

**Государственное образовательное учреждение
"Приднестровский государственный университет им. Т.Г. Шевченко"**

Инженерно-технический институт

**Кафедра программного обеспечения вычислительной техники
и автоматизированных систем**

УТВЕРЖДАЮ

Заведующий кафедрой ПОВТ и АС

 **С.Г. Федорченко**

«28» августа 2022 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Б1.В.07 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

на 2022/2023 учебный год

Направление

2.09.03.01 Информатика и вычислительная техника

Профиль

Вычислительные машины, комплексы, системы и сети

Квалификация

бакалавр

Форма обучения
очная, заочная

2021 ГОД НАБОРА

ст. преп. кафедры ПОВТ и АС,

 **Г.С. Федорченко**

«28» августа 2022 г.

Тирасполь, 2022

Паспорт фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Объектно-ориентированное программирование» у обучающегося должны быть сформированы следующие компетенции:

Задача ПД	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции
Обязательные профессиональные компетенции выпускников и индикаторы их достижения		
Тип задач профессиональной деятельности: <i>производственно-технологический</i>		
Разработка документов для тестирования и анализа качества покрытия; разработка стратегии тестирования и управление процессом тестирования. Обеспечение информационной безопасности на уровне баз данных. Разработка технических документов, адресованных специалистам по информационным технологиям. Администрирование сетевых устройств и программного обеспечения инфокоммуникационной системы, включая администрирование безопасности; проведение регламентных работ на сетевых устройствах и программном обеспечении	ПК-11. Способен осуществлять управление программно-аппаратными средствами информационных служб инфокоммуникационной системы организации, администрирование сетевой подсистемы инфокоммуникационной системы организации	ИД-1 _{ПК-11} Знать методы управления программно-аппаратными средствами информационных служб инфокоммуникационной системы организации, администрирования сетевой подсистемы инфокоммуникационной системы организации
		ИД-2 _{ПК-11} Уметь анализировать состояние программно-аппаратных средств информационных служб инфокоммуникационной системы организации, осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации
		ИД-3 _{ПК-11} Владеть способами управления программно-аппаратными средствами информационных служб инфокоммуникационной системы организации, администрирования сетевой подсистемы инфокоммуникационной системы организации

2. Программа оценивания контролируемой компетенции:

Текущая аттестация	Контролируемые модули, разделы (темы) дисциплины их название	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
РУБЕЖНЫЙ КОНТРОЛЬ	Раздел 1. Классы в C#, наследование классов	ПК-11	Контрольная работа №1
			Лабораторная работа №1 Лабораторные работы №2 Лабораторные работы №3 Лабораторные работы №4 Лабораторные работы №5 Контрольная работа №1 Лабораторные работы №6
РУБЕЖНАЯ АТТЕСТАЦИЯ	Раздел 2. Функциональный тип в C#. Делегаты, события.		Лабораторная работа №7 Лабораторные работы №8 Лабораторные работы №9 Контрольная работа №2
	Раздел 3. Универсальные классы. Отладка программ		Лабораторная работа №10 Лабораторные работы №11 Контрольная работа №2
Промежуточная аттестация		Код контролируемой компетенции (или ее части)	Наименование оценочного средства
№1		ПК-11	Экзамен
№2		ПК-11	Курсовая работа

3. Показатели и критерии оценивания компетенции по этапам формирования, описание шкал оценивания

Этапы оценивания компетенции	Показатели достижения заданного уровня освоения компетенции	Критерии оценивания результатов обучения			
		2	3	4	5
Первый этап	ИД-1 _{ПК-11} Знать методы управления программно-аппаратными средствами информационных служб инфокоммуникационной системы организации, администрирования сетевой подсистемы инфокоммуникационной системы организации	Не знает	Знает современные технологии разработки ПО, но затрудняется с их применением в реальных задачах	Знает современные технологии разработки ПО, но делает ошибки, не влияющие на результаты	Знает современные технологии разработки ПО. Умеет адекватно их применять
Второй этап	ИД-2 _{ПК-11} Уметь анализировать состояние программно-аппаратных средств информационных служб инфокоммуникационной системы организации, осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации	Не умеет	Умеет использовать современные технологии разработки ПО, но допускает много ошибок при их применении для решения профессиональных задач	Умеет использовать современные технологии разработки ПО, но выбирает не оптимальный путь решения задачи	Умеет использовать современные технологии разработки ПО
Третий этап	ИД-3 _{ПК-11} Владеть способами управления программно-аппаратными средствами информационных служб инфокоммуникационной системы организации, администрирования сетевой подсистемы инфокоммуникационной системы организации	Не владеет	Не в полной мере владеет использованием современных технологий разработки ПО	Владеет навыками использования современных технологий разработки ПО, но допускает не критические ошибки	Владеет использованием современных технологий разработки ПО

4. Шкала оценивания

Согласно Положению «О порядке организации аттестации в ИТИ ПГУ им. Т.Г. Шевченко, итоговая оценка представляет собой сумму баллов, полученных студентом по итогу освоения дисциплины (модуля):

Оценка в традиционной шкале	Оценка в 100-балльной шкале	Буквенные эквиваленты оценок в шкале 3Е (% успешно аттестованных)
5 (отлично)	88–100	А (отлично) – 88-100 баллов
4 (хорошо)	70–87	В (очень хорошо) – 80-87 баллов
		С (хорошо) – 70-79 баллов
3 (удовлетворительно)	50–69	Д (удовлетворительно) – 60-69 баллов
		Е (посредственно) – 50-59 баллов
2 (неудовлетворительно)	0–49	FX – неудовлетворительно, с возможной пересдачей – 21-49 баллов
		F – неудовлетворительно, с повторным изучением дисциплины – 0-20 баллов

Расшифровка уровня знаний, соответствующего полученным баллам, дается в таблице, указанной ниже

А	“Отлично” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.
В	“Очень хорошо” - теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом в основном сформированы, все предусмотренные программой обучения учебные задания выполнены, качество выполнения большинства из них оценено числом баллов, близким к максимальному.
С	“Хорошо” - теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками.
Д	“Удовлетворительно” - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки.
Е	“Посредственно” - теоретическое содержание курса освоено частично, некоторые практические навыки работы не сформированы, многие предусмотренные программой обучения учебные задания не выполнены, либо качество выполнения некоторых из них оценено числом баллов, близким к минимальному.
FX	“Условно неудовлетворительно” - теоретическое содержание курса освоено частично, необходимые практические навыки работы не сформированы, большинство предусмотренных программой обучения учебных заданий не выполнено, либо качество их выполнения оценено числом баллов, близким к минимальному; при дополнительной самостоятельной работе над материалом курса возможно повышение качества выполнения учебных заданий.
F	“Безусловно неудовлетворительно” - теоретическое содержание курса не освоено, необходимые практические навыки работы не сформированы, все выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над материалом курса не приведет к какому-либо значимому повышению качества выполнения учебных заданий.

5. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций при изучении учебной дисциплины в процессе освоения образовательной программы

5.1 Типовой вариант лабораторной работы № 1.

ЗАДАНИЕ к лабораторной работе № 1

Разработать класс, который ведет учет результатов сессии для конкретного студента и рекомендует: а) отчислить, б) сформировать график сдачи задолженностей, в) начислить стипендию, г) объявить благодарность, д) вынести выговор.

5.2 Типовой вариант лабораторной работы № 2.

ЗАДАНИЕ к лабораторной работе № 2

Разработать согласно заданию класс. Предусмотреть конструктор по умолчанию и с параметрами. Создать методы, работающие с полями класса. Предусмотреть наличие методов-свойств, статических полей и методов. Продемонстрировать работу объектов класса.

Описания членов-данных пользовательских классов:

1. СТУДЕНТ: ФИО, курс, пол, оценки. Решить задачу определения студентов, которые будут получать стипендию, вычислить размер стипендии. Учесть, что стипендия начисляется в условных единицах (РУМЗП) и в момент расчета учитывается текущее значение РУМЗП. Величина стипендии зависит от полученных оценок, но ряд студентов получают ее вне зависимости от оценок, например староста и студенты-сироты. В программе рассмотреть случай группы из 10 студентов.

5.3 Типовой вариант лабораторной работы № 3.

ЗАДАНИЕ к лабораторной работе № 3

Разработать класс, содержащий вложенные объекты других классов. Обратиться к методам этих классов. Провести полное тестирование разработанных классов, убедившись в том, что все методы работоспособны.

Класс Сессия. Вложенные объекты: **группа, предмет, аудитория, преподаватель.**

Сформировать расписание сессии (4 экзамена).

Выдать на экран по требованию оператора:

- данные о том, когда какой экзамен, какому преподавателю сдает группа;
- список группы;
- данные об аудитории, где принимают экзамен (номер, корпус, количество мест);
- данные о преподавателе, принимающем экзамен;
- данные о группе, сдающей экзамен (факультет, курс, специальность, кол-во студентов).

Программа должна обеспечить возможность изменения некоторых данных.

5.4 Типовой вариант контрольной работы № 1.

Разработать интерфейсные классы и сформировать класс (классы) – наследники. Интерфейсы должны иметь два метода с одинаковыми именами. Разрешить коллизию имен.

Тема: Человек, билет пассажир, поезд (организовать учет пассажиров поезда, поиск вакантных мест, сообщение, до какого пункта едет пассажир).

5.5 Типовой вариант лабораторной работы № 4.

Задание к лабораторной работе № 4

Разработать иерархию классов. Выделить родительский класс. Описать конструкторы в каждом классе. Создать перегруженные и виртуальные методы, а также методы скрывающие методы родителя. Продемонстрировать работу объектов классов в приложе-

нии.

Перечень классов:

- 1) студент, преподаватель, персона;

5.6 Типовой вариант лабораторной работы № 5.

Задание к лабораторной работе № 5

Разработать программу, используя абстрактный класс и образуя его потомки:

- 1) написать программу для расчета объема и площади поверхности следующих геометрических фигур: шар, конус;

5.7 Типовой вариант лабораторной работы № 6.

Задание к лабораторной работе № 6

1. Для полученного задания реализовать по 2 интерфейсных класса и наследовать их.
2. Наследовать стандартный интерфейс *IComparable*.
3. При реализации в наследнике методов интерфейса использовать обертывание и кастинг.
4. Предусмотреть, чтобы один из методов имел одинаковое имя в обоих интерфейсных классах. Реализовать множественное наследование с разрешением коллизии имен.
5. Реализовать еще одного наследника класса и решить проблему наследование от общего предка.
6. В приложении реализовать с помощью интерфейсных элементов управления ввод данных в объекты и вывод информации.

5.8 Типовой вариант лабораторной работы № 7.

Задание к лабораторной работе № 7

Написать программу, которая организует диалог между оператором и программой на одном из 3- языков (например, русский, английский и украинский). Программа содержит три функции, каждая из которых отвечает за диалог на одном из языков. Функция получает номер сообщения и выводит его на экран. В программе объявлен делегат, через который организуется весь диалог с пользователем. В начале работы программы пользователь определяет язык, на котором будет производиться диалог. Программа, по итогам этого диалога формирует экземпляр делегата. Всего реализовать не менее 5 сообщений.

5.9 Типовой вариант лабораторной работы № 8.

Задание к лабораторной работе № 8

Написать программу, которая вводит с файла/клавиатуры данные и выдает их на экран, сортируя по одному из трех признаков, указанному оператором. Данные содержат информацию о:

1. Студенте, его успеваемости.

5.10 Типовой вариант лабораторной работы № 9.

Задание к лабораторной работе № 9

Написать программу, которая организует диалог между оператором и программой на одном из 3- языков (например, русский, английский и украинский). Программа содержит три функции, каждая из которых отвечает за диалог на одном из языков. Функция получает номер сообщения и выводит его на экран. В программе объявлен делегат, через который организуется весь диалог с пользователем. В начале работы программы пользователь определяет язык, на котором будет производиться диалог. Программа, по итогам этого диалога формирует экземпляр делегата. Всего реализовать не менее 5 сообщений.

5.11 Типовой вариант лабораторной работы № 10.

Задание к лабораторной работе № 10

Разработать программу, поддерживающую небольшую информационную систему. Программа должна поддерживать два списка. Содержимое 1-го списка просматривает пользователь и выбирает одну запись. Из нее считывается ключ, и производится поиск по второму списку с использованием этого ключа.

Тема:

1. Библиотека;

5.12 Типовой вариант лабораторной работы № 11.

Задание к лабораторной работе № 11

Разработать программу, выбрасывающую исключение и использующую не менее 3-х обработчиков:

1. Студент знает ответы на 10 экзаменационных вопросов из 30. Если заданный вопрос ему не известен, то он: а) пытается вспомнить; б) пробует воспользоваться шпаргалкой; в) несет окошечку.

5.11 Типовой вариант контрольной работы № 2.

Написать программу, которая:

- 1) Просит оператора указать, к какому виду пользователей он относится: администратор, гость, студент или anonymous. В зависимости от выбора оператора программа по разному организует свой диалог с ним: запрашивает или нет, пароль, сообщает о том, что ему можно делать, а что нельзя и т.д. Использовать для реализации этой части задания делегат, который обращается к одному из 4-х методов, созданных вами.
- 2) Программа ведет диалог с пользователем, в процессе которого она выясняет, значение какой функции, в каких пределах и с каким шагом он хочет вычислить. Программа выбирает, с помощью делегата, одну из 5 функций, созданных вами, и формирует на экране таблицу ее значений. Виды 5 функций задать самостоятельно.
- 3) Программа различным образом прощается с пользователем, например, спрашивает, когда он еще посетит их, или наоборот, просит экономить машинное время, или предлагает зайти через месяц, когда появится новая версия программы и т.д. всего версий диалога окончания работы программы должно быть 5-6. Выбор версии осуществляется программой с помощью делегата, в зависимости от результатов диалога в п.1 и п.2. Данную зависимость разработать самостоятельно.

5.12 Типовой тест к экзамену

1. Класс – это:

- 1) тип данных, задающий реализацию некоторой абстракции данных;
- 2) модуль, архитектурная единица построения программы;
- 3) структура, содержащая набор данных;
- 4) набор функций (методов), обрабатывающий конкретный набор данных.

2. Модификаторы доступа класса:

- 1) private;
- 2) protected;
- 3) static;
- 4) public;
- 5) global.

3. Конструктор класса, это:
 - 1) метод, осуществляющий инициализацию полей объекта класса;
 - 2) метод, создающий объект класса;
 - 3) метод, создаваемый автоматически компилятором, если он не объявлен программистом.
4. Методы класса, которые можно вызвать, не создавая объект класса, имеют модификатор:
 - 1) internal;
 - 2) static;
 - 3) global;
 - 4) private.
5. Деструктор имеет имя:
 - 1) назначенное разработчиком;
 - 2) "~" + имя класса;
 - 3) имя класса;
 - 4) длиной не более 10 символов.
6. Класс А является клиентом класса В, если:
 - 1) класс А является наследником класса В;
 - 2) классы А и В имеют ряд общих методов;
 - 3) класс А содержит объект класса В;
 - 4) класс А обращается к полям класса В.
7. При наследовании класс – наследник получает от класса-родителя:
 - 1) поля;
 - 2) методы;
 - 3) конструктор;
 - 4) деструктор;
 - 5) индексатор.
8. Класс – наследник может:
 - 1) изменить имена полей класса- родителя;
 - 2) переопределить методы, полученные от класса-родителя;
 - 3) добавить новые методы;
 - 4) добавить новое поле;
 - 5) изменить конструктор класса-родителя.
9. Способность одного и того же программного текста вида Х.М выполняться по-разному, в зависимости от того, с каким объектом связан Х, это:
 - 1) полиморфизм;
 - 2) наследование;
 - 3) инкапсуляция;
 - 4) множественное наследование.
10. Абстрактный класс это:
 - 1) если он имеет хотя бы один абстрактный метод;
 - 2) класс, содержащий не полностью определенные методы;
 - 3) класс, не позволяющий создавать потомков;
 - 4) класс, имеющий более одного конструктора.
11. Интерфейсный класс:
 - 1) класс, содержащий только абстрактные методы;

- 2) класс, не содержащий конструктора;
 - 3) класс, требующий от класса - наследника полностью переопределить его методы;
 - 4) класс, содержащий статические поля и методы.
12. Обертывание позволяет:
- 1) организовать доступ к закрытым методам класса-родителя;
 - 2) создать новый объект класса-родителя;
 - 3) изменить метод класса-родителя.
13. Клонирование это:
- 1) создание копии объекта;
 - 2) удаление объекта;
 - 3) модификация объекта;
 - 4) переименование объекта.
14. Сериализация объектов:
- 1) сохранение состояния объекта на магнитном носителе;
 - 2) восстановление состояния объекта с магнитного носителя;
 - 3) сортировка объектов по критерию, заданному программистом;
 - 4) сравнение объектов по заданному алгоритму.
15. Делегат это:
- 1) функциональный тип (класс) данных;
 - 2) класс-наследник другого класса;
 - 3) класс, содержащий функции высшего порядка;
 - 4) класс, экземплярами которого являются функции.
16. Над делегатами определены следующие операции:
- 1) Combine;
 - 2) Remove;
 - 3) +=;
 - 4) -=;
 - 5) * =;
 - 6) /=.
17. Event это:
- 1) ключевое слово при объявлении события;
 - 2) модификатор доступа к методу;
 - 3) используется при описании виртуальных методов;
 - 4) позволяет объявлять On-процедуры.
18. Список вызова (Invocation list) это:
- 1) список полей данного класса;
 - 2) список объектов, подключенных через индексатор;
 - 3) список методов, вызываемых при обращении к экземпляру делегата;
 - 4) список классов-потомков и соответствующих классов-родителей.
19. Для захвата исключения применяется:
- 1) блок catch;
 - 2) блок if;
 - 3) блок switch;
 - 4) блок while.
20. Универсальность класса:

- 1) способность класса объявлять типы переменных, модулей как аргументы;
 - 2) возможность обработки исключительных ситуаций;
 - 3) возможность обращения к объектам интерфейсных классов;
 - 4) способность работы с методами класса-родителя.
21. Ограниченная универсальность позволяет:
- 1) ограничить возможности класса-наследника;
 - 2) ввести дополнительные операции, разрешенные для объектов класса-наследника;
 - 3) обеспечить защиту полей класса.
22. Предложение using позволяет:
- 1) породить от универсального класса не универсальный класс-наследник;
 - 2) обеспечить создание универсальной структуры;
 - 3) объявить универсальный интерфейс.
23. Спецификация это:
- 1) описание требований, предъявляемых к значениям входных и выходных данных программы (модуля);
 - 2) описание алгоритма, реализуемого модулем;
 - 3) описание внешних вызываемых модулей.
24. Законы программотехники:
- 1) корректность системы – недостижима;
 - 2) каждая ошибка системы – следствие незнания ее спецификации;
 - 3) новичок способен подвесить любую систему;
 - 4) дурак опаснее врага;
- 25) Функция высшего порядка, это:
- 1) Функция, использующая в качестве переменной объект некоторого класса;
 - 2) Функция, имеющая в качестве аргумента ссылку на другую функцию, определенную вне данной функции
 - 3) Функция, возвращающая значение в виде объекта некоторого класса
 - 4) Функция, к которой обращаются из другого класса
- 26) On-процедура выполняет:
- 1) Объявление события
 - 2) Зажигание события
 - 3) Подключение обработчика события
 - 4) Отключение обработчика события

Ключи:

1. 1), 2).
2. 1), 2), 4).
3. 2), 3).
4. 2).
5. 2).
6. 3).
7. 1), 2).
8. 2), 3), 4).
9. 1).
10. 1), 2).

11. 1),2)
12. 1).
13. 1).
14. 1).
15. 1),4).
16. 1),2),3)4).
17. 1)
18. 3).
19. 1)
20. 1).
21. 2)
22. 1)
23. 1)
24. 1), 2), 3)
25. 2)
26. 2)

5.9 Примерная тематика курсовых работ

1. Программа, демонстрирующая обмен данными в топологии «шина».
2. Программа, иллюстрирующая решение транспортной задачи.
3. Информационная система выпускников вуза.
4. Программа, иллюстрирующая тему «Структура ЭВМ».
5. Формирование трехмерного вида с различным разрешением с перемещающейся точки взгляда в трехмерном пространстве.
6. Синтез элементов ландшафта.
7. Программа редактирования матрицы.
8. Разработка информационной системы «Расписание занятий»
9. Разработка информационной системы «Справочник учебных заведений»
10. Разработка информационной системы «Справочник предприятий»
11. Разработка информационной системы «Сотрудники организации»
12. Разработка информационной системы «Студенты»
13. Разработка информационной системы «Учет ЭВМ в организации»

Вопросы к экзамену по дисциплине «Объектно-ориентированное программирование»

1. Понятие класса и объекта класса.
2. Модификаторы доступа.
3. Элементы класса: поля, методы, свойства.
4. Конструкторы и деструкторы.
5. Статические поля и методы.
6. Индексаторы.
7. Свойства класса.
8. Клиенты класса.
9. Наследование класса.
10. Наследование классов. Коллизия имен.
11. Абстрактные классы.
12. Наследование, определение производного класса, изменение методов родителя.
13. Статические поля и методы класса.
14. Механизмы, обеспечивающие полиморфизм
15. Статический контроль типов и динамическое связывание.
16. Инкапсуляция, наследование, полиморфизм.
17. Полиморфизм, примеры его использования.
18. Включение в класс объектов другого класса.
19. Интерфейсы, преобразование к классу интерфейса.

20. Множественное наследование.
21. Сериализация объектов.
22. Встроенный интерфейс *IComparable*.
23. Встроенный интерфейс *ICloneable*.
24. Встроенный интерфейс *IComparable*.
25. Кастинг и обертывание, пример их использования.
26. Функции обратного вызова.
27. Построение программных систем методом «раскрутки».
28. Понятие делегата, экземпляр делегата.
29. Функции высших порядков.
30. Делегаты как свойства.
31. Операции над делегатами.
32. Классы с событиями, объявление событий, вызов и обработка события.
33. Переопределение значений аргументов события.
34. Наследование и универсальность
35. Родовое порождение класса.
36. Обработка исключительных ситуаций.
37. Выбрасывание исключений, захват исключений.
38. Синтаксис универсального класса, класс с универсальными методами.
39. Ограниченная универсальность.
40. Универсальные структуры.
41. Корректность и устойчивость программных систем.
42. Законы программирования