МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА
(филиал) федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский технологический университет «МИСИС»
СТИ НИТУ «МИСИС»

Рабочая программа утверждена
решением Ученого совета
СТИ НИТУ «МИСИС»
от «20» июня 2023 г.
протокол № 5

Рабочая программа дисциплины

Программирование на языках высокого уровня

Закрепленная кафедра **Кафедра автоматизированных и информационных систем управления**

Квалификация **Бакалавр**

Форма обучения **Очная**

Общая трудоемкость **__4__ ЗЕТ**

Формы контроля в семестре:4

Часов по учебному плану 144

в том числе:

Экзамен, Курсовая работа, 4

аудиторные занятия 51

самостоятельная работа 48

часов на контроль 45

Семестр изучения ___4_

Распределение часов дисциплины по курсам

Семестр	4		Итого	
Вид занятий	УП	РП	УП	РП
Лекции	17	17	17	17
Лабораторные	17	17	17	17
Практические занятия.	17	17	17	17
Сам. работа	48	48	48	48
Часов на контроль	45	45	45	45
Итого:	144	144	144	144

Год набора 2020г.
В редакции 2023 г.

Программу составил: ассистент кафедры. АИСУ

Жуков Пётр Игоревич _____
Должность, уч.ст., уч.зв ФИО полностью *подпись*

Рабочая программа дисциплины

Программирование на языках высокого уровня

наименование

Разработана в соответствии с ОС ВО НИТУ «МИСИС»:
Самостоятельно устанавливаемый образовательный стандарт высшего образования – бакалавриат Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСИС» по направлению подготовки 09.03.02 Информационные системы и технологии (приказ от 05.03.2020 г. № 95 о.в.)

Составлена на основании учебного плана 2020 года набора.
09.03.02 Информационные системы и технологии,
Профиль: Информационные системы и технологии, утвержденного Ученым советом СТИ НИТУ «МИСИС» 22.06.2020 г., протокол № 23.

Рабочая программа одобрена на заседании кафедры Автоматизированных и информационных систем управления

Протокол от «24» мая 2023 г. № 3

И.о. зав. кафедрой АИСУ _____ Д.А. Полещенко
аббревиатура наименования кафедры *подпись* *И.О. Фамилия*

«24» мая 2023 г.

Руководитель ОПОП ВО
И.о. зав. каф. АИСУ, кандидат технических наук, доцент _____ Д.А. Полещенко
подпись *И.О. Фамилия*

1. ЦЕЛИ ОСВОЕНИЯ
Цель дисциплины получить теоретические знания и практические навыки по основам объектно-ориентированного программирования на языках высокого уровня, способствовать формированию представлений о языках программирования высокого уровня, формировать четкое представления об архитектуре современного программного обеспечения. Задачи дисциплины: 1. Научить основных подходам объектно-ориентированного программирования; 2. Развить алгоритмическое мышление; 3. Научить взаимодействовать со средами разработки программ; 4. Научить кодировать программы в соответствии с объектно-ориентированной парадигмой.

2. МЕСТО В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.ДВ.
2.1	**Требования к предварительной подготовке обучающихся:**
2.1.1	Компьютерное обеспечение специальности
2.1.2	Проектирование информационных систем
2.1.3	Современные информационные технологии
2.1.4	Системный анализ
2.1.5	Интеллектуальные системы управления
2.2	**Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее:**
2.2.1	Научно-исследовательская работа
2.2.2	Преддипломная практика для выполнения выпускной квалификационной работы
2.2.3	Защита выпускной квалификационной работы, включая подготовку к процедуре защите и процедуру защиты

3. ИНДИКАТОРЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ, СОВМЕЩЕННЫЕ С РЕЗУЛЬТАТАМИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.	
Знать:	ОПК-6-З1 – Современные методики проектирования ПО, модели и нотации.
Уметь:	ОПК-6-У1 – Выбирать соответствующие организационные и программно-аппаратные средства для реализации программного решения.
Владеть:	ОПК-6-В1 – Навыками проектирования ПО при помощи современных моделей и нотаций.
ОПК-8: Способен применять математические модели, методы и средства проектирования информационных и автоматизированных систем.	
Знать:	ОПК-8-З1 – Математические основы алгоритмов и алгоритмизации ОПК-8-З2 – Организационно-экономические модели жизненного цикла ПО
Уметь:	ОПК-8-У1 – Интерпретировать математические формулировки алгоритмических задач ОПК-8-У2 – Осуществлять выбор моделей жизненного цикла
Владеть:	ОПК-8-В1 – Навыками применения объектно-ориентированного подхода в решении практических задач
ПК-1: Способен участвовать в работах по: • **доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных технологий;**	

• **поддержанию работоспособности информационных систем и технологий в заданных функциональных характеристиках и в соответствии с критериями качества;** • **управлению программно-аппаратными средствами информационных служб инфокоммуникационной системы организации; - обеспечению безопасности и целостности данных информационных систем и технологий, а также условий жизненного цикла информационных систем;** • **адаптации приложений к изменяющимся условиям функционирования;** • **администрированию сетевой подсистемы инфокоммуникационной системы организации.** **Использовать технологии разработки объектов профессиональной деятельности в различных областях.**	
Знать:	ПК-1-З1 – Синтаксис языка высокого уровня (C#) и его основных библиотек
Уметь:	ПК-1-У1 – Тестировать ПО, написанное на языках высокого уровня, и выявлять уязвимые места
Владеть:	ПК-1-В1 – Инструментами отладки и тестирования кода языков высокого уровня.

4. СТРУКТУРА И СОДЕРЖАНИЕ						
Код занятия	**Наименование разделов и тем /вид занятия/**	**Семестр**	**Часов**	**Компетен-ции**	**Литература и эл. ресурсы**	**Примечание**
	Раздел 1. Структура dotNetCore и знакомство с C#					
1.1.	Структура программы в dotNetCore. Базовые сведений о CLR/Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
1.2	Выявление ошибок в ПО и причин их появления /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
1.3	Основные типы данных и операторы языка C#/Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э2 Э3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	
1.4	Проработка лекционного материала. Самостоятельное изучение литературы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
1.5	Контроль знаний по разделу /Кр/	4	4	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 2. Основы ООП					
2.1	Классы и объекты. Элементы классов /Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
2.2	Тестирование «белым ящиком» /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
2.3	Массивы, кортежи и строки / Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	

2.4	Проработка лекционного материала. Самостоятельное изучение литературы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
2.5	Контроль знаний по разделу /Кр/	4	5	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 3. Наследование					
3.1	Принципы наследования классов/Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
3.2	Тестирование «черным ящиком» /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
3.3	Работа со строками и файлами/Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	
3.4	Проработка лекционного материала. Самостоятельное изучение литературы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
3.5	Контроль знаний по разделу /Кр/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 4. Инкапсуляция					
4.1	Инкапсуляция в ООП. Свойства классов. Геттеры и Сеттеры /Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
4.2	Модульное тестирование /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
4.3	Простейшие классы /Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	

4.4	Проработка лекционного материала. Самостоятельное изучение литературы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
4.5	Контроль знаний по разделу /Кр/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 5. Основной объектно-ориентированный код					
5.1	Особые приемы объектно-ориентированного кода /Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
5.2	Интеграционное тестирование Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
5.3	Структуры в C#/Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	
5.4	Проработка лекционного материала. Самостоятельное изучение литературы. Выполнение курсовой работы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
5.5	Контроль знаний по разделу /Кр/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 6. Расширенный объектно-ориентированный подход					
6.1	Модификаторы доступа. Свойства. Nullable-контекст /Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
6.2	Классы эквивалентности и ручное тестирование /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
6.3	Классы и операции в C#/Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	

				ОПК-8-В1		
6.4	Проработка лекционного материала. Самостоятельное изучение литературы. Выполнение курсовой работы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
6.5	Контроль знаний по разделу /Кр/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 7. Полиморфизм					
7.1	Нестандартные виды полиморфизма. Перегрузка перегрузки операторов. Нюансы полиморфизма в C#/Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
7.2	Тестовое окружение /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
7.3	Абстрактные классы в C# /Лр/	4	2	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	
7.4	Проработка лекционного материала. Самостоятельное изучение литературы. Выполнение курсовой работы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
7.5	Контроль знаний по разделу /Кр/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	
	Раздел 8. Повышенный уровень владения C#					
8.1	Введение в интерфейсы /Лек/	4	1	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
8.2	Делегаты. Лямбда-выражения /Лек/	4	2	ОПК-6-З1 ОПК-8-З1 ОПК-8-З2 ПК-1-З1 ПК-1-У1	Э 3 Л 1.1 Л 1.2 Л 1.3 Л 2.1	
8.3	Разработка документации для программного кода /Пр/	4	2	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	

8.4	Формальные инспекции кода /Пр/	4	1	ПК-1-У1 ПК-1-В1	Э 2 Э 3 Л 2.2 Л 3.1	
8.5	Интерфейсы в C# /Лр/	4	3	ОПК-6-З1 ОПК-6-У1 ОПК-6-В1 ПК-1-З1 ОПК-8-З1 ОПК-8-У1 ОПК-8-В1	Э 2 Э 3 Л 1.3 Л 2.1 Л 2.3 Л 3.3	
8.6	Проработка лекционного материала. Самостоятельное изучение литературы. Выполнение курсовой работы. Подготовка к лабораторным и практическим работам/Ср/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1	Э 2 Э 3 Л 1.1 Л 1.3 Л 2.1 Л 2.2 Л 2.3 Л 3.1 Л 3.2 Л 3.3	
8.7	Контроль знаний по разделу /Кр/	4	6	ОПК-8-З1 ОПК-8-З2 ОПК-6-З1 ПК-1-З1 ОПК-8-В1 ОПК-6-В1 ПК-1-В1	Л 3.1 Л 3.2 Л 3.3	

5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

5.1. Вопросы для самостоятельной подготовки к экзамену (зачёту с оценкой)

5.1.1. Перечень вопросов для подготовки к экзамену.
(промежуточный контроль успеваемости)

1. Что такое точка входа в приложение? (ОПК-6-З1);
2. Опишите семантику определения метода. (ОПК-8-В1);
3. Что такое пространство имен, и каково его назначение? (ОПК-6-З1);
4. Типы значений и ссылочные типы. (ОПК-8-В1);
5. В чем отличие в передаче аргументов методу по значению от передачи аргументов по ссылке? (ОПК-8-В1)
6. Что такое аргументы метода? (ОПК-8-В1*)*;
7. Что такое пространство объявления идентификатора? (ОПК-8-В1);
8. В чем заключается концепция рекурсивных методов? (ОПК-6-З1);
9. В чем заключается концепция перегрузки методов? (ОПК-6-З1);
10. Что такое сигнатура метода? (ОПК-8-В1);
11. Что такое модификаторы параметров? (ОПК-6-З1);
12. Что такое передача параметров по значению? (ОПК-6-З1);
13. Опишите концепцию необязательных аргументов. (ОПК-6-З1);
14. Что такое именованные аргументы? (ОПК-8-В1);
15. Определение массива объектов. (ОПК-6-З1);
16. Возможно ли передать массив в качестве аргумента? Если да, то как? (ОПК-6-З1);
17. Какой тип данных используется для перечисления? Опишите его семантику. (ОПК-6-В1);
18. Какой тип по умолчанию используется для хранения значений перечисления? (ОПК-6-З1);
19. Типы структур. Определение структур и их использование. (ОПК-6-В1);
20. Что такое класс? (ОПК-8-В1);
21. Ключевое слово new. (ПК-1-З1);
22. Модификаторы доступа public и private. (ПК-1-З1);
23. Модификаторы доступа protected и internal. (ПК-1-З1);
24. Экземпляр класса. (ОПК-6-З1);
25. Конструктор класса. (ОПК-6-З1);
26. Ключевое слово this. (ПК-1-З1);
27. Вложенные типы. (ОПК-8-З1);
28. Наследование свойств класса. (ОПК-8-З1);
29. Инкапсуляция с помощью свойств. (ОПК-8-З1);
30. Понятие транслятора. Интерпретатор и компилятор (ОПК-6-З1);
31. Модификаторы доступа в C#. Какой модификаторы доступа установлен для struct по умолчанию? (ПК-1-З1);
32. Виды массивов в C#. Как инициализируется каждый вид массива? Рассказать про класс Array, привести примеры его статичных методов (ПК-1-З1);
33. Что такое индексаторы в классах? Чем метод for отличается от foreach? (ПК-1-З1);
34. Полиморфизм и наследование. Возможность переопределять методы класса-родителя в классе-потомке – это полиморфизм или наследование. Обосновать почему. (ОПК-8-В1);

35. Инкапсуляция в ООП. Основные подходы к ней в C#. Каким образом происходит инкапсуляция полей в классах C#? (ОПК-8-В1, ПК-1-З1);
36. Конструкторы и деструкторы классов в C#. Как явным образом объявить деструктор в классе C#? (ПК-1-З1);
37. Коллекции в C#. Какие бывают и чем отличаются друг от друга? Как обратится к элементу коллекции в C#? (ПК-1-З1);
38. Произвольный тип данных var, каким образом транслятор понимает к какому типу данных его нужно привести? (ПК-1-З1);
39. Приведение типов данных в C#. Какие типы данных можно привести явно, а для каких требуется неявное приведение? От какого типа от наследованы все типы данных в C#? (ПК-1-З1);
40. Статический класс, чем он отличается от обычного? Что такое сигнатура с массивом параметров? (ПК-1-З1);
41. Можно ли применить сеттер (set) без указания геттера (get)? Почему? (ОПК-8-В1);
42. Перегрузка операндов. Какие операнды можно перегружать, а какие нельзя? Что такое перегрузка перегрузки операнда? (ОПК-8-В1);
43. Что такое LINQ? Для чего нужен? (ПК-1-З1);
44. Чем виртуальный метод отличается от абстрактного? (ОПК-8-В1);
45. «Стек» и «Куча». В чем отличие и зачем нужно это разделение? (ПК-1-З1);
46. Интерфейсы и абстрактные классы. Для чего используется интерфейс? Чем отличается интерфейс от абстрактного класса? (ОПК-8-В1);
47. Какой интерфейс отвечает за работу с деструкторами в C#? Как работает оператор using? (ПК-1-З1).

5.1.2. Перечень контрольных заданий, используемых при формировании экзаменационных билетов при оценке знаний обучающихся на экзамене

1. Написать программу, которая меняет местами значения в двух массивах, не прибегая к третьему (временному массиву) и не используя реализации статичных классов C# (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);
2. Написать программу, которая находит оптимальную загрузку судна (Vmax = 2000 м3), при условии, что имеется всего два продукта (А1 и А2) с фиксированной ценой СА1 = 80 000р за контейнер и СА2 = 32 000р за контейнер, при этом занимаемый каждым контейнером объем определен: VA1 = 50м3, VA2 = 10 м3. (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);
3. Написать программу, которая превращает число А в число Б, при условии, что можно выполнять только две операции: умножение на 2 и добавление к имеющемуся числу справа цифру 1. Число А и Б задаются с клавиатуры, в качестве результата должны выводиться на экран последовательности заданных операций, либо «Нет ответа», если приведение А к Б не возможно (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);.
4. В качестве сувениров на соревновании было решено вручить футболки. Всего в типографии были напечатаны футболки шести размеров: S, M, L, XL, XXL, XXXL (размеры перечислены в порядке возрастания). Для каждого размера от S до XXXL вам известно количество футболок такого размера. Во время регистрации организаторы попросили каждого из n участников указать размер футболки. Если участник колебался между двумя размерами, то он мог указать два соседних — это означает, что ему подойдет футболка любого из двух размеров. Напишите программу, которая определит, возможно ли из напечатанных в типографии футболок сделать подарок каждому участнику соревнования. Конечно, каждому участнику должна достаться футболка его размера: требуемого размера, если указан один размер; любого из двух размеров, если указаны два соседних размера. В случае положительного ответа программа должна найти любой из вариантов раздачи футболок (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);.
5. Пусть имеется зубчатый массив из n подмассивов. Размерность первого подмассива 2, размерность каждого последующего +1. Массивы заполняются числами Фибоначчи, примерно, следующим образом:
   • 1 1
   • 1 2 1
   • 1 3 2 3 1
   • 1 4 3 5 2 5 3 4 1
   Написать программу, которая самостоятельно заполняет массив числами Фибоначчи, и которая по заданному числу n определит сколько раз число n будет встречаться в n-ом массиве. (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);
6. Пусть дана последовательность целых чисел размерностью N, значения которой определены в диапазоне от -100 до 100. Написать программу, которая находит максимальную сумму между двумя отрицательными простыми числами. (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);
7. Пусть дана последовательность целых чисел размерностью N, значения которой определены в диапазоне от -100 до 100. Написать программу, которая находит максимальную сумму между двумя отрицательными простыми числами. Имеется строка s на входе, разделение этой строки на 2 непустые подстроки p и q называется хорошим тогда и только тогда, когда конкатенация этих строк дает строку s и количество различных элементов в подстроке p равно количеству различных в q (без повторений). Написать программу, которая на выходе возвращает количество "хороших" разделений строки s, при условии, что s содержит элементы из множества ['a', 'b'...'z'] и
   s.Length >1;
   s.Length <1E+05.
   Пример для отладки:
   p = "aa", q = "caba" - не является "хорошей";
   p = "aac", q = "aba" - является "хорошей";
   p = "aaca", q = "ba" - является "хорошей" (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);

8. Пусть имеется необходимость отслеживать количество бракованных аккумуляторов. Известно, что каждый из приготовленных аккумуляторов может оказаться бракованный с вероятностью в 2%. При этом, на этапе упаковки, аккумулятор рабочий может быть определен как бракованный с вероятностью 1%, при этом неработающие аккумуляторы определяются с вероятностью в 98%. Известно, что с момента запуска производства было поставлено K партий аккумуляторов из N штук. Цена одного аккумулятора 10000р, написать программу, которая находит общую стоимость издержек на неисправные аккумуляторы. (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);
9. Пусть дан массив целых чисел, в количестве от 0 до 100*N. Значения чисел в нем сгенерированы случайно в диапазоне от 0 до 1000*N. Написать программу, которая вычисляет количество цифр «2», которые встречаются во всем диапазоне чисел. (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);
10. Пусть даны две строки S1 и S2 произвольной длины. Написать такую программу, которая определяет являются ли строки анаграммами. S1 является анаграммой к S2, если из S1 можно без разрывов (пробелов) получить S2. (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-У1, ОПК-8-В1, ПК-1-З1, ПК-1-У1);

5.1.3. ПЕРЕЧЕНЬ ВОПРОСОВ К ЗАЩИТЕ ЛАБОРАТОРНЫХ РАБОТ

Лабораторная работа №1

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. Что такое точка входа в приложение?
2. Опишите семантику определения метода.
3. Что такое пространство имен, и каково его назначение?
4. Типы значений и ссылочные типы.

Лабораторная работа №2

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. В чем отличие в передаче аргументов методу по значению от передачи аргументов по ссылке?
2. Аргументы метода.
3. Пространство объявления идентификатора.
4. Рекурсивный метод.

Лабораторная работа №3

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. Рекурсивный метод.
2. Концепция перегрузки методов
3. Сигнатура метода.
4. Модификаторы параметров.

Лабораторная работа №4

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. Передача параметров по значению.
2. Необязательные аргументы.
3. Именованные аргументы.
4. Определение массива объектов.

Лабораторная работа 5

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. Возможно ли передать массив в качестве аргумента? Если да, то как?
2. Какой тип данных используется для перечисления? Опишите его семантику.
3. Какой тип по умолчанию используется для хранения значений перечисления?
4. Типы структур. Определение структур и их использование

Лабораторная работа 6

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. Что такое класс?
2. Ключевое слово new.
3. Модификаторы доступа public и private.
4. Модификаторы доступа protected и internal

Лабораторная работа 7-8

(ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1)

1. Экземпляр класса.
2. Конструктор класса.
3. Ключевое слово this.
4. Вложенные типы.
5. Наследование свойств класса.
6. Инкапсуляция с помощью свойств.

5.2. Перечень работ, выполняемых по дисциплине

В перечень работ, выполняемых в процессе изучения дисциплины в семестре 4 включены:

1. **Лабораторные работы** ((ОПК-6-З1, ОПК-6-З1, УК-2-В2, ОПК-6, ОПК-8 ПК-1-З1, ПК-1-У1, ПК-1-В1);
2. **Практические работы (**ОПК-6-З1, ОПК-6-У1, ОПК-6-В1, ПК-1-З1, ПК-1-У1, ПК-1-В1**);**
3. **Курсовая работа** (ОПК-6-З1, ОПК-6-З1, ОПК-6-З1, ОПК-6-З1, ОПК-6-В1, ОПК-8-З1, ОПК-8-З2, ОПК-8-В1, ПК-1-З1, ПК-1-У1, ПК-1-В1)

Тема: «Проектная реализация программного продукта по индивидуальному ТЗ»

Задание для данной курсовой работы предполагает проверку всех знаний синтаксиса языка C# и навыков алгоритмизации, полученных в результате выполнения лабораторных работ по данному предмету. Курсовая работа предполагает творческое задание на подобии реальной задачи программирования, упрощенной до базового уровня владения языком. Выдача академических «учебных» задач и распределение по вариантам не предусматривается.

Пример ТЗ на естественном языке:

- Спроектировать и реализовать игру «Крестики-нолики» (Tik-tac-toe) в графическом интерфейсе (или псевдографическом) для игры вдвоем, где пользователи делают ходы по очереди. Предусмотреть наличие счетчика побед и сохранения результата в файл. Подход к проектированию свободный. Ограничение – язык C#;

Курсовая работа должна содержать:

Отчет должен включать:

1. Титульный лист
2. Лист консультаций (см. приложение А Л 3.2 данного РПД);
3. Лист задания (см. приложение Б Л 3.2 данного РПД);
4. Содержание;
5. Введение;
6. Аналитическая часть:
    a. Постановка задачи;
    b. Декомпозиция задачи на подзадачи;
    c. Проектирование классов:
        i. Разработка классовой структуры;
        ii. Выделение основных классов;
        iii. Создание UML диаграммы классов (статический вариант);
        iv. Определения типов и видов наследования;
        v. Формальное описание каждого класса на естественном языке с описанием полей, методов, свойств.
    d. Проектирование бизнес-логики:
        i. Разработка алгоритмов в соответствии с подзадачами (п.5b);
        ii. Формальное описание алгоритмов с детализацией функционала (если имеются формулы для расчета, то указать их).
7. Практическая часть:
    a. Описание используемых библиотек;
    b. Листинг кода реализации отдельных классов (код должен быть за комментирован);
    c. Листинг программы с точкой входа (код должен быть за комментирован);
8. Заключение;
9. Список использованных источников (не менее 10, включая корректно оформленные ссылки на интернет-источники. См. ГОСТ 7.1-2003 и ГОСТ 7.0.100-2018).

Подход к реализации, выбору программно-аппаратных средств и технологий свободный. Защита курсовой работы осуществляется путем защиты кода программы и алгоритмов в нотации блок-схем.

5.2.1 Перечень тестовых вопросов

(ОПК-6-З1, ОПК-6-З1, ОПК-6-З1, ОПК-6-В1, ОПК-8-З1, ОПК-8-З2, ОПК-8-В1, ПК-1-З1, ПК-1-В1)

Алгоритмизация (ОПК-6-З1, ОПК-6-З1, ОПК-6-В1, ОПК-8-З1, ОПК-8-В1):

1. Что такое алгоритм:
    a) Указание на выполнение действий;
    b) Процесс выполнения вычислений, приводящих к решению задачи;
    c) Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи;
2. Одно из свойств алгоритма:
    a) Цикличность;
    b) Массовость;
    c) Оперативность.
3. Одно из свойств алгоритма:
    a) Информативность;
    b) Оперативность;
    c) Определенность.
4. Одно из свойств алгоритма:
    a) Дискретность
    b) Цикличность
    c) Информативность
5. Алгоритм может быть задан следующим способом:
    a) Последовательностью байтов;
    b) Словесно-графическим;
    c) На алгоритмическом языке;
6. Алгоритм может быть задан следующим способом:
    a) Словесно-графическим;

b) Графическим;
c) Последовательностью байтов.

7. Как называется определенная последовательность действий, которую нужно выполнить для решения конкретной задачи:
   a) Алгоритм;
   b) Программа;
   c) Система команд исполнителя.
8. О каком свойстве алгоритма идет речь в данном примере: алгоритм должен быть применим для целого класса подобных задач, отвечающих общим условиям:
   a) Дискретность;
   b) Однозначность;
   c) Массовость;
9. В каком варианте записана команда присваивания:
   a) A>D;
   b) X:=X+Y;
   c) X+Y:=x.
10. Алгоритм должен состоять из отдельных шагов, как называется свойство:
   a) Дискретность;
   b) Однозначность;
   c) Массовость;
11. Как называется алгоритм, записанный на понятном компьютеру языке:
   a) Блок-схемой;
   b) Программой;
   c) Исполнителем;
12. Выберите верное утверждение:
   a) Значение переменной может изменяться в процессе выполнения алгоритма;
   b) При присваивании переменной какого-либо значения предыдущее её значение сохраняется автоматически;
   c) Одна величина может иметь несколько типов;
13. Выберите верное утверждение:
   a) Одна величина может иметь несколько типов;
   b) При присваивании переменной какого-либо значения предыдущее её значение сохраняется автоматически;
   c) Величина логического типа может принимать всего два значения.
14. Укажите логическое выражение:
   a) X+7>=0;
   b) X:=7;
   c) X-7.
15. Укажите логическое выражение:
   a) X+7;
   b) N=10;
   c) X:=7.
16. Алгоритм называется линейным, если:
   a) Он предполагает многократное повторение одних и тех же действий;
   b) Ход его выполнения зависит от истинности тех или иных условий;
   c) Его команды выполняются в порядке следования друг за другом;
17. Геометрическая фигура ромб используется в блок-схемах для обозначения:
   a) Начала или конца алгоритма;
   b) Принятия решения;
   c) Ввода или вывода.
18. Алгоритм включает в себя ветвление, если:
   a) Он предполагает многократное повторение одних и тех же действий;
   b) Его команды записаны в табличной форме;
   c) Ход его выполнения зависит от истинности тех или иных условий.
19. Геометрическая фигура параллелограмм используется в блок-схемах для обозначения:
   a) Выполнения действия;
   b) Ввода или вывода;
   c) Принятия решения.
20. Необходимо выбрать верное представление арифметического выражения на алгоритмическом языке:
   a) (x + 3*y) / (5*x*y);
   b) x + 3*y / 5*xy;
   c) x + 3y / 5xy.
21. Геометрическая фигура прямоугольник используется в блок-схемах для обозначения:
   a) Принятия решения;
   b) Выполнения действия;
   c) Ввода или вывода.
22. Что можно считать алгоритмом:

a) Список класса в журнале;
b) Расписание уроков в школе;
c) Описание решения квадратного уравнения.

23. Свойство алгоритма, означающее, что он всегда приводит к результату через конечное, возможно, очень большое, число шагов, называется:
    a) Дискретность;
    b) Результативность;
    c) Понятность.
24. Свойство алгоритма, означающее, что он задан с помощью таких предписаний, которые исполнитель может воспринимать и по которым может выполнять требуемые действия, называется:
    a) Определенность;
    b) Дискретность;
    c) Понятность.
25. Наибольшей наглядностью обладают такие формы записи алгоритмов:
    a) Словесные;
    b) Графические;
    c) Рекурсивные.
26. Геометрическая фигура овал используется в блок-схемах для обозначения:
    a) Начала или конца алгоритма;
    b) Ввода или вывода;
    c) Выполнения действия.
27. Программа-интерпретатор выполняет:
    a) Поиск файлов на диске;
    b) Полное выполнение программы;
    c) Пооператорное выполнение программы.

Язык программирования C# (ОПК-6-З1, ПК-1-З1, ПК-1-В1, ОПК-8-З1, ОПК-8-В1):

28. Какой тип переменной используется в коде: int a = 5:
    a) Знаковое 8-бит целое;
    b) Беззнаковое 1-байт целое;
    c) Знаковое 8-байт целое;
    d) Беззнаковое 32-бит целое.
29. Что делает оператор «%»
    a) Возвращает процент от суммы;
    b) Возвращает остаток от деления;
    c) Возвращает тригонометрическую функцию;
    d) Ни чего из вышеперечисленного.
30. Что сделает программа, выполнив следующий код: Console.WriteLine(«Hello, World!»):
    a) Напишет на новой строчке Hello, World!;
    b) Напишет Hello, World!;
    c) Удалит все значения с Hello, World!;
    d) Вырежет слово Hello, World! из всего текста.
31. Как сделать инкрементацию числа
    a) ++
    b) --
    c) %%
    d) !=
32. Как сделать декрементация числа
    a) %%
    b) --
    c) !=
    d) ++
33. Как найти квадратный корень из числа x
    a) Sqrt(x);
    b) Summ.Koren(x);
    c) Arifmetic.sqrt(x);
    d) Math.Sqrt(x).
34. Обозначения оператора «НЕ»
    a) Not
    b) No
    c) !
    d) !=
35. Обозначение оператора «ИЛИ»
    a) !
    b) !=

c) ||
d) Or

36. 9. Обозначение оператора «И»
    a) and
    b) &&
    c) &
    d) Все вышеперечисленные.
37. Чему будет равен c, если int a = 10; int b = 4; int c = a % b;
    a) 11;
    b) 2;
    c) 3;
    d) 1.
38. Чему будет равен c, если int a = 10; int b = 4; bool c = (a == 10 && b == 4);
    a) True;
    b) False;
    c) Null;
    d) 14;
39. Чему будет равен c, если int a = 0; int c = a--;
    a) Null;
    b) -1;
    c) 0;
    d) 1.
40. Чему будет равен c, если int a = 0; int c = --a;
    a) Null;
    b) -1;
    c) 0;
    d) 1.
41. Чему равен d, если int a = 0; int b = a++; int c = 0; int d = a + b + c + 3;
    a) 3;
    b) True;
    c) False;
    d) 4.
42. Для чего нужны условные операторы
    a) Чтобы устанавливать условия пользователю;
    b) Для ветвления программы;
    c) Для оптимизации программы;
    d) Чтобы были.
43. Как называется оператор «?:»
    a) Вопросительный;
    b) Прямой оператор;
    c) Тернарный оператор;
    d) Территориальный оператор.
44. Что такое массив
    a) Набор однотипных данных, которые располагаются в памяти последовательно друг за другом;
    b) Набор текстовых значений в формате Unicode, которые расположены в случайном порядке;
    c) Набор данных типа int (32-бит целое);
    d) Переменная.
45. Какие бывают массивы?
    a) Одноранговые и многоранговые;
    b) Сложные и простые;
    c) Одномерные и многомерные.
    d) Определенные и неопределенные.
46. Что такое цикл и для чего они нужны
    a) Циклы нужны для многократного запуска программы;
    b) Циклы нужны для многократного выполнения кода;
    c) Циклы нужны для многократного размещения данных;
    d) Циклы нужны чтобы выполнить код без ошибок.
47. Какие бывают циклы?
    a) Большие и маленькие;
    b) Цикл, Форич, Двойной цикл, Многократный;
    c) for, while, do-while, foreach;
    d) ref, out, static, root.
48. Какой оператор возвращает значение из метода?
    a) veni;
    b) return;

c) out;
d) end;

49. Что такое константа?
    a) Переменная типа string$
    b) Переменная, которая может быть изменена в любое время$
    c) Глобальная переменная$
    d) Переменная значение которой нельзя изменить.
50. Что обозначает ключевое слово var?
    a) Объявляет пустую переменную;
    b) Обозначает что переменная без явного типа данных;
    c) Обозначает что переменная имеет явный тип данных;
    d) Такого слова нет в синтаксисе C#;
51. Что такое Куча?
    a) Это структура данных;
    b) Именованная область памяти;
    c) Область динамической памяти;
    d) Куча переменных.
52. Алгоритмы, целиком используемые в составе других алгоритмов, называются…
    a) Линейными;
    b) Определенными;
    c) Разветвляющимися;
    d) Вспомогательными;
    e) Циклическими.
53. Выберите выражение, в результате которого получено значение c=2, если a=14 и b=4
    a) c:= a mod b;
    b) c:= b div a;
    c) c:= a div b;
    d) c:= b mod a;
    e) c:= a/b;
54. Сплошной массив байтов, располагающийся в старших адресах сразу за областью памяти, которую занимает тело программы
    a) Куча;
    b) Сегмент данных;
    c) Стек данных;
    d) Тело программы;
    e) Динамика.

Объектно-ориентированный подход (ОПК-6-З1, ОПК-6-З1, ОПК-8-З1, ОПК-8-В1, ПК-1-З1):

55. Свойство, при котором объекты содержат описание атрибутов и действий одновременно:
    a) Наследование;
    b) Полиморфизм;
    c) Инкапсуляция.
56. Свойство, при котором объекты содержат описание атрибутов и действий одновременно:
    a) Наследование;
    b) Полиморфизм;
    c) Инкапсуляция.
57. Один из принципов объектно-ориентированного программирования:
    a) Инкапсуляция;
    b) Ингаляция;
    c) Инструкция;
58. Метод определения объектов, при котором производные объекты наследуют свойства от своих потомков:
    a) Монорфизм;
    b) Полиморфизм;
    c) Наследование.
59. Свойство объектов, при котором действие с одинаковыми именами вызывает различное поведение для различных объектов:
    a) Полиморфизм;
    b) Передача;
    c) Монорфизм.
60. Данные, характеризующие состояние объекта:
    a) Доли объекта;
    b) Части объекта;
    c) Атрибуты объекта.
61. Под объектами понимают:
    a) Всю абстрактную сущность, заданную набором имен атрибутов и имен методов поведения;

b) Некоторую абстрактную сущность, заданную набором имен атрибутов и имен методов поведения;
c) Некоторую видимую сущность, заданную набором имен атрибутов и имен методов поведения.

62. Совокупность объектов, характеризующаяся общностью методов и свойств:
    a) Класс;
    b) Вид.
63. в) Род Изменение состояния объекта в ответ на какое-либо действие:
    a) Значение;
    b) Событие;
    c) Данность.
64. Действие, которое может выполнить объект:
    a) Метод;
    b) Событие;
    c) Свойство.
65. Характеристика объекта:
    a) Событие;
    b) Данность;
    c) Свойство.
66. Совокупность свойств и методов:
    a) Объект;
    b) Свойство;
    c) Событие.
67. Тип, соответствующий классу:
    a) Объективный тип;
    b) Объектный тип;
    c) Видимый тип.
68. Компоненты, которые видны во время работы приложения, с ними напрямую может взаимодействовать пользователь, называются:
    a) Абстрактными;
    b) Видимыми;
    c) Визуальными.
69. Какая из перечисленных функций не может быть конструктором:
    a) String (const int a);
    b) String (String & s);
    c) void String ().
70. Свойство Name отвечает за:
    a) Название компонента;
    b) Имя компонента;
    c) Назначение компонента.
71. Имя формы, используется для управления формой и доступа к компонентам формы:
    a) а) Свойство формы Name;
    b) б) Значение формы Name;
    c) в) Следствие формы Name.
72. Расстояние от верхней границы формы до верхней границы экрана:
    a) а) Значение формы Top;
    b) б) Следствие формы Top;
    c) в) Свойство формы Top.
73. Этот компонент предназначен для вывода текста на поверхность формы:
    a) Label;
    b) Edit;
    c) Button.
74. Компонент, представляющий собой поле ввода-редактирования строки символов:
    a) Memo;
    b) Edit;
    c) CheckBox.

Нотации и средства проектирования (ОПК-6-З1, ОПК-8-З2, ОПК-8-В1, ПК-1-З1, ПК-1-В1):

75. Модель жизненного цикла ПО, в которой отсутствует стадия промежуточного тестирования:
    a) Круговая;
    b) Спиральная;
    c) Каскадная;
    d) Одиночная.
76. Выберите неверное утверждение:
    a) Модель жизненного цикла используется для определения состава и последовательности выполнения работ;
    b) Модель жизненного цикла используется для определения времени эксплуатации ИС;
    c) Модель жизненного цикла используется для определения количества ресурсов, необходимое при

разработке.

77. Какой модели жизненного цикла не существует:
    a) Ветвистой;
    b) V-образной;
    c) Спиральной;
    d) Каскадной.
78. Какая из UML диаграмм используется для проектирования классовой структуры кода:
    a) Статическая диаграмма UML;
    b) Use-Case диаграмме;
    c) Диаграмма последовательностей;
    d) Динамическая диаграмма UML.
79. Вид связи, при котором производные классы могут существовать только вместе с существованием исходного класса называется…
    a) Агрегаций;
    b) Композиция;
    c) Обобщение;
    d) Наследование.
80. Вид связи, при котором производные классы семантически (по смыслу) указывают друг на друга называется…
    a) Агрегация;
    b) Ассоциация;
    c) Обобщение;
    d) Композиция.
81. Вид связи, при котором производные классы обобщаются видом «частное-целое» называется…
    a) Агрегация;
    b) Обобщение;
    c) Композиция;
    d) Ассоциация.
82. В нотации UML статических классов используется символ «#». Для чего он?
    a) Описывает атрибут класса;
    b) Описывает свойства класса;
    c) Описывает модификатор доступа к элементу класса;
    d) Описывает метод класса.
83. В UML статических классов для отметки элемента, как public (общий) используют следующий знак:
    a) #;
    b) –;
    c) +;
    d) &.
84. В UML статических классов для отметки элемента, как private (частный) используют следующий знак:
    a) #;
    b) –;
    c) +;
    d) &.
85. В UML статических классов для отметки элемента, как protected (защищенный) используют следующий знак:
    a) #;
    b) –;
    c) +;
    d) &.
86. Какого типа связи не существует в UML:
    a) Зависимость;
    b) Ассоциация
    c) Реализация;
    d) Использование.
87. На схемах классовых структур UML стрелка с закрашенным ромбом символизирует связь вида…
    a) Композиция;
    b) Агрегация;
    c) Реализация;
    d) Зависимость.
88. На схемах классовых структур UML стрелка с не закрашенным ромбом символизирует связь вида…
    a) Композиция;
    b) Агрегация;
    c) Реализация;
    d) Зависимость.
89. На схемах классовых структур UML пунктирная стрелка символизирует связь вида…
    a) Композиция;
    b) Агрегация;

c) Реализация;
d) Зависимость.

90. Пунктирная линия на блок-схемах алгоритмов обозначает…
    a) Потоки управлений;
    b) Потоки сообщений;
    c) Потоки преобразований;
    d) Информационные потоки.
91. Интерфейсы в нотации UML должны иметь связь:
    a) Агрегация;
    b) Композиция;
    c) Реализация;
    d) Ассоциация.
92. Какой этап в каскадной модели жизненного цикла не предусмотрен?
    a) Разработка требований;
    b) Проектирование;
    c) Реализация;
    d) Тестирование;
    e) Сопровождение.
93. Вид тестирования ПО, при котором неизвестно внутреннее устройство кода или к нему не обращаются:
    a) Черный ящик;
    b) Белый ящик;
    c) Модульное тестирование;
    d) Интегральное тестирование.
94. Вид тестирования ПО, при котором известно внутреннее устройство кода или к нему не обращаются:
    a) Черный ящик;
    b) Белый ящик;
    c) Модульное тестирование;
    d) Интеграционное тестирование.
95. Альфа-тестирование и бета-тестирование является подкатегорией следующего вида тестирования:
    a) Модульное тестирование;
    b) Интеграционное тестирование;
    c) Системное тестирование;
    d) Блочное тестирование.
96. Какой вид тестирования предшествует интеграционному?
    a) Модульное тестирование;
    b) Альфа-тестирование;
    c) Системное тестирование;
    d) Бета-тестирование.
97. К какому классу тестирования относится тестирование безопасности?
    a) По объекту тестирования;
    b) По степени автоматизации;
    c) По степени изолированности;
    d) По времени тестирования.
98. Какого вида тестирования не существует:
    a) Тестирование черного ящика;
    b) Тестирование серого ящика;
    c) Тестирование белого ящика;
    d) Тестирование синего ящика.
99. За что отвечает юзабилити-тестирование?
    a) За работоспособность отдельных блоков кода;
    b) За эргономичность готового продукта;
    c) За безопасность использования готового продукта;
    d) За оценку количества потребляемых ресурсов продуктом.
100. Выберите верное утверждение:
    a) Альфа-тестирование проводят после модульного, но переде интеграционным тестированием;
    b) Альфа-тестирование проводят после интеграционного, но перед модульным тестированием;
    c) Альфа-тестирование проводят после интеграционного, выполненного после модульного тестирования;
    d) Альфа-тестирование проводят после модульного, выполненного после интеграционного тестирования.
101. Выберете верное утверждение:
    a) Тестирование программы позволяет отловить все ошибки;
    b) При тестировании имеется практическая невозможность трассировки всех путей выполнения программы;
    c) Тестовый код занимает много меньше места, чем исходный;
    d) Существует абсолютно универсальные сценарии тестирования кода.

5.3. Оценочные материалы, используемые для экзамена (описание билетов, тестов и т.п.)

Экзаменационный билет включает в себя фундаментальный теоретический вопрос и прикладной теоретический вопрос из установленного перечня контрольных вопросов, используемых при формировании экзаменационных билетов при оценке знаний обучающихся на экзамене, а также практическое задание из установленного перечня контрольных заданий, используемых при формировании экзаменационных билетов при оценке знаний обучающихся на экзамене
Пример экзаменационного билета:

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ им. А.А.Угарова
(филиал) федерального государственного автономного образовательного учреждения
высшего образования
«Национальный исследовательский технологический университет «МИСиС»
Кафедра «Автоматизированные и информационные системы управления»
09.03.02 Информационные системы и технологии
Профиль – Информационные системы и технологии

Дисциплина **«Программирование на языках высокого уровня»**

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 10

1. Интерфейсы и абстрактные классы. Для чего используется интерфейс? Чем отличается интерфейс от абстрактного класса?
2. Какой интерфейс отвечает за работу с деструкторами в C#? Как работает оператор using
3. Пусть даны две строки *S1* и *S2* произвольной длины. Написать такую программу, которая определяет являются ли строки анаграммами.
S1 является анаграммой к *S2*, если из *S1* можно без разрывов (пробелов) получить *S2*.

Экзаменатор _____ П.И. Жуков

Утверждено на заседании кафедры АИСУ

Протоколом №5 от 26.05.2022 г.

И.о. заведующего кафедрой АИСУ _____ Д.А. Полещенко

Билеты в бумажном виде хранятся на кафедре АИСУ и утверждены ее заведующим (или заместителем зав. Кафедрой).

5.4. Методика оценки освоения дисциплины

№ п/п	Вид оценочного средства	Критерий	Оценка
1.	Выполнение и защита курсовой работы	Обучающийся самостоятельно выполняет полное и аргументированное решение индивидуальных заданий, не допустив ошибок. При защите заданий отвечает развернуто и исчерпывающе на все вопросы.	«Отлично»
		Обучающийся практически самостоятельно выполняет полное решение заданий, но не может аргументировать свое решение. При защите заданий допускает незначительные неточности.	«Хорошо»
		Обучающийся в целом правильно решает задание, но не может аргументировать свое решение. При защите заданий допускает значительные неточности. Обучающийся правильно понимает способ решения заданий, но допускает ошибки при их решении. Задание выполнено частично. При защите заданий допускает значительные неточности.	«Удовлетворительно»
		Обучающийся не может решить задание.	«Неудовлетворительно»
2.	Устный опрос по материалам лекционных занятий	Обучающийся уверенно, логически связно, динамично, грамотно отвечает на поставленные вопросы. Демонстрирует абсолютное понимание темы обсуждаемой предметной области, достаточно полные знания в объеме прослушанного теоретического материала. Грамотно владеет и адекватно использует терминологию предметной области.	«Зачтено»
		Обучающийся не верно интерпретирует поставленные вопросы, не владеет терминологией предметной области, не понимает сущности обсуждаемой проблемы. Невнятно, невразумительно, неуверенно, неверно формулирует ответы на дополнительные и наводящие вопросы и/или не отвечает на них.	«Не зачтено»
3.	Выполнение и защита лабораторных	Обучающийся показывает твердые и достаточно полные знания в объеме прослушанного теоретического материала, владеет требуемым математическим аппаратом, методиками	«Зачтено»

	работ	решения, необходимыми остаточными знаниями по изученным фундаментальным дисциплинам (математика, физика); демонстрирует умения и практические навыки владения информационными технологиями, позволяющими оптимизировать экспериментальную и аналитическую часть лабораторного исследования. Логически связно, динамично, грамотно и последовательно излагает методику выполнения лабораторной работы и обработки результатов моделирования. Ошибаясь, уверенно исправляется после дополнительных и наводящих вопросов.	
		Обучающийся допускает грубые ошибки в ответе, не понимает сущности излагаемого вопроса, не умеет применять теоретические знания на практике и/или не владеет требуемыми знаниями. Невнятно, невразумительно, неуверенно, неверно формулирует ответы на дополнительные и наводящие вопросы и/или не отвечает на них.	«Не зачтено»
4.	Экзамен	Компетенции УК-2, ПК-1 ОПК-6, ОПК-8 сформированы. Обучающийся демонстрирует: - глубокие знания содержания изученной дисциплины во взаимосвязи с другими дисциплинами; - способность использовать теоретические знания при выполнении практических заданий; - аргументированные, исчерпывающие ответы на все вопросы по билету, а также дополнительные вопросы экзаменатора; - умение выполнять и обосновывать решение практических заданий высокого уровня сложности; - наличие собственной обоснованной позиции по обсуждаемым вопросам; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.	«Отлично»
		Компетенции УК-2, ПК-1 ОПК-6, ОПК-8 сформированы. Обучающийся демонстрирует: - знание основных терминов по содержанию изученной дисциплины; - твердые знания теоретического материала; - умение дать четкие ответы на поставленные вопросы; - умение решать практические задания; - владение основной литературой, рекомендованной программой дисциплины. Допускаются незначительные неточности в ответах на теоретические вопросы и при выполнении практических заданий.	«Хорошо»
		Компетенции УК-2, ПК-1 ОПК-6, ОПК-8 сформированы. Обучающийся демонстрирует: - знания теоретического материала по изученной дисциплине; - неполные ответы на основные вопросы, допуская ошибки в ответе; недостаточное понимание сущности излагаемых вопросов; - неточные ответы на дополнительные вопросы; - умение выполнять практические задания без грубых ошибок; - недостаточное владение литературой, рекомендованной программой дисциплины.	«Удовлетворительно»
		Компетенции УК-2, ПК-1 ОПК-6, ОПК-8 не сформированы. Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - принципиальные ошибки при ответе на основные вопросы билета, отсутствие знаний и понимания основных терминов и определений; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие навыка или существенные ошибки при выполнении практических заданий; - незнание литературы, рекомендованной программой дисциплины.	«Неудовлетворительно»
5	Оценка тестовых	Обучающийся дал 90% и более правильных ответов	«Отлично»

		заданий	Обучающийся дал 65-89% правильных ответов	«Хорошо»
			Обучающийся дал 50-64% правильных ответов	«Удовлетворительно»
			Обучающийся дал менее 50% правильных ответов	«Неудовлетворительно»

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

6.1. Рекомендуемая литература

6.1.1. Основная литература

Обозначе-ние	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 1.1	Николаев В. Т., Купцов С. В. , Тикменов В. Н.	Практика программирования в инженерных расчётах: учебное пособие	ЭБС «Университетская библиотека онлайн» URL: https://biblioclub.ru/index.php?page=book_red&id=485295	Москва: Физматлит, 2018
Л 1.2	Чеповский А.М., Макаров А.В., Скоробогатов С.Ю.	Common Intermediate Language и системное программирование в Microsoft .NET: курс лекций	ЭБС «Университетская библиотека онлайн» URL: https://biblioclub.ru/index.php?page=book_red&id=429250	Москва: Национальный Открытый Университет «ИНТУИТ», 2016
Л 1.3	Абрамян А. В. , Абрамян М. Э.	Разработка пользовательского интерфейса на основе технологии Windows Presentation Foundation : учебник	ЭБС «Университетская библиотека онлайн» URL: https://biblioclub.ru/index.php?page=book_red&id=499453	Ростов-на-Дону, Таганрог: Южный федеральный университет, 2018

6.1.2. Дополнительная литература

Обозначе-ние	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 2.1	Гусеева Е.Н.	Математика и информатика: практикум: учебное пособие	ЭБС «Университетская библиотека онлайн» URL: https://biblioclub.ru/index.php?page=book_red&id=83437	Москва: ФЛИНТА, 2021
Л 2.2	Плаксин М. А.	Тестирование и отладка программ для профессионалов будущих и настоящих: учебное пособие	ЭБС «Университетская библиотека онлайн» URL: https://biblioclub.ru/index.php?page=book_red&id=445811	Москва: Лаборатория знаний, 2020
Л 2.3	Лауферман О.В. Лыгина Н. И.	Разработка программного продукта: профессиональные стандарты, жизненный цикл, командная работа: учебное пособие	ЭБС «Университетская библиотека онлайн» URL: https://biblioclub.ru/index.php?page=book_red&id=576397	Новосибирск: Новосибирский государственный технический университет, 2019

6.1.3. Методические разработки

Обозначе-ние	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 3.1	Жуков П.И.	Методические указания по выполнению практических работ	https://disk.yandex.ru/d/Pji31sUJCpdfjQ	
Л 3.2	Жуков П.И.	Методическое указания по выполнению курсовой работы	https://disk.yandex.ru/d/Pji31sUJCpdfjQ	

Л 3.3	Жуков П.И..	Программирование на языках высокого уровня. Учебное пособие для выполнения лабораторных работ	https://disk.yandex.ru/d/Pji31sUJCpdfjQ	

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»	
Э 1	Открытое образование [Электронный ресурс]: https://openedu.ru/
Э 2	LMS Canvas [Электронный ресурс]: https://lms.misis.ru
Э 3	ЭБС «Университетская библиотека онлайн» [Электронный ресурс]: https://biblioclub.ru
6.3. Перечень программного обеспечения	
П 1	Microsoft Windows
П 2	Microsoft Office
П 3	Microsoft Visual Studio 2022
6.4. Перечень информационных справочных систем и профессиональных баз данных	
И 1	eLIBRARY.RU - НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА [Электронный ресурс]: https://elibrary.ru/

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	
7.1	**Аудитория №406** **«Лаборатория прикладного программирования»** Перечень основного оборудования, учебно-наглядных пособий: • монитор - 9шт.; • персональный компьютер - 9шт.; • проектор; • экран настенный; • усилитель-распределитель; • комплект учебной мебели на 25 посадочных мест. В помещении имеется подключение к сети «Интернет» и доступ в электронную информационно-образовательную среду организации.
7.2	**Аудитория №416** **«Лаборатория искусственного интеллекта»** Перечень основного оборудования, учебно-наглядных пособий: • Персональный компьютер ASUS H81M с монитором- 9 шт. • Экран ELV240 / ACP-DINON-ELV240 / 180x240 MW 3:4 - 1шт • Мультимедиа проектор Epson EB-485W - 1шт • Комплект учебной мебели на 25 посадочных мест. • Лабораторный стенд «Исследование сервоприводов» - 1 шт., ввод в эксплуатацию: 2011 год. В помещении имеется подключение к сети «Интернет» и доступ в электронную информационно-образовательную среду организации.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Для успешного освоения дисциплины обучающемуся необходимо:

- Посещать все виды занятий;
- Своевременно зарегистрироваться на рекомендованные электронные ресурсы;
- При возникновении любых вопросов по содержанию дисциплины и организации работы своевременно обращаться к преподавателю (в часы очных консультаций, через MS Teams или LMS Canvas);
- Отчеты по практическим работам рекомендуется выполнять с использованием MS Office или LibreOffice;
- Активно работать с научными базами в сети Интернет;

6. Качественное освоение дисциплины возможно только при систематической самостоятельной работе, что поддерживается системой текущей аттестации.

Самостоятельная работа студентов осуществляется и контролируется с помощью:

- Индивидуального опроса студентов при защите практических работ и лабораторных работ
- Сдачи курсовой работы.

По результатам выполнения КР обучающиеся оформляют отчеты (согласно индивидуальным требования в Л 3.2).
Аттестацией по предмету является экзамен, допуск к которому возможен только после защиты курсовой работы.
Основные вопросы и задачи, подлежащие обсуждению в рамках экзамена приведены в п. 5.1, п. 5.3 данного РПД