

Энгельсский технологический институт (филиал) федерального государственного  
бюджетного образовательного учреждения  
высшего образования  
«Саратовский государственный технический университет имени Гагарина Ю.А.»

Кафедра «Естественные и математические науки»

## **РАБОЧАЯ ПРОГРАММА**

по дисциплине

### **Б.1.1.9 Программирование**

направления подготовки

09.03.04 «Программная инженерия»

Профиль «Управление разработкой программных проектов»

форма обучения – очная  
курс – 1,2  
семестр – 1,2,3,4  
зачетных единиц – 15 (3,6,3,3)  
часов в неделю – 5,3,3,3  
всего часов – 540(108,216,108,108)  
в том числе:  
лекции – 80 (32,16,16,16)  
практические занятия – 144 (48,32,32,32)  
лабораторные занятия – нет  
самостоятельная работа – 316 (28,168,60,60)  
экзамен – 2,4 семестр  
зачет – 1,3 семестр  
РГР – нет  
курсовая работа – нет  
курсовой проект – нет

Рабочая программа обсуждена на заседании кафедры ЕМН  
«27» июня 2022 года, протокол № 9

Заведующий кафедрой  /Жилина Е.В./

Рабочая программа обсуждена на УМКН ИВЧТ  
«27» июня 2022 года, протокол № 5

Председатель УМКН  /Жилина Е.В./

## 1. Цели и задачи дисциплины

Целью преподавания дисциплины Б.1.1.9 «Программирование» является:

- формирование у студентов алгоритмического мышления;
- приобретение теоретических знаний и практических навыков в области основных методологий разработки программ с помощью языков программирования высокого уровня (процедурно-ориентированной и объектно-ориентированной);
- изучение современных инструментальных сред, предназначенных для разработки программ с помощью языков программирования высокого уровня.

## 2. Место дисциплины в структуре ОПОП ВО

Дисциплина Б.1.1.9 «Программирование» входит в обязательную часть блока 1 учебного плана основной профессиональной образовательной программы бакалавриата по направлению 09.03.04 «Программная инженерия».

Дисциплина читается на 1 и 2 курсе. Зачетных единиц 15. Продолжительность курса составляет 224 аудиторных учебных часов, образованных 80 часами лекций и 144 часами практических и лабораторных занятий. Помимо этого, 316 часов в курсе отводится под самостоятельную работу студентов.

Требования к «входным» знаниям, умениям и навыкам обучающегося, необходимым для освоения данной дисциплины: в рамках объема школьных знаний по информатике и математике.

Дисциплина Б.1.1.9 «Программирование» имеет логическую и содержательно-методическую взаимосвязь с дисциплинами, для освоения которых необходимы теоретические и практические знания основ информатики и информационных технологий.

Основные положения дисциплины будут использованы в дальнейшем при изучении следующих дисциплин учебного плана: «Базы данных», «Функциональное и логическое программирование», «Объектно-ориентированное программирование», «Структуры и алгоритмы обработки данных», «Тестирование программного обеспечения».

## 3. Требования к результатам освоения дисциплины

Результаты освоения дисциплины Б.1.1.9 «Программирование» направлены на овладение следующими компетенциями:

- способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности (ОПК-2);
- способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-6).

В результате освоения дисциплины студент должен:

### **Знать:**

- основные понятия информатики и программирования;
- основные управляющие структуры и способы описания алгоритмов;
- основные методы разработки алгоритмов, особенности их реализации;
- понятие типа данных, форматы представления данных при решении задач с помощью компьютера, а также средства конструирования новых типов на основе стандартных типов, используемых в языках программирования;

- основы организации файлов, выполнения операций над файлами, возможности их использования при решении задач;
- методы организации, хранения и эффективной обработки абстрактных структур данных;
- основные технологии программирования: процедурно-ориентированное и объектно-ориентированное программирование с использованием языка высокого уровня;

**Уметь:**

- решать задачи, используя различные методы разработки алгоритмов и выбирая наиболее подходящие алгоритмы и средства их реализации в зависимости от постановки задачи;
- разрабатывать программные продукты: разрабатывать программы средней сложности на языках программирования высокого уровня;
- использовать основные концепции процедурного и объектно-ориентированного программирования; отлаживать и тестировать программы.

**Владеть:**

- навыками разработки и анализа алгоритмов решения типовых задач (сортировки и поиска данных, поиска решений и пр.), исследования их свойств;
  - методами и инструментальными средствами разработки программ: разработки программ средней сложности, их тестирования и отладки;
- навыками самостоятельного решения задач с помощью компьютеров, изучения новых средств разработки программ.

Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

Код и наименование компетенции (результат освоения)	Код и наименование индикатора достижения компетенции (составляющей компетенции)
ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ИД-1 <sub>ОПК-2</sub> Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.
	ИД-2 <sub>ОПК-2</sub> Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.
	ИД-3 <sub>ОПК-2</sub> Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных	ИД-1 <sub>ОПК-6</sub> Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.
	ИД-2 <sub>ОПК-6</sub> 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,

Код и наименование компетенции (результат освоения)	Код и наименование индикатора достижения компетенции (составляющей компетенции)
продуктов	ведения баз данных и информационных хранилищ.
	ИД-3 <sub>ОПК-6</sub> Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач.

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ИД-1 <sub>ОПК-2</sub> Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.	<b>Знает:</b> основные понятия информатики и программирования
ИД-2 <sub>ОПК-2</sub> Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности.	<b>Умеет:</b> решать задачи, используя различные методы разработки алгоритмов и выбирая наиболее подходящие алгоритмы и средства их реализации в зависимости от постановки задачи
ИД-3 <sub>ОПК-2</sub> Имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	<b>Владеет:</b> навыками применения инструментальных средств современных информационных технологий и программ для решения профессиональных задач
ИД-1 <sub>ОПК-6</sub> Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	<b>Знает:</b> <ul style="list-style-type: none"> <li>- основные управляющие структуры и способы описания алгоритмов;</li> <li>- понятие типа данных, форматы представления данных при решении задач с помощью компьютера, а также средства конструирования новых типов на основе стандартных типов, используемых в языках программирования;</li> <li>- основы организации файлов, выполнения операций над файлами, возможности их использования при решении задач;</li> <li>- методы организации, хранения и эффективной обработки абстрактных структур данных;</li> <li>- основные технологии программирования на языках высокого уровня.</li> </ul>
ИД-2 <sub>ОПК-6</sub> 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	<b>Умеет:</b> <ul style="list-style-type: none"> <li>- разрабатывать программы средней сложности на языках программирования высокого уровня;</li> <li>- использовать основные концепции процедурного и объектно-ориентированного программирования;</li> <li>отлаживать и тестировать программы</li> </ul>

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
хранилищ.	
ИД-3 <sub>ОПК-6</sub> Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач.	<p><b>Владеет:</b></p> <ul style="list-style-type: none"> <li>- навыками разработки и анализа алгоритмов решения типовых задач (сортировки и поиска данных, поиска решений и пр.), исследования их свойств;</li> <li>- методами и инструментальными средствами разработки программ: разработки программ средней сложности, их тестирования и отладки;</li> <li>- навыками самостоятельного решения задач с помощью компьютеров, изучения новых средств разработки программ.</li> </ul>

#### 4. РАСПРЕДЕЛЕНИЕ ТРУДОЕМКОСТИ (ЧАС.) ДИСЦИПЛИНЫ ПО ТЕМАМ И ВИДАМ ЗАНЯТИЙ

##### 1 семестр

№ Модуля	№ Неде ли	№ Те мы	Наименование раздела	Часы				
				Всего	Лек-ции	Коллок-виумы	Практические	СРС
1	2	3	4	5	6	7	8	9
1	1	1.1	Жизненный цикл программного обеспечения.	13	2	-	4	7
1	2	1.2	Общая характеристика системы MSDN.	13	2	-	4	7
1	3	1.3	Состав языка программирования высокого уровня.	9	2	-	4	3
1	4-5	1.4	Основные операторы	14	4	-	6	4
1	6-7	1.5	Процедурно-ориентированная технология разработки программ	10	4	-	6	-
2	8-9	2.1	Массивы, строки	10	4	-	6	-
2	10-12	2.2	Динамические структуры данных (списки, бинарные деревья)	12	6	-	6	-
2	13-14	2.3	Файлы	10	4	-	6	-
2	15-16	2.4	Визуальное проектирование Windows приложений	17	4	-	6	7
<b>Всего 1 семестр</b>				<b>108</b>	<b>32</b>		<b>48</b>	<b>28</b>

##### 2 семестр

№ Мо-ду-ля	№ Неде-ли	№ Те-мы	Наименование раздела	Часы				
				Всего	Лек-ции	Коллок-виумы	Практи-ческие	СРС
1	2	3	4	5	6	7	8	9
1	1	3.1	Классы, объекты, методы. Перегрузка методов	44	2	-	8	34
1	2	3.2	Основные свойства ООП: инкапсуляция	42	2	-	6	34
1	3-4	3.3	Основные свойства ООП: наследование, полиморфизм	44	4	-	6	34
2	5-7	4.1	Абстрактные типы данных. Коллекции	29	6	-	8	15
2	8	4.2	Обобщенное программирование	21	2	-	4	15
	-		Подготовка к экзамену	36	-	-	-	36
<b>Всего 2 семестр</b>				<b>216</b>	<b>16</b>	<b>-</b>	<b>32</b>	<b>168</b>

### 3 семестр

1	2	3	4	5	6	7	8	9
1	1-2	5.1	Обработка исключительных ситуаций.	28	4	-	6	16
1	3-5	5.2	О О программа как программа, управляемая событиями	36	6	-	10	18
2	6-7	6.1	Дополнительные возможности LINQ to objects	24	4	-	10	8
2	8	6.2	Запросы LINQ to objects	28	2	-	6	18
<b>Всего 3 семестр</b>				<b>108</b>	<b>16</b>	<b>-</b>	<b>32</b>	<b>60</b>

### 4 семестр

1	2	3	4	5	6	7	8	9
2	3-5	7.1	Создание потоков	40	8	-	16	12
2	6-8	7.2	Синхронизация потоков	40	8	-	16	12
			Подготовка к экзамену	36	-	-	-	36
<b>Всего 4 семестр</b>				<b>108</b>	<b>16</b>	<b>-</b>	<b>32</b>	<b>60</b>

## 5. Содержание лекционного курса

№ темы	Всего часов	№ недели	Тема лекции. Вопросы, отрабатываемые на лекции	Учебно-методическое обеспечение
1	2	3	4	5
<b>1 семестр</b>				
1.1	2	1	<b>Жизненный цикл программного обеспечения.</b> Постановка задачи и спецификация программы, способы записи алгоритма. Программа на языке высокого уровня. Понятия тестирования и отладки. Критерии качества программы.	<b>3,4,7,8</b>

			Диалоговые программы, дружелюбность интерфейса. Стиль программирования. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма. Типовые алгоритмы вычислительных процессов	
1.2	2	2	<b>Общая характеристика системы MSDN.</b> Понятия системы программирования и среды программирования. Основные понятия. Выполнение программы в .NET. Структура платформы .NET. Интегрированная среда разработки MS Visual Studio.NET. Обобщенная система типов.	3,4,7,8
1.3	2	3	<b>Состав языка программирования высокого уровня.</b> Структура программы на языке высокого уровня. Алфавит языка. Лексемы. Понятие типа данных. Классификация типов. Значимые типы и ссылочные типы данных. Классификация по способу хранения. Встроенные типы данных. Переменные. Выражения. Основные операции. Использование стандартных функций. Приведение типов.	3,4,7,8
1.4	4	4-5	<b>Основные операторы.</b> Операторы и основные управляющие структуры: итерация, ветвление, повторение. Решение задач с использованием основных управляющих структур. Вложенные циклы. Рекуррентные вычисления.	3,4,7,8
1.5	4	6-7	<b>Процедурно-ориентированная технология разработки программ.</b> Понятие функции. Спецификатор static. Фактические и формальные параметры функции. Способы передачи параметров в функцию (по значению, по ссылке, выходные параметры, параметры-массивы).	3,4,7,8
2.1	4	8-9	<b>Массивы, строки.</b> Массивы в C#. Многомерные массивы. Рваные массивы. Цикл foreach для перебора массивов. Статические элементы класса Array для работы с массивами. Типы задач по обработке массивов. Простые методы сортировки. Поисковые задачи. Двумерные массивы. Решение задач с использованием массивов. Массивы в C#. Многомерные массивы. Рваные массивы. Цикл foreach для перебора массивов. Статические элементы класса Array для работы с массивами. Типы задач по обработке массивов. Простые методы сортировки. Поисковые задачи. Двумерные массивы. Решение задач с	3,4,7,8

			использованием массивов.	
2.2	6	10-12	<b>Динамические структуры данных (списки, бинарные деревья)</b>	3,4,7,8
2.3	4	13-14	<b>Файлы.</b> Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками. Потоки байтов FileStream. Потоки символов (StreamWriter и StreamReader). Двоичные файлы (BinaryWriter, BinaryReader).	3,4,7,8
2.4	4	15-16	<b>Визуальное проектирование Windows приложений.</b> Основные особенности Windows. Типы программ в ОС Windows. Сообщения. Структура приложения с обработкой сообщений. Шаблон Windows – приложения. Основные типы Windows.Forms. Элементы управления. Диалоговые окна.	3,4,7,8
	<b>32</b>			
			<b>2 семестр</b>	
3.1	2	1	<b>Классы, объекты, методы. Перегрузка методов.</b> Определение класса. Поля. Методы. Объекты. Конструкторы. Деструктор. Сбор мусора. Ключевое слово this. Ключевое слово static. Статические конструкторы. Статические классы. Перегрузка операций (унарные и бинарные операции, приведение типов).	1,2,6,7,8
3.2	2	2	<b>Основные свойства ООП: Инкапсуляция.</b> Инкапсуляция. Спецификаторы доступа. Свойства. Автоматические свойства. Индексаторы.	1,2,6,7,8
3.3	4	3-4	<b>Основные свойства ООП: наследование, полиморфизм.</b> Иерархии классов. Наследование. Виртуальные методы. Механизм позднего связывания. Абстрактные классы. Класс object. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET.	1,2,6,7,8
4.1	6	5-7	<b>Абстрактные типы данных. Коллекции.</b> Основные абстрактные структуры данных. Физическое представление данных. Массив, список, двоичное дерево, хеш-таблица, стек, очередь Пространство имен System.Collections. Интерфейсы. Коллекции общего назначения. Основные элементы класса ArrayList. Основные элементы класса Hashtable. Основные элементы класса SortedList. Основные элементы класса Stack. Основные элементы класса Queue. Доступ к коллекциям с помощью нумератора.	1,2,6,7,8



4.2	2	8	<b>Обобщенное программирование.</b> Классы-прототипы. Основные преимущества использования обобщений. Ограничения. Параметризованные коллекции библиотеки .NET Создание собственных классов-прототипов. Обобщенные методы.	1,2,6,7,8
	<b>16</b>			
			<b>3 семестр</b>	
5.1	4	1-2	<b>Обработка исключительных ситуаций.</b> События, нарушающие нормальный ход выполнения программы. Специальный объект-исключение.	1,2,6,7,8
5.2	6	3-5	<b>ОО программа как программа, управляемая событиями.</b> Понятие делегата. Описание и использование. Многоадресная передача. События. События для многоадресной передачи. Многопоточные приложения.	1,2,6,7,8
5.3	4	6-8	<b>Дополнительные возможности LINQ to objects.</b> Неявная типизация локальных переменных. Синтаксис инициализации объектов и коллекций. Лямбда-выражения. Расширяющие методы. Анонимные типы.	1,2,6,7,8
	2		<b>LINQ to objects.</b> Применение запросов LINQ к элементарным массивам. Возврат результатов LINQ запросов. Применение запросов LINQ к объектам коллекций. Применение запросов LINQ к необобщенным коллекциям. Внутреннее представление операторов запросов LINQ.	1,2,6,7,8
	<b>16</b>			
			<b>4 семестр</b>	
7.1	8	1-4	<b>Создание потоков.</b> Введение в многопоточность. Класс Thread. Создание потоков. Потоки с параметрами	5,7,8
7.2	8	5-8	<b>Синхронизация потоков.</b> Проблема синхронизации потоков. Мониторы. Мьютексы. Семафоры. Использование таймеров.	5,7,8
	<b>16</b>			

**6. Содержание коллоквиумов**  
(не предусмотрены учебным планом)

**7. Перечень практических занятий**

№	Всего	Наименование практических занятий. Задания,	Учебно-
---	-------	---	---------

<b>темы</b>	<b>часов</b>	<b>вопросы, отрабатываемые на практических занятиях.</b>	<b>методическое обеспечение</b>
<b>1</b>	<b>2</b>	<b>4</b>	<b>3</b>
<b>1 семестр</b>			
1.1	4	Жизненный цикл программного обеспечения.	3,4,7
1.2	4	Общая характеристика системы MSDN.	3,4,7
1.3	4	Состав языка программирования высокого уровня.	3,4,7,9
1.4	6	Основные операторы	3,4,7
1.5	6	Процедурно-ориентированная технология разработки программ	3,4,7
2.1	6	Массивы, строки	3,4,7,9
2.2	6	Динамические структуры данных (списки, бинарные деревья)	3,4,7
2.3	6	Файлы	3,4,7,9
2.4	6	Визуальное проектирование Windows приложений	3,4,7
	<b>48</b>		
<b>2 семестр</b>			
3.1	8	Классы, объекты, методы. Перегрузка методов	1,2,6,7
3.2	6	Основные свойства ООП: инкапсуляция	1,2,6,7
3.3	6	Основные свойства ООП: наследование, полиморфизм	1,2,6,7
4.1	8	Абстрактные типы данных. Коллекции	1,2,6,7
4.2	4	Обобщенное программирование	1,2,6,7
	<b>32</b>		
<b>3 семестр</b>			
5.1	6	Обработка исключительных ситуаций.	1,2,6,7
5.2	10	О О программа как программа, управляемая событиями	1,2,6,7
6.1	10	Дополнительные возможности LINQ to objects	1,2,6,7
6.2	6	Запросы LINQ to objects	1,2,6,7
	<b>32</b>		
<b>4 семестр</b>			
7.1	16	Создание потоков.	5,7
7.2	16	Синхронизация потоков в	5,7
	<b>32</b>		

**8. Перечень лабораторных работ**  
(не предусмотрены учебным планом)

**9. Задания для самостоятельной работы студентов**

9.

№ темы	Всего часов	Задания, вопросы, для самостоятельного изучения (задания)	Учебно-методическое обеспечение
1	2	3	4
<b>1 семестр</b>			
1.1	7	Модели жизненного цикла ПО. Стадии цикла разработки ПО.	3,4,7
1.2	7	Характеристика системы MSDN. Оптимизация поиска библиотека MSDN.	3,4,7
1.3	3	Классификация языков программирования по парадигмам программирования Классификация по ориентации на предметную область. Классификация по семейственной принадлежности	3,4,7
1.4	4	Типы операторов для языков Visual Basic, Python, JavaScript. Выражения и операторы.	3,4,7
2.4	7	Визуальное проектирование Windows приложений. Автоматизация пользовательского интерфейса. Продукты, реализующих специальные возможности. Диалоговые окна.	3,4,7
	<b>28</b>		
<b>2 семестр</b>			
3.1	34	Методы, их параметры, взаимодействие и перегрузка в Java и в языке C#.	1,2,6,7
3.2	34	Инкапсуляция в Хакслет, C#, Java, Python	1,2,6,7
3.3	34	Наследование, полиморфизм в Хакслет, C#, Java, Python	1,2,6,7
4.1	15	Упражнения на реализацию абстрактных типов данных (АТД) с помощью классов языка C#	1,2,6,7
4.2	15	Стандартная библиотека шаблонов и ее компоненты. Упражнения.	1,2,6,7
	36	Подготовка к экзамену	1,2,6,7
	<b>168</b>		
<b>3 семестр</b>			
5.1	16	Обработка исключительных ситуаций. Выполнение дополнительных заданий по теме.	1,2,6,7
5.2	18	ОО программа как программа, управляемая событиями	1,2,6,7
6.1	8	Дополнительные возможности LINQ to objects	1,2,6,7
6.2	18	Запросы LINQ to objects	1,2,6,7
	<b>60</b>		
<b>4 семестр</b>			
7.1	16	Создание потоков. Управление потоком.	5,7
7.2	16	Синхронизация потоков в C#, Java, Python	5,7
	36	Подготовка к экзамену	5,7

	60		
--	----	--	--

**10. Расчетно-графическая работа** (не предусмотрена учебным планом)

**11. Курсовая работа** (не предусмотрена учебным планом)

**12. Курсовой проект** (не предусмотрен учебным планом).

**13. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)**

В процессе обучения студент должен полностью выполнить учебный план, предусмотренный в рабочей программе дисциплины, по всем видам учебных занятий и набрать 15 зачетных единиц трудоемкости.

В результате освоения образовательной программы у студентов формируются компетенции ОПК-2, ОПК-6.

Уровни освоения компетенции

Индекс ОПК-2	Формулировка:		
	способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности		
Ступени освоения компетенции	Отличительные признаки	Технологии формирования	Средства и технологии оценки
Пороговый (удовлетворительный)	Знает: основные понятия структурного программирования; Умеет: определять требования к создаваемой программе; Владеет: простейшими навыками структурного программирования	Лекции, практические занятия, СРС	Опрос, отчет по практическим работам
Продвинутый (хорошо)	Знает: элементы структурного языка программирования и их классификацию, типы данных и их классификацию, управляющие конструкции и их особенности, принципы использования и разработки функций, виды трансляции программ и их особенности; Умеет: применять принципы структурного программирования при разработке консольных приложений средней сложности; Владеет: навыками структурного программирования с использованием принципа модульности.		
Высокий	Знает: особенности принципов		

(отлично)	<p>различных парадигм программирования, элементы структурного языка программирования, их классификацию, синтаксическое и семантическое значение, типы данных, их классификацию и особенности их представления в памяти компьютера, управляющие конструкции, их особенности и варианты взаимозамены, особенности использования и разработки функций, виды трансляции программ и их особенности;</p> <p>Умеет: эффективно применять принципы структурного программирования при разработке консольных приложений средней сложности и обосновывать применение данных принципов;</p> <p>Владеет: навыками структурного программирования с использованием принципа модульности и нисходящего проектирования</p>		
-----------	---	--	--

Индекс ОПК-6	Формулировка: способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов		
Ступени освоения уровней компетенции	Отличительные признаки	Технологии формирования	Средства и технологии оценки
Пороговый (удовлетворительный)	<p>Знает: схему подготовки программ, основы синтаксиса, семантики, структуру программ на языках высокого уровня;</p> <p>Умеет: писать и читать исходный текст простейших программ, компилировать и запускать программу;</p> <p>Владеет: навыками создания простейших консольных приложений.</p>	Лекции, практические занятия, СРС	Опрос, защита практических работ
Продвинутый (хорошо)	<b>Знает:</b> основные управляющие структуры и способы описания		

	<p>алгоритмов; понятие типа данных; методы организации, хранения и эффективной обработки абстрактных структур данных;</p> <p><b>Умеет:</b> разрабатывать программы средней сложности на языках программирования высокого уровня;</p> <p><b>Владеет:</b> навыками разработки программ средней сложности</p>		
Высокий (отлично)	<p><b>Знает:</b> основные технологии программирования на языках высокого уровня;</p> <p><b>Умеет:</b> разрабатывать программы на языках программирования высокого уровня, использовать основные концепции процедурного и объектно-ориентированного программирования; отлаживать и тестировать программы;</p> <p><b>Владеет:</b> навыками разработки программ средней сложности, их тестирования и отладки.</p>		

### Примеры заданий промежуточной аттестации

1. Описать иерархию классов: **Pair**( int first, int second) – базовый класс, **Time** (минуты, секунды) – производный класс. Для каждого класса определить данные, свойства, конструкторы без параметров, с параметрами, метод для инициализации объектов, метод для вывода данных объекта.
2. В основной программе создать массив, в который записать объекты разных классов (базового и производного).
3. Подсчитать количество объектов каждого класса в массиве.
4. Выполнить сортировку объектов в массиве по возрастанию.
5. Выполнить клонирование объектов типа Time.

#### Вопросы для зачета (1 семестр)

1. Жизненный цикл программного обеспечения.
2. Каскадная схема жизненного цикла ПО.
3. Итерационная схема жизненного цикла ПО.
4. Основные участники разработки ПО.
5. Основные парадигмы программирования.
6. Процедурно-ориентированная парадигма программирования.
7. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма.
8. Понятие системы программирования. Система программирования VS.Net.
9. Общая характеристика платформы MSDN.
10. Структура программы на языке программирования C#.
11. Основные элементы языка программирования.
12. Массивы.

13. Оператор foreach.
14. Размерность массива. Многомерные массивы.
15. Рваные массивы.
16. Типы задач по обработке массивов.
17. Бинарный поиск в массиве
18. Сортировка простым обменом
19. Сортировка простым выбором
20. Сортировка простым включением.
21. Методы и свойства класса Array.
22. Регулярные и буквальные строковые литералы.
23. Ссылки типа string.
24. Операции над строками
25. Методы и свойства класса String.
26. Форматирование строк.
27. Неизменяемость объектов класса String.
28. Тип System.Text.StringBuilder.

### Вопросы для зачета (3 семестр)

1. Понятие делегата. Описание и использование.
2. Многоадресная передача.
3. События.
4. События для многоадресной передачи.
5. Многопоточные приложения.
6. Механизм обработки исключений.
7. Системные исключения и их обработка.
8. Свойства и методы класса Exception Исключения, создаваемые программистом.
9. Основные абстрактные структуры данных.
10. Физическое представление данных.

### Вопросы для экзамена (2 семестр)

1. Определение класса.
2. **Поля. Методы.**
3. Объекты.
4. Конструкторы.
5. Деструктор. Сбор мусора.
6. Ключевое слово this. Ключевое слово static. Статические конструкторы.
7. Основные свойства ООП.
8. Инкапсуляция. Спецификаторы доступа.
9. Свойства. Автоматические свойства.
10. Индексаторы.
11. Иерархии классов.
12. Наследование.
13. Виртуальные методы. Механизм позднего связывания.
14. Абстрактные классы.
15. Класс object.
16. Интерфейсы.
17. Работа с объектами через стандартные интерфейсы .NET
18. Пространство имен System.Collections.
19. Интерфейсы пространства имен System.Collections.
20. Коллекции общего назначения.
21. Основные элементы класса ArrayList.
22. Основные элементы класса Hashtable.

23. Основные элементы класса SortedList.
24. Основные элементы класса Stack.
25. Основные элементы класса Queue.
26. Доступ к коллекциям с помощью нумератора.
27. Классы-прототипы.
28. Основные преимущества использования обобщений.
29. Ограничения.
30. Параметризованные коллекции библиотеки .NET.
31. Создание собственных классов-прототипов.
32. Обобщенные методы.

### **Вопросы для экзамена (4 семестр)**

1. Применение потоков
2. Классическая модель потоков
3. Реализация потоков в пользовательском пространстве
4. Реализация потоков в ядре
5. Действия, необходимые для создания потока
6. Назначение потоков.
7. Управление потоками в операционной системе
8. Раскройте сущность задания. Какие ограничения на процесс можно наложить, включив его в задание?
9. Каким образом можно завершить процессы и потоки?
10. Назовите основные параметры функции создания процесса.
11. Каково внутреннее устройство потока?
12. Каким образом можно осуществить планирование потоков и задание их приоритетов.
13. Как осуществляется привязка потоков к процессорам?
14. Какие функции можно использовать для планирования потоков?
15. Для каких целей используются семафоры?
16. В чем отличие между критическим разделом и мьютексом?
17. Приведите порядок действий при создании и использовании объекта "критический раздел".
18. Приведите порядок действий при создании и использовании объекта "мьютекс".
19. Приведите порядок действий при создании и использовании объекта "семафор".
20. Приведите порядок действий при создании и использовании объекта "событие".
21. Каким образом программа может использовать "семафор" или "событие", созданные другой программой?
22. Какие из существующих в операционной системе Windows wait-функций Вы знаете?
23. Дайте краткую характеристику основным методам синхронизации параллельно работающим программам.

### **14. Образовательные технологии**

На лекциях используется «проблемный» подход к изложению материала: материал каждой лекции иллюстрируется примерами, рассматриваются нестандартные ситуации, требующие решения с использованием рассматриваемого материала. При этом студенты должны активно участвовать в обсуждении вопросов, выработке решений. Для самостоятельного изучения предлагается использовать электронные ресурсы.

На практических занятиях используются следующие методы обучения и контроля усвоения материала:



- выполнение практических работ предполагает решение индивидуальных задач по программированию, по работе оформляется отчет, описывающий процесс решения задачи в соответствии с жизненным циклом программной системы;

- каждую практическую работу студент защищает преподавателю и получает оценку за защиту, в рамках защиты обсуждаются различные варианты решения, предложенные студентами, сравнение решений, анализ возможных ситуаций, code review.

## 15. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

1. Мейер, Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 с. — ISBN 978-5-4486-0513-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/79706.html>
2. Кауфман, В. Ш. Языки программирования. Концепции и принципы / В. Ш. Кауфман. — 2-е изд. — Саратов : Профобразование, 2019. — 464 с. — ISBN 978-5-4488-0137-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/88014.html>. — Режим доступа: для авторизир. пользователей
3. Павловская, Т. А. Программирование на языке высокого уровня C# : учебное пособие / Т. А. Павловская. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 245 с. — ISBN 978-5-4497-0862-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102051.html>. — Режим доступа: для авторизир. Пользователей
4. Свиркин, М. В. Программирование под Windows в среде Visual C++ 2005 : учебное пособие / М. В. Свиркин, А. С. Чуркин. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 215 с. — ISBN 978-5-4497-0866-3. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102053.html>. — Режим доступа: для авторизир. Пользователей
5. Гунько, А. В. Программирование (в среде Windows) : учебное пособие / А. В. Гунько. — Новосибирск : Новосибирский государственный технический университет, 2019. — 155 с. — ISBN 978-5-7782-3890-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/99209.htm>. — Режим доступа: для авторизир. Пользователей
6. Биллиг, В. А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) : учебник / В. А. Биллиг. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 409 с. — ISBN 978-5-4497-0880-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102029.html>. — Режим доступа: для авторизир. Пользователей
7. Биллиг, В. А. Основы программирования на C# : учебное пособие / В. А. Биллиг. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 573 с. — ISBN 978-5-4497-0893-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102033.html>. — Режим доступа: для авторизир. Пользователей.
8. Рацеев, С. М. Программирование на языке Си : учебное пособие для вузов / С. М. Рацеев. — Санкт-Петербург : Лань, 2022. — 332 с. — ISBN 978-5-8114-8585-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/193320>. — Режим доступа: для авториз. пользователей.

9. Фарафонов, А. С. Программирование на языке высокого уровня : методические указания к проведению лабораторных работ по курсу «Программирование» / А. С. Фарафонов. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2013. — 32 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/22912.html> . — Режим доступа: для авторизир. Пользователей

#### ИНТЕРНЕТ-РЕСУРСЫ

10. - Электронная библиотека учебно-методической литературы для общего и профессионального образования <http://window.edu.ru/>
11. - Электронная библиотека <http://www.twirpx.com/library/>
12. - Научная электронная библиотека <http://elibrary.ru/>
13. - ИТ-образование в России <http://www.edu-it.ru/>

### 16. Материально-техническое обеспечение

#### 1. Учебная аудитория для проведения занятий лекционного типа, текущего контроля и промежуточной аттестации, групповых и индивидуальных консультаций.

Укомплектована специализированной мебелью и техническими средствами обучения: 22 стола, 44 стула; рабочее место преподавателя; маркерная доска; проектор BENQ 631, рулонный проекционный экран, ноутбук Lenovo 560 (I3/4Гб/500, мышь), подключенный в сеть с выходом в Интернет и доступом в информационно-образовательную среду ЭТИ (филиал) СГТУ имени Гагарина Ю.А., учебно-наглядные пособия, обеспечивающие тематические иллюстрации по рабочей программе дисциплины.

Программное обеспечение: Microsoft Windows 7, Microsoft Office 2010 (Word, Excel, PowerPoint), Google Chrome.

#### 2. Учебная аудитория для проведения практических занятий, групповых и индивидуальных консультаций, текущего контроля.

Укомплектована специализированной мебелью и техническими средствами обучения: 12 столов, 12 стульев; рабочее место преподавателя; маркерная доска, 12 компьютеров (I 3/ 8 Гб/ 500), мониторы 22' LG, клавиатура, мышь). Компьютеры объединены в локальную сеть с выходом в Интернет и доступом в информационно-образовательную среду ЭТИ (филиал) СГТУ имени Гагарина Ю.А., учебно-наглядные пособия, обеспечивающие тематические иллюстрации по рабочей программе дисциплины.

Программное обеспечение: Microsoft Windows10, Microsoft Office 2010 (Word, Excel, PowerPoint), Visual Studio; VS code, Google Chrome.

Рабочую программу  
составил  
доцент кафедры ЕМН



/Ершов А.С./

### 17. Дополнения и изменения в рабочей программе

Рабочая программа пересмотрена на заседании кафедры  
«\_\_\_\_\_» \_\_\_\_\_ 20 \_\_\_\_ года, протокол № \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_/\_\_\_\_\_

Внесенные изменения утверждены на заседании УМКС/УМКН  
«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ года, протокол № \_\_\_\_\_  
Председатель УМКС/УМКН \_\_\_\_\_/\_\_\_\_\_/