

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

«29» августа 2022 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Б1.О.13 «ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ»

Направление подготовки
2.09.03.04 Программная инженерия

Профиль подготовки
Разработка программно-информационных систем

Квалификация (степень)

выпускника:

бакалавр

Форма обучения:

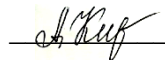
очная, заочная

Год набора:

2022 г.

Разработал:

к.п.н., доцент кафедры ПОВТ и АС,



/А.В.Кирсанова

«29» августа 2022 г.

Тирасполь, 2022

Паспорт фонда оценочных средств по учебной дисциплине

- 1. В результате изучения дисциплины: Б1.О.13 «Прикладное программирование» у обучающегося должны быть сформированы следующие компетенции:**

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения универсальной компетенции
Обязательные профессиональные компетенции и индикаторы их достижения		
	ОПК-2. Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ИД-1 _{ОПК-2} Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
		ИД-2 _{ОПК-2} Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
		ИД-3 _{ОПК-2} Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
	ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ИД-1 _{ОПК-5} Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем
		ИД-2 _{ОПК-5} Умеет выполнять параметрическую настройку информационных и автоматизированных систем
		ИД-3 _{ОПК-5} Имеет навыки инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
	ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ИД-1 _{ОПК-6} Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий

		ИД-2 _{ОПК-6} Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ
		ИД-3 _{ОПК-6} Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач
	ПК-7. Способность оценивать временную и емкостную сложность программного обеспечения	ИД-1 _{ПК-7} Знает методы оценки временной и емкостной сложности программного обеспечения
		ИД-2 _{ПК-7} Умеет вычислять временную и емкостную сложность ПО
		ИД-3 _{ПК-7} Имеет навыки оценки временной и емкостной сложности ПО

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	<i>Раздел 1. Технологии разработки прикладного программного обеспечения.</i>	ОПК-2, ОПК-5, ОПК-6, ПК-7	КР1
	<i>Раздел 2. Основы прикладного программирования с использованием языка программирования высокого уровня, например, C#.</i>	ОПК-2, ОПК-5, ОПК-6, ПК-7	КР1, ЛБ1-ЛБ3
	<i>Раздел 3. Реализация объектно-ориентированного программирования.</i>	ОПК-2, ОПК-5, ОПК-6, ПК-7	КР1, ЛБ3
РУБЕЖНАЯ АТТЕСТАЦИЯ	<i>Раздел 4. Пользовательский интерфейс прикладных программ.</i>	ОПК-2, ОПК-5, ОПК-6, ПК-7	КР2, ЛБ3

	Раздел 5. Организация разработки прикладного программного обеспечения.	ОПК-2, ОПК-5, ОПК-6, ПК-7	КР2, ЛБ4-ЛБ8
Промежуточная аттестация		ОПК-2, ОПК-5, ОПК-6, ПК-7	Экзамен

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	Знать ОПК-2	Не знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности, но допускает существенные ошибки	Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности, но допускает неточности	Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
Второй этап	Уметь ОПК-2	Не умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности	Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности, но допускает существенные ошибки	Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности, но допускает неточности	Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
Третий этап	Владеть ОПК-2	Не имеет навыки применения современных информационных технологий и программных средств	Имеет навыки применения современных информационных технологий и программных средств, в том	Имеет навыки применения современных информационных технологий и программных средств, в том	Имеет навыки применения современных информационных технологий и программных средств, в том

		средств, в том числе отечественного производства, при решении задач профессиональной деятельности	числе отечественного производства, при решении задач профессиональной деятельности, но допускает существенные ошибки	числе отечественного производства, при решении задач профессиональной деятельности, но допускает неточности	числе отечественного производства, при решении задач профессиональной деятельности
Первый этап	Знать ОПК-5	Не знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем	Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем, но допускает существенные ошибки	Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем, но допускает неточности	Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем
Второй этап	Уметь ОПК-5	Не умеет выполнять параметрическую настройку информационных и автоматизированных систем	Умеет выполнять параметрическую настройку информационных и автоматизированных систем, но допускает существенные ошибки	Умеет выполнять параметрическую настройку информационных и автоматизированных систем, но допускает неточности	Умеет выполнять параметрическую настройку информационных и автоматизированных систем
Третий этап	Владеть ОПК-5	Не имеет навыки инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем	Имеет навыки инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем, но допускает существенные ошибки	Имеет навыки инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем, но допускает неточности	Имеет навыки инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем
Первый этап	Знать ОПК-6	Не знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки

		программные среды разработки информационных систем и технологий	информационных систем и технологий, но допускает существенные ошибки	информационных систем и технологий, но допускает неточности	информационных систем и технологий
Второй этап	Уметь ОПК-6	Не умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ, но допускает существенные ошибки	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ, но допускает неточности	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ
Третий этап	Владеть ОПК-6	Не имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач, но допускает существенные ошибки	Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач, но допускает неточности	Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач
Первый этап	Знать ПК-7	Не знает методы оценки временной и емкостной сложности программного обеспечения	Знает методы оценки временной и емкостной сложности программного обеспечения, но допускает грубые ошибки	Знает методы оценки временной и емкостной сложности программного обеспечения, но допускает неточности	Знает методы оценки временной и емкостной сложности программного обеспечения

Второй этап	Уметь ПК-7	Не умеет вычислять временную и емкостную сложность ПО	Умеет вычислять временную и емкостную сложность ПО, но допускает грубые ошибки	Умеет вычислять временную и емкостную сложность ПО, но допускает неточности	Умеет вычислять временную и емкостную сложность ПО
Третий этап	Владеть ПК-7	Не умеет навыки оценки временной и емкостной сложности ПО	Имеет навыки оценки временной и емкостной сложности ПО, но допускает грубые ошибки	Имеет навыки оценки временной и емкостной сложности ПО, но допускает неточности	Имеет навыки оценки временной и емкостной сложности ПО
Первый этап	Знать ПК-8	Не знает способы создания программных интерфейсов	Знает способы создания программных интерфейсов, но допускает грубые ошибки	Знает способы создания программных интерфейсов, но допускает неточности	Знает способы создания программных интерфейсов
Второй этап	Уметь ПК-8	Не умеет создавать интуитивно понятные программные интерфейсы	Умеет создавать интуитивно понятные программные интерфейсы, но допускает грубые ошибки	Умеет создавать интуитивно понятные программные интерфейсы, но допускает неточности	Умеет создавать интуитивно понятные программные интерфейсы
Третий этап	Владеть ПК-8	Не имеет навыки в создании современных программных интерфейсов	Имеет навыки в создании современных программных интерфейсов, но допускает грубые ошибки	Имеет навыки в создании современных программных интерфейсов, но допускает неточности	Имеет навыки в создании современных программных интерфейсов

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 ЗЕ (% успешно аттестованных)
5 (отлично)	88–100	А (отлично) – 88-100 баллов
4 (хорошо)	70–87	В (очень хорошо) – 80-87баллов
		С (хорошо) – 70-79 баллов
3 (удовлетворительно)	50–69	D(удовлетворительно) – 60-69 баллов
		E(посредственно) – 50-59 баллов
2 (неудовлетворительно)	0–49	Fх– 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

A	“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.
B	“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.
C	“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.
D	“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.
E	“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 задания на контрольную работу 1.

1. Какие компоненты включает в себя *.NET Framework*?

2. Расшифруйте и сформулируйте назначение следующих аббревиатур *MSIL*, *JIT*, *CTS*.
3. Дайте определения понятиям: инкапсуляция, наследование и полиморфизм. Приведите пример полиморфизма.
4. Каково «взаимоотношение» между классом и объектом?
5. Показать на примере в чем состоит отличие в работе оператора множественного выбора в C++ и в C#.
6. Решить одну из предложенных задач (задачи даны по возрастанию количества баллов)
 - 6.1. Стороны одного прямоугольника равны A и B . Стороны другого равны X и Y . Написать алгоритм и программу проверки прямоугольников на равенство.
 - 6.2. Тело имеет массу M г и объем V см³. Будет ли оно плавать в жидкости, объем которой равен V_1 см³, а масса M_1 г? (Указание: найдите плотности тела и жидкости, а затем сравните их).
 - 6.3. Смекалкин, уходя в школу, вышел из дому на n минут позже младшего брата. Расстояние до школы S метров. Смекалкин идет со скоростью V_1 м/мин, а его брат – V_2 м/мин. Догонит ли Смекалкин брата, прежде чем тот придет в школу?

5.2 Типовой вариант задания на контрольную работу 2

1. Тип `char[]` и его отличительные особенности. Преобразование типа `String` в тип `char[]` и наоборот. Методы класса `char[]`.
2. Написать программу (одну из трех), которая
 - 4.1) находит в строке количество знаков пунктуации.
 - 4.2) в строке после каждого слова добавляет число, указывающее сколько раз встретилось это слово в строке. Например: Роза2 красивый2 цветок2. Да1 роза2 самый1 красивый2 цветок2!
 - 4.3) находит в строке самое длинное слово и определяет, сколько раз оно встретилось в тексте. Считать, что такое слово одно.
3. Напишите программу, в которой выполняются следующие действия:
 1. В папке `D:\Stud` создайте папки `Kat1` и `Kat2`.
 2. В папке `Kat1`:
 - а) создайте файл `file1.txt`, в который запишите следующий текст: *г. Тирасполь, ПМР*
 - б) создайте файл `file2.txt`, в который запишите следующий текст: *г. Бендеры, ПМР*
 3. В папке `Kat2` создайте файл `file3.txt`, в который перепишите вначале текст из файла `file1.txt`, а затем из `file2.txt`
 4. Выведите развернутую информацию о созданных файлах.
 5. Файл `file2.txt` перенесите в папку `Kat2`.
 6. Файл `file1.txt` скопируйте в папку `Kat2`.
 7. Папку `Kat2` переименуйте в `Kat`, а папку `Kat1` удалите.
 8. Вывести полную информацию о файлах папки `Kat`.

5.3. Типовой вариант задания на лабораторную работу 1. *Visual Studio.NET*. Создание консольного проекта. Типы данных и операторы языка C#

1. Дано целое число. Определить:
 - а) является ли оно четным;
 - б) оканчивается ли оно цифрой 7;
 - в) оканчивается ли оно четной цифрой (составное условие не использовать).
2. Известны площади круга и квадрата. Определить:
 - а) уместится ли круг в квадрате;
 - б) уместится ли квадрат в круге.
3. Дано двузначное число. Определить:
 - а) какая из его цифр больше, первая или вторая;
 - б) одинаковы ли его цифры.
4. Даны две точки: $A(x_1, y_1)$ и $B(x_2, y_2)$. Составить алгоритм, определяющий, которая из точек находится ближе к началу координат.
5. Даны два угла треугольника (в градусах). Определить, существует ли такой треугольник, и если да, то является ли он прямоугольным.

6. Рис расфасован в два пакета. Масса первого - m кг, 2-го - n кг. Составить программу, определяющую:
 - а) какой пакет тяжелее - первый или второй;
 - б) массу более тяжелого пакета.
7. Написать программу, которая анализирует данные о возрасте и относит человека к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст вводится с клавиатуры.

5.4 Типовой вариант задания на лабораторную работу 2. Функции и процедуры.

Различные способы передачи параметров в функцию.

1. Разработать метод $f(x)$, который в трехзначном числе меняет местами первую с последней цифрой, а остальные числа оставляет без изменения. Продемонстрировать работу данного метода на примере.
2. Два числа называются «дружественными», если одно из них равно сумме всех делителей другого (кроме его самого) (например, числа 220 и 284). Найти все пары «дружественных чисел», которые не больше данного числа N .
3. Два простых числа называются «близнецами», если они отличаются друг от друга на 2 (например, 41 и 43). Напечатать все пары «близнецов» из отрезка $[n, 2n]$, где n – заданное натуральное число больше 2.
4. Проверить, являются ли данные три числа взаимно простыми.
5. Вычислить площадь правильного шестиугольника со стороной a , используя функцию вычисления площади треугольника.
6. Даны две дроби $\frac{A}{B}$ и $\frac{C}{D}$ (A, B, C, D – натуральные числа). Составить программу:
 - деления дроби на дробь;
 - умножения дроби на дробь;
 - сложения этих дробей.

Ответ должен быть несократимой дробью.

7. Написать функцию, которая вычисляет сопротивление цепи, состоящей из двух резисторов. Параметрами функции являются величины сопротивлений и тип соединения (последовательное или параллельное). Функция должна проверять корректность параметров: если неверно указан тип соединения, то функция должна возвращать -1.

8. Написать функцию, обеспечивающую решение квадратного уравнения. Параметрами функции должны быть коэффициенты и корни уравнения. Значение, возвращаемое функцией, должно передавать в вызывающую программу информацию о наличии у уравнения корней: 2 – два разных корня, 1 – корни одинаковые, 0 – уравнение не имеет решения. Кроме того, функция должна проверять корректность исходных данных. Если исходные данные неверные, то функция должна возвращать -1.

9. Разработать метод $f(x1, y1, x2, y2)$, который вычисляет длину отрезка по координатам вершин $(x1, y1)$ и $(x2, y2)$, и метод $max(a, b)$, который вычисляет максимальное из чисел a, b . С помощью данных методов определить, какая из трех точек на плоскости наиболее удалена от начала координат.

10. Разработать метод $f(x1, y1, x2, y2)$, который вычисляет длину отрезка по координатам вершин $(x1, y1)$ и $(x2, y2)$, и метод $t(a, b, c)$, который проверяет, существует ли треугольник с длинами сторон a, b, c . С помощью данных методов проверить, сколько различных треугольников можно построить по четырем заданным точкам на плоскости.

5.5. Типовой вариант задания на лабораторную работу 3. Обработка исключений

1. Постройте таблицу значений функции $y = \frac{x}{\sqrt{2x-1}}$ для $x \in [a, b]$ с шагом h . При решении данной задачи проводить обработку возможных исключений.
2. Постройте таблицу значений функции $y = \frac{1}{x-1} + \frac{2}{1-x}$ для $x \in [a, b]$ с шагом h . При решении данной задачи проводить обработку возможных исключений.

5.6. Типовой вариант задания на лабораторную работу 4. Одномерные и многомерные массивы в C#. Методы класса Array

1. Удалить элемент массива целых чисел, удовлетворяющий условию: остаток от деления на 2 равен 3. Если таких элементов нет, выдать сообщение «Элементы для удаления не найдены».
2. На k -е место одномерного массива вещественных чисел вставить элемент, равный среднему арифметическому элементов массива.
3. Задан массив. Создать два массива, в одном из которых находятся все четные по номеру элементы, в другом нечетные по номеру элементы массива.
4. Заданы два массива. Создать один массив, в котором все элементы расположены в порядке возрастания.
5. У прилавка магазина выстроилась очередь из n покупателей. Время обслуживания i -го покупателя равно t_i ($i = 1, \dots, n$). Определить время Cr пребывания $г$ -го покупателя в очереди.
6. Каждый солнечный день улитка, сидящая на дереве, поднимается вверх на 2 см, а каждый пасмурный день опускается вниз на 1 см. В начале наблюдения улитка находилась на расстоянии A см от земли на 5-метровом дереве. Имеется 30-элементный массив, содержащий сведения о том, был ли соответствующий день наблюдения пасмурным или солнечным. Написать программу, определяющую местоположение улитки к концу 30-го дня наблюдения.
7. Даны координаты n точек на плоскости $(x_1, y_1), \dots, (x_n, y_n)$ ($n < 30$). Найти номера пары точек, расстояние между которыми наибольшее (считать, что такая пара единственная).
8. Даны два упорядоченных массива целых чисел M_1 и M_2 . Объединить их в один упорядоченный массив. Рассмотреть случаи, когда массивы M_1 и M_2 упорядочены одинаково и когда — по-разному.
9. При диспансеризации школьников определялись их рост и вес. В результате были получены массивы значений роста $R(n)$ и веса $W(n)$. Написать алгоритм и программу определения школьников с максимальным и минимальным ростом и их номеров в таблице, а также номер школьника с наибольшим дефицитом веса (наименьшим отношением веса к росту). Кроме того, следует упорядочить по возрастанию массив весов. Вывести результаты работы программы.
10. На соревнованиях по прыжкам в высоту и в длину получены два массива результатов: $H(n)$ и $D(n)$. Определить три лучших и три худших результата в каждом виде соревнований.

5.7. Типовой вариант задания на лабораторную работу 5. Строки в C#. Классы Char[], String, StringBuilder и их методы

1. Составить программу, которая будет вводить строку в переменную string. Подсчитать, сколько различных символов встречаются в ней. Вывести их на экран.
2. Составить программу, которая будет вводить строку в переменную string. Найти в ней те слова, которые начинаются и оканчиваются одной и той же буквой.
3. Составить программу, которая будет вводить строку в переменную string. Определить, сколько раз в строке встречается заданное слово.
4. Строка, содержащая произвольный русский текст, состоит не более чем из 200 символов. Написать, какие буквы и сколько раз встречаются в этом тексте. Ответ должен приводиться в грамматически правильной форме: например: а — 25 раз, к — 3 раза и т.д.
5. Двумерный массив $n \times m$ содержит некоторые буквы русского алфавита, расположенные в произвольном порядке. Написать программу, проверяющую, можно ли из этих букв составить данное слово S . Каждая буква массива используется не более одного раза.
6. Дана строка, содержащая текст и арифметические выражения вида $a \otimes b$, где $\otimes$ — один из знаков $+$, $-$, $*$, $/$. Выписать все арифметические выражения и вычислить их значения.
7. Составить программу, которая будет вводить строку в переменную string. Удалить из нее все лишние пробелы, оставив между словами не более одного. Результат поместить в новую строку.

5.8. Типовой вариант задания на лабораторную работу 6. Регулярные выражения.

1. Задан текст. Определить входит ли в него заданное слово и сколько раз.

2. Задан текст. Определить является ли он кодом HTML : содержит теги <html>, <form>, <h1>.
3. Задан текст. Определить является ли он программой на PHP: содержит скобки <? и ?>.
4. Задан текст. Определить является ли он текстом на английском языке.
5. Задан текст. После каждой буквы «о» вставить сочетание «ок».
6. Задан текст. Определить является ли он текстом на русском языке.
7. Задан текст. Определить содержит ли он цифры.
8. Задан текст. Определить сколько предложений начинается со слова “Информатика”.
9. Задан текст. Выбрать из него все номера телефонов.
10. Задан текст. Определить содержит ли он цифры.
11. Задан текст. Выбрать из него все e-mail адреса.
12. Задан текст, содержащий буквы и цифры. Найти произведение все чисел в тексте.
13. Задано предложение. Распечатать все слова в столбик.
14. Задан текст. Определить количество согласных букв в нем и распечатать их.

5.9. Типовой вариант задания на лабораторную работу 7. Байтовый, символьный и двоичный потоки. Перенаправление стандартных потоков

Вариант 1. Сформировать файл последовательности 15 чисел, в которой каждый i -й компонент определяется по формуле $y = \sin(i \cdot \pi / 8)$, если $i < 8$; $y = 4 \cos(i(\pi + 1)/5)$, если $i > 8$. Определить количество положительных значений, содержащихся в сформированном файле.

Составить программу, определяющую правильность следования скобок в строке символов, хранящейся в файле.

Вариант 2. Сформировать файл последовательности 20 чисел, в которой каждый i -й компонент определяется по формуле $y = \sin(i \cdot \pi / 8)$, если $i < 8$; $y = 4 \cos(i(\pi + 1)/5)$, если $i > 8$. Определить количество отрицательных значений, содержащихся в сформированном файле.

Подсчитать количество двоек символов 'ee', 'nn', 'll' в тексте, расположенном в текстовом файле.

Вариант 3. Сформировать файл из значений случайных величин: 0,324; 0,524; 0,789; 0,556; 0,761; 0,248; 0,345; 0,911; 0,216, Определить для данной последовательности среднее арифметическое компонент, значения которых меньше 0,5.

Разбить произвольный текст, находящийся в файле, на строки определенной длины. При переносе слова предусмотреть вывод дефиса.

Вариант 4. Откорректировать текст, расположенный в текстовом файле, заменив в нем все вхождения одной буквы на другую.

Произвести сортировку файла целых чисел методом пузырька.

Вариант 5. Сформировать файл, содержащий фамилии нескольких студентов. Добавить к полученному файлу фамилии еще 2-3 студентов.

Найти в текстовом файле самое длинное и самое короткое слово.

Вариант 6. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий. Добавить в начало файла оценки, полученные на вступительных экзаменах.

Из строки, расположенной в текстовом файле, исключить все символы, входящие в нее более одного раза.

5.10. Типовой вариант задания на лабораторную работу 8. Работа с каталогами. Работа с дисками. Работа с файлами: запись и чтение данных в/из файла.

Вариант 1. Написать программу, которая считывает из текстового файла три предложения и выводит их в обратном порядке.

Вариант 2. Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие введенное с клавиатуры слово.

Вариант 3. Написать программу, которая считывает текст из файла и выводит на экран только строки, содержащие двузначные числа.

Вариант 4. Написать программу, которая считывает английский текст из файла и выводит на экран слова, начинающиеся с гласных букв.

Вариант 5. Написать программу, которая считывает текст из файла и выводит его на экран, меняя местами каждые два соседних слова.

Вариант 6. Написать программу, которая считывает текст из файла и выводит на экран только предложения, не содержащие запятых.

Вариант 7. Написать программу, которая считывает текст из файла и определяет, сколько в нем слов, состоящих не более чем из четырех букв.

Вариант 8. Написать программу, которая считывает текст из файла и выводит на экран только цитаты, то есть предложения, заключенные в кавычки.

Вариант 9. Написать программу, которая считывает текст из файла и выводит на экран только предложения, состоящие из заданного количества слов.

Вариант 10. Написать программу, которая считывает английский текст из файла и выводит на экран слова текста, начинающиеся и оканчивающиеся на гласные буквы.

5.11 Типовой тест промежуточной аттестации

1. Из указанных типов выберите типы данных, хранящих свое значение в стеке

a. Числовые типы

c. Строки

e. Перечисления

b. Массивы

d. Структуры

f. Классы

g. Логический тип

2. Какие из следующих фрагментов ошибочны

a. `int k; ...`

c. `int a, b, k; ...`

e. `int a, b, k; ...`

`if (k=4) ...`

`if((a>=k)&&(!b)) ...`

`if((a>b)&&(b<k)) ...`

b. `int a, b; ...`

d. `int k; ...`

f. `int k; ...`

`if(!(a==b)) ...`

`if (k) ...`

`if (k== 'f') ...`

3. Выберите допустимые прототипы метода Main.

a. `static void Main() {}`

d. `static void Main(string[] args) {}`

b. `static int Main() {}`

e. `static int main() {}`

c. `static void main() {}`

f. `static int Main(string[] args) {}`

4. Какое из объявлений строк верное:

a. `string s2 = 'www';`

b. `StringBuilder s1 = new StringBuilder ("The best");`

c. `string s2 = "abc hju";`

d. `string s2 = new string ("abc");`

e. `StringBuilder s = "def";`

5. Что появится на экране в результате работы следующего фрагмента
namespace test

```
{
    class Test
    {
        static void Main()
        {
            string s1 = "led", s2 = "voda", s3 = "par";
            System.Console.WriteLine(String.Compare(s1, s1));
            System.Console.WriteLine(String.Compare(s1, s2));
            System.Console.WriteLine(String.Compare(s1, s3));
        }
    }
}
```

[illegible]

6. Какое из объявлений массива верное:

- a.** `int[] k = new int [3];`
b. `int[] k = new int [] {1, 2, 3, 5, 8};`
c. `int[] k = new int {4, 6};`
d. `int k[] = new int [] {1, 2, 3, 5, 8};`
e. `int[] k = {1, 2, 3, 5, 8};`

7. Что будет результатом работы программы namespace test

```
{
class someClass
{
    public static int SomeFunc()
    {
        return 5;
    }
    public static int SomeFunc(int k)
    {
        return k*k;
    }
    public static float SomeFunc(float k)
    {
        return k;
    }
    static void Main(string[] args)
    {
        Console.WriteLine(SomeFunc());
        Console.WriteLine(SomeFunc( 10));
        Console.WriteLine(SomeFunc((float)10));
    }
}
}
```

8. Верными являются утверждения

- a.** в структуре не допускается объявлений членов класса со спецификаторами доступа *protected* и *protected internal*
- b.** в структуре допускается объявление членов класса с любым спецификатором доступа
- c.** при объявлении структуры возможно описание разных конструкторов: с параметрами и без параметров
- d.** при объявлении структуры не допускается объявление конструктора без параметров
- e.** структура принадлежит к категории типов — тип-значение

9. Что такое перегрузка функции?

- a.** Множество функций с одним именем, но разной сигнатурой
- b.** Множество функций с разными именами, но одинаковой сигнатурой
- c.** Множество функций с разными именами и разной сигнатурой.

10. Из указанных типов выберите ссылочные типы данных

- | | | |
|--------------------------------|---------------------|--------------------------|
| a. Числовые типы данных | c. Структуры | e. Перечисления |
| b. Массивы | d. Строки | f. Классы |
| | | g. Логический тип |

11. Назначение операции new

- a.** Переименование объекта
- b.** Присваивание объекту нового значения
- c.** Создание нового объекта и вызов соответствующего конструктора

12. Что используют в языке C # для объявления набора связанных по смыслу констант (например, набора цветов) ясным и безопасным способом.

- a.** цикл for (...) **b.** struct **c.** enum **d.** одномерный массив

13. В результате выполнения фрагмента программы:

```
class Program
{
    static void F(int a)
    {
        ++a;
    }
    static void Main()
    {
        int a=5;
        F(a);
        Console.WriteLine( a);
    }
}
```

на экран будет выведено значение: **a.** 0 **b.** 5 **c.** 6

14. Нам необходимо написать метод `set_zero`, который будет использоваться для инициализации двух переменных целого типа. Он должен устанавливать в 0 их значения и возвращать в таком виде вызывающему методу. Выберите правильный заголовок для такого метода

- a.** `void set_zero(int x, int y);` **c.** `void set_zero(out int x, out int y);`
b. `void set_zero(ref int x, ref int y);` **d.** `int set_zero(int x, int y);`

15. Предположим, что у нас имеются два определения функции `sum` со следующими параметрами:

```
double sum(double x, double y);
double sum(double x, int y);
```

Какое из них будет использоваться в вызове `s=sum(5,10)`?

- a.** `double sum(double x, double y);`
b. `double sum(double x, int y);` **c.** такой вызов ошибочен

16. В результате выполнения фрагмента программы:

```
class Program
{
    static void F(ref int a)
    {
        ++a;
    }

    static void Main()
    {
        int a=5;
        F(ref a);
        Console.WriteLine( a);
    }
}
```

на экран будет выведено значение: **a.** 0 **b.** 5 **c.** 6

17. В результате выполнения фрагмента программы:

```
class Program
{
    static int F( int a)
    {
```

```

        return a%10;
    }
    static int F( int a, int b)
    {
        return (a+b)%10;
    }
    static void Main()
    {
        int a=17, b=21;
        Console.WriteLine(F(a, b));
    }
}

```

на экран будет выведено значение: **a.** 1 **b.** 7 **c.** 8

18. Дан рекурсивный метод:

```

static void F(int n)
{
    if (n>0) F(n-1);
    Console.Write(n+ " ");
}

```

Что будет выведено на экран при вызове метода F(5)?

a. 0 1 2 3 4 5 **b.** 5 4 3 2 1 0 **c.** 1 2 3 4 5 **d.** 5 4 3 2 1

19. Дан рекурсивный метод:

```

static void F(int n)
{
    Console.Write(2*n+ " ");
    if (n>1) F(n-2);
}

```

Что будет выведено на экран при вызове метода F(6)?

a. 0 4 8 12 **b.** 12 8 4 0 **c.** 4 8 12 **d.** 12 8 4

20. Дан рекурсивный метод:

```

static void F(int n)
{
    if (n>0) {F(n-2); Console.Write(n*n+ " ");}
}

```

Что будет выведено на экран при вызове метода F(4)?

a. 0 4 16 **b.** 16 4 0 **c.** 4 16 **d.** 16 4

21. При выполнении фрагмента программы:

```

static void Main(){
    try
    {
        int x = 5;
        if (x >0) throw new Exception();
        Console.WriteLine("ok");
    }
    catch
    {
        Console.WriteLine("введено недопустимое значение");
    }
}

```


на экран будет выведено:

a. ok

b. введено недопустимое значение

22. При выполнении фрагмента программы:

```
byte x = 20; byte y = 13;
try
{
    byte result = unchecked((byte)(x * y));
    Console.WriteLine(result);
}
catch (OverflowException)
{
    Console.WriteLine("возникло переполнение");
}
```

на экран будет выведено: **a.** 260

b. возникло переполнение

c. 6

d. 4

e. 5

23. Исключение в C# реализуется как

a. строка

b. объект

c. массив

d. число

24. Что будет выведено на экран после выполнения следующего программного фрагмента

```
try
{
    Console.WriteLine("Введите два числа");
    int x=int.Parse(Console.ReadLine());
    int y=int.Parse(Console.ReadLine());
    Console.WriteLine(x + "/" + y + "=" + x / y);
}
catch (FormatException)
{
    Console.WriteLine("Нужно ввести число!");
}
catch (DivideByZeroException)
{
    Console.WriteLine("Делить на нуль нельзя!");
}
```

если с клавиатуры будут введены числа

12345678987654321

12345678987654321

a. нужно ввести число!

c. случится аварийное прерывание

b. делить на нуль нельзя!

d. 1

25. Массив определен следующим образом:

```
int [][]a=new int [4][];
for (int i=0; i<a.Length; ++i)
    a[i]=new int [2*i-1]
```

В строке с индексом 2 содержится _____ элементов.

26. В метод массив передается:

a. по значению

b. по ссылке

c. в качестве выходного параметра.

27. В результате выполнения фрагмента программы:

```
int []a={1, 2, 4, 3, 1, 2};
int s=0;
for (int i=0; i<a.Length; i+=2)
    s+=a[i];
Console.WriteLine(s);
```

на экран будет выведено значение: _____

28. Массив объявлен следующим образом

```
int[] [] myArray = new int[3][];
myArray[0] = new int[2];
myArray[1] = new int[4];
myArray[2] = new int[3];
```

Какой из следующих программных фрагментов выводит на экран элементы массива построчно, следующим образом

```
0      1
10     11     12     13
```

a. for (int i = 0; i < myArray.Length; i++)
 {
 foreach (int x in myArray[i])
 Console.WriteLine("{0}\t",x);
 Console.WriteLine();
 }

c. for (int i = 0; i < myArray.Length; i++)
 {
 foreach (int x in myArray[i])
 Console.WriteLine("{0:D5}",x);
 Console.WriteLine();
 }

b. foreach (int x in myArray)
 Console.WriteLine("{0}\t",x);
 Console.WriteLine();

d. foreach (int x[j] in myArray[i])
 Console.WriteLine("{0}\t",x[i][j]);
 Console.WriteLine();

29. В результате выполнения фрагмента программы:

```
char [] a="abcde".ToCharArray();
Console.WriteLine(a[2]);
```

на экран будет выведено:

a. b

b. c

c. ab

30. В результате выполнения фрагмента программы:

```
string a = "кол около колокола";
char[] b = {'o'};
string[] c = a.Split(b);
Console.WriteLine(c.Length);
```

на экран будет выведено:

a. 2

b. 3

c. 7

d. 8

31. При объявлении строки типа StringBuilder вызов конструктора можно не использовать:

a. да

b. нет

32. В результате выполнения фрагмента программы:

```
StringBuilder str=new StringBuilder("курс доллара=");
str.AppendFormat("{0:c}", 10.5393);
Console.WriteLine(str);
```

на экран будет выведено:

a. курс доллара=10,5393 **b.** курс доллара=10,5393р. **c.** курс доллара=10,54
доллара=10,54р.

d. курс

33. В результате выполнения фрагмента программы:

```
StringBuilder str=new StringBuilder("кол около колокола");  
Console.WriteLine(str.Capacity);
```

на экран будет выведено:

a. 18

b. 32

34. Выберите верные высказывания:

- a.** индексатор можно передавать в метод как ref-параметр
- b.** индексатор можно перегружать
- c.** индексатор должен использовать базовый массив
- d.** можно создавать только одномерный индексатор

35. После выполнения следующего фрагмента кода на экран будет выведено

```
Hashtable hash = new Hashtable();  
hash.Add(1, "gold medal");  
hash.Add(2, "silver medal");  
hash.Add(3, "bronze medal");  
hash.Add(4, "no medal");  
hash.Remove(2);  
foreach (DictionaryEntry d in hash)  
    Console.WriteLine("{0} = {1}", d.Key, d.Value);
```

a.

4 = no medal
3 = bronze medal
2 = silver medal
1 = gold medal

b.

4 = no medal
3 = bronze medal
2 =
1 = gold medal

c.

4 = no medal
3 = bronze medal
1 = gold medal

36. Выберите справедливые утверждения:

- a.** массивы-экземпляры класса ArrayList могут расширяться
- b.** в классе ArrayList есть метод для упорядочивания элементов массива
- c.** индексы массива – экземпляра класса ArrayList не обязательно целые числа
- d.** массив - экземпляр класса ArrayList может иметь только одно измерение

37. Предположим, что коллекция SortedList хранит пары наименование товара (ключ)/вид товара (значение), например:

```
SortedList goods = new SortedList();  
goods.Add("кукла", "игрушка");  
goods.Add("тетрадь", "канцтовары");  
goods.Add("лото", "игра");
```

Каким из предложенных способов можно получить значение для "лото":

- a.** s1["лото"]
- b.** s1["игра"]
- c.** s1[goods]
- d.** s1[3]
- e.** s1.value

38. Поток FileStream является: **a.** байтовым **b.** двоичным **c.** символьным

39. Поток FileStream является потоком:

- a.** с последовательным доступом
- b.** с произвольным доступом

40. Для направления выходного потока FileStream на файл t.txt можно использовать конструктор:

- a.** new FileStream("t.txt", FileMode.Create, FileAccess.Write)
- b.** new FileStream("t.txt", FileMode.Truncate)
- c.** new FileStream("t.txt", FileMode.CreateNew , FileAccess.Read)

41. Для работы с потоками нужно использовать библиотеку:

- a.** System
- b.** System.IO
- c.** System.Text

42. Значение переменной c после выполнения фрагменты программного фрагмента

```
float a = 3, b = 2, c = a / b;
```

равно **a.** 1 **b.** фрагмент содержит ошибку **c.** 2 **d.** 1,5

43. Значение выражения $6.0 * 3 / 4$ равно... (Вычисления производить по правилам языка C#)

- a.** 4,5
- b.** 5,0
- c.** 4,0
- d.** 0
- e.** 6,0

44. Объект является образцом для создания класса:

- a.** да
- b.** нет

45. Полиморфизм проявляется сокрытием некоторых деталей реализации класса:

- a.** да
- b.** нет

46. Базовым классом в C# является класс System.Object:

- a.** да
- b.** нет

47. Значение переменной y после выполнения фрагмента программы:

```
int x,y;
```

```
x=14; y=x+10;  
y -= 4; y/=4;
```

равно: **a. 5** **b. 6** **c. 7** **d. 8**

48. Значение переменной m после выполнения фрагмента программы:

```
a=5; b=7;  
if (a<b) m=b-a;  
else m=a-b;
```

равно: **a. -2** **b. 0** **c. 2** **d. 12**

49. Значение переменной y после выполнения фрагмента программы:

```
double x,y;  
x=2;  
if (x<2) y=Math.Pow(x+1,2);  
else if (x>2) y=x+3;  
else y=0;
```

равно: **a. 0** **b. 3** **c. 5** **d. 9**

50. Значение переменной y после выполнения фрагмента программы:

```
double x,y;  
x=0;  
if (x<2) y=Math.Pow(x+1,2);  
if (x>2) y=x+3;  
else y=0;
```

равно: **a. 0** **b. 3** **c. 5** **d. 9**

51. Значение переменной y после выполнения фрагмента программы:

```
int x,y;  
x=3;  
x=10-2*x;  
switch (x)  
{  
  case 10: y=0; break;  
  case 8: case 9: y=1; break;  
  case 7: y=2; break;  
  default: y=3; break;  
}
```

равно: **a. 0** **b. 1** **c. 2** **d. 3**

52. Значение переменной y после выполнения фрагмента программы:

```
x=1;  
x=10-2*x;  
switch (x)  
{
```

```

    case 10: y=0; break;
    case 8: case 9: y=1; break;
    case 7: y=2; break;
    default: y=3; break;
}

```

равно: **a.** 0 **b.** 1 **c.** 2 **d.** 3

53. После выполнения фрагмента программы:

```

for (int i=1; i<=5; i++)
    Console.Write("{0}", i-2)

```

на экране будет напечатано: **a.** 1 2 3 4 5 **b.** -1 0 1 2 3 **c.** 5 4 3 2 1 **d.** 3 2 1 0 -1.

54. После выполнения фрагмента программы:

```

for (int i=1; i<=5; i++)
{
    for (int j=1; j<=i; j++)
        Console.Write (" {0} ", j);
    Console.WriteLine();
}

```

на экране будет напечатано:

a.	1 1 1 1 1	b.	1 2 3 4 5	c.	1	d.	1
	2 2 2 2 2		1 2 3 4 5		2 2		1 2
	3 3 3 3 3		1 2 3 4 5		3 3 3		1 2 3
	4 4 4 4 4		1 2 3 4 5		4 4 4 4		1 2 3 4
	5 5 5 5 5		1 2 3 4 5		5 5 5 5 5		1 2 3 4 5

55. После выполнения фрагмента программы:

```

int I=1;
while (I<=4)
{
    Console.WriteLine ("I ",I);
    I++;
}

```

на экране будет напечатано:

a.	I	b.	I I I I	c.	1	d.	1 2 3 4
	I				2		
	I				3		
	I				4		

5.12 Вопросы к экзамену

1. Типы C#. Категории, на которые делятся все типы C#. Перечислить базовые типы.
2. Явное и неявное (автоматическое) преобразование типов. Привести примеры.
3. Проверяемые преобразования. Операции checked и unchecked. Привести примеры.

4. Виды констант. Привести примеры.
5. *Перечисления*: назначение, синтаксис, примеры. Привести примеры.
6. *Организация ввода данных в программу и вывода данных*. Привести примеры.
7. Управление форматом вывода данных. Спецификаторы вывода. Привести примеры.
8. Процедуры и функции. Отличия. Привести примеры.
9. Описание методов (процедур и функций). Синтаксис. Привести примеры.
10. Описание формальных аргументов. Синтаксис. Виды аргументов метода. Привести примеры.
11. Перегрузка методов. Привести примеры.
12. Виды исключений. Стандартные исключения. Привести примеры.
13. Способы объявления и инициализации одномерных массивов. Привести примеры.
14. Методы и свойства класса Array. Привести примеры использования.
15. Способы объявления и инициализации многомерных массивов: прямоугольный, и ступенчатые массивы.
16. Символьный тип char, методы класса Char().
17. Неизменяемые строки string. Способы создания. Методы класса String().
18. Изменяемые строки StringBuilder. Способы создания. Методы класса StringBuilder().
19. Метасимволы в регулярных выражениях.
20. Конструкторы для описания регулярного выражения. Поиск в тексте по шаблону (методы).
21. Редактирование текста (методы класса Regex()).
22. Организация C#-системы ввода-вывода: дать определение потока, объяснить иерархию потоков ввода-вывода. Перенаправление стандартных потоков.
23. Работа с каталогами: абстрактный класс FileSystemInfo, классы DirectoryInfo и Directory.
24. Работа с дисками. Класс DriveInfo.
25. Работа с файлами: абстрактный класс FileSystemInfo, классы FileInfo и File.