационных систем, структуры, конфигурации информационных систем, общую характеристику процесса проектирования информационных систем, основные этапы, методологию, технологию и средства проектирования информационных систем; модели, методы, стандарты и инструменты интеграции при построении и сопровождении корпоративных информационных систем.	Лек, Пр., Ср лекция	1.1, 1.2, 1.3, 1.4., 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10	Контрольные вопросы	Ответы на контрольные вопросы; Выполнение практической работы и ее защита по контрольным вопросам в форме собеседования
	Умеет разрабатывать программные приложения и сервисы; обосновывать организационно-технические мероприятия по защите информации в ИС; использовать технологические и	Использовать архитектурные и детализированные решения при проектировании систем; проводить выбор исходных данных для проектирования	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа	

	функциональные стандарты, современные модели и методы при отладке программных средств; программировать и тестировать программы	Информационных систем, проводить сборку информационной системы из готовых компонентов, адаптировать приложения к изменяющимся условиям функционирования				
	Владеет навыками работы с инструментальными средствами проектирования баз данных и знаний, управления проектами ИС и защиты информации; использования функциональных и технологических стандартов ИС и сервисов	Моделями и средствами разработки архитектуры информационных систем; технологиями построения и сопровождения инфокоммуникационных систем и сетей.	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа	
ПК-5	Знает технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах; основы объектно-ориентированного подхода к программированию	Принципы моделирования, классификацию способов представления моделей систем; приемы, методы, способы формализации объектов, процессов, явлений и реализацию их на компьютере; достоинства и недостатки различных способов представления моделей систем; разработку алгоритмов фиксации и обработки результатов	Лек, Пр., Ср лекция	1.1, 1.2, 1.3, 1.4., 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10	Контрольные вопросы	Ответы на контрольные вопросы; Выполнение практической работы и ее защита по контрольным вопросам в форме собеседования

		моделирования систем; способы планирования машинных экспериментов с моделями.			
	Умеет работать с современными системами программирования, включая объектно-ориентированные; ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования	Использовать технологии моделирования; представлять модель в математическом и алгоритмическом виде	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа
	Владеет языками процедурного и объектноориентированного программирования, навыками разработки и отладки и тестирования программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня	Построением имитационных моделей информационных процессов; получением концептуальных моделей систем; построением моделирующих алгоритмов	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа

## **1.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

Оценивание результатов обучения по дисциплине осуществляется в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся.

По дисциплине «Объектно- ориентированное программирование» предусмотрены следующие виды контроля: текущий контроль (осуществление контроля всех видов аудиторной и внеаудиторной деятельности обучающегося с целью получения первичной информации о ходе усвоения отдельных элементов содержания дисциплины); промежуточная аттестация (оценивается уровень и качество подготовки по дисциплине в целом).

## **1.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы обучающихся. Текущий контроль служит для оценки объёма и уровня усвоения обучающимся учебного материала одного или нескольких разделов дисциплины (модуля) в соответствии с её рабочей программой и определяется результатами текущего контроля знаний обучающихся.

Текущий контроль осуществляется два раза в семестр по календарному графику учебного процесса.

Текущий контроль предполагает начисление баллов за выполнение различных видов работ. Результаты текущего контроля подводятся по шкале балльно-рейтинговой системы. Регламент балльно-рейтинговой системы определен Положением о системе «Контроль успеваемости и рейтинг обучающихся».

Текущий контроль является результатом оценки знаний, умений, навыков и приобретенных компетенций обучающихся по всему объёму учебной дисциплины, изученному в семестре, в котором стоит форма контроля в соответствии с учебным планом.

Текущий контроль успеваемости предусматривает оценивание хода освоения дисциплины: теоретических основ и практической части.

Промежуточная аттестация по дисциплине «Объектно-ориентированное программирование» проводится в форме курсового проекта и экзамена.

В табл. 2 приведено весовое распределение баллов и шкала оценивания по видам контрольных мероприятий.



Таблица 2 – Весовое распределение баллов и шкала оценивания по видам контрольных мероприятий

Текущий контроль (50 баллов <sup>1</sup> )				Промежуточная аттестация (50 баллов)	Итоговое количество баллов по результатам текущего контроля и промежуточной аттестации
Блок 1		Блок 2			
Лекционные занятия (X <sub>1</sub> )	Практические занятия (Y <sub>1</sub> )	Лекционные занятия (X <sub>2</sub> )	Практические занятия (Y <sub>2</sub> )	от 0 до 50 баллов	Менее 41 балла – неудовлетворительно; 41-60 баллов – удовлетворительно; 61-80 баллов – хорошо; 81-100 баллов – отлично
5	15	5	25		
Сумма баллов за 1 блок = X <sub>1</sub> + Y <sub>1</sub> = 20		Сумма баллов за 2 блок = X <sub>2</sub> + Y <sub>2</sub> = 30			

Для определения фактических оценок каждого показателя выставляются следующие баллы (табл.3):

Таблица 3– Распределение баллов по дисциплине

Вид учебных работ по дисциплине	Количество баллов	
	1 блок	2 блок
<i>Текущий контроль (50 баллов)</i>		
Выполнение практических работ в виде проведения исследования, оформления отчета	<b>20</b>	<b>20</b>
Защита практических работ в форме собеседования по контрольным вопросам	<b>5</b>	<b>5</b>
<i>Промежуточная аттестация (50 баллов)</i>		
<p>Экзамен по дисциплине «Объектно- ориентированное программирование» проводится в письменной форме в виде ответов на вопросы для промежуточной аттестации. Задание для экзамена состоит из 3 вопросов. Первый и второй вопрос позволяют проконтролировать знания обучающегося, третий – умения и навыки. Правильные ответы на первый и второй вопросы оцениваются в 15 баллов, третий – в 20 баллов. За неверно выполненное задание – 0 баллов.</p>		

<sup>1</sup> Вид занятий по дисциплине (лекционные, практические, лабораторные) определяется учебным планом. Количество столбцов таблицы корректируется в зависимости от видов занятий, предусмотренных учебным планом.

Распределение баллов по блокам, по каждому виду занятий в рамках дисциплины определяет преподаватель. Распределение баллов по дисциплине утверждается протоколом заседания кафедры. По заочной форме обучения мероприятия текущего контроля не предусмотрены.

<b>Сумма баллов по дисциплине 100 баллов</b>
--

Экзамен является формой итоговой оценки качества освоения обучающимся образовательной программы по дисциплине в целом или по разделу дисциплины. По результатам экзамена обучающемуся выставляется оценка «отлично», «хорошо», «удовлетворительно», или «неудовлетворительно».

Оценка «отлично» (81-100 баллов) выставляется обучающемуся, если:

- обучающийся набрал по текущему контролю необходимые и достаточные баллы для выставления оценки автоматом ;
- обучающийся знает, понимает основные положения дисциплины, демонстрирует умение применять их для выполнения задания, в котором нет явно указанных способов решения;
- обучающийся анализирует элементы, устанавливает связи между ними, сводит их в единую систему, способен выдвинуть идею, спроектировать и презентовать свой проект (решение);
- ответ обучающегося по теоретическому и практическому материалу, содержащемуся в вопросах экзаменационного билета, является полным, и удовлетворяет требованиям программы дисциплины;
- обучающийся продемонстрировал свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей дисциплины;
- на дополнительные вопросы преподавателя обучающийся дал правильные ответы.

Компетенция (и) или ее часть (и) сформированы на высоком уровне (уровень 3) (см. табл. 1).

Оценка «хорошо» (61-80 баллов) выставляется обучающемуся, если:

- обучающийся знает, понимает основные положения дисциплины, демонстрирует умение применять их для выполнения задания, в котором нет явно указанных способов решения; анализирует элементы, устанавливает связи между ними;
- ответ по теоретическому материалу, содержащемуся в вопросах экзаменационного билета, является полным, или частично полным и удовлетворяет требованиям программы, но не всегда дается точное, уверенное и аргументированное изложение материала;
- на дополнительные вопросы преподавателя обучающийся дал правильные ответы;
- обучающийся продемонстрировал владение терминологией соответствующей дисциплины.

Компетенция (и) или ее часть (и) сформированы на среднем уровне (уровень 2) (см. табл. 1).

Оценка «удовлетворительно» (41-60 баллов) выставляется обучающемуся, если:

- обучающийся знает и воспроизводит основные положения дисциплины в соответствии с заданием, применяет их для выполнения типового задания в котором очевиден способ решения;

- обучающийся продемонстрировал базовые знания важнейших разделов дисциплины и содержания лекционного курса;

- у обучающегося имеются затруднения в использовании научно-понятийного аппарата в терминологии курса;

- несмотря на недостаточность знаний, обучающийся имеется стремление логически четко построить ответ, что свидетельствует о возможности последующего обучения.

Компетенция (и) или ее часть (и) сформированы на базовом уровне (уровень 1) (см. табл. 1).

Оценка «неудовлетворительно» (менее 41 балла) выставляется обучающемуся, если:

- обучающийся имеет представление о содержании дисциплины, но не знает основные положения (темы, раздела, закона и т.д.), к которому относится задание, не способен выполнить задание с очевидным решением, не владеет навыками применения средств проектирования информационных систем и технологий.

- у обучающегося имеются существенные пробелы в знании основного материала по дисциплине;

- в процессе ответа по теоретическому материалу, содержащемуся в вопросах экзаменационного билета, допущены принципиальные ошибки при изложении материала.

Компетенция(и) или ее часть (и) не сформированы.

После выполнения курсового проекта студент сдает его на кафедру для проверки, затем назначается дата и время защиты. На защите автор кратко сообщает о результатах выполнения проекта и отвечает на дополнительные вопросы. Комиссия в составе руководителя и членов кафедры оценивает курсовой проект.

Оценка «отлично» ставится студенту, который в срок, в полном объеме и на высоком уровне выполнил курсовой проект. При защите и написании работы студент продемонстрировал вышеперечисленные навыки и умения. Тема, заявленная в работе раскрыта полностью, все выводы студента подтверждены материалами исследования и расчетами. Отчет подготовлен в соответствии с предъявляемыми требованиями. Отзыв руководителя положительный.

Оценка «хорошо» ставится студенту, который выполнил курсовой проект, но с незначительными замечаниями, был менее самостоятелен и инициативен. Тема работы раскрыта, но выводы носят поверхностный характер, практические материалы обработаны не полностью. Отзыв руководителя положительный.

Оценка «удовлетворительно» ставится студенту, который допускал просчеты и ошибки в работе, не полностью раскрыл заявленную тему, делал поверхностные выводы, слабо продемонстрировал аналитические способности и навыки работы с теоретическими источниками. Отзыв руководителя с замечаниями.

Оценка «неудовлетворительно» ставится студенту, который не выполнил курсовой проект, либо выполнил с грубыми нарушениями требований, не раскрыл заявленную тему, не выполнил практической части работы.

Оценка проставляется в зачетной книжке студента и в ведомости. При неудовлетворительной оценке автор проекта устраняет недочеты или получает новое задание.

Критерии оценки:

Критерий	Показатель	Максимальное количество баллов
1 Сроки, качество доклада и защиты курсового проекта	<ul style="list-style-type: none"> <li>- соответствие содержания теме проекта;</li> <li>- полнота и глубина раскрытия основных понятий проблемы;</li> <li>- умение работать с литературой, систематизировать и структурировать материал;</li> <li>- умение обобщать, сопоставлять различные точки зрения по рассматриваемому вопросу, аргументировать основные положения и выводы;</li> <li>- умение защищать результаты расчетов.</li> </ul>	10
2 Соблюдение требований по оформлению пояснительной записки	<ul style="list-style-type: none"> <li>- правильное оформление пояснительной записки, ссылок на используемые литературные источники;</li> <li>- соблюдение требований к объему проекта.</li> </ul>	5
Соблюдение требований по оформлению графической части проекта	<ul style="list-style-type: none"> <li>- чертежи выполнены в соответствии с требованиями ЕСКД;</li> <li>- количество соответствует требованиям задания.</li> </ul>	5

Тематика курсовых проектов определяется преподавателем кафедры, осуществляющим руководство курсовой работы. Студент выбирает тему проекта в соответствии со своими интересами, о чем лично сообщает преподавателю. В ходе предварительного обсуждения выбранной темы с преподавателем она может быть изменена по согласованию между

преподавателем и студентом. Выбор должен быть сделан в течение первых двух недель семестра текущего учебного года.

### **Примерная тематика курсовых проектов:**

1. Разработать модель работы салона по прокату транспортных средств (автомобили, мотоциклы). Спроектировать иерархию классов: Транспортное средство, Автомобиль, Мотоцикл.

При разработке модели предусмотреть выполнение следующих задач (информация по задачам выводится в текстовый файл):

a. Создать текстовый файл, содержащий информацию о транспортных средствах (не менее 20 объектов).

b. Обеспечить вывод информации по имеющимся транспортным средствам: Марка, Год выпуска, Цвет, Номер, Пробег за время аренды (в км), Цена аренды (за 1сутки– для автомобиля, за 1 час – для мотоцикла).

c. Реализовать модель работы салона в течении 1 месяца (30 дней). Информацию о работе салона вывести в текстовый файл.

d. Определить сколько раз были сданы в прокат транспортные средства в указанный период времени по каждому виду (автомобиль, мотоцикл).

e. Определить выручку предприятия за месяц. При оплате проката учесть, что в случае превышения времени проката клиент платит штраф: удвоенная сумма стоимости за каждый лишний день (или час для мотоцикла) эксплуатации. Предусмотреть возможность определения размера штрафа.

f. Определить транспортное средство, с наибольшим пробегом по итогам месяца.

2. Разработать модель работы станции ТО транспортных средств (ТС).

Спроектировать иерархию классов: Транспортное средство – Заявка.

При разработке модели предусмотреть выполнение следующих задач (информация по задачам выводится в текстовый файл):

a. Создать текстовые файлы, содержащие:

информацию о клиентах и их ТС (не менее 20 объектов);

список механиков СТО (не менее 5).

□ Обеспечить вывод информации по имеющимся заявкам: Номер, Дата поступления, транспортное средство, Ф.И.О. клиента, Ф.И.О. механика, сумма ТО, дата выполнения заявки, Позиция заявки (в обработке, закрыта, отказ).

в. Реализовать модель работы СТО в течении 1 месяца (30 дней). Информацию о работе предприятия вывести в текстовый файл.

с. Определить три наиболее часто обсуживаемые марки ТС.

д. Определить выручку предприятия за месяц.

е. Определить Ф.И.О. механика, обслужившего наибольшее число ТС по итогам месяца.

#### 4. Структура курсового проекта

Курсовой проект начинается с титульного листа стандартной формы, состоящего из трёх частей (Приложение 1), за которым следует лист с оглавлением работы и состоит из введения, разделов, заключения, списка использованной литературы и Интернет-источников, приложений.

*Введение* содержит общий обзор работы, цель и задачи работы, позволяющий составить общее представление об исследуемой проблеме и полученных результатах. Во введении также может быть предложена краткая аннотация отдельных разделов работы.

В первом разделе, который может быть назван, например, аналитической частью, анализом предметной области и т.п., следует дать характеристику предметной области, для которой решается задача, описать используемые в ней предметные технологии, обосновать необходимость их автоматизации с применением вычислительной техники, сделать постановку задачи.

Во втором разделе, который может быть назван проектной частью, следует привести проектные решения задачи, поставленной в предыдущем разделе: дать подробное описание информационного, программного и технологического обеспечения разработанной системы с использованием иллюстрационного материала.

Таким образом, структура содержания курсового проекта может быть следующей:

Введение

##### *1. Аналитическая часть*

1.1. Анализ предметной области 1.2. Постановка цели и задач выполняемой работы 1.3. Метод реализации процесса проектирования 1.4. Обоснование проектных решений

## 2. Проектная часть

2.1. Информационное обеспечение информационной системы

2.2. Программное обеспечение информационной системы

2.3. Технологическое обеспечение информационной системы

**2 Контрольные задания (демоверсии) для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

### 2.1 Задания для оценивания результатов обучения в виде знаний

Контроль знаний по дисциплине «Объектно-ориентированное программирование» осуществляется посредством тестовых заданий.

База тестовых вопросов по материалу курса

1. Какой диапазон кодов символов используется в кодировке ASCII?

От 0 до 127.

От 0 до 255.

От 0 до 65535.

2. Пусть значения целочисленных переменных  $x$  и  $y$  равны 20 и 10 соответственно. Указать значение логического выражения

$y \neq 0$  и  $x/y \leq 1$

Ложь.

Истина.

3. Пусть  $x$  и  $y$  — вещественные переменные типа `double`. Может ли произойти прерывание из-за деления на ноль при вычислении логического выражения

$y > 0.1$  и  $x / y \geq 1.0$ ?

Может.

Не может.

4. Указать, что произойдет с элементами массива  $a$  в результате выполнения следующего фрагмента программы:

```
вещ  $a[100]$ ; цел  $i$ ;
```

```
...
```

```
 $i := 0$ ;
```

```
цикл пока  $i < 99$ 
```

```
|  $a[i+1] := a[i]$ ;
```

```
|  $i := i+1$ ;
```

```
конец цикла
```

```
 $a[0] := a[99]$ ;
```

Элементы массива циклически сдвинутся вправо.

Все элементы массива станут равными элементу с индексом 0.

5. В каком алгоритмическом языке текстовая строка представляется последовательностью байтов, в которой первый байт содержит длину строки, а далее следуют коды символов, составляющих строку?

В языке Си.

В языке Паскаль.

6. В каком алгоритмическом языке — в Паскале или в Си — операция поиска конкретного символа в строке реализуется более эффективно?

В Паскале.

В Си.

Нет существенной разницы.

7. В какой кодировке под символ отводится 2 байта?

В кодировке ASCII.

В кодировке UNICODE.

8. Пусть значения целочисленных переменных  $x$  и  $y$  равны 1 и 2 соответственно. Указать значение логического выражения

$(x \geq 1 \text{ и } y < 0)$  или  $(x \leq 1 \text{ и } y > 0)$

Ложь.

Истина.



9. Какой из перечисленных подходов к программированию является наиболее поздним (зрелым)?

структурный

объектно-ориентированный

компонентно-ориентированный

10. Что отличает императивные языки программирования от декларативных?

степень зависимости от среды реализации

стиль программирования

структура программы

## 2.2 Задания для оценивания результатов в виде владений и умений

Контроль умений и навыков по дисциплине «Объектно-ориентированное программирование» осуществляется посредством подготовки и выполнения практических работ и экзамена.

Примерная тематика практических работ:

1. Язык программирования C#.
2. Процедурное программирование. Функции
3. Реализация базовых и дополнительных алгоритмических структур.
4. Алгоритмы поиска.
5. Алгоритмы сортировки
6. Объектно-ориентированное программирование
7. Разработка пользовательского интерфейса

Типовой пример практического задания по дисциплине «Объектно-ориентированное программирование».

### **Объектно-ориентированное программирование в MS VS C#**

**Цель работы:** Изучить создание классов на языке C#. Ознакомиться с основными свойствами классов, изучить структуру их построения.

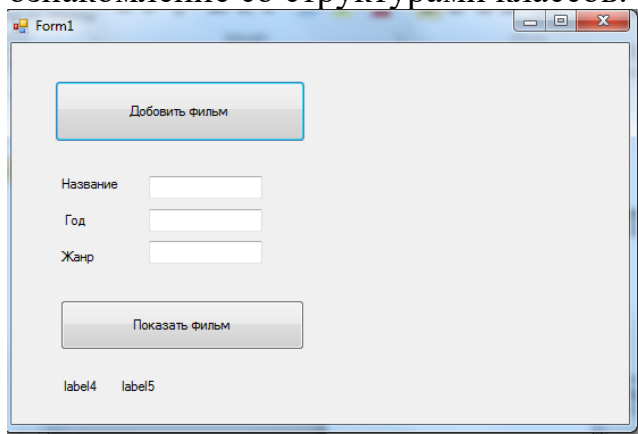
**Задание:** Создать класс по произвольной тематике (задается индивидуально преподавателем). На его основе создать структуру, по которой к нему можно будет обращаться. Добавить метод, который будет возвращать что либо, и метод, не возвращающий значения.

**Ход работы:**

1. Создаем пустой проект с шаблоном Windows Forms.
2. Создаем класс под названием student. Присвоим ему значения, создадим два метода.

```
public class student//название класса
{
    public string fam_name;//фамилия
    public string name;//имя
    public string o_name;//отчество
    public string fullname;//строка для метода
    public int num;//номер для метода
    public void Plus()//метод, увеличивающий номер на 1
    {
        num++;
    }
    public string Getfullname()//метод, возвращающий полное имя
    {
        fullname = name + " " + year + " " + o_name;
        return fullname;
    }
}
```

3. Далее создаем интерфейс программы. Разместим на форме поля ввода значений, поле вывода результата, текстовые сообщения и исполняющую кнопку. Отметим, что данная программа направлена исключительно на ознакомление со структурами классов. Сохраняем получившийся результат.



4. Мы получили интерфейс программы. Однако для его работоспособности необходимо прописать код. Просматриваем получившийся код программы:

```
namespace John_project__4
{
    public partial class Form1 : Form
    {
        public class student
        {
            public string name;
            public string year;
```

```

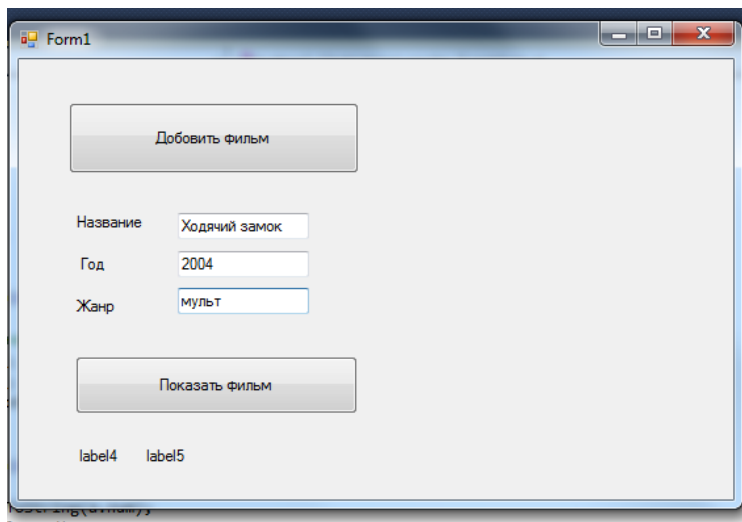
public string o_name;
public string fullname;
public int num;
public void Plus()
{
    num++;
}
public string Getfullname()
{
    fullname = name + " " + year + " " + o_name;
    return fullname;
}
}
student a = new student();
public Form1()
{
    InitializeComponent();
}

private void button1_Click(object sender, EventArgs e)
{
    a.Plus();//присвоение номера в классе
    a.name = textBox1.Text;
    a.year = textBox2.Text;
    a.o_name = textBox3.Text;
}

private void button2_Click(object sender, EventArgs e)
{
    Label4.Text = Convert.ToString(a.num);
    Label5.Text = a.Getfullname();
}
}
}

```

5. Проводим отладку программы. Проверим выполнение класса.



## 2.3 Типовые экзаменационные материалы

### Перечень примерных вопросов к экзамену

1. Какие парадигмы программирования Вы знаете?
2. Сформулируйте принципы логического программирования.
3. Какие языки программирования поддерживают логическое программирование?
4. На чем основано функциональное программирование? Какие языки его поддерживают?
5. В чем заключаются принципы структурного проектирования и программирования?
6. Сформулируйте принципы модульного программирования.
7. Дайте определение объекта, класса, экземпляра.
8. Сформулируйте принципы объектно-ориентированного подхода.
9. Что такое наследование? Какие виды наследования Вы знаете?
10. В чем заключаются инкапсуляция и полиморфизм?
11. Приведите примеры объектно-ориентированных языков.
12. Дайте определение компонентных данных и компонентных функций.
13. Сформулируйте принципы доступности компонент класса.
14. Дайте определение конструкторов и деструкторов класса.
15. Как и с какой целью осуществляется перегрузка операций?
16. Дайте определение виртуальных функций и классов. Когда они применяются?
17. Как осуществляется ввод-вывод в языке C#?
18. Перечислите основные особенности программирования в оконных средах.
19. Что такое среда программирования?
20. Опишите среду разработки C# под Windows.
21. Как происходит отладка и тестирование программ в среде C# под Windows?
22. Дайте определение технологии визуального программирования?
23. Что такое GDI и MDI?
24. Что такое окно диалога? Как его создать средствами C#?

25.Опишите объекты Windows.

26.Что такое событие и как его обработать?

Критерий оценки:

Полнота ответа на поставленный вопрос, умение использовать термины, формулы, приводить примеры, делать выводы и анализировать конкретные ситуации.

Шкала оценивания

Максимальное количество баллов, которое обучающийся может получить за промежуточную аттестацию (зачет) составляет 100 баллов.

Менее 41 балла –  
неудовлетворительно;  
41-60 баллов –  
удовлетворительно;  
61-80 баллов –  
хорошо;  
81-100 баллов –  
Отлично

**Задачи для экзамена по курсу «Объектно- ориентированное программирование»**

**Утверждено**

**Протокол № 1 от 28.08.17**

1. Чему равно значение целочисленной переменной  $x$  в результате выполнения приведенного ниже фрагмента программы?

$x := 64;$

цикл пока  $x * x > 100$

|  $x := -(x / 2);$

конец цикла

Значение  $x = 16.$

Значение  $x = 8.$

Значение  $x = -8.$

Значение  $x = 4.$

2. Завершится ли когда-нибудь выполнение цикла в приведенном ниже фрагменте программы?

$x := 1;$

цикл пока  $x \neq 144$

|  $x := x * 13;$

| если  $x \leq 299$

|| то  $x := x - 299;$

| конец если

конец цикла

Завершится.

Не завершится.

3. Содержимое двухбайтового слова можно интерпретировать либо как неотрицательное целое число в диапазоне 0...65535, либо как число со знаком в диапазоне -32768...32767. Какое число со знаком имеет тот же двоичный код, что и неотрицательное число 65533?

Число -1.

Число -2.

Число -3.

4. Целочисленная переменная  $x$  представляет короткое целое число со знаком в диапазоне -128...127 и занимает 1 байт. Чему равно значение  $x$  после выполнения приведенного ниже фрагмента программы?

```
x := 30;
```

```
x := x * 5;
```

Значение  $x$  равно -106.

Значение  $x$  равно -22.

Значение  $x$  равно 22.

5. Сколько двоичных разрядов отводится для хранения порядка в двоичном коде вещественного числа типа `double` длиной 8 байтов?

8 разрядов.

11 разрядов.

6. Всегда ли равны выражения

$$(x + y) + z, \quad x + (y + z)$$

для произвольных вещественных переменных  $x$ ,  $y$ ,  $z$  типа `double`?

Да, всегда равны.

Нет, могут быть неравными.

7. Какой диапазон кодов символов используется в кодировке ASCII?

От 0 до 127.

От 0 до 255.

От 0 до 65535.

8. Пусть значения целочисленных переменных  $x$  и  $y$  равны 20 и 10 соответственно. Указать значение логического выражения

$$y \neq 0 \text{ и } x/y \leq 1$$

Ложь.

Истина.

9. Пусть  $x$  и  $y$  — вещественные переменные типа `double`. Может ли произойти прерывание из-за деления на ноль при вычислении логического выражения

$y > 0.1$  и  $x / y \geq 1.0$ ?

Может.

Не может.

10. Указать, что произойдет с элементами массива  $a$  в результате выполнения следующего фрагмента программы:

вещ  $a[100]$ ; цел  $i$ ;

...

$i := 0$ ;

цикл пока  $i < 99$

|  $a[i+1] := a[i]$ ;

|  $i := i+1$ ;

конец цикла

$a[0] := a[99]$ ;

Элементы массива циклически сдвинутся вправо.

Все элементы массива станут равными элементу с индексом 0.

Структура экзаменационного билета

Экзаменационный билет включает в себя 2 теоретических вопроса и 1 практическое задание.

Пример экзаменационного билета



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО  
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

**(Институт технологий (филиал) ДГТУ в г. Волгодонске)**

Факультет «Технологии и менеджмент»

Кафедра «Технический сервис и информационные технологии»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1\_\_**

на 2022/ 2023 учебный год

Дисциплина Объектно-ориентированное программирование

1. Объектно-ориентированная декомпозиция \_\_\_\_\_
2. Языки программирования. Общие понятия \_\_\_\_\_
3. Задача \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_ Н.В. Кочковая

15.12.2023

подпись

дата

АКТУАЛЬНО НА

20\_\_/20\_\_уч.год

подпись

Ф.И.О. зав. каф.

20\_\_/20\_\_уч.год

подпись

Ф.И.О. зав. каф.

20\_\_/20\_\_уч.год

подпись

Ф.И.О. зав. каф.

20\_\_/20\_\_уч.год

подпись

Ф.И.О. зав. каф.

### Карта тестовых заданий

ПК-2: Способность проводить тестирование компонентов программного обеспечения ИС.

Дисциплина: Объектно-ориентированное программирование

#### Описание теста:

1. Тест состоит из 70 заданий, которые проверяют уровень освоения компетенций обучающегося. При тестировании каждому обучающемуся предлагается 30 тестовых заданий по 15 открытого и закрытого типов разных уровней сложности.

2. За правильный ответ тестового задания обучающийся получает 1 условный балл, за неправильный ответ – 0 баллов. По окончании тестирования, система автоматически определяет «заработанный итоговый балл» по тесту, согласно критериям оценки

3 Максимальная общая сумма баллов за все правильные ответы составляет – 100 баллов.

4. Тест успешно пройден, если обучающийся правильно ответил на 70% тестовых заданий (61 балл).

5. На прохождение тестирования, включая организационный момент, обучающимся отводится не более 45 минут. На каждое тестовое задание в среднем по 1,5 минуты.

6. Обучающемуся предоставляется одна попытка для прохождения компьютерного тестирования.

**Кодификатором** теста по дисциплине является раздел рабочей программы «4. Структура и содержание дисциплины (модуля)»

#### Задания закрытого типа

#### Задания альтернативного выбора

*Выберите один правильный ответ*

#### Простые (1 уровень)

1. Ключевое слово `base` используется в `C#` для \_\_\_\_\_.

**А) вызова методов базового класса из переопределенных методов производного класса**

Б) вызова методов производного класса

В) описания родительского класса

Г) описания конструктора

2. Создание текстов программ на языках программирования - это:

А) Аттестация программного средства



**Б) Кодирование программного средства**

В) Конструирование программного средства

3. В коллекции \_\_\_\_\_ на языке C# реализуется принцип FIFO.

**А) Queue**

Б) Stack

В) List

Г) LinkedList

4. Метод \_\_\_\_\_ может генерировать исключение (язык C#).

**А) int.Parse**

Б) int.TryParse

В) Math.Sin

5. Способность программного средства решать нужный набор задач - \_\_\_\_\_.

А) Точность

Б) Функциональная пригодность

В) Соответствие стандартам и правилам

Г) Способность к взаимодействию

6. начальный проект приложения типа WindowsForms на языке C# по умолчанию содержит \_\_\_\_\_.

**А) Класс Form1**

Б) Класс Window

В) пространство имен Form1

Г) пространство имен Form

**Средне-сложные (2 уровень)**

7. В объектно-ориентированном программировании на C# разрешено множественное наследование \_\_\_\_\_.

**А) от интерфейсов**

Б) от классов

В) от конструкторов

Г) от виртуальных классов

8. Совокупность процессов, которая отражает различные состояния программного средства, начиная с момента принятия решения о необходимости создания этого программного средства и заканчивая его полным изъятием из эксплуатации - это \_\_\_\_\_.

А) Рекурсия программного средства

Б) Процесс разработки программного средства

**В) Жизненный цикл программного средства**

Г) Эксплуатационный период программного средства

9. Императивное программирование - \_\_\_\_\_

**А) программирование вычислительной машины путем описания последовательности инструкций, выполняемых одна за другой; программист однозначно определяет последовательность шагов, которые необходимо выполнить для достижения результата.**

Б) программирование вычислительной машины путем описания искомого результата; программист определяет только требования, которым должен удовлетворять результат, но не последовательность шагов для его

достижения.

В) программирование вычислительной машины путем графического отображения интерфейса.

10. Модель жизненного цикла программного средства, которая представляет собой процесс частичной реализации всей системы и постепенного наращивания функциональных возможностей.

А) Итерационная

Б) Спиральная

В) **Инкрементная**

Г) Каскадная

11. Объектно-ориентированное программирование относится к \_\_\_\_\_

А) функциональному программированию

Б) **императивному программированию**

В) абстрактному программированию

Г) логическому программированию

12. На языке C# для делегата программист должен объявить \_\_\_\_\_.

А) хотя бы один конструктор

Б) хотя бы один метод

В) хотя бы одно событие

Г) хотя бы одно поле

Д) хотя бы одно событие и один метод

Е) **все варианты не верны**

13. Коллекция в C#, в которую можно добавлять элементы в середину.

А) **List**

Б) Queue

В) Stack

14. В очереди реализуется дисциплина обслуживания \_\_\_\_\_

А) **первый пришел - первый ушел**

Б) последний пришел - первый ушел

В) последний пришел - последний ушел

Г) последний пришел - последний увидел

Д) первый пришел - первый увидел

15. Ключевые слова, которые используются для работы с исключениями на C#.

А) **try, catch, finally**

Б) try, catch, final

В) try, final, throw

16. Загвоздка событие в программировании – это значит \_\_\_\_\_.

А) **послать сообщение о событии всем объектам – получателям сообщения**

Б) вызвать специальную процедуру, обрабатывающую событие

В) закончить работу программы

Г) сгенерировать исключение

17. Метод, который является обобщенным.

А) **void Swap<T>(T a, T b)**

Б) void Swap(Object a, Object b)

В) void Swap(object a, object b)

Г) void Swap(ref int a, ref int b)

18. Ключевое слово struct в ограничениях на обобщенный тип означает, что тип \_\_\_\_\_.

**А) должен быть значимым**

Б) должен быть ссылочным

В) структурным

Г) числовым

19. Слово "Del" на языке C# в строке delegate void Del(string s) означает \_\_\_\_\_.

**А) Название делегата**

Б) Имя метода

В) Имя экземпляра метода

Г) Ключевое слово

Д) Обработчик события

20. Выберите экземпляры делегата delegate double Del (double s), которые объявлены правильно на языке C#.

**А) Del d= Math.Sin;**

Б) delegate Del d= new (Math.Sin);

21. В строке

Del d += new(Math.Sin); d - это \_\_\_\_\_ .

**А) многоадресный делегат**

Б) делегат-массив

В) переменная типа double

Г) обработчик события

22. Коллекция в языке C# \_\_\_\_\_ поддерживает произвольный доступ.

**А) Dictionary**

Б) Stack

В) Queue

Г) LinkedList

23. Абстрактный класс должен \_\_\_\_\_.

А) содержать неабстрактные методы

Б) содержать абстрактные методы

В) содержать неабстрактные методы

Г) содержать виртуальные методы

Д) содержать виртуальные методы

**Е) нет верных вариантов**

**Сложные (3 уровень)**

24. Методы, которые являются экземплярами делегата delegate void Del(string s) на языке C#.

**А) void S(string s);**

Б) int S(string s2);

В) string Del(string s)

Г) abstract void Del(string s)

25. В технологии LINQ к отложенным операциям относятся \_\_\_\_\_.

**A) Union, Reverse, OrderBy**

Б) ToArray, ToDictionary

В) Skip, Take, Any, All

Г) Skip, Take, Last

Д) Min, Max, Count

**Задания на установление соответствия.**

*Установите соответствие между левым и правым столбцами.*

**Простые (1 уровень)**

26. Восстановите соответствие названию коллекции имя класса в языке C#.

**1) Очередь**

**A) Queue**

**2) Стек**

**Б) Stack**

**3) Двусвязный список**

**В) LinkedList**

Г) Tree

Д) Dequeue

Е) Deque

**Ж) List**

**(1:A,2:Б,3:В)**

27. Расположите модификаторы доступа по возрастанию уровня открытости.

**1) 1**

**A) private**

**2) 2**

**Б) protected**

**3) 3**

**В) public**

**(1:A,2:Б,3:В)**

28. Восстановите соответствие определениям.

**1) Императивное программирование**

**A) описание последовательности действий.**

**2) Декларативное программирование**

**Б) описание требуемого результата**

**В) представление программы в виде объектов.**

**Г) создание концепции абстрактных классов.**

**(1:A,2:Б)**

29. Восстановите соответствие названию коллекции используемую дисциплину обслуживания в C#.

**1) Очередь**

**A) беспriorитетная FIFO**

**2) Стек**

**Б) LIFO**

**В) FILO**

**Г) приоритетная FIFO**

**Д) циклическая**

**Е) динамическая**

**Ж) DynamicTime**

**(1:A,2:Б)**

**Средне-сложные (2 уровень)**

30. Восстановите соответствие функции LINQ ее назначение.

- |              |  |
|--------------|--|
| 1) Where     | А) возвращает удовлетворяющие условию элементы последовательности.   |
| 2) OrderBy   | Б) Выполняет упорядочение элементов последовательности в порядке возрастания ключа.  |
| 3) Take      | В) Возвращает указанное число подряд идущих элементов с начала последовательности.   |
| 4) TakeWhile | Г) Возвращает удовлетворяющие условию подряд идущие элементы с начала последовательности.<br>Д) Выполняет упорядочение элементов последовательности в порядке убывания ключа.<br>Е) Пропускает указанное число подряд идущих элементов с начала последовательности.<br>Ж) Пропускает удовлетворяющие условию подряд идущие элементы с начала последовательности.<br>З) Выполняет дополнительное упорядочение элементов последовательности в порядке возрастания ключа. |

(1:А,2:Б,3:В,4:Г)

31. Восстановите соответствие названия исключений и причин их возникновения.

- |                             |  |
|-----------------------------|--|
| 1) OverflowException        | А) При переполнении в арифметических выражениях  |
| 2) OutOfMemoryException     | Б) При неудачной попытке выделить память   |
| 3) IndexOutOfRangeException | В) При выходе за пределы массива<br>Г) При несоответствии формата<br>Д) При невозможности преобразования типа<br>Е) При попытке деления на нуль<br>Ж) При недопустимом значении аргумента при вызове функции<br>З) При переполнении рабочего стека |

(1:А,2:Б,3:В)

32. Восстановите соответствие назначения регулярного выражения.

- |  |  |
|--|--|
| 1) Для выделения из текста подстроки, в которой x и y разделены операций + или - | А) x[+-]y  |
| 2) Почтовый индекс РФ  | Б) d{6}  |
| 3) Номер месяца  | В) (1[0-2])[1-9]<br>Г) {d}{6}<br>Д) [d]{6}<br>Е) [1]?[12]<br>Ж) [1-12] |

- З) `\d\d`
- И) `x[+|-]y`
- К) `x[+|-]y`

(1:А,2:Б, 3:В)

33. Восстановите соответствие между конструкциями, используемыми в программе на языке C# и используемыми технологиями.

- |                                |                                  |
|--------------------------------|----------------------------------|
| 1) <code>Select</code>         | А) LINQ                          |
| 2) <code>try</code>            | Б) исключительные ситуации       |
| 3) <code>base</code>           | В) ООП                           |
| 4) <code>Stack&lt;T&gt;</code> | Г) обобщенное программирование   |
|                                | Д) динамическое программирование |
|                                | Е) структурное программирование  |
|                                | Ж) к программирование            |

(1:А,2:Б,3:В,4:Г)

**Сложные (3 уровень)**

34. Восстановите последовательность по правильному расположению нескольких блоков `catch` после блока `try` на языке C#, где `MyException` - производный класс от `ArgumentException`.

- |      |  |
|------|--|
| 1) 1 | А) <code>catch(MyException e) { }</code>       |
| 2) 2 | Б) <code>catch(ArgumentException e) { }</code> |
| 3) 3 | В) <code>catch(Exception e) { }</code>         |

(1:А,2:Б,3:В)

35. Восстановите соответствие между переменными и их значениями после выполнения программного кода

```
int f(ref int [] m, ref int a, double d) { m=new int {12,14}; d=4; a=1; }
int a=2;
double z=3;
int c=new int {1,2,3};
f(ref c, ref a, z);
```

- |                   |             |
|-------------------|-------------|
| 1) <code>a</code> | А) 1        |
| 2) <code>z</code> | Б) 3        |
| 3) <code>m</code> | В) {12, 14} |
| 4)                | Г) 2        |
| 5)                | Д) 4        |

(1:А,2:Б,3:В)

**Задания открытого типа**

**Задания на дополнение**

*Напишите пропущенное слово.*

**Простые (1 уровень)**

36. Одно из основных понятий объектно-ориентированного программирования.

(класс, объект)

37. Поля в классе на языке C# рекомендуется описывать с модификатором доступа \_\_\_\_\_.

(private)

38. Метод класса Queue в библиотеке .NET для добавления одного элемента в очередь.

(Enqueue)

39. Коллекция в библиотеке .NET для хранения набора битовых значений.

(BitArray)

40. Stack s = new Stack(); s.Push(2); s.Push(3); int z=s.Pop(); Console.WriteLine(s.Count); На экран будет выведено \_\_\_\_.

(1)

41. Ключевое слово для генерации исключительной ситуации

(throw)

42. Технология программирования, при которой программу можно запустить на других платформах вычислительной системы

(межплатформенное, кроссплатформенное, межплатформенная, кроссплатформенная, межплатформенное программирование, кроссплатформенное программирование, межплатформенная технология, кроссплатформенная технология, межплатформенная технология программирования, кроссплатформенная технология программирования)

43. Специальный метод в ООП, вызываемый автоматически при удалении объекта.

(деструктор)

**Средне-сложные (2 уровень)**

44. Регулярное выражение, которое находит любой из указанных символов: a, b, или c.

([a-c], [abc], a|b|c, @"[a-c]", @"[abc]", @"a|b|c", "[a-c]", "[abc]", "a|b|c", Regex("[a-c]"), Regex("[abc]"), Regex("a|b|c"), Regex("[a-c]"), Regex("[abc]"), Regex("a|b|c"), new Regex("[a-c]"), new Regex("[abc]"), new Regex("a|b|c"), Regex("[a-c]");, Regex("[abc]");, Regex("a|b|c");, Regex("[a-c]");, Regex("[abc]");, Regex("a|b|c");, new Regex("[a-c]");, new Regex("[abc]");, new Regex("a|b|c");)

45. Оператор функционального программирования "переходит в" на C# записывается символами \_\_\_\_\_.

(=>)

46. Регулярное выражение, означающее "не цифра".

(\D, [^0-9], "\\D", "[^0-9]", @"[^0-9]", @"\D", Regex("[^0-9]");, Regex(@"[^0-9]");, Regex(@"\D");, Regex("\\D"); Л) Regex("[^0-9]"), Regex(@"[^0-9]"), Regex(@"\D"), Regex("\\D"), new Regex("[^0-9]");, new Regex(@"[^0-9]");, new Regex(@"\D");, new Regex("\\D");, new

**Regex("[^0-9]"), new Regex(@"[^0-9]"), new Regex(@"\D"), new Regex("\\D"))**

47. Метод для чтения всего содержимого текстового файла в строковую переменную string str = File.\_\_\_\_\_("test.txt");

**(ReadAllText, string str = File.ReadAllText("test.txt"))**

48. int[] numbers = { -3, -2, -1, 0, 1, 2, 3 };  
var result = numbers.Skip(4).Select(a=>a+3);

Напишите значение numbers через пробел.  
**(4 5 6)**

49. Название обязательного интерфейса COM-объекта.

**(IUnknown)**

50. Название одного из методов, осуществляющих подсчет количества пользователей COM-объекта.

**(AddRef, Release)**

51. Для поиска и замены подстрок в строках используется технология \_\_\_\_\_.

**(regular expression, regular expressions, Regex, регулярные выражения, регулярных выражений)**

52. Недостающий фрагмент для упорядочивания чисел по возрастанию на LINQ var m = m.\_\_\_\_\_ =>a).

**(OrderBy(a))**

53. Напишите название метода, который помещает элемент в вершину стека на C#.

**(Push)**

54. Напишите недостающую строку функции на языке C#, предназначенную для сложения двух значений

T f<T>(T a, T b) { \_\_\_\_\_ }

**(return a+b;, return a + b;, return a+ b;, return a +b;)**

55. Ключевое слово в ограничениях на обобщенный тип требует, чтобы у типа был конструктор без параметров (язык C#).

**(new(), new)**

56. Для описания события на языке C# используется ключевое слово \_\_\_\_\_.

**(event)**

57. На языке C# Для связывания событий с обработчиком можно применять только операцию \_\_\_\_\_.

**(+=)**

58. Ключевое слово, которое накладывает ограничение на тип обобщения в языке C#.

**(where)**

59. Напишите недостающий фрагмент для получения всех чисел до первого нулевого на LINQ

var m = m.\_\_\_\_\_ a! =0).

**(TakeWhile(a=>)**

60. Для программной генерации исключения используется ключевое слово \_\_\_\_\_.

**(throw)**



61. Название метода, который вставляет один элемент в произвольную позицию списка на языке C#.

**(Insert)**

62. `int[] numbers = { -3, -2, -1, 0, 1, 2, 3 }; var result = numbers.Skip(4).Take(3);`

Напишите полученные элементы numbers через пробел.

**(1 2 3)**

63. Класс \_\_\_\_\_ служит для представления результатов очередного применения всего регулярного выражения к исходной строке.

**(Match)**

64. `int[] numbers = { -3, -2, -1, 0, 1, 2, 3 }; var result = numbers.TakeWhile(a=>a<1).Take(2).Select(a=>a+3);`

Напишите полученные элементы numbers через пробел.

**(0 1)**

65. Метод, который может быть переопределен в производных классах называется \_\_\_\_\_.

**(виртуальный, виртуальный метод, virtual)**

66. Для сокращения методов используется ключевое слово \_\_\_\_\_.

**(new)**

67. Объявите делегат с именем De для функции Math.Sin.

**(delegate double De(double\*))**

68. `int[] numbers = { -3, -2, -1, 0, 1, 2, 3 }; var result = numbers.TakeWhile(a=>a<1).Take(3);`

Напишите значение result.

**(-3 -2 -1)**

**Сложные (3 уровень)**

69. Занести в список m2 квадраты всех положительных элементов массива m с помощью программного кода

`m2=m.Where(a=>_____).ToList();`

(впишите фрагмент без лишних пробелов).

**(a>0).Select(a=>a\*a), a>0).Select(a=>Math.Pow(a,2))**

70. Напишите выражение, которое привязывает к событию onR объекта k класса Kor обработчик события

`void UvelR(Kor k, int oR).` (Без пробелов.)

**(k.onR+=UvelR;, k.onR+=UvelR, k.onR+= UvelR, k.onR+= UvelR;, k.onR += UvelR, k.onR += UvelR;, k.onR +=UvelR, k.onR +=UvelR;)**

### Карта учета тестовых заданий

Компетенция	ПК-2: Способность проводить тестирование компонентов программного обеспечения ИС.			
Дисциплина	Объектно-ориентированное программирование			
Уровень освоения	Тестовые задания			Итого
	Закрытого типа		Открытого типа	
	Альтернативный выбор	Установление соответствия/последовательности	На дополнение	
1.1.1 (20%)	6	4	7	17
1.1.2 (70%)	17	4	24	45
1.1.3 (10%)	2	2	4	8
Итого:	25 шт.	10 шт.	35 шт.	70 шт.

### Критерии оценивания

#### Критерии оценивания тестовых заданий

Критерии оценивания: правильное выполнение одного тестового задания оценивается 1 баллом, неправильное – 0 баллов.

Максимальная общая сумма баллов за все правильные ответы составляет наивысший балл – 100 баллов.

#### Шкала оценивания результатов компьютерного тестирования обучающихся (рекомендуемая)

Оценка	Процент верных ответов	Баллы
«удовлетворительно»	70-79%	61-75 баллов
«хорошо»	80-90%	76-90 баллов
«отлично»	91-100%	91-100 баллов

## Ключи ответов

- 1 - А)вызова методов базового класса из переопределенных методов производного класса
- 2 - Б)кодирование программного средства
- 3 - А)Queue
- 4 - А)int.Parse
- 5 - Б) Функциональная пригодность
- 6 - А)Класс Form1
- 7 - А)от интерфейсов
- 8 - В) Жизненный цикл программного средства
- 9 - А)программирование вычислительной машины путем описания последовательности инструкций, выполняемых одна за другой; программист однозначно определяет последовательность шагов, которые необходимо выполнить для достижения результата.
- 10 - В) Инкрементная
- 11 - Б)императивному программированию
- 12 - Е)все варианты не верны
- 13 - А)List
- 14 - А)первый пришел - первый ушел
- 15 - А)try, catch, finally
- 16 - А)послать сообщение о событии всем объектам – получателям сообщения
- 17 - А)void Swap<T>(T a, T b)
- 18 - А)должен быть значимым
- 19 - А)Название делегата
- 20 - А)Del d= Math.Sin;
- 21 - А)многоадресный делегат
- 22 - А)Dictionary
- 23 - Е)нет верных вариантов
- 24 - А)void S(string s);
- 25 - А)Union, Reverse, OrderBy
- 26 - 1:А,2:Б,3:В
- 27 - 1:А,2:Б,3:В
- 28 - 1:А,2:Б
- 29 - 1:А,2:Б
- 30 - 1:А,2:Б,3:В,4:Г
- 31 - 1:А,2:Б,3:В
- 32 - 1:А,2:Б,3:В
- 33 - 1:А,2:Б,3:В,4:Г
- 34 - 1:А
- 35 - 1:А,2:Б,3:В
- 36 - класс, объект
- 37 - private
- 38 - Enqueue
- 39 - BitArray

40 - 1  
41 - throw  
42 - межплатформенное, кроссплатформенное, межплатформенная, кроссплатформенная, межплатформенное программирование, кроссплатформенное программирование, межплатформенная технология, кроссплатформенная технология, межплатформенная технология программирования, кроссплатформенная технология программирования  
43 - деструктор  
44 - [a-c], [abc], a|b|c, @"[a-c]", @"[abc]", @"a|b|c", "[a-c]", "[abc]", "a|b|c", Regex("[a-c]"), Regex("[abc]"), Regex("a|b|c"), Regex("[a-c]"), Regex("[abc]"), Regex("a|b|c"), new Regex("[a-c]"), new Regex("[abc]"), new Regex("a|b|c"), Regex("[a-c]");, Regex("[abc]");, Regex("a|b|c");, Regex("[a-c]");, Regex("[abc]");, Regex("a|b|c");, new Regex("[a-c]");, new Regex("[abc]");, new Regex("a|b|c");  
45 - =>  
46 \D, [^0-9], "\\D", "[^0-9]", @"[^0-9]", @"\D", Regex("[^0-9]");, Regex(@"[^0-9]");, Regex(@"\D");, Regex("\\D"); Л) Regex("[^0-9]"), Regex(@"[^0-9]"), Regex(@"\D"), Regex("\\D"), new Regex("[^0-9]");, new Regex(@"[^0-9]");, new Regex(@"\D");, new Regex("\\D");, new Regex("[^0-9]"), new Regex(@"[^0-9]"), new Regex(@"\D"), new Regex("\\D")  
47 - ReadAllText, string str = File.ReadAllText("test.txt")  
48 - 4 5 6  
49 - IUnknown  
50 - AddRef, Release  
51 - regular expression, regular expressions, Regex, регулярные выражения, регулярных выражений  
52 - OrderBy(a  
53 - Push  
54 - return a+b;, return a + b;, return a+ b;, return a +b;  
55 - new(), new  
56 - event  
57 - +=  
58 - where  
59 - TakeWhile(a=>  
60 - throw  
61 - Insert  
62 - 1 2 3  
63 - Match  
64 - 0 1  
65 - виртуальный, виртуальный метод, virtual  
66 - new  
67 - delegate double De(double\*  
68 - -3 -2 -1  
69 - =a>0).Select(a=>a\*a), =a>0).Select(a=>Math.Pow(a,2))

70 -  $k \cdot \text{onR} += \text{UvelR};$ ,  $k \cdot \text{onR} += \text{UvelR}$ ,  $k \cdot \text{onR} += \text{UvelR}$ ,  $k \cdot \text{onR} += \text{UvelR};$ ,  $k \cdot \text{onR} += \text{UvelR}$ ,  $k \cdot \text{onR} += \text{UvelR};$ ,  $k \cdot \text{onR} += \text{UvelR}$ ,  $k \cdot \text{onR} += \text{UvelR};$