

**МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

УТВЕРЖДАЮ
Проректор по учебной работе

(подпись) **Василенко В.Н.**
(Ф.И.О.)

«25» мая 2023 г.

**РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ**

Параллельные вычисления и параллельное программирование

Направление подготовки

09.03.03 Прикладная информатика

Направленность (профиль) подготовки

Цифровизация бизнес-процессов

Квалификация выпускника

Бакалавр

Воронеж

1. Цели и задачи дисциплины

Целями освоения дисциплины «Параллельные вычисления и параллельное программирование» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

06 Связь, информационные и коммуникационные технологии в сфере исследования, разработки, внедрения и сопровождения информационных технологий и систем.

Дисциплина направлена на решение задач профессиональной деятельности следующих типов:

- производственно-технологический;
- организационно-управленческий;
- проектный.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 «Прикладная информатика».

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

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений
2	ПКв-2	Способность разрабатывать и адаптировать прикладное программное обеспечение.	ИД3 _{ПКв-2} - демонстрирует навыки технического и рабочего проектирования компонентов информационных систем в соответствии со спецификой профиля подготовки
3	ПКв-9	Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.	ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования

Код и наименование индикатора достижения	Результаты обучения (показатели оценивания)
ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений	Знает: методы и алгоритмы параллельного программирования для решения задач в рамках поставленной цели.
	Умеет: проводить сравнительный анализ последовательных и параллельных программных средств и оценивать их эффективность, принимать конкретные решения для повышения эффективности разработки программных приложений, использовать полученные знания для создания параллельных программ в различных предметных областях исходя из имеющихся ресурсов и ограничений.
	Владеет: навыками реализации параллельных алгоритмов и их использования для решения прикладных задач;

ИДЗ _{ПКв-2} - демонстрирует навыки технического и рабочего проектирования компонентов информационных систем в соответствии со спецификой профиля подготовки	Знает: подходы для проведения анализа требований к параллельным программам, архитектурные принципы реализации параллельной обработки в вычислительных машинах
	Умеет: использовать программные инструменты для написания, отладки, тестирования и запуска параллельных приложений, производить трансформацию параллельных и распределенных приложений с целью повышения эффективности вычислений
	Владеет: приемами разработки параллельных программ на языках параллельного программирования для решения задач исходя из действующих правовых норм, имеющихся ресурсов и ограничений, навыками настройки высокопроизводительных вычислительных систем и комплексов
ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования	Знает: основные команды и функции языка программирования
	Умеет: использовать различные прикладные средства разработки параллельного программного обеспечения
	Владеет: прикладным инструментарием для разработки параллельных алгоритмов в соответствии с назначением объекта проектирования

3. Место дисциплины в структуре ОП ВО

Дисциплина относится к части, формируемой участниками образовательных отношений Блока Б1 ООП. Дисциплина является обязательной к изучению.

Изучение дисциплины основано на знаниях, умениях и навыках, полученных при изучении обучающимися дисциплин Программирование на языках высокого уровня, Объектно-ориентированные системы программирования, Технологии программирования.

Дисциплина является предшествующей для изучения дисциплин выполнения выпускной квалификационной работы.

4. Объем дисциплины и виды учебных занятий

Общая трудоемкость дисциплины составляет 3 зачетных единиц.

Виды учебной работы	Всего ак. ч	Распределение трудоемкости по семестрам, ак. ч
		7 семестр
Общая трудоемкость дисциплины	108	108
Контактная работа в т.ч. аудиторные занятия:	61,6	61,6
Лекции	30	30
<i>в том числе в форме практической подготовки</i>	-	-
Практические занятия (ПЗ)	30	30
<i>в том числе в форме практической подготовки</i>	30	30
Консультации текущие	1,5	1,5
Виды аттестации (зачет)	0,1	0,1
Самостоятельная работа:	46,4	46,4
Проработка материалов по конспекту лекций (собеседование, тестирование)	15	15
Проработка материалов по учебникам (собеседование, тестирование)	15	15
Изучение материалов к практическим работам, оформление отчета по практической работе	16,4	16,4

5. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

5.1 Содержание разделов дисциплины

№ п/п	Наименование разделов дисциплины	Содержание раздела (указывается в дидактических единицах)	Трудоемкость раздела,
1.	Цели и задачи введения параллельной обработки данных	Ограничения максимальной производительности однопроцессорных ЭВМ. История вве деня параллелизма. Различие многозадач ных, параллельных и распределенных вы числений. Проблемы использования паралл лелизма.	14
2	Принципы разработки параллельных алгоритмов и программ	Пути достижения параллелизма. Способы построения многопроцессорных вычисли тельных систем. Виды параллельных вычис лительных систем. Оценка производитель ности МВС.	14
3	Проектирование параллельных алгоритмов	Проектирование параллельных алгоритмов. Цели проектирования. Графовое представв ление. Общая схема проектирования. Раз деление задачи на подзадачи. Определение связей между подзадачами. Масштабироваа ние подзадач. Распределение подзадач на узлы вычислительной системы. Анализ эфф фективности.	15
4	Основы языка программирования C#	Платформа .NET. Преимущества платформы .NET. Понятие процесса операционной системы. Понятие потока операционной системы. Потoki и стратегия управления временем процессоров. Приоритеты потоков. Жизненный цикл потока. Классы и объекты. Принципы ООП. Пространство имен. Описание класса.	31,7
5	Создание многопоточных приложений на языке программирования C#	Модель вычислений в виде графа «операции-операнды». Схема параллельного выполнения алгоритма. Определение времени выполнения параллельного алгоритма. Показатели эффективности параллельного алгоритма. Вычисление частных сумм последовательности числовых значений. Оценка максимально достижимого параллелизма. Анализ масштабируемости параллельных вычислений. Алгоритмы маршрутизации в много процессорных системах. Методы передачи данных в много процессорных системах. Анализ трудоемкости основных операций передачи данных между двумя процессорами. Анализ трудоемкости основньх операций передачи данных от одного процессора всем остальным процессорам сети. Циклический сдвиг при передаче данных между процессорами.	31,7

	Методы логического представления топологии коммуникационной среды. Оценка трудоемкости операций передачи данных для кластерных систем. Использование виртуальных функций. Абстрактные базовые классы. Концепция интерфейсов. Полиморфизм в С#. Понятие делегата. Описание делегатов. Дженерики. Многоадресная передача в делегате. Обобщенные делегаты. Понятие события. Многоадресная передача события. Использование событийных средств доступа. Программа, управляемая событиями. Анонимные методы в С#. Лямбда-выражение. Блоки операторов в лямбда-выражении. Создание потока в С#. Гонка данных в потоках. Оператор lock. Критические секции. Понятие клинча. Мягкие методы блокировки. Мониторы. Семафоры. Задача об «обедающих философях». Класс ThreadPool. Распараллеливание циклов (Класс Parallel)	
<i>Консультации текущие</i>		1,5
<i>Зачет</i>		0,1

5.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, ак. час	ПЗ, ак. час	СРО, ак. час
1.	Цели и задачи введения параллельной обработки данных	4	4	6
2.	Принципы разработки параллельных алгоритмов и программ	4	4	6
3.	Проектирование параллельных алгоритмов	6	2	7
4.	Основы языка программирования С#	8	10	13,7
5.	Создание многопоточных приложений на языке программирования С#	8	10	13,7
<i>Консультации текущие</i>				1,5
<i>Зачет</i>				0,1

5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятий	Трудоемкость, ак. час
1	Цели и задачи введения параллельной обработки данных	Ограничения максимальной производительности однопроцессорных ЭВМ. История введения параллелизма. Различие многозадачных, параллельных и распределенных вычислений. Проблемы использования параллелизма.	4
2	Принципы разработки параллельных алгоритмов и программ	Пути достижения параллелизма. Способы построения многопроцессорных вычислительных систем. Виды параллельных вычислительных систем. Оценка производительности МВС.	4

3	Проектирование параллельных алгоритмов	Проектирование параллельных алгоритмов. Цели проектирования. Графовое представление. Общая схема проектирования. Разделение задачи на подзадачи. Определение связей между подзадачами. Масштабирование подзадач. Распределение подзадач на узлы вычислительной системы. Анализ эффективности.	6
4	Основы языка программирования C#	Платформа .NET. Преимущества платформы .NET. Понятие процесса операционной системы. Понятие потока операционной системы. Поток и стратегия управления временем процессоров. Приоритеты потоков. Жизненный цикл потока. Классы и объекты. Принципы ООП. Пространство имен. Описание класса.	8
5	Создание многопоточных приложений на языке программирования C#	Модель вычислений в виде графа «операции-операнды». Схема параллельного выполнения алгоритма. Определение времени выполнения параллельного алгоритма. Показатели эффективности параллельного алгоритма. Вычисление частных сумм последовательности числовых значений. Оценка максимально достижимого параллелизма. Анализ масштабируемости параллельных вычислений. Алгоритмы маршрутизации в много процессорных системах. Методы передачи данных в много процессорных системах. Анализ трудоемкости основных операций передачи данных между двумя процессорами. Анализ трудоемкости основных операций передачи данных от одного процессора всем остальным процессорам сети. Циклический сдвиг при передаче данных между процессорами. Методы логического представления топологии коммуникационной среды. Оценка трудоемкости операций передачи данных для кластерных систем. Использование виртуальных функций. Абстрактные базовые классы. Концепция интерфейсов. Полиморфизм в C#. Понятие делегата. Описание делегатов. Дженерики. Многоадресная передача в делегате. Обобщенные делегаты. Понятие события. Многоадресная передача события. Использование событийных средств доступа. Программа, управляемая событиями. Анонимные методы в C#. Лямбда-выражение. Блоки операторов в лямбда-выражении. Создание потока в C#. Гонка данных в потоках. Оператор lock. Критические секции. Понятие клинча. Мягкие методы блокировки. Мониторы. Семафоры. Задача об «обедающих философях». Класс ThreadPool. Распараллеливание циклов (Класс Parallel)	8

5.2.2 Практические занятия

№ п/п	Наименование раздела дисциплины	Наименование практических работ	Трудоемкость,
1.	Цели и задачи введения параллельной обработки данных	Практическая работа по расчету сложности разработки параллельных алгоритмов. Практическая работа по анализу конвейерных и векторных вычислений	4

2.	Принципы разработки параллельных алгоритмов и программ	Практическая работа по анализу параллелизма данных и параллелизма задач. Практическая работа по анализу показателя эффективности распараллеливания	4
3.	Основы языка программирования C#	Разработка параллельных программ с использованием технологии многопоточного программирования.	2
4.	Создание многопоточных приложений на языке программирования C#	Практикум по разработке параллельных алгоритмов и программ для решения задач вычислительной математики.	10
5	Основы языка программирования C#	Разработка параллельных программ с использованием технологии многопоточного программирования.	10

5.2.3 Лабораторный практикум

Не предусмотрен

5.2.4 Самостоятельная работа обучающихся (СРО)

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудовые часы
1	Цели и задачи введения параллельной обработки данных	Проработка материала по конспекту лекций (собеседование, тестирование)	2
		Проработка материала по учебнику (собеседование, тестирование)	2
		Изучение материалов к практическим работам, оформление отчета по практической работе	2
2	Принципы разработки параллельных алгоритмов	Проработка материала по конспекту лекций (собеседование, тестирование)	2
		Проработка материала по учебнику (собеседование, тестирование)	2
		Изучение материалов к практическим работам, оформление отчета по практической работе	2
3	Проектирование параллельных алгоритмов	Проработка материала по конспекту лекций (собеседование, тестирование)	3
		Проработка материала по учебнику (собеседование, тестирование)	3
		Изучение материалов к практическим работам, оформление отчета по практической работе	1
4	Основы языка программирования C#	Проработка материала по конспекту лекций (собеседование, тестирование)	4
		Проработка материала по учебнику (собеседование, тестирование)	4
		Изучение материалов к практическим работам, оформление отчета по практической работе	5,7
5	Создание многопоточных приложений на языке	Проработка материала по конспекту лекций (собеседование, тестирование)	4

	программирования С#	Проработка материала по учебнику (собое сдование, тестирование)	4
		Изучение материалов к практическим работам, оформлнение отчета по практической работе	5,7

6. Учебно-методическое и информационное обеспечение дисциплины

Для освоения дисциплины обучающийся может использовать:

6.1. Основная литература

1. Биллиг, В.А. Параллельные вычисления и многопоточное программирование / В.А. Биллиг. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУУ ИТ», 2016. – 311 с. : ил., схем. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=428948>

2. Основы высокопроизводительных вычислений : учебное пособие / К.Е. Афанасьев, С.В. Стуколов, В.В. Малышенко и др. – Кемерово : Кемеровский государственный унии верситет, 2012. – Том 2. Технологии параллельного программирования. – 412 с. – Режим доступа: по подписке. – URL:

3. Карепова, Е.Д. Основы многопоточного и параллельного программирования : учебное пособие / Е.Д. Карепова ; Сибирский федеральный университет, Институт вычислительного моделирования Сибирского отделения Российской академии наук, Сибирский научно-образовательный центр суперкомпьютерных технологий. – Красноярск : Сибирский федеральный университет (СФУ), 2016. – 355 с. : ил. – Режим доступа: по подд писке. – URL: <https://biblioclub.ru/index.php?page=book&id=497217>

6.2 Дополнительная литература

4. Кузнецов, А.С. Теория вычислительных процессов : учебник / А.С. Кузнецов, Р.Ю. Царев, А.Н. Князьков ; Сибирский федеральный университет. – Красноярск : Сибирский федее ральный университет (СФУ), 2015. – 184 с. : табл., схем. – Режим доступа: по подписке. – URL:

5. Гома, Х. UML. Проектирование систем реального времени, параллельных и распрее деленных приложений. [Электронный ресурс] — Электрон.дан. — М. : ДМК Пресс, 2007. — 704 с. — Режим доступа: <http://e.lanbook.com/book/1232>— Загл. с экрана.

6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся

6. Ашарина, И.В. Объектно-ориентированное программирование вС++: лекции и упражнения. [Электронный ресурс] — Электрон.дан. — М. : Горячая линия-Телеком, 2012. — 320 с. — Режим доступа: <http://e.lanbook.com/book/5115>

7. Энтони, У. Параллельное программирование на С++ в действии. Практика разработт ки многопоточных программ. [Электронный ресурс] — Электрон.дан. — М. : ДМК Пресс, 2012. — 672 с. — Режим доступа: <http://e.lanbook.com/book/4813>— Загл. с экрана.

8. Федотов, И.Е. Модели параллельного программирования. [Электронный ресурс] — Электрон.дан. — М. : СОЛОН-Пресс, 2012. — 384 с. — Режим доступа: <http://e.lanbook.com/book/13807>— Загл. с экрана.

6.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Наименование ресурса сети «Интернет»	Электронный адрес ресурса
«Российское образование» - федеральный портал	http://www.edu.ru/index.php

Научная электронная библиотека	http://www.elibrary.ru/defaulttx.asp?
Федеральная университетская компьютерная сеть России	http://www.runnet.ru/
Информационная система «Единое окно доступа к образовательным ресурсам»	http://www.window.edu.ru/
Электронная библиотека ВГУИТ	http://biblos.vsu.ru/megapro/web
Сайт Министерства науки и высшего образования РФ	http://minobrnauki.gov.ru
Портал открытого on-line образования	http://npoad.ru
Информационно-коммуникационные технологии в образовании. Система федеральных образовательных порталов	http://www.ict.edu.ru/
Электронная образовательная среда ФГБОУ ВО «ВГУИТ»	http://education.vsu.ru
Поисковая система «Google»	http://www.google.ru

6.5. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

При освоении дисциплины используется лицензионное и открытое программное обеспечение и информационные справочные системы: ОС MS Windows, MS Office (MS Word, MS PowerPoint), программа для создания и редактирования видео Corel VideoStudio, ПС Google, СУБД «Рейтинг студентов», информационная среда для дистанционного обучения «Moodle», виртуальная машина Oracle VM Virtual Box. Локальная сеть университета и глобальная сеть Internet.

7. Материально-техническое обеспечение дисциплины

Ауд. 334 для проведения лекционных занятий, оснащенная комплектом мебели для учебного процесса – 50 шт., проектором Epson EH-TW650.

Ауд. 339 для проведения практических и лабораторных работ:

Количество ПК – 16 (IntelCore i5 – 4570), проектор – 1 (ViewSonicPJ5255).

Microsoft Windows 7 Microsoft Open License Microsoft Windows Professional 7 Russian Upgrade Academic OPEN 1 License No Level#47881748 от 24.12.2010г. <http://eopen.microsoft.com>.

Microsoft VisualStudio 2010 Сублицензионный договор № 42082/VRN3 от 21 августа 2013 г. на право использование программы DreamSparkElectronicSoftwareDeliver;

Microsoft Office 2007 Standar Microsoft Open License Microsoft Office 2007 Russian Academic OPEN No Level #44822753 от 17.11.2008г <http://eopen.microsoft.com>.

1С: Предприятие. Бухгалтерия 8 Лицензионное соглашение с ЗАО «1С» Регистрационный номер 9985964 1С: Предприятие 8. Комплект для обучения в высших и средних учебных заведениях.

8. Оценочные материалы для промежуточной аттестации обучающихся по дисциплине

Оценочные материалы (ОМ) для дисциплины (модуля) включают в себя:

- перечень компетенций с указанием индикаторов достижения компетенций, этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности.

ОМ представляются в виде отдельного документа и входят в состав рабочей программы дисциплины (модуля) в виде приложения.

Оценочные материалы формируются в соответствии с П ВГУИТ «Положение об оценочных материалах».

ПРИЛОЖЕНИЕ
к рабочей программе

1. Организационно-методические данные дисциплины для очно-заочной или заочной форм обучения

1.1 Объемы различных форм учебной работы и виды контроля в соответствии с учебным планом

Общая трудоемкость дисциплины (модуля) составляет 3 зачетных единицы.

Виды учебной работы	Всего ак. ч	Распределение трудоемкости по семестрам, ак. ч
		8 семестр
Общая трудоемкость дисциплины	108	108
Контактная работа в т.ч. аудиторные занятия:	9,5	9,5
Лекции	4	4
<i>в том числе в форме практической подготовки</i>	-	-
Практические занятия (ПЗ)	4	4
<i>в том числе в форме практической подготовки</i>	4	4
Консультации текущие	1,4	1,4
Виды аттестации (зачет)	0,1	0,1
Самостоятельная работа:	94,6	94,6
Проработка материалов по конспекту лекций (собеседование, тестирование)	10	10
Проработка материалов по учебникам (собеседование, тестирование)	70,6	70,6
Выполнение контрольной работы	10	10
Оформление текста контрольной работы	4	4
Контроль	3,9	3,9

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

по дисциплине

Параллельные вычисления и параллельное программирование

1 Перечень компетенций с указанием этапов их формирования

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений
2	ПКв-2	Способность разрабатывать и адаптировать прикладное программное обеспечение.	ИД3 _{ПКв-2} - демонстрирует навыки технического и рабочего проектирования компонентов информационных систем в соответствии со спецификой профиля подготовки
3	ПКв-9	Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.	ИД1 _{ПКв-9} - использование прикладного инструментария для разработки программного обеспечения в соответствии с назначением объекта проектирования

Код и наименование индикатора достижения	Результаты обучения (показатели оценивания)
ИД2 _{УК-1} – Решает поставленные задачи, используя системный подход, на основе критического анализа и синтеза информации и оценивает последствия возможных решений	Знает: методы и алгоритмы параллельного программирования для решения задач в рамках поставленной цели.
	Умеет: проводить сравнительный анализ последовательных и параллельных программных средств и оценивать их эффективность, принимать конкретные решения для повышения эффективности разработки программных приложений, использовать полученные знания для создания параллельных программ в различных предметных областях исходя из имеющихся ресурсов и ограничений.
	Владеет: навыками реализации параллельных алгоритмов и их использования для решения прикладных задач;
ИД3 _{ПКв-2} - демонстрирует навыки технического и рабочего проектирования компонентов информационных систем в соответствии со спецификой профиля подготовки	Знает: подходы для проведения анализа требований к параллельным программам, архитектурные принципы реализации параллельной обработки в вычислительных машинах
	Умеет: использовать программные инструменты для написания, отладки, тестирования и запуска параллельных приложений, производить трансформацию параллельных и распределенных приложений с целью повышения эффективности вычислений
	Владеет: приемами разработки параллельных программ на языках параллельного программирования для решения задач исходя из действующих правовых норм, имеющихся ресурсов и ограничений, навыками настройки высокопроизводительных вычислительных систем и комплексов
ИД1 _{ПКв-9} - использование прикладного инструментария для разработки	Знает: основные команды и функции языка программирования

программного обеспечения в соответствии с назначением объекта проектирования	Умеет: использовать различные прикладные средства
	Владеет: прикладным инструментарием для разработки параллельных алгоритмов в соответствии с назначением объекта проектирования

2 Паспорт оценочных материалов по дисциплине

№ п/п	Разделы дисциплины	Индекс контролируемой компетенции (или ее части)	Оценочные материалы		Технология/ процедура оценивания (способ контроля)
			наименование	№№ задания	
1	Цели и задачи введения параллельной обработки данных	УК-1	<i>Банк тестовых заданий (промежуточное тестирование, экзамен)</i>	1-15	Бланочное тестирование
		УК-1	Собеседование (защита практических работ)	16-30	Защита практической работы
		УК-1	Кейс-задание (тестирование, экзамен)	111-116, 173-202	Проверка преподавателем
2	Принципы разработки параллельных алгоритмов и программ	ПКв-2	<i>Банк тестовых заданий (промежуточное тестирование, экзамен)</i>	31-70	Бланочное тестирование
		ПКв-2	Собеседование (защита практических работ)	117-132	Защита практической работы
		ПКв-2	Кейс-задание (тестирование, экзамен)	203-212	Проверка преподавателем
3	Технология параллельного программирования OpenMP	ПКв-2	<i>Банк тестовых заданий (промежуточное тестирование, экзамен)</i>	252	Бланочное тестирование
		ПКв-2	Собеседование (защита практических работ)	31-70	Защита практической работы
		ПКв-2	Кейс-задание (тестирование, экзамен)	213-227	Проверка преподавателем
4	Технология	ПКв-9	<i>Банк тестовых заданий (промежуточное</i>	71-110	Бланочное тестирование

параллельно ого программиро вания MPI		<i>тестирование, экзамен)</i>		
	ПКв-9	Собеседование (защита практических работ)	133-172	Защита практической работы
	ПКв-9	Кейс-задание (тестирование, экзамен)	228-250	Проверка преподавателем

3 Оценочные материалы для промежуточной аттестации

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

Испытание промежуточной аттестации обучающегося по дисциплине «Безопасность жизнедеятельности» в форме тестирования, решения кейс-заданий, собеседования и выполнения реферата. Собеседование применяется при защите практических работ. В течение семестра проводятся промежуточные тестирования.

Каждый вариант тестовых заданий включает в себя:

- 15 контрольных тестовых заданий, из них 8 на проверку знаний, 4 на проверку умений и 3 на проверку навыков;
- одну кейс-задачу на проверку умений или навыков.

Экзамен проводится в форме устного ответа на вопросы билета, включающего кейс-задание или прохождения теста из 30 тестовых заданий (форма проведения экзамена в форме ответа на вопросы билета или прохождения теста выбирается студентом).

Каждый билет включает в себя: 2 теоретических вопроса по курсу лекций и 1-но кейс-задание.

Каждый билет в форме теста включает в себя:

- 30 контрольных тестовых заданий, из них 20 на проверку знаний, 5 на проверку умений и 5 на проверку навыков;
- Одно кейс-задание на проверку умений.

3.1 Тесты (задания для промежуточного тестирования)

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

№ задания	примеры тестовых заданий
1.	<p>Строгая модель консистентности памяти определяется следующим условием:</p> <p>1) все обращения к ячейкам памяти абсолютно упорядочены во времени</p> <p>2) все процессы наблюдают все обращения к ячейкам памяти в одном и том же порядке. Обращения не упорядочены по времени</p> <p>3) все процессы наблюдают все обращения к ячейкам памяти, связанные причинно-следственной связью в одном и том же порядке</p> <p>4) неверно ни одно из утверждений</p>

2.	<p>Использование операторов перехода (goto) в структурном блоке OpenMP возможно:</p> <p>1) если оператор перехода не выводит за пределы блока</p> <p>2) поскольку в стандарте нет никаких ограничений на операторы перехода</p> <p>3) в некоторых OpenMP-компиляторах (зависит от реализации)</p> <p>4) неверно ни одно из утверждений</p>
3.	<p>Найдите ошибку в следующем фрагменте программы: <code>#pragma omp parallel default(shared) { int i, j; #pragma omp for for (i=0; i<n; i++) { #pragma omp for for (j=0; j<n; j++) work(i, j); } }</code></p> <p>1) используются конструкции распределения работ, которые непосредственно вложены одна в другую</p> <p>2) в результате использования клаузы default(shared), счетчики циклов i и j являются общими для всех нитей</p> <p>3) в данном фрагменте программы ошибки нет</p> <p>4) используются конструкции распределения работ, которые непосредственно не вложены одна в другую</p>
4.	<p>Директива master</p> <p>1) определяет блок операторов в программе, который будет выполнен одной нитью группы. Остальные нити группы дожидаются завершения выполнения этого блока</p> <p>2) определяет блок операторов в программе, который будет выполнен master-нитью. Остальные нити группы дожидаются завершения выполнения этого блока</p> <p>3) определяет блок операторов в программе, который будет выполнен master-нитью. Остальные нити группы не дожидаются завершения выполнения этого блока</p>
5.	<p>Определите способ распределения витков цикла между нитями для следующего фрагмента программы: <code>#define N 100 void work(int i); int main () { #pragma omp parallel { #pragma omp for for (int i=0; i<N; i++) work (i); } }</code></p> <p>1) статический (static)</p> <p>2) зависит от значения переменной окружения OMP_SCHEDULE</p> <p>3) зависит от реализации компилятора</p>
6.	<p>Найдите ошибку в следующем фрагменте программы: <code>#define N 1000 int main (void){ float a[N], tmp; #pragma omp parallel { #pragma omp for for(int i=0; i<N; i++) { tmp= a[i]*a[i]; a[i]=1-tmp; } } }</code></p> <p>1) в директиве for отсутствует клауза private(i)</p> <p>2) в директиве for отсутствует клауза private(tmp)</p> <p>3) в данном фрагменте программы ошибки нет</p>
7.	<p>Выберите наилучшую стратегию распределения витков цикла между нитями, которая для следующего фрагмента программы даст минимальное время выполнения: <code>#include <omp.h> #include <unistd.h> #define msec 1000 int main (void){ omp_set_num_threads (8); #pragma omp parallel { #pragma omp for schedule (runtime) for(int i=0; i<100; i++) { sleep (i*msec); } } }</code></p>

	<p>1) <code>export OMP_SCHEDULE="static"</code></p> <p>2) <code>export OMP_SCHEDULE="static,10"</code></p> <p>3) <code>export OMP_SCHEDULE="dynamic"</code></p>
8.	<p>Директива SHARABLE в технологии Intel Cluster OpenMP:</p> <p>1) определяет разделяемые нитями переменные, доступ к которым возможен только на одном из узлов кластера</p> <p>2) определяет разделяемые нитями переменные, доступ к которым обрабатывается механизмом DSM</p> <p>3) задает, какие переменные, на какой узел кластера необходимо отобразить</p>
9.	<p>PRAM модель консистентности памяти определяется следующим условием:</p> <p>1) все процессы наблюдают все обращения к ячейкам памяти в одном и том же порядке. Обращения не упорядочены по времени</p> <p>2) все процессы наблюдают все обращения к ячейкам памяти, связанные причинно-следственной связью в одном и том же порядке</p> <p>3) все процессы наблюдают операции записи любого процесса в порядке их выполнения. Операции записи различных процессов могут наблюдаться разными процессами в разном порядке</p>
10.	<p>Параллельная область в OpenMP создается при помощи:</p> <p>1) вызова функции <code>omp_set_max_active_levels</code></p> <p>2) вызова функции <code>omp_set_num_threads</code></p> <p>3) директивы <code>parallel</code></p>
11.	<p>Найдите ошибку в следующем фрагменте программы: <code>int i, j; #pragma omp parallel default(shared) { #pragma omp for collapse (3) for (i=0; i<n; i++) { for (j=0; j < n; j++) work(i, j); } }</code></p> <p>1) в результате использования клаузы <code>default(shared)</code>, счетчики циклов - переменные <code>i</code> и <code>j</code> являются общими для всех нитей</p> <p>2) количество заголовков циклов не соответствуют значению, указанному в клаузе <code>collapse</code></p> <p>3) в данном фрагменте программы ошибки нет</p>
12.	<p>Найдите ошибку в следующем фрагменте программы: <code>#define N 10 int A[N],B[N]; #pragma omp parallel default(shared) { int i; #pragma omp master for (i=0; i<N; i++) { A[i]=0; } #pragma omp barrier B[N-1]=B[N-1] + A[N-1]; }</code></p> <p>1) оператор <code>for</code> не может быть использован внутри конструкции <code>master</code></p> <p>2) оператор, в котором изменяется значение общей переменной <code>B[N-1]</code> выполняется без какой-либо синхронизации</p> <p>3) в данном фрагменте программы ошибки нет</p>
13.	<p>Определите способ распределения витков цикла между нитями для следующего фрагмента программы: <code>#define N 100 #include "omp.h" void work(int i); int main () { #pragma omp parallel { omp_set_schedule (omp_sched_static); #pragma omp for for (int i=0;i<N;i++) work (i); } }</code></p>

	<p>1) статический (static)</p> <p>2) зависит от значения переменной окружения OMP_SCHEDULE</p> <p>3) зависит от реализации компилятора</p>																		
14.	<p>Найдите ошибку в следующем фрагменте программы: <code>int main (void){ int a, i; #pragma omp parallel shared(a) private(i) { #pragma omp master a = 0; #pragma omp for reduction(+:a) for (i = 0; i < 10; i++) { a += i; } } }</code></p> <p>1) в директиве parallel клауза shared(a) должна быть заменена на private(a)</p> <p>2) перед директивой for отсутствует директива barrier</p> <p>3) в данном фрагменте программы ошибки нет</p>																		
15.	<p>Выберите наилучшую стратегию распределения витков цикла между нитями, которая для следующего фрагмента программы даст минимальное время выполнения: <code>#include <omp.h> #include <unistd.h> #define msec 1000 int main (void){ omp_set_num_threads (8); #pragma omp parallel { #pragma omp for schedule (runtime) for(int i=0; i<80;i++) { sleep (msec); } } }</code></p> <p>1) export OMP_SCHEDULE="static,10"</p> <p>2) export OMP_SCHEDULE="static,15"</p> <p>3) export OMP_SCHEDULE="static,20"</p>																		
16.	<p>Использование технологии Intel Cluster OpenMP целесообразно:</p> <p>1) для OpenMP-программ, с хорошей локализацией данных и требующих малой синхронизации</p> <p>2) для OpenMP-программ с нерегулярными вычислениями, проводимыми над общими данными</p> <p>3) для любых OpenMP-программ</p>																		
17.	<p>Пусть $W(x)a$ – операция записи в переменную x значения a. Пусть $R(x)a$ – операция чтения переменной x, где a – прочитанное значение переменной x. Пусть S - операция синхронизации. Следующая последовательность событий:</p> <table border="1" data-bbox="279 1417 1121 1682"> <tr> <td>P1: $W(x)a$</td> <td>$W(x)b$</td> <td>S</td> <td></td> <td></td> <td></td> </tr> <tr> <td>P2:</td> <td></td> <td></td> <td>$R(x)a$</td> <td>$R(x)b$</td> <td>S</td> </tr> <tr> <td>P3:</td> <td></td> <td></td> <td>$R(x)b$</td> <td>$R(x)a$</td> <td>S</td> </tr> </table> <p>1) допустима при слабой модели консистентности</p> <p>2) не допустима при слабой модели консистентности</p> <p>3) допустима при модели консистентности по выходу</p>	P1: $W(x)a$	$W(x)b$	S				P2:			$R(x)a$	$R(x)b$	S	P3:			$R(x)b$	$R(x)a$	S
P1: $W(x)a$	$W(x)b$	S																	
P2:			$R(x)a$	$R(x)b$	S														
P3:			$R(x)b$	$R(x)a$	S														
18.	<p>При реализации компилятором редукционного оператора, описанного при помощи клаузы <code>reduction (+: sum)</code>, где переменная <code>sum</code> имеет тип <code>integer</code>, для каждой нити создается</p>																		

	<p>локальная копия переменной <code>sum</code>, начальное значение которой будет инициализировано:</p> <p>1) 0</p> <p>2) <code>-MAXINT</code> (минимально возможное целое число)</p> <p>3) <code>MAXINT</code> (максимально возможное целое число)</p>
19.	<p>Найдите ошибку в следующем фрагменте программы: <code>#define N 10 float c[N]; float sum = 0.0; #pragma omp parallel shared(sum, c) { #pragma omp for reduction (+: sum) nowait for (int i=0; i<N; i++) { sum += c[i]; } #pragma omp single printf ("Sum of array=%4.2f\n", sum); }</code></p> <p>(1) редуцирующая переменная <code>sum</code> объявлена общей(<code>shared</code>)</p> <p>(2) в директиве <code>for</code> нельзя одновременно указать клаузу <code>reduction</code> и клаузу <code>nowait</code></p> <p>3) до использования значения редуцирующей переменной <code>sum</code> в операторе печати в блоке <code>single</code> отсутствует барьерная синхронизация нитей</p>
20.	<p>Найдите ошибку в следующем фрагменте программы: <code>#define N 10 int A[N],B[N], sum; #pragma omp parallel default(shared) num_threads(10) { int iam=omp_get_thread_num(); if (A[iam] > 0) { #pragma omp critical (update_a) sum +=A[iam]; } if (B[iam] > 0) { #pragma omp critical (update_b) sum +=B[iam]; } }</code></p> <p>1) чтение общих переменных <code>A[iam]</code> и <code>B[iam]</code> в операторах вида <code>A[iam]>0</code> и <code>B[iam]>0</code> выполняется без какой-либо синхронизации</p> <p>2) изменение общей переменной <code>sum</code> внутри критических секций <code>update_a</code> и <code>update_b</code> может выполняться одновременно несколькими нитями, поскольку названия критических секций отличаются</p> <p>3) в данном фрагменте программы ошибки нет</p>
21.	<p>Определите количество нитей, между которыми будет распределена работа в параллельной области: <code>#define N 100 #include "omp.h" void work(int i); int main () { omp_set_num_threads(2); #pragma omp parallel num_threads(4) { #pragma omp for for (int i=0;i<N;i++) work (i); } }</code></p> <p>(1) 2</p> <p>2) 4</p> <p>(3) зависит от значения переменной окружения <code>OMP_NUM_THREADS</code></p>
22.	<p>Найдите ошибку в следующем фрагменте программы: <code>#include <omp.h> int numproc; #pragma omp threadprivate(numproc) int main (void){ numproc=omp_get_num_procs(); #pragma omp parallel { if (numproc < 4) do_small_work(); else do_big_work (); } }</code></p> <p>1) функция <code>omp_get_num_procs</code> не может быть вызвана вне параллельной области</p> <p>2) в директиве <code>parallel</code> отсутствует клауза <code>copyin(numproc)</code></p> <p>3) в данном фрагменте программы ошибки нет</p>
23.	<p>Выберите наиболее подходящую оптимизацию, которая позволит сократить время выполнения следующего фрагмента программы: <code>#include <omp.h> #include <stdio.h> #define N 100 float c[N]; float sum = 0.0; int main (void) { omp_set_num_threads (8); #pragma omp parallel shared(sum, c) { #pragma omp for for (int i=0; i<N; i++) { #pragma omp critical sum</code></p>

	<pre>+= c[i]; } } printf ("Sum of array=%4.2f\n", sum); }</pre> <p>1) заменить директиву <code>critical</code> на директиву <code>atomic</code></p> <p>2) заменить директиву <code>critical</code> на директиву <code>critical (<имя_критической_секции>)</code></p> <p>3) добавить к директиве <code>for</code> клаузу <code>reduction(+:sum)</code> и убрать директиву <code>critical</code></p>
24.	<p>Найдите ошибку в следующем фрагменте программы: <code>#define N 10 int i; #pragma omp parallel { int tmp = 0; #pragma omp for private(tmp) for (i=0; i<N; i++) { tmp += i; } }</code></p> <p>1) не определен класс для переменной <code>i</code> (отсутствует клауза <code>private (i)</code>)</p> <p>2) начальное значение переменной <code>tmp</code> в параллельном цикле не определено</p> <p>3) в данном фрагменте программы ошибки нет</p>
25.	<p>Клауза <code>copyin</code>:</p> <p>1) может быть использована только для переменных, указанных в клаузе <code>private</code></p> <p>2) может быть использована только для переменных, указанных в директиве <code>threadprivate</code></p> <p>3) может быть использована как для переменных указанных в директиве <code>threadprivate</code>, так и для переменных, указанных в клаузе <code>private</code></p>
26.	<p>Найдите ошибку в следующем фрагменте программы: <code>#pragma omp parallel default(shared) { int i; #pragma omp for lastprivate(i) for (i=0; i!=n; i++) { work(i); } }</code></p> <p>1) в заголовке цикла, витки которого распределяются между нитями при помощи директивы <code>for</code> вместо оператора отношения (типа <code><</code>, <code><=</code>, <code>></code>, <code>>=</code>) указан оператор сравнения (<code>!=</code>)</p> <p>2) индексная переменная цикла, витки которого распределяются между нитями при помощи директивы <code>for</code>, не может быть указана в клаузе <code>lastprivate</code></p> <p>3) в данном фрагменте программы ошибки нет</p>
27.	<p>Найдите ошибку в следующем фрагменте программы: <code>int numt=0; #pragma omp parallel { #pragma omp master { #pragma omp critical { numt=omp_get_num_threads(); } #pragma omp barrier } }</code></p> <p>1) директива <code>barrier</code> не может быть использована внутри конструкции <code>master</code></p> <p>2) директива <code>critical</code> не может быть использована внутри конструкции <code>master</code>;</p> <p>3) в данном фрагменте программы ошибки нет</p>
28.	<p>Пусть следующая программа скомпилирована компилятором, поддерживающим вложенный параллелизм. <code>#include <stdio.h> #include "omp.h" int counter; int main() { counter=0; omp_set_nested(0); #pragma omp parallel num_threads(2) { if (omp_get_thread_num() == 0) { #pragma omp parallel num_threads(2) { #pragma omp atomic counter++; } } } printf("Counter=%d\n",counter); }</code> Определите значение переменной <code>counter</code> по завершении выполнения этой программы:</p> <p>1) 1</p>

	<p>2) 2</p> <p>3) 4</p>
29.	<p>Найдите ошибку в следующем фрагменте программы: <code>#pragma omp parallel { int me; me = omp_get_thread_num (); if (me == 0) goto Master; #pragma omp barrier Master: #pragma omp single }</code></p> <p>1) в директиве parallel отсутствует клауза private(me)</p> <p>2) дедлок - взаимная блокировка нитей, возникающая в результате того что master-нить не попадает на директиву barrier</p> <p>3) оператор goto не может быть использован внутри конструкции parallel</p>
30.	<p>Выберите наиболее подходящую оптимизацию, которая позволит сократить время выполнения следующего фрагмента программы: <code>#include <omp.h> #include <unistd.h> #define msec 1000 int main (void){ omp_set_num_threads (8); #pragma omp parallel { #pragma omp for for(int i=0; i<80;i++) { sleep (i*msec); } #pragma omp for for(int i=0; i<80;i++) { sleep ((80-i)*msec); } }</code></p> <p>1) для циклов, выполнение витков которых распределяется между нитями при помощи директивы for, добавить клаузу schedule(static,10)</p> <p>2) для циклов, выполнение витков которых распределяется между нитями при помощи директивы for, добавить клаузу nowait</p> <p>3) для циклов, выполнение витков которых распределяется между нитями при помощи директивы for, добавить клаузу schedule(dynamic)</p>

ПКв-2 Способность разрабатывать и адаптировать прикладное программное обеспечение.

№ задания	примеры тестовых заданий
31.	<p>На каких языках программирования можно работать в Visual Studio .Net?</p> <p>1) только на языке C#</p> <p>2) на четырех языках - Visual C++, C#, Visual Basic, J#</p> <p>3) число языков не ограничено</p>
32.	<p>Пусть x – объект класса int[]. В вызове x.M роль M может играть:</p> <p>1) свойство Length</p> <p>2) метод GetLength</p> <p>3) метод CopyTo</p>
33.	<p>Отметьте правильные объявления:</p> <p>1) char a = new char("a");</p> <p>2) char b = "b";</p> <p>3) char[] c = new char[5];</p>

	4) char[] d = {'a','b','c','d'};
34.	<p>Объект класса string можно создать:</p> <p>1) конструктором по умолчанию</p> <p>2) без явного вызова конструктора</p> <p>3) конструктором, которому передается объект string</p> <p>4) конструктором, которому передается объект char[]</p>
35.	<p>Класс имеет:</p> <p>1) только один конструктор</p> <p>2) только конструктор по умолчанию</p> <p>3) не более одного статического конструктора</p> <p>4) не более одного закрытого конструктора</p> <p>5) несколько конструкторов с одинаковой сигнатурой</p>
36.	<p>Какие утверждения справедливы для развернутых и ссылочных типов</p> <p>1) структуры относятся к развернутым типам</p> <p>2) классы относятся к ссылочным типам</p> <p>3) в языке C# есть модификатор expand, позволяющий отнести класс к развернутому типу</p> <p>4) объектам развернутого типа память отводится в куче</p>
37.	<p>Ключевое слово interface в языке C# задает описание:</p> <p>1) пользовательского интерфейса класса</p> <p>2) открытой части класса</p> <p>3) частного случая класса</p> <p>4) абстрактного класса</p>
38.	<p>Отметьте корректные объявления делегата – функционального класса:</p> <p>1) public class D = delegate void(string s)</p> <p>2) Delegate D1 = new Delegate(void(string s)</p> <p>3) public delegate void D2 (string s)</p> <p>4) delegate int T(int x1, ref int x2)</p> <p>5) public delegate T1(int x)</p>
39.	<p>Обработчик события</p> <p>1) всегда принадлежит классу, вызывающему событие</p> <p>2) никогда не принадлежит классу, вызывающему событие</p> <p>3) может принадлежать классу, вызывающему событие</p> <p>4) принадлежит только одному классу, слушающему событие</p>

40.	<p>Универсальный класс – это:</p> <ol style="list-style-type: none"> 1) класс, способный решать любые задачи пользователя 2) класс, методы которого могут иметь родовые параметры 3) класс с родовыми параметрами 4) класс, функциональные возможности которого превосходят возможности его потомков
41.	<p>Классы Debug и Trace:</p> <ol style="list-style-type: none"> 1) имеют разный набор свойств и методов, используемых для класса Debug в интересах отладки, для класса Trace – в интересах трассировки 2) в Debug конфигурации проекта игнорируются методы и свойства класса Trace 3) в Release конфигурации проекта игнорируются методы и свойства класса Debug 4) классы Trace и Debug имеют одинаковый набор свойств и методов
42.	<p>Объект класса Graphics можно получить</p> <ol style="list-style-type: none"> 1) конструктором класса Graphics 2) статическим методом класса Graphics 3) используя статическое свойство класса Form 4) методом CreateGraphics класса Control 5) используя аргумент, передаваемый обработчику некоторых событий
43.	<p>Преобразования между значимыми и ссылочными типами</p> <ol style="list-style-type: none"> 1) всегда допустимы 2) никогда не допустимы 3) должны быть явными 4) используют операции boxing и unboxing
44.	<p>Отметьте правильные объявления переменной:</p> <ol style="list-style-type: none"> 1) int x; 2) int x(77); 3) int x =77; 4) int x = new int(77); 5) int x = new int();
45.	<p>Верно, что приоритет операций выше:</p> <ol style="list-style-type: none"> 1) умножения чем сложения 2) отношения чем умножения 3) эквивалентности чем конъюнкции 4) унарных чем бинарных

46.	<p>Операциями присваивания языка C# являются:</p> <p>1) +=</p> <p>2) ++</p> <p>3) &=</p> <p>4) &&=</p> <p>5) >>=</p>
47.	<p>Метод можно описать на уровне:</p> <p>1) класса</p> <p>2) пространства имен</p> <p>3) проекта</p> <p>4) решения</p>
48.	<p>Отметьте истинные высказывания:</p> <p>1) для языков программирования, входящих в Visual Studio .Net, каждый компилятор использует собственную библиотеку классов FCL</p> <p>2) для языков программирования, входящих в Visual Studio .Net, все компиляторы используют одну и ту же библиотеку классов FCL</p> <p>3) библиотека классов FCL является частью Visual Studio .Net</p> <p>4) библиотека классов FCL является частью каркаса Framework .Net, отделенного от Visual Studio</p>
49.	<p>Отметьте истинное высказывание:</p> <p>1) в рекурсивной процедуре должна присутствовать нерекурсивная ветвь</p> <p>2) многопроцессорный современный компьютер решит задачу "Ханойские башни" в течение суток</p> <p>3) квадратичные методы сортировки, требующие $O(n^2)$ операций всегда работают медленнее, чем сортировки с порядком операций $O(n \cdot \log(n))$</p> <p>4) каждый цикл имеет только один инвариант</p>
50.	<p>Отметьте истинные высказывания:</p> <p>1) массив массивов является более общей структурой данных, чем многомерный массив</p> <p>2) в языке C# переменные m1 и m2, являющиеся массивами с разным числом элементов всегда принадлежат разным классам</p> <p>3) в языке C# переменные m1 и m2, являющиеся массивами с разными типами элементов всегда принадлежат разным классам</p> <p>4) процедура – функция в C# может возвращать массив в качестве результата</p>
51.	<p>Отметьте истинные высказывания:</p> <p>1) все классы-массивы являются прямыми потомками только одного родительского класса</p>

	<p>2) все классы-массивы наследуют свойства и методы своих предков - двух классов и четырех интерфейсов</p> <p>(3) метод Sort позволяет сортировать любые массивы</p> <p>(4) если S и T принадлежат разным классам – массивам, то присваивание S=T недопустимо</p>
52.	<p>Отметьте истинные высказывания:</p> <p>1) в языке C#, также как и в C++, строка завершается нулевым кодом</p> <p>2) в языке C# можно объявить строку типа char*</p> <p>3) у класса char нет конструкторов с параметрами</p> <p>4) существуют неявные взаимно-обратные преобразования между типами char и string</p> <p>5) существует неявное преобразование символов char в целочисленный тип</p>
53.	<p>Отметьте правильные высказывания</p> <p>1) у класса Regex есть метод Match</p> <p>2) образец @"w+" распознает непустую последовательность букв и цифр</p> <p>3) образец @"w+" распознает непустую последовательность букв и цифр, начинающуюся с буквы</p> <p>4) свойство Success метода Match в случае успеха распознавания возвращает найденную строку</p> <p>5) Знаки + и - в образце должны задаваться в виде \+ и \-</p>
54.	<p>Отметьте истинные высказывания:</p> <p>1) каждый объект содержит набор из всех полей, определенных в классе</p> <p>2) создание объекта выполняется операцией new с вызовом конструктора класса</p> <p>3) для реализации стратегии доступа к закрытым полям класса используются методы-свойства</p>
55.	<p>Отметьте истинные высказывания:</p> <p>(1) сложение точки – структуры Point с размером – структурой Size дает новую точку</p> <p>(2) у структуры нет конструкторов</p> <p>(3) у структуры нет наследников</p> <p>(4) сложение двух точек – структур Point создает линию – структуру Line</p>
56.	<p>Отметьте истинное высказывание:</p> <p>1) класс называется абстрактным, если он не вводит собственных полей данных</p> <p>2) если метод класса A объявлен с модификатором virtual, то для него применяется статическое связывание</p> <p>3) у класса может быть несколько непосредственных родительских классов</p> <p>4) у класса может быть только один непосредственный потомок</p>

	5) в проектах на C# контроль типов выполняется на этапе компиляции
57.	<p>Отметьте истинные высказывания</p> <p>1) слово «интерфейс» имеет разный смысл в зависимости от контекста</p> <p>2) множественное наследование интерфейсов дает те же возможности, что и множественное наследование классов</p> <p>3) при наследовании интерфейса ICloneable необходимо реализовать метод MemberwiseClone</p> <p>4) при наследовании двух интерфейсов имена их методов должны быть различными</p> <p>5) несколько интерфейсов могут быть наследниками одного и того же интерфейса</p>
58.	<p>Отметьте истинные высказывания:</p> <p>1) каждый объявленный функциональный класс (delegate) является наследником класса Delegate</p> <p>2) сигнатура метода, связываемого с экземпляром делегата, в точности должна совпадать с сигнатурой, определенной делегатом</p> <p>3) один из аргументов функции высшего порядка является экземпляром делегата</p> <p>4) для делегатов определена операция вычитания</p> <p>5) определение делегата в точности эквивалентно определению указателя на функцию</p>
59.	<p>Отметьте истинные высказывания</p> <p>1) все события имеют одинаковую сигнатуру из двух аргументов с одними и теми же типами</p> <p>2) все события имеют сигнатуру из двух аргументов, но с отличающимися типами</p> <p>3) все события, не имеющие собственных аргументов, передаваемых обработчику, соответствуют стандартному встроенному делегату EventHandler</p> <p>4) для класса с большим числом событий целесообразно применять динамическое связывание</p> <p>5) для связывания событий с обработчиком можно применять только операцию +=</p>
60.	<p>Отметьте истинные высказывания:</p> <p>1) наследник универсального класса всегда универсален</p> <p>2) для универсального класса не может указываться атрибут сериализации</p> <p>3) родовые параметры универсального класса могут задаваться как на уровне класса, так и на уровне методов</p> <p>4) класс без родовых параметров может быть наследником универсального интерфейса</p> <p>5) делегаты могут иметь родовые параметры</p>
61.	<p>Отметьте истинные утверждения:</p> <p>1) в тексте охраняемого блока всегда должен присутствовать оператор throw, выбрасывающий исключение</p> <p>2) никакая программа не может быть корректной по отношению к произвольным</p>

	<p>спецификациям</p> <p>3) вывод, поступающий от методов класса Debug, может быть направлен только одному, заранее выбранному слушателю</p> <p>4) если блок finally сопровождает охраняемый блок, то он всегда будет выполняться</p> <p>5) правильно организованная отладка позволяет доказать корректность программы</p>
62.	<p>Отметьте истинные высказывания:</p> <p>1) модальное окно нельзя покинуть, не закрыв его</p> <p>2) цвет может быть выбран из специального окна, открывающегося в процессе работы с объектом класса ColorDialog</p> <p>3) у класса Menu нет потомков</p>
63.	<p>Отметьте истинные высказывания:</p> <p>1) существует неявное преобразование переменных арифметического типа в объекты</p> <p>2) существует неявное преобразование объектов в переменные арифметического типа</p> <p>3) возможно явное преобразование объектов в переменные арифметического типа</p> <p>4) переменная арифметического типа в определенном контексте воспринимается как объект</p>
64.	<p>Отметьте истинные высказывания:</p> <p>1) существует неявное преобразование переменных арифметического типа в объекты</p> <p>2) существует неявное преобразование объектов в переменные арифметического типа</p> <p>3) существует явное преобразование объектов в переменные арифметического типа</p> <p>4) в зависимости от контекста переменная арифметического типа представляет собой объект</p>
65.	<p>Отметьте истинные высказывания:</p> <p>1) глобальные переменные в C# могут быть объявлены только в пространстве имен</p> <p>2) поля класса играют роль глобальных переменных для методов класса</p> <p>3) имя локальной переменной не должно совпадать ни с именем аргумента метода, ни с именем поля класса</p> <p>4) в непересекающихся блоках процедуры допускаются одинаковые имена переменных</p>
66.	<p>Отметьте истинные высказывания:</p> <p>1) все методы класса Math являются статическими</p> <p>2) для вызова методов класса Random нужно предварительно создать экземпляр этого класса</p> <p>3) класс Random позволяет получить только повторяющиеся серии случайных чисел</p> <p>4) все переменные, входящие в выражение присваивания, должны быть явно</p>

	инициализированы
67.	<p>Отметьте истинное высказывание:</p> <p>1) присваивание является операцией языка C#</p> <p>2) в теле оператора foreach текущий элемент позволяет получать и изменять значения элементов, хранящихся в контейнере</p> <p>3) в операторе if число ключевых слов if должно совпадать с числом слов else</p> <p>4) case-выражение в операторе switch может задавать диапазон значений</p>
68.	<p>Спецификации CLS задают:</p> <p>1) спецификации, предназначенные для разработчиков трансляторов</p> <p>2) спецификации модуля, предназначенного для использования в проектах на разных языках программирования</p> <p>3) спецификации, которые безусловно должны выполняться программистами, работающими в среде Visual Studio .Net</p> <p>4) спецификации для программистов, работающих на языке C#</p>
69.	<p>Оператор throw</p> <p>1) передает управление catch-блоку, следующему за try-блоком</p> <p>2) может вызываться без аргументов</p> <p>3) имеет два аргумента</p> <p>4) вызывает прерывание процесса вычислений охраняемого блока</p> <p>5) создает исключение – объект класса, производного от Exception</p>
70.	<p>Цель и источник согласованы по присваиванию, если</p> <p>1) базовый тип цели и тип источника совпадают</p> <p>2) тип источника является потомком базового типа цели</p> <p>3) базовый тип цели является потомком типа источника</p> <p>4) базовый тип цели – object</p>

ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

№ задания	примеры тестовых заданий
71.	<p>В проверяемом (checked) блоке проверяется корректность</p> <p>(1) всех преобразований типов</p> <p>(2) преобразований внутри арифметического типа</p> <p>(3) только преобразований, выполняемых методом Parse</p>

	(4) только преобразований, выполняемых методами класса Convert
72.	<p>Отметьте правильные объявления константы</p> <p>(1) const double size = 55;</p> <p>(2) const int size = 55;</p> <p>(3) const byte size = 55;</p> <p>(4) const double size = 5.5;</p> <p>(5) const size = 55;</p>
73.	<p>Отметьте истинные высказывания:</p> <p>(1) операция typeof может выполняться только в небезопасных блоках</p> <p>(2) метод Parse определен для всех арифметических типов</p> <p>(3) метод Parse позволяет преобразовать строковый тип в арифметический</p> <p>(4) метод GetType возвращает результат того же типа, что и операция typeof</p>
74.	<p>В каком фрагменте возникнет ошибка:</p> <p>(1) sbyte sb1=1, sb2= 2, sb3 = sb1+ sb2</p> <p>(2) ulong ul1 =1, ul2= 2, ul3 = ul1 + ul2</p> <p>(3) int x; if (5 >4) x= 4</p> <p>(4) int x=5, y; if(x>4) y=x</p>
75.	<p>Проект на C#:</p> <p>1) может быть написан и скомпилирован только в среде Visual Studio .Net</p> <p>2) может быть создан в любом текстовом редакторе и скомпилирован без использования Visual Studio .Net</p> <p>3) может быть написан и скомпилирован в любой среде разработки, включающей компилятор языка C#</p>
76.	<p>Размерность массива определяется:</p> <p>1) числом его элементов</p> <p>2) числом его индексов</p> <p>3) по максимальному значению элементов</p>
77.	<p>В Visual Studio 2005 (Framework 2.0) для класса разрешается определять:</p> <p>1) только унарные операции</p> <p>2) только бинарные операции</p> <p>3) унарные и бинарные операции</p> <p>4) операции произвольной арности</p>
78.	Перечисление – это

	<p>1) абстрактный класс</p> <p>2) класс без собственных методов</p> <p>3) класс без собственных полей</p>
79.	<p>Тег <summary>, заданный для метода M:</p> <p>1) используется в подсказке при вызове метода M</p> <p>2) является комментарием и не используется в других целях</p> <p>3) представляет часть документации проекта при построении XML-отчета</p>
80.	<p>Экземпляр делегата следует рассматривать как:</p> <p>1) единственную нетипизированную ссылку на метод</p> <p>2) единственную ссылку на метод с жестко фиксированной сигнатурой, заданной делегатом</p> <p>3) совокупность ссылок - на метод с жестко фиксированной сигнатурой, заданной делегатом, на объект, вызвавший метод, на список присоединенных делегатов</p>
81.	<p>Сообщение о событии получает:</p> <p>1) только один объект одного класса, имеющего обработчик события, присоединенный к объекту, задающему событие</p> <p>2) несколько объектов одного класса, имеющего обработчик события, присоединенный к объекту, задающему событие</p> <p>3) все объекты одного класса, имеющего обработчик события, присоединенный к объекту, задающему событие</p> <p>4) все объекты всех классов, имеющих обработчики события, присоединенные к событию</p>
82.	<p>Для ограничения универсальности</p> <p>1) задаются специальные методы класса</p> <p>2) задаются специальные поля класса</p> <p>3) родовые параметры могут быть объявлены наследниками некоторого класса и или интерфейсов</p>
83.	<p>Отладка позволяет</p> <p>1) доказать, что в системе нет ошибок</p> <p>2) доказать, что в системе есть ошибки</p> <p>3) устранить все ошибки, существующие в системе</p>
84.	<p>Меню в формах:</p> <p>1) создать нельзя</p> <p>2) можно создать только вручную</p> <p>3) можно создать только программно</p> <p>4) можно создать вручную и программно</p>

85.	<p>Операции упаковки и распаковки (boxing, unboxing)</p> <p>1) преобразуют объекты значимого типа в ссылочный и обратно</p> <p>2) обе операции выполняются автоматически, не требуя явного задания преобразования типов</p> <p>3) задают преобразование строк из кодировки Unicode в кодировку ASCII и обратно</p>
86.	<p>Необходимость в преобразованиях может возникать:</p> <p>1) в выражениях</p> <p>2) при присваивании</p> <p>3) при замене формальных аргументов фактическими</p>
87.	<p>При объявлении переменной</p> <p>1) обязательно указание модификатора доступа</p> <p>2) обязательно указание типа</p> <p>3) обязательна ее инициализация</p> <p>4) обязательно указание атрибутов</p>
88.	<p>Под перегруженной операцией понимается операция:</p> <p>1) для которой существует несколько реализаций</p> <p>2) применимая к операндам разного типа</p> <p>3) сложность которой превышает сложность операции сложения</p>
89.	<p>В языке C# присваивание:</p> <p>1) это оператор языка, не применимый в выражениях</p> <p>2) это операция, применимая в выражениях</p> <p>3) синтаксически является операцией, чаще всего используемой в роли оператора</p>
90.	<p>Метод является:</p> <p>1) функцией, если тип возвращаемого значения отличен от void</p> <p>2) функцией, если указан модификатор метода function</p> <p>3) процедурой, если указан void как тип возвращаемого значения</p> <p>4) процедурой, если указан модификатор метода procedure</p>
91.	<p>Пространство имен – это:</p> <p>1) множество имен, включающих имена переменных проекта</p> <p>2) множество имен, включающих имена переменных класса</p> <p>3) модуль, содержащий встроенные пространства имен и классы, имена которых уникальны в пространстве имен</p>
92.	<p>Правда ли, что:</p> <p>1) корректность программы можно определить, только по отношению к ее</p>

	<p>спецификациям</p> <p>2) рекурсивной называется программа с циклами</p> <p>3) предусловие метода накладывает ограничение на входные данные</p>
93.	<p>Правда ли, что:</p> <p>1) над массивами определены операции поиска элементов и сортировки</p> <p>2) класс Array является абстрактным классом</p> <p>3) массив можно рассматривать как коллекцию и использовать для просмотра цикл foreach</p>
94.	<p>Правда ли, что:</p> <p>1) статический метод IsLetterOrDigit класса char возвращает true, если символ является буквой или цифрой</p> <p>2) большинство статических методов класса char перегружены</p> <p>3) все методы класса char являются статическими</p>
95.	<p>Правда ли, что:</p> <p>1) метод Join позволяет воссоздать строку в исходном виде, расцепленную методом Split</p> <p>2) метод Format может вызываться с разным числом параметров</p> <p>3) существуют константы класса StringBuilder</p>
96.	<p>Правда ли, что:</p> <p>1) статические поля представляют общий информационный пул для всех объектов классов, позволяя извлекать и создавать общую информацию</p> <p>2) конструктор не может быть объявлен с атрибутом private</p> <p>3) для класса C# нельзя задать деструктор</p>
97.	<p>Какой класс применяется для показа меню, отображаемого по нажатию правой кнопки мыши:</p> <p>1) ContextMenuStrip</p> <p>2) ToolStrip</p> <p>3) Pointer</p>
98.	<p>Тип данных, представляющий локальный набор таблиц и информацию об отношениях между ними, носит название:</p> <p>1) DataSet</p> <p>2) DataManager</p> <p>3) DataProvider</p> <p>4) DataAdapter</p>
99.	<p>Какие языки программирования могут использоваться в среде разработки Visual Studio:</p> <p>1) Managed C++</p>

	<p>2) Visual Basic.NET</p> <p>3) Java.NET</p> <p>4) C#</p> <p>(5) Pascal</p> <p>(6) Prolog</p>
100.	<p>Какой элемент интерфейса используется для вывода текстовой или графической информации, относящейся к работе приложения:</p> <p>1) MenuStrip</p> <p>2) StatusStrip</p> <p>3) ToolStrip</p>
101.	<p>Какие управляемые провайдеры реализованы в технологии ADO.NET:</p> <p>1) только SQL</p> <p>2) только OleDb</p> <p>3) как SQL, так и OleDb</p>
102.	<p>Какое ключевое слово в определении класса Form1 предназначено для определения класса, структуры или интерфейса, распределенных по нескольким файлам:</p> <p>1) private</p> <p>2) partial</p> <p>3) static</p>
103.	<p>Какой класс из перечисленных ниже является общим предком всех классов, обеспечивающим все производные классы общим набором важнейших возможностей:</p> <p>1) Form</p> <p>2) FormatControl</p> <p>3) Control</p>
104.	<p>Какой тип пространства имен System.Data представляет набор всех отношений между таблицами в DataSet:</p> <p>1) ConstraintCollection</p> <p>2) DataTableCollection</p> <p>3) DataRelationCollection</p>
105.	<p>Какой метод класса Application прекращает обработку сообщений для текущего потока:</p> <p>1) Exit()</p> <p>2) Thred()</p> <p>3) ExitThred()</p>
106.	<p>Выберите из списка элементы управления, поддерживающие сложную привязку данных:</p>

	1) ListBox 2) DataList 3) TextBox
107.	Определите неверный способ комментирования кода в C#: 1) # comment 2) // comment 3) /* comment */
108.	Какое свойство элемента TextBoxBase определяет, будет ли текст в TextBox автоматически переноситься на новую строку при достижении предельной длины строки: 1) AcceptsTab 2) Wordwrap 3) Multiline
109.	Укажите свойство элемента управления ComboBox, которое позволяет узнать или задать источник данных для элемента управления: 1) DisplayMember 2) ValueMember 3) DataSource
110.	Каким образом должны именоваться переменные в соответствии с соглашениями в C#: (1) sResult (2) Result (3) pszResult

**3.2. Вопросы к собеседованию (текущие опросы на практических занятиях)
УК-1 Способен осуществлять поиск, критический анализ и синтез информации,
применять системный подход для решения поставленных задач**

Номер вопроса	Пример вопросов в билете
111.	Понятие параллельных вычислений
112.	Необходимость параллельных вычислений
113.	Сдерживающие факторы параллельных вычислений
114.	Характеристика необходимых знаний и умений
115.	Пути достижения параллелизма
116.	Примеры параллельных вычислительных систем

ПКв-2 Способность разрабатывать и адаптировать прикладное программное обеспечение.

Номер вопроса	Пример вопросов в билете
117.	Суперкомпьютеры
118.	Кластеры
119.	Мультипроцессорные вычислительные системы
120.	Типовые схемы коммуникации процессоров
121.	Системные платформы для построения кластеров
122.	Платформа .NET
123.	Преимущества платформы .NET
124.	Понятие процесса операционной системы
125.	Понятие потока операционной системы
126.	Потоки и стратегия управления временем процессоров
127.	Приоритеты потоков
128.	Жизненный цикл потока
129.	Классы и объекты
130.	Принципы ООП
131.	Пространство имен
132.	Описание класса

ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

Номер вопроса	Пример вопросов в билете
133.	Модель вычислений в виде графа «операции-операнды»
134.	Схема параллельного выполнения алгоритма
135.	Определение времени выполнения параллельного алгоритма
136.	Показатели эффективности параллельного алгоритма
137.	Вычисление частных сумм последовательности числовых значений
138.	Оценка максимально достижимого параллелизма
139.	Анализ масштабируемости параллельных вычислений
140.	Алгоритмы маршрутизации в много процессорных системах
141.	Методы передачи данных в много процессорных системах
142.	Анализ трудоемкости основных операций передачи данных между двумя процессорами
143.	Анализ трудоемкости основных операций передачи данных от одного процессора всем остальным процессорам сети
144.	Циклический сдвиг при передаче данных между процессорами
145.	Методы логического представления топологии коммуникационной среды
146.	Оценка трудоемкости операций передачи данных для кластерных систем
147.	Использование виртуальных функций
148.	Абстрактные базовые классы
149.	Концепция интерфейсов
150.	Полиморфизм в С#
151.	Понятие делегата
152.	Описание делегатов
153.	Дженерики
154.	Многоадресная передача в делегате
155.	Обобщенные делегаты

156.	Понятие события
157.	Многоадресная передача события
158.	Использование событийных средств доступа
159.	Программа, управляемая событиями
160.	Анонимные методы в C#
161.	Лямбда-выражение
162.	Блоки операторов в лямбда-выражении
163.	Создание потока в C#
164.	Гонка данных в потоках
165.	Оператор lock. Критические секции.
166.	Понятие клинча
167.	Мягкие методы блокировки
168.	Мониторы
169.	Семафоры
170.	Задача об «обедающих философам»
171.	Класс ThreadPool
172.	Распараллеливание циклов (Класс Parallel)

3.3 Кейс- задания

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

Номер вопроса	Примеры текст задания
172.	<p>В результате объявления: <code>int x=1,y=2,z=((x y + ++x)>5)? x y : ++x +2</code> ; переменные x,y,z получат значения</p> <p>1) <code>x=1; y=2; z=4</code></p> <p>2) <code>x=2; y=2; z=4</code></p> <p>3) <code>x=3; y=2; z=5</code></p> <p>4) возникнет ошибка на этапе трансляции</p> <p>5) возникнет ошибка на этапе выполнения</p> <p>6) <code>x=1; y=2; z=3</code></p>

173.	<p>Какая из процедур вызовет ошибку периода трансляции?</p> <p>1) <code>public static void L1_V1_Q31(int[] A, int S) { S=0; for(int i =0; i<A.Length; i++) S+=A[i]; }</code></p> <p>2) <code>public static void L1_V1_Q32(ref int[] A, ref int S) { S=0; for(int i =0; i<A.Length; i++) S+=A[i]; }</code></p> <p>3) <code>public void L1_V1_Q33(val int[] A, val int S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; return(S); }</code></p> <p>4) <code>public static int L1_V1_Q34(object[] A) { int S=0; foreach(int i in A) S+=(int)i; return(S); }</code></p>
174.	<p>Пусть процедура ItWorks имеет следующий заголовок: <code>public void ItWorks(object[] A)</code>. Какой массив может быть передан ей в качестве аргумента?</p> <p>1) <code>int[] x</code></p> <p>2) <code>string[] s</code></p> <p>3) <code>Person[] p;</code> (Person – класс, определенный пользователем)</p> <p>4) <code>Anketa[] a;</code> (<code>Anketa – структура, определенная пользователем</code>)</p>
175.	<p>Какие высказывания справедливы для класса <code>StringBuilder</code>?</p> <p>1) существует конструктор класса, которому можно передать аргумент класса <code>string</code></p> <p>2) переменные этого класса можно создавать без явного вызова конструктора</p> <p>3) набор операций класса тот же, что и у класса <code>string</code></p> <p>4) классы <code>string</code> и <code>StringBuilder</code> относятся к неизменяемым классам</p> <p>5) методы <code>Remove</code>, <code>Insert</code>, <code>Replace</code> возвращают новую строку, представляющую результат операции</p>
176.	<p>Метод-Свойство класса</p> <p>1) может иметь атрибут доступа <code>private</code></p> <p>2) может включать методы <code>get</code> и <code>set</code></p> <p>3) включает только один из методов <code>get</code> или <code>set</code></p> <p>4) позволяет реализовать различные стратегии доступа к закрытым полям класса</p> <p>5) обязательно должно быть указано для каждого закрытого поля класса</p>
177.	<p>При создании объектов перечислений</p> <p>1) нельзя вызывать конструктор</p> <p>2) требуется их явная инициализация</p> <p>3) объекты получают значение, даже если инициализация не задана</p>
178.	<p>В родительском классе описан метод <code>public virtual void M(int x) {}</code>. Какое объявление в классе потомке вызовет ошибку на этапе компиляции, не допускающую выполнения проекта?</p> <p>1) <code>override void M(int x){}</code></p> <p>2) <code>public override void M(int item){}</code></p>

	3) <code>public new void M(int x){}</code>
179.	Интерфейс <code>ISerializable</code> 1) автоматически реализует глубокую сериализацию 2) позволяет управлять процессом сериализации 3) имеет два метода, которые должен реализовать класс, наследующий интерфейс 4) конфликтует с атрибутом класса <code>Serializable</code>
180.	Решение – это: 1) специальный тип проекта 2) специальный компонент проекта 3) модуль, содержащий проекты и другие ресурсы 4) стартовый проект на языке C#
181.	Функция обратного вызова – это: (1) функция, прямо или косвенно вызывающая саму себя (2) функция f, вызывающая функцию g, переданную ей в качестве одного из аргументов, с сигнатурой, удовлетворяющей контракту, заданному функцией f (3) пара функций, взаимно вызывающих друг друга (4) функция высшего порядка
182.	Объявление события в классе может представлять собой: (1) объявление метода класса (2) объявление поля класса (3) объявление метода-свойства класса, включающего методы <code>get</code> и <code>set</code> (4) объявление метода-свойства класса, включающего методы <code>add</code> и <code>remove</code>
183.	Какие фрагменты кода корректны в предположении, что используемые родительские классы и интерфейсы определены корректно? 1) <code>class A<B, C> { B b; C c;}</code> 2) <code>class A1<A1, B> { A1 a1; B b;}</code> 3) <code>interface IC<T> { } class A2 : IC<T> { }</code> 4) <code>class D<T1, T2>{ } class A3 where B : D<int, int> { }</code>
184.	В каких фрагментах возникнет ошибка компиляции: 1) <code>int x=5, y =2*2+1, z; if (y=x) z=2*x; else z=x+y;</code> 2) <code>int a=2, b=3,c=5, x;if(a<b)if(c>a)if(c>a+b) x=a+b+c;</code> 3) <code>int x, y, z; if (5 >4) x=5; y=x; z=y;</code>

185.	<p>В каких вызовах возникнет ошибка, если задано описание <code>int x=1; int z=0; int p(int x, out int y){...}</code></p> <p>1) <code>x=p(out 77,z)</code></p> <p>2) <code>x=p(77+z, out z)</code></p> <p>3) <code>p(77+z, out z)</code></p> <p>4) <code>x=p(77, out 77+z)</code></p>
186.	<p>Три массива: <code><int[] ar1 = new int[5]; double[] ar2 = {5.5, 6.6, 7.7}; int[,] ar3 = new Int32[3,4]; ></code> принадлежат</p> <p>1) одному и тому же классу</p> <p>2) разным классам, но у которых есть общий родительский класс <code>Array</code></p> <p>3) разным классам, общность которых определяется только родительским классом <code>object</code></p>
187.	<p>Для выделения из текста подстроки, состоящей из трех символов, в которой первый и третий символы являются буквами т или к, а второй символ произвольной буквой кириллицы в нижнем регистре, следует использовать шаблон:</p> <p>1) <code>@ "т к.т к"</code></p> <p>2) <code>@ "[т к][а-я][т к]"</code></p> <p>3) <code>@ "[т к]?[т к]"</code></p>
188.	<p>Изменить значение строки класса <code>string</code></p> <p>1) можно только используя статические методы класса</p> <p>2) можно только используя динамические методы класса</p> <p>3) вообще невозможно</p>
189.	<p>Для выделения из текста подстроки, в которой x и y разделены знаком операции + или -, следует использовать шаблон:</p> <p>1) <code>@ "(x+y) (x-y)"</code></p> <p>2) <code>@ "x[+ -]y"</code></p> <p>3) <code>@ "x[+ -]y"</code></p>
190.	<p>Какой параметр позволяет задать текст заголовка диалогового окна <code>MessageBox</code>: <code>public static DialogResult Show { string message, string caption, MessageBoxButtons btns, MessageBoxIcon icon, MessageBoxDefaultButton defButton, MessageBoxOptions opt };</code></p> <p>1) <code>message</code></p> <p>2) <code>caption</code></p> <p>3) <code>icon</code></p>
191.	<p>Какие программы корректны по отношению к своим спецификациям?</p> <p>1) <code>{n=5}int fact(int n){int f=0; for (int i=0; i< n; i++) f*=i; return(f);}{result = n!}</code></p> <p>2) <code>{n=5}int fact(int n){return(120);}{result = n!}</code></p>

	<p>3) {false}int fact(int n){int f=0; for (int i=0; i<n; i++) f*=i; return(f);}{result = n!}</p> <p>4) {(n целое) & (n>0)}int fact(int n){int f=1; for (int i=2; i<n+1; i++) f*=i; return(f);}{result = n!}</p>
192.	<p>Отметьте правильные объявления:</p> <p>1) char a = new char("a");</p> <p>2) char b = "b";</p> <p>3) char[] c = new char[5];</p> <p>4) char[] d = {'a','b','c','d'};</p>
193.	<p>Отметьте корректные объявления делегата – функционального класса:</p> <p>1) public class D = delegate void(string s)</p> <p>2) Delegate D1 = new Delegate(void(string s)</p> <p>3) public delegate void D2 (string s)</p> <p>4) delegate int T(int x1, ref int x2)</p> <p>5) public delegate T1(int x)</p>
194.	<p>В результате объявления: int x=1,y=2,z=((x y + ++x)>5)? x y : ++x +2 ; переменные x,y,z получают значения</p> <p>1) x=1; y=2; z=4</p> <p>2) x=2; y=2; z=4</p> <p>3) x=3; y=2; z=5</p> <p>4) возникнет ошибка на этапе трансляции</p> <p>5) возникнет ошибка на этапе выполнения</p> <p>6) x=1; y=2; z=3</p>
195.	<p>Какая из процедур вызовет ошибку периода трансляции?</p> <p>1) public static void L1_V1_Q31(int[] A, int S) { S=0; for(int i =0; i<A.Length; i++) S+=A[i]; }</p> <p>2) public static void L1_V1_Q32(ref int[] A, ref int S) { S=0; for(int i =0; i<A.Length; i++) S+=A[i]; }</p> <p>3) public void L1_V1_Q33(val int[] A, val int S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; return(S); }</p> <p>4) public static int L1_V1_Q34(object[] A) { int S=0; foreach(int i in A) S+=(int)i; return(S); }</p>
196.	<p>Какие фрагменты кода корректны в предположении, что используемые родительские классы и интерфейсы определены корректно?</p> <p>1) class A<B, C> { B b; C c;}</p> <p>2) class A1<A1, B> { A1 a1; B b;}</p> <p>3) interface IC<T> { } class A2 : IC<T> { }</p> <p>4) class D<T1, T2>{ } class A3 where B : D<int, int> { }</p>

197.	<p>В каком фрагменте возникнет ошибка:</p> <p>1) sbyte sb1=1, sb2= 2, sb3 = sb1+ sb2</p> <p>2) ulong ul1 =1, ul2= 2, ul3 = ul1 + ul2</p> <p>3) int x; if (5 >4) x= 4</p> <p>4) int x=5, y; if(x>4) y=x</p>
198.	<p>В каких вызовах возникнет ошибка, если задано описание int x=1; int z=0; int p(int x, out int y){...}</p> <p>1) x=p(out 77,z)</p> <p>2) x=p(77+z, out z)</p> <p>3) p(77+z, out z)</p> <p>4) x=p(77, out 77+z)</p>
199.	<p>Три массива: <int[] ar1 = new int[5]; double[] ar2 ={5.5, 6.6, 7.7}; int[,] ar3 = new Int32[3,4]; > принадлежат</p> <p>1) одному и тому же классу</p> <p>2) разным классам, но у которых есть общий родительский класс Array</p> <p>3) разным классам, общность которых определяется только родительским классом object</p>
200.	<p>Для выделения из текста подстроки, в которой x и y разделены знаком операции + или -, следует использовать шаблон:</p> <p>1) @"(x+y) (x-y)"</p> <p>2) @"x[+ -]y"</p> <p>3) @"x[+ -]y"</p>
201.	<p>Верно ли, что:</p> <p>(1) код на C# <byte b = 1; b+=1;> эквивалентен коду <byte b = 1; b = b+1;></p> <p>2) класс Random имеет несколько конструкторов</p> <p>3) операция присваивания выполняется справа налево</p>

ПКв-2 Способность разрабатывать и адаптировать прикладное программное обеспечение.

Номер вопроса	Примеры текст задания
202.	<p>Для программы, вычисляющей сумму первых n элементов массива: S=A[0]; k=0; while(k !=(n-1)) { k++; S+=A[k];} инвариантом цикла являются</p>

	<p>(1)</p> $S = \sum_{i=0}^{n-1} A[i]$ <p>(2)</p> $S = \sum_{i=0}^{k-1} A[i]$ <p>3) $S = \sum_{i=0}^k A[i]$</p> <p>4) $k \geq 0$</p> <p>5) $S = A[k]$</p>
203.	<p>Отметьте правильные объявления:</p> <p>(1) double[] v1 = {1,2,3}, v2={3,4,5}, v3={6,7,8};</p> <p>(2) double[,] w1 = new double[3,3];</p> <p>(3) double[][] z1 = new double[3][];</p> <p>(4) object[][] z2 = new object[3][];</p>
204.	<p>Пусть массив x принадлежит классу double[[]]. В вызове x.M роль M может играть:</p> <p>1) метод Sort</p> <p>2) свойство Length</p> <p>3) метод GetLength</p> <p>4) метод Copy</p> <p>5) метод CopyTo</p>
205.	<p>Даны объявления: char ch = 'A'; char[] ar ={'B', 'C', 'D'}, ar1= new char[3]; Отметьте синтаксически корректные операторы:</p> <p>1) ar1 = char.Parse("BCD")</p> <p>2) ar.CopyTo(ar1,0)</p> <p>3) ch = char.Copy("B")</p> <p>4) ar1= Array.Copy(ar)</p>
206.	<p>Словом будем называть непустую последовательность букв кириллицы, обрамленную белыми пробелами. Какой образец распознает слова, соответствующие этому определению?</p> <p>1) @"\\s\\W+\\s"</p> <p>2) @"\\s*[а-я]+\\s"</p>

	<p>3) @"[A-Яa-я]+"</p> <p>4) @"\s[A-Яa-я]+\s"</p> <p>5) @"\s[A-Яa-я]*\s"</p>
207.	<p>Пусть задано описание интерфейсов: interface IN{string M(string s);} interface IP{string M(string s); string M1(int s);} interface IQ{int M(int s);}. Какие из объявлений классов содержат ошибки</p> <p>1) public class C1:IP{string IP.M(string s){return (s+s);} string IP.M1(int x){return x.ToString();}public int M (int s) { return s++;}}</p> <p>2) public class C1:IP,IN{string IP.M(string s){return (s+s);} string IP.M1(int x){return x.ToString();}}</p> <p>3) public class C1:IP,IN{public string M(string s){return (s+s);} public string M1(int x){return x.ToString();}}</p> <p>4) public class C1:IP,IN,IQ{public string M(string s){return (s+s);} public string M1(int x){return x.ToString();}}</p>
208.	<p>Дано объявление делегата: public delegate double D (double x); Какие объявления его экземпляров будут корректны в определенном контексте?</p> <p>1) D d1 = new D(Math.Sin)</p> <p>2) D d3 = new D(x.f1)</p> <p>3) D d4 = new D(Person.f2)</p> <p>4) D d5 = new D(f3)</p>
209.	<p>Чему будет равно x в результате выполнения следующего фрагмента int a=2, b=3,c=5, x=0; if(a>b)if(c>a)if(c>a+b) x=a+b+c; else x=7;else x=12;</p> <p>1) 0</p> <p>2) 7</p> <p>3) 10</p> <p>4) 12</p>
210.	<p>В результате объявления: int x=2,y=5,z=((x y +x++)>=9) ? x y : ++x +2 ; переменные x, y, z получают значения</p> <p>1) x=3; y=5; z=7</p> <p>2) x=4; y=5; z=6</p> <p>3) x=2; y=5; z=7</p> <p>4) возникнет ошибка на этапе трансляции</p> <p>5) возникнет ошибка на этапе выполнения</p> <p>6) x=3; y=5; z=8</p>
211.	<p>Какие из процедур корректно вычислят сумму переданного им массива int?</p>

	<p>1) <code>public static void L1_V1_Q31(int[] A, int* S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; }</code></p> <p>2) <code>public static void L1_V1_Q32(ref int[] A, ref int S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; }</code></p> <p>3) <code>public void L1_V1_Q33(val int[] A, val int S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; return(S); }</code></p> <p>4) <code>public static int L1_V1_Q34(object[] A) { int S=0; foreach(int i in A) S+=(int)i; return(S); }</code></p>
212.	<p>Пусть процедура <code>ItWorks</code> имеет следующий заголовок: <code>public void ItWorks(Array A)</code>. Какой массив может быть передан ей в качестве аргумента?</p> <p>(1) <code>int[] x</code></p> <p>(2) <code>int [][] y</code></p> <p>(3) <code>string[] s</code></p> <p>(4) <code>Person[] p</code>; (<code>Person</code> – класс, определенный пользователем)</p> <p>(5) <code>Anketa[] a</code>; (<code>Anketa</code> – структура, определенная пользователем)</p>
213.	<p>Какие слова принадлежат языку, заданному регулярным выражением: <code>@ "[a-zA-Z]\w*\s*=\s*\w+(\s*\+\s*\w+)+"</code></p> <p>1) <code>x=y+z</code></p> <p>2) <code>XYZ = 2 + 3 + 7</code></p> <p>3) <code>Agent007 = James + Bond</code></p> <p>4) <code>x5 = 5x</code></p> <p>5) <code>5y= y5</code></p>
214.	<p>В родительском классе описан метод <code>public virtual void M(int x) {}</code>. Какое объявление в классе потомке вызовет ошибку на этапе компиляции?</p> <p>(1) <code>public override void M(int item){}</code></p> <p>(2) <code>public new void M(int x){}</code></p> <p>(3) <code>public override virtual void M(int x){}</code></p> <p>(4) <code>public virtual void M(int x, int y){}</code></p>
215.	<p>В каких фрагментах кода возникнут ошибки (предупреждения) на этапе компиляции?</p> <p>1) <code>public class B<T> { } public class S : B<int> { }</code></p> <p>2) <code>public class Test1:IComparable<T> {public T x, y;}</code></p> <p>3) <code>public class B1<T> {public T M<T>(T x) { return x; } }</code></p> <p>4) <code>public class B2 <T> {public T1 M<T1>(T1 x) { return x; } }</code></p>
216.	
217.	В каких фрагментах возникнет ошибка:

	<p>1) double x; x=Math.Sin(1);</p> <p>2) Math M1 = new Math(); int x; x= M1.Sin(1);</p> <p>3) int x; x=Random.Next();</p> <p>4) int x; Random R1 = new Random(); x=R1.Next();</p>
218.	<p>В каких фрагментах возникнет ошибка:</p> <p>1) int x=5, y = 2*2+1, z; if(y >=x) z=2*x; else z=x+y;</p> <p>2) int x=5; {int y = 7;} int z = x+y;</p> <p>3) string s, s1, s2, s3; s=s1=s2=s3="око"; switch(s) { case s1: x=3; break; case s2: case s3: x=5; break; }</p> <p>4) for (int Sum=0, i=0; i<N; i++) Sum += Arr[i]; (N и Arr определены должным образом)</p>
219.	<p>Дано перечисление enum TC {red, blue, yellow} и три переменных c1, c2, c3 этого класса. В результате присваиваний: c1 = TC.yellow; c2 = TC.blue; c3 = c1 c2; эти переменные получают следующие значения базового типа:</p> <p>1) 0, 1, 2</p> <p>2) 2, 1, 3</p> <p>3) 1, 2, 3</p> <p>4) 2, 1, 0</p>
220.	<p>Отметьте вызовы процедур, в которых переменная sum получит значение 21. int sum =0; int[] mas = {7, 6, 8}; object[] mas2 = {3, 9, 9}; Описания процедур:public static void L1_V1_Q31(int[] A, int S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; } public static void L1_V1_Q32(ref int[] A, ref int S) { S=0; for(int i =0; i< A.Length; i++) S+=A[i]; } public static int L1_V1_Q34(object[] A) { int S=0; foreach(int i in A) S+=(int)i; return(S); }</p> <p>1) Questions.L1_V1_Q31(mas, sum)</p> <p>2) Questions.L1_V1_Q32(ref mas, ref sum)</p> <p>3) sum = Questions.L1_V1_Q34(mas)</p> <p>4) sum = Questions.L1_V1_Q34(mas2)</p>
221.	<p>Пусть процедура ItWorks имеет следующий заголовок: public void ItWorks(int[] A). Какой массив может быть передан ей в качестве аргумента?</p> <p>1) int[] x</p> <p>2) byte[] y</p> <p>3) string[] s</p> <p>4) Person[] p; (Person – класс, определенный пользователем).</p>
222.	<p>Даны объявления структур: Point pt1 = new Point(3,5),pt2; Size sz1 = new Size(7,10), sz2; Какие операторы присваивания ошибочны?</p> <p>1) pt2 = pt1 +sz1;</p>

	<p>2) <code>sz2 = pt2;</code></p> <p>3) <code>pt2 +=pt1;</code></p> <p>4) <code>pt2 = pt1 +1;</code></p>
223.	<p>В родительском классе описан метод <code>public void M(int x) {}</code>. Какие объявления в классе потомке вызовут ошибки или предупреждения на этапе компиляции?</p> <p>1) <code>public override void M(int x) {}</code></p> <p>2) <code>public new void M(int x) {}</code></p> <p>3) <code>public new void M(int x, int y){}</code></p> <p>4) <code>public void M(int x){}</code></p> <p>5) <code>private new void M(int x){}</code></p>
224.	<p>В каких строках кода возникнут ошибки (предупреждения) на этапе трансляции?</p> <p>1) <code>abstract public class A5<T> { abstract public void M(T x); }</code></p> <p>2) <code>public class SA5<T> : A5<T> { public override void M(T x) { T y = x; } }</code></p> <p>3) <code>public class B6 { SA5<int> obj1 = new SA5<int>(); }</code></p> <p>4) <code>public class B7 { SA5<T> obj1 = new SA5<T>(); }</code></p>
225.	<p>В каких фрагментах возникнет ошибка:</p> <p>1) <code>float f1=1.0, f2= 2.0, f3 = f1+ f2;</code></p> <p>2) <code>double d1 =1.0, d2= 2.0, d3 = d1 + d2;</code></p> <p>3) <code>int u =5, v, w; if (u >0) v= 4; w=v;</code></p> <p>4) <code>int x=5, y, z; if(x>4) y=x; else y=4; z=y;</code></p>
226.	<p>В каких фрагментах возникнет ошибка:</p> <p>1) <code>x=;; y=7</code></p> <p>2) <code>int x=5, z; {int y=7; z=y+1;} int w = x+z;</code></p> <p>3) <code>string s="кок"; switch(s) { case "око": break; case "кок": case "пок": s="тук"; s=s+s; break; }</code></p> <p>4) <code>int x=5, y, z; if(x>4) y=x; else y=4; z=y;</code></p>

ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

Номер вопроса	Пример вопросов в билете
227.	<p>В каких фрагментах не возникнет ошибка?</p> <p>1) <code>char[] s = new char[3]; s = "abc".ToCharArray();</code></p> <p>2) <code>char[] s = new char[3]; s = "abc";</code></p>

	3) char[] s = new char[3] {'b', 'c', 'd'};
228.	<p>Дано перечисление enum TC {red, blue, yellow} и три переменных c1, c2, c3 этого класса. В результате присваиваний: c1 = TC.yellow; c2 = TC.blue; c3 = c1 c2; переменная c3 получит значение</p> <p>1) red</p> <p>2) blue</p> <p>3) yellow</p> <p>4) ни одно из этих значений</p>
229.	<p>Для программы, вычисляющей сумму первых n элементов массива S=A[0]; k=0; while(k !=(n-1)) { k++; S+=A[k];} подходящими инвариантами цикла являются</p> <p>(1)</p> $S = \sum_{i=0}^{n-1} A[i]$ <p>(2)</p> $S = \sum_{i=0}^{k-1} A[i]$ <p>3) <u>$S = \sum_{i=0}^k A[i]$</u></p> <p>(4) S= A[k]</p>
230.	<p>Отметьте правильные объявления:</p> <p>1) int[][.] mx = { new int[2,3], new int[3,2] };</p> <p>2) object[][] mz = { new int[3], new double[4] };</p> <p>3) int n = 10; string[,] ms = new string[n,2*n];</p> <p>4) double[] md = new double[5] = {1.5, 2.7, 3.9};</p> <p>5) float[] mf = {1.5, 2.7, 3.9};</p>
231.	<p>Отметьте правильные объявления:</p> <p>1) char[] e = new char[4]{'a', 'b', 'c', 'd'};</p> <p>2) int n = 4; char[] g = new char[n]{'a','b','c','d'};</p> <p>3) char f1;</p>

232.	<p>Отметьте фрагменты, в которых не возникнут ошибки при компиляции и выполнении:</p> <p>1) string c1= new string();</p> <p>2) string c2= new string("ABC");</p> <p>3) int n = 5; string c3= new string ('a', 2*n);</p> <p>4) char[] car = new char[3]; car[1] ='a'; car[2] ='b'; string s =new string(car,0,2);</p>
233.	<p>Пусть задано описание интерфейса и класса: interface IP{string M(string s); string M1(int s);} public class C1:IP{string IP.M(string s){return (s+s);} string IP.M1(int x){return x.ToString();}public int M (int s) { return (s++);}}. Какие из объявлений в клиентском классе выполнимы</p> <p>1) C1 it1 = new C1(); it1.M(7777)</p> <p>2) C1 it2 = new C1(); string s ="ss"; s =it2.IP.M(s)</p> <p>3) C1 it3 = new C1(); string s ="ss"; s =((IP)it3).M(s)</p> <p>4) IP it4 = new IP(); string s= "ss"; s = it4.M(s)</p> <p>5) IP it5 = (IP) new C1(); string s= "ss"; s = it5.M(s)</p>
234.	<p>Отметьте правильные объявления внутри метода:</p> <p>1) int x, double y;</p> <p>2) int x, y, z = 77;</p> <p>3) int x =77, y = new int();</p>
235.	<p>Тело цикла for (int i = 1, j=-10; j<10; i--) j++;</p> <p>1) ни разу не выполнится</p> <p>2) выполнится 3 раза</p> <p>3) произойдет зацикливание</p> <p>4) выполнится конечное число раз, большее 3-х</p>
236.	<p>В результате объявления: int x=2, y=3, z= ((x --y +y)>5) ? x y : ++x +2 ; переменные x, y, z получают значения</p> <p>1) x=1; y=2; z=4</p> <p>2) x=2; y=2; z=4</p> <p>3) x=2; y=2; z=2</p> <p>4) возникнет ошибка на этапе трансляции</p> <p>5) возникнет ошибка на этапе выполнения</p>

	6) x=2; y=3; z=3								
237.	<p>Даны объявления: string s = "Hello, Dolly!"; const int size =15; char[] c0 = new char[size], c1= new char[size]; char[] c2 = new char[size], c3= new char[size]; char[] c4 = new char[size]; Группа операторов выполняется после этих объявлений. Отметьте операторы, не приводящие к ошибке на этапе выполнения:</p> <p>1) c0= s.ToCharArray()</p> <p>2) c1=c0</p> <p>3) Array.Copy(c0,c2,2*size)</p> <p>4) c0.CopyTo(c3,0)</p> <p>5) c1[7]='P'</p>								
238.	<p>Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс после каждой итерации.</p> <table border="1" data-bbox="316 875 1182 1061"> <thead> <tr> <th>Номер варианта</th> <th>N</th> <th>M</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>9</td> <td>200</td> <td>208</td> <td>200</td> </tr> </tbody> </table> <pre> using System; using System.Collections.Generic; using System.Linq; using System.Text; using System.Threading; namespace задача1 { class Program { private static int Fp=0; private static int N = 200; private static int M = 208; static void Main(string[] args) { </pre>	Номер варианта	N	M	Q	9	200	208	200
Номер варианта	N	M	Q						
9	200	208	200						

```

Thread[] Threadfirst=new Thread[5];

for (int i = 0; i < 5; i++){

    Fp = i + 1;
    Threadfirst[i] = new Thread(stream);
    Threadfirst[i].Start();

}

}

static void write(int n)
{
    for (int i = N; i <= M; i++)
    {
        Console.Write("итерация {0}, поток №
{1}",i.ToString(),n.ToString()+"\n");
    }
}

static void stream()
{
    while (true)
    {
        write(Fp);
        Thread.Sleep(50);
    }
}

}

```

```
}
```

239.

Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс после каждой итерации.

Номер варианта	N	M	Q
2	34	100	140

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;

namespace задача1
{
    class Program
    {
        private static int N = 34;
        private static int M = 100;
        private static int pN = 0;

        static void Main(string[] args)
        {
            Thread[] Threadfirst=new Thread[5];
```

```

        for (int i = 0; i < 5; i++){

            pN = i + 1;
            Threadfirst[i] = new Thread(stream);
            Threadfirst[i].Start();

        }
    }

    static void write(int n)
    {
        for (int i = N; i <= M; i++)
        {
            Console.Write("итерация {0}, поток №
{1}",i.ToString(),n.ToString()+"\n");
        }
    }

    static void stream()
    {
        while (true)
        {
            write(pN);
            Thread.Sleep(5000);
        }
    }
}

```

```
}  
}
```

240.

Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс после каждой итерации.

Номер варианта	N	M	Q
3	30	50	260

```
using System;  
  
using System.Collections.Generic;  
  
using System.Linq;  
  
using System.Text;  
  
using System.Threading;  
  
namespace задача1  
{  
    class Program  
    {  
  
        private static int N = 30;  
        private static int M = 50;  
        private static int pN = 0;  
  
        static void Main(string[] args)  
        {  
            Thread[] Threadfirst=new Thread[5];  
            for (int i = 0; i < 5; i++){  
  
                pN = i + 1;  
                Threadfirst[i] = new Thread(stream);  
                Threadfirst[i].Start();  
            }  
        }  
    }  
}
```

```

        Thread.Sleep(50);

    }
}

static void write(int n)
{
    for (int i = N; i <= M; i++)
    {
        Console.Write("итерация {0}, поток №
{1}", i.ToString(), n.ToString()+"\n");
    }
}

static void stream()
{
    while (true)
    {
        write(pN);
    }
}
}
}

```

241.

Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс

после каждой итерации.

Номер варианта	N	M	Q
5	120	158	300

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading;
```

```
namespace задача
```

```
{
```

```
    class Program
```

```
    {
```

```
        private static int k=0;
```

```
        static void Main(string[] args)
```

```
        {
```

```
            Thread[] Threadfirst=new Thread[5];
```

```
            for (int i = 0; i < 5; i++){
```

```
                k = i;
```

```
                Threadfirst[i] = new Thread(stream);
```

```
                Threadfirst[i].Start();
```

```
            }
```

```
        }
```



```

static void write(int n)
{
    int N = 120;
    int M = 158;

    for (int i = N; i <= M; i++)
    {
        Console.WriteLine("итерация {0}, поток №
{1}", i.ToString(), n.ToString()+"\n");
        Thread.Sleep(50);
    }
}

static void stream()
{
    while (true)
    {
        write(k);
    }
}
}
}

```

242.

Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс

после каждой итерации.

Номер варианта	N	M	Q
8	108	400	390

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;

namespace Задание1
{
    class Program
    {
        private static int N = 108;
        private static int M = 400;
        private static int Npotoka = 0;

        static void Main(string[] args)
        {
            Thread Thread1 = new Thread(potok);
            Thread1.Name = "1 поток";
            Thread1.Start();

            Thread Thread2 = new Thread(potok);
            Thread2.Name = "2 поток";
            Thread2.Start();

            Thread Thread3 = new Thread(potok);
```

```

Thread3.Name = "3 поток";

Thread3.Start();

Thread Thread4 = new Thread(potok);

Thread4.Name = "4 поток";

Thread4.Start();

Thread Thread5 = new Thread(potok);

Thread5.Name = "5 поток";

Thread5.Start();

}

static void write(int n)
{
    for (int i = N; i <= M; i++){
        Console.Write("итерация {0}, {1}", i.ToString(),
Thread.CurrentThread.Name + "\n");
    }
}

static void potok()
{
    while (true){
        write(Npotoka);

        Thread.Sleep(50);
    }
}
}

```

243.

Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс после каждой итерации.

Номер варианта	N	M	Q
9	200	208	200

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

namespace задача1
{
    class Program
    {
        private static int Fp=0;

        private static int N = 200;

        private static int M = 208;

        static void Main(string[] args)
        {
            Thread[] Threadfirst=new Thread[5];

            for (int i = 0; i < 5; i++){

                Fp = i + 1;

                Threadfirst[i] = new Thread(stream);

                Threadfirst[i].Start();

            }

        }

        static void write(int n)
        {
```

```

        for (int i = N; i <= M; i++)
        {
            Console.WriteLine("итерация {0}, поток №
{1}",i.ToString(),n.ToString()+"\n");
        }
    }

    static void stream()
    {
        while (true)
        {
            write(Fp);
            Thread.Sleep(50);
        }
    }
}
}
}

```

244.

Создание приложения, которое в начале своей работы устанавливает максимальное количество рабочих потоков и потоков ввода-вывода в пуле потоков. Реализовать цикл, состоящий из K итераций, в котором происходил бы вызов метода, запрошенного из пула потоков. Данный метод выводит на экран случайное число от N до M, а также идентификатор потока.

Номер варианта	N	M	Q
9	200	208	200

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```

using System.Text;

using System;

using System.Threading;

namespace Задача2
{

class Program
{
    private static int N = 200;

    private static int M = 208;

    static void Main(string[] args)
    {
        int Q=200;

        int nWorkerThreads;

        int nCompletionThreads;

        ThreadPool.SetMaxThreads(Q,Q);

        ThreadPool.GetMaxThreads(out nWorkerThreads, out nCompletionThreads);

        Console.WriteLine("Максимальное количество потоков: " + nWorkerThreads
            + "\nПотоков ввода-вывода доступно: " + nCompletionThreads);

        for (int i = 0; i <= Q; i++)

            ThreadPool.QueueUserWorkItem(JobForAThread);

        Thread.Sleep(3000);

        Console.ReadLine();

    }
}

```

```

static void JobForAThread(object state)
{
    Random random=new Random();
    Console.WriteLine("число = {0}, выполнение внутри потока из пула {1}",
                      random.Next(N, M),
Thread.CurrentThread.ManagedThreadId);
    Thread.Sleep(50);
}
}
}

```

245.

Создание приложения, реализующего одновременную параллельную работу Q таймеров, каждый из которых с периодичностью раз в (N*10) мс выводит на экран случайное число от N до M и номер текущего потока.

Номер варианта	N	M	Q
9	200	208	200

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Timers;
using System.Threading;

namespace задача3
{
    class MyTimer

```

```

{

    private System.Timers.Timer Tim;

    private int Number=0;

    private int N=0;

    private int M=0;

    System.Random r;

    public MyTimer(int num, int n, int m, System.Random R)
    {
        Number = num;

        r = R;

        N = n;

        M = m;

        Tim = new System.Timers.Timer();

        Tim.Interval=10*N;

        Tim.Elapsed += tEvent;

        Tim.AutoReset = true;

        Tim.Enabled = true;

    }

    private void tEvent(Object S, ElapsedEventArgs e)
    {
        Console.WriteLine("Таймер №{0} число от {1} до {2} ={3}",
        Number.ToString(), N.ToString(), M.ToString(), Convert.ToString(r.Next(N,M)));

    }

}

class Program

```



```

{
    private static System.Random r;

    static void Main(string[] args)
    {
        r = new Random((int)DateTime.Now.Ticks);

        int Q = 200;

        int N = 200;

        int M = 208;

        for (int i = 0; i < Q; i++)
        {
            MyTimer myTimer = new MyTimer(i,N,M,r);
        }

        Console.ReadLine();
    }
}

```

246.

Сгенерировать случайным образом целочисленную матрицу $A(x,y)$ размерами $N \times M$, элементы которой находятся в диапазоне $[0; 255]$. $0 \leq x \leq N-1$ и $0 \leq y \leq M-1$

Для каждого пикселя (кроме крайних, где $x = 0$ или $N-1$; $y = 0$ или $M-1$) выполнить вычисление:

$Q_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 A_{x+i,y+j} * H_{X_{i+1,j+1}}$, где матрица H_X выглядит следующим образом:

$$H_x = \begin{matrix} f_{00} & f_{10} & f_{20} & -1 & 0 & +1 \\ f_{01} & f_{11} & f_{21} & -2 & 0 & +2 \\ f_{02} & f_{12} & f_{22} & -1 & 0 & +1 \end{matrix}$$

Q – результирующая матрица размерами NxM.

Номер варианта	Способ распараллеливания	N	M
9	Создать 2 потока переднего плана, где каждый выполняет вычисления для половины матрицы	1000	600

Решение без потоков

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading;

namespace Задача5
{
    class Program
    {
        public static int[,] Massive;
        public static int[,] Q;
        public static int[,] A={{-1,0,1},{-2,0,2},{1,0,1}};
        public static int[,] tempMas = { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
        public static int N;
```

```

public static int M;

public static int im=0;

public static int jm=0;

static IEnumerable<int[,]> GetBlocks(int[,] array, int blockSize)
{
    for (int i = 0; i <= array.GetLength(0) - blockSize; i += blockSize)
        for (int j = 0; j <= array.GetLength(1) - blockSize; j += blockSize)
            yield return GetMatrix3(array, i, j, blockSize);
}

static int[,] GetMatrix3(int[,] array, int i, int j, int size)
{
    int[,] block = new int[size, size];

    for (int bi = 0; bi < size; bi++)
        for (int bj = 0; bj < size; bj++)
            block[bi, bj] = array[i + bi, j + bj];

    im = i;
    jm = j;
    return block;
}

static void SetMatrix3(int[,] array, int[,] block, int i, int j, int size)
{
    for (int bi = 0; bi < size; bi++)
        for (int bj = 0; bj < size; bj++)
            array[i + bi, j + bj]=block[bi, bj];
}

```

```
static void PrintMatrix(int[,] array, int Size)
{
    for (int i = 0; i < Size; i++)
    {
        for (int j = 0; j < Size; j++)
            Console.Write(array[i, j] + " ");
        Console.WriteLine();
    }
}

static void Main(string[] args)
{
    N = 1000;
    M = 600;

    Massive = new int[N, M];
    Q = new int[N, M];

    Random random = new Random((int)DateTime.Now.Ticks);

    for (int i = 0; i < N; i++)
        for (int j = 0; j < M; j++){
            Massive[i, j] = random.Next(0, 255);
            Q[i, j] = 0;
        }

    var res = GetBlocks(Massive, 3);
```

```

        foreach (var block in res)
        {
            for (int i = 0; i < 3; i++)
                for (int j = 0; j < 3; j++){
                    tempMas[i, j] = block[i, j];
                }
            MetodThread();
        }

        PrintMatrix(Q, 9);

        Console.ReadKey();
    }

    private static void MetodThread()
    {
        int S=0;
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                S += tempMas[i, j] * A[i, j];
            }
        }
        tempMas[1, 1] = S;

        SetMatrix3(Q, tempMas, im, jm, 3);
    }
}

```

```
}
```

Решение с использованием потков

```
using System;
```

```
using System.Collections.Generic;
```

```
using System.Linq;
```

```
using System.Text;
```

```
using System.Threading;
```

```
namespace Задача5
```

```
{
```

```
    class Program
```

```
    {
```

```
        public static int[,] Massive;
```

```
        public static int[,] Q;
```

```
        public static int[,] A={{-1,0,1},{-2,0,2},{1,0,1}};
```

```
        public static int[,] tempMas = { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };
```

```
        public static int N;
```

```
        public static int M;
```

```
        public static int im=0;
```

```
        public static int jm=0;
```

```
        static IEnumerable<int[,]> GetBlocks(int[,] array, int blockSize)
```

```
        {
```

```
            for (int i = 0; i <= array.GetLength(0) - blockSize; i += blockSize)
```

```
            for (int j = 0; j <= array.GetLength(1) - blockSize; j += blockSize)
```

```
                yield return GetMatrix3(array, i, j, blockSize);
```

```
        }
```

```

static int[,] GetMatrix3(int[,] array, int i, int j, int size)
{
    int[,] block = new int[size, size];
    for (int bi = 0; bi < size; bi++)
        for (int bj = 0; bj < size; bj++)
            block[bi, bj] = array[i + bi, j + bj];
    im = i;
    jm = j;
    return block;
}

static void SetMatrix3(int[,] array, int[,] block, int i, int j, int size)
{
    for (int bi = 0; bi < size; bi++)
        for (int bj = 0; bj < size; bj++)
            array[i + bi, j + bj]=block[bi, bj];
}

static void PrintMatrix(int[,] array, int Size)
{
    for (int i = 0; i < Size; i ++)
    {
        for (int j = 0; j < Size; j ++)
            Console.Write(array[i, j] + " ");
        Console.WriteLine();
    }
}

```

```

static void Main(string[] args)
{
    N = 1000;
    M = 600;

    Massive = new int[N, M];
    Q = new int[N, M];

    Random random = new Random((int)DateTime.Now.Ticks);

    for (int i = 0; i < N; i++)
    for (int j = 0; j < M; j++){
        Massive[i, j] = random.Next(0, 255);
        Q[i, j] = 0;
    }

    var res = GetBlocks(Massive, 3);

    int k = 0;
    foreach (var block in res)
    {
        k++;
        for (int i = 0; i < 3; i++)
            for (int j = 0; j < 3; j++){
                tempMas[i, j] = block[i, j];
            }

        if (k % 2 == 0)
        {
            Thread Thread1 = new Thread(new
ParameterizedThreadStart(MetodThread));

```



```

        Thread1.Name = string.Format("поток {0}", 1);

        Thread1.Start();

    }

    else

    {

        Thread Thread2 = new Thread(new
ParameterizedThreadStart(MetodThread));

        Thread2.Name = string.Format("поток {0}", 2);

        Thread2.Start();

    }

}

PrintMatrix(Q, 9);

Console.ReadKey();

}

private static void MetodThread(object obj)
{

    int S=0;

    for (int i = 0; i < 3; i++)

    {

        for (int j = 0; j < 3; j++)

        {

            S += tempMas[i, j] * A[i, j];

        }

    }

    tempMas[1, 1] = S;

    Console.WriteLine(Thread.CurrentThread.Name);

    SetMatrix3(Q, tempMas, im, jm, 3);

```

```

    }

}

}

```

247.

Создание приложения, которое в начале своей работы устанавливает максимальное количество рабочих потоков и потоков ввода-вывода в пуле потоков. Реализовать цикл, состоящий из K итераций, в котором происходил бы вызов метода, запрошенного из пула потоков. Данный метод выводит на экран случайное число от N до M, а также идентификатор потока.

Номер варианта	N	M	Q
2	34	100	140

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System;
using System.Threading;

namespace Задача2
{
    class Program
    {
        private static int Q = 140;

```

```

private static int N = 34;

private static int M = 100;

static void Main(string[] args)
{

    int nWorkerThreads;
    int nCompletionThreads;

    ThreadPool.SetMaxThreads(Q,Q);
    ThreadPool.GetMaxThreads(out nWorkerThreads, out nCompletionThreads);
    Console.WriteLine("Максимальное количество потоков: " + nWorkerThreads
        + "\nПотоков ввода-вывода доступно: " + nCompletionThreads);

    for (int i = 0; i <= Q; i++)
        ThreadPool.QueueUserWorkItem(JobForAThread);

    Thread.Sleep(300);
    Console.ReadLine();
}

static void JobForAThread(object state)
{
    Random random=new Random();
    Console.WriteLine("число = {0}, выполнение внутри потока из пула {1}",
        random.Next(N, M),
        Thread.CurrentThread.ManagedThreadId);
    Thread.Sleep(50);
}
}

```

```
}
```

248.

Создание приложения, реализующего одновременную параллельную работу Q таймеров, каждый из которых с периодичностью раз в $(N \cdot 10)$ мс выводит на экран случайное число от N до M и номер текущего потока.

Номер варианта	N	M	Q
2	34	100	140

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Timers;

using System.Threading;

namespace задача3
{
    class Timer
    {
        private System.Timers.Timer Tim;

        private int Number=0;

        System.Random r;

        int Q = 0;

        int N = 0;
```

```

int M = 0;

public Timer(int num, int n, int m, System.Random R)
{
    Number = num;
    r = R;
    N = n;
    M = m;

    Tim = new System.Timers.Timer();
    Tim.Interval=10*N;
    Tim.Elapsed += tEvent;
    Tim.AutoReset = true;
    Tim.Enabled = true;
}

private void tEvent(Object S, ElapsedEventArgs e)
{
    Console.WriteLine("Таймер №{0} число от {1} до {2} ={3}",
Number.ToString(), N.ToString(), M.ToString(), Convert.ToString(r.Next(N,M)));
}

}

class Program
{
    private static System.Random r;
    private static int Q = 140;
    private static int N = 34;
    private static int M = 100;
}

```

```

static void Main(string[] args)
{
    r = new Random((int)DateTime.Now.Ticks);

    for (int i = 0; i < Q; i++)
    {
        Timer timer = new Timer(i,N,M,r);
    }
    Console.ReadLine();
}
}
}

```

249.

Сгенерировать случайным образом целочисленную матрицу A(x,y) размерами NxM, элементы которой находятся в диапазоне [0; 255]. 0<=x<=N-1 и 0<=y<=M-1

Для каждого пикселя (кроме крайних, где x = 0 или N-1; y = 0 или M-1) выполнить вычисление:

$Q_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 A_{x+i,y+j} * H_{X_{i+1,j+1}}$, где матрица H_x выглядит следующим образом:

$$H_x = \begin{matrix}
& f_{00} & f_{10} & f_{20} & -1 & 0 & +1 \\
f_{01} & f_{11} & f_{21} & = -2 & 0 & +2 \\
f_{02} & f_{12} & f_{22} & -1 & 0 & +1
\end{matrix}$$

Q – результирующая матрица размерами NxM.

Требуется написать две программы. Первая выполняет все операции вычисления последовательно, а вторая использует распараллеливание

Номер варианта	Способ распараллеливания	N	M
2	Создать N потоков переднего плана, где каждый выполняет вычисления для M пикселей	290	500

```

    Решение программа 1
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

namespace Задача4
{

    class Program
    {
        public static int[,] Massive;

        public static int[,] Q;

        public static int[,] A = { { -1, 0, 1 }, { -2, 0, 2 }, { 1, 0, 1 } };

        public static int[,] Block = { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };

        public static int N;

        public static int M;

        public static int im = 0;

        public static int jm = 0;

        static IEnumerable<int[,]> GetBlocks(int[,] array, int blockSize)
        {
            for (int i = 0; i <= array.GetLength(0) - blockSize; i += blockSize)
                for (int j = 0; j <= array.GetLength(1) - blockSize; j += blockSize)
                    yield return GetMatrix3(array, i, j, blockSize);
        }
    }
}

```

```

static int[,] GetMatrix3(int[,] array, int i, int j, int size)
{
    int[,] block = new int[size, size];
    for (int bi = 0; bi < size; bi++)
        for (int bj = 0; bj < size; bj++)
            block[bi, bj] = array[i + bi, j + bj];

    im = i;
    jm = j;
    return block;
}

static void SetMat3x3(int[,] array, int[,] block, int i, int j, int size)
{
    for (int bi = 0; bi < size; bi++)
        for (int bj = 0; bj < size; bj++)
            array[i + bi, j + bj] = block[bi, bj];
}

static void Main(string[] args)
{
    N = 290;
    M = 500;

    Massive = new int[N, M];
    Q = new int[N, M];

    Random random = new Random();

    for (int i = 0; i < N; i++)

```



```

        for (int j = 0; j < M; j++)
        {
            Massive[i, j] = random.Next(0, 255);
            Q[i, j] = 0;
        }

    var bigMatrix = GetBlocks(Massive, 3);

    int k = 0;
    foreach (var block in bigMatrix)
    {

        for (int i = 0; i < 3; i++)
            for (int j = 0; j < 3; j++)
            {
                Block[i, j] = block[i, j];
            }

        MetodThread();

        k++;

    }

    Console.ReadKey();
}

private static void MetodThread()
{

```

```

int S = 0;

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        S += Block[i, j] * A[i, j];
    }
}

Block[1, 1] = S;
SetMat3x3(Q, Block, im, jm, 3);

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
        Console.Write(Block[i, j] + " ");
    Console.WriteLine();
}

}

}
}

```

Программа 2

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;

using System.Text;

using System.Threading;

namespace Задача5
{

    class Program
    {

        public static int[,] Massive;

        public static int[,] Q;

        public static int[,] A={{-1,0,1},{-2,0,2},{1,0,1}};

        public static int[,] Block = { { 0, 0, 0 }, { 0, 0, 0 }, { 0, 0, 0 } };

        public static int N;

        public static int M;

        public static int im=0;

        public static int jm=0;

        static IEnumerable<int[,]> GetBlocks(int[,] array, int blockSize)
        {
            for (int i = 0; i <= array.GetLength(0) - blockSize; i += blockSize)
                for (int j = 0; j <= array.GetLength(1) - blockSize; j += blockSize)
                    yield return GetMatrix3(array, i, j, blockSize);
        }

        static int[,] GetMatrix3(int[,] array, int i, int j, int size)
        {
            int[,] block = new int[size, size];

            for (int bi = 0; bi < size; bi++)
                for (int bj = 0; bj < size; bj++)

```

```

        block[bi, bj] = array[i + bi, j + bj];

        im = i;
        jm = j;
        return block;
    }

    static void SetMat3x3(int[,] array, int[,] block, int i, int j, int size)
    {
        for (int bi = 0; bi < size; bi++)
            for (int bj = 0; bj < size; bj++)
                array[i + bi, j + bj]=block[bi, bj];
    }

    static void Main(string[] args)
    {
        N = 290;
        M = 500;

        Massive = new int[N, M];
        Q = new int[N, M];

        Random random = new Random();

        for (int i = 0; i < N; i++)
            for (int j = 0; j < M; j++){
                Massive[i, j] = random.Next(0, 255);
                Q[i, j] = 0;
            }

        var bigMatrix = GetBlocks(Massive, 3);
    }

```

```

Thread[] Thread =new Thread[N];

int k = 0;
foreach (var block in bigMatrix)
{

    for (int i = 0; i < 3; i++)
        for (int j = 0; j < 3; j++){
            Block[i, j] = block[i, j];
        }

    if (k<N-1)
    {
        Thread[k] = new Thread(new
ParameterizedThreadStart(MetodThread));
        Thread[k].Name = string.Format("поток {0}", k);

        Thread[k].Start();
    }
    else
    {
        Thread[N-1] = new Thread(new
ParameterizedThreadStart(MetodThread));
        Thread[N-1].Name = string.Format("поток {0}", k);
        Thread[N-1].Start();
    }
    k++;
}

Console.ReadKey();

```

```

    }

    private static void MetodThread(object obj)
    {
        int S=0;
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
            {
                S += Block[i, j] * A[i, j];
            }
        }

        Block[1, 1] = S;
        Console.WriteLine(Thread.CurrentThread.Name);
        SetMat3x3(Q, Block, im, jm, 3);
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
                Console.Write(Block[i, j] + " ");
            Console.WriteLine();
        }
    }
}
}

```

--	--

3.4 Экзамен (примеры заданий)

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

№ задания	примеры заданий																																												
250.	<p data-bbox="298 696 1487 801">Создание многопоточного приложения, состоящего из 5 потоков, организующих бесконечный вывод чисел от N до M и временную приостановку потока на 50 мс после каждой итерации.</p> <table border="1" data-bbox="298 875 1163 1886"> <thead> <tr> <th data-bbox="298 875 515 969">Номер варианта</th> <th data-bbox="515 875 730 969">N</th> <th data-bbox="730 875 946 969">M</th> <th data-bbox="946 875 1163 969">Q</th> </tr> </thead> <tbody> <tr> <td data-bbox="298 969 515 1064">0</td> <td data-bbox="515 969 730 1064">1</td> <td data-bbox="730 969 946 1064">10</td> <td data-bbox="946 969 1163 1064">100</td> </tr> <tr> <td data-bbox="298 1064 515 1158">1</td> <td data-bbox="515 1064 730 1158">10</td> <td data-bbox="730 1064 946 1158">26</td> <td data-bbox="946 1064 1163 1158">200</td> </tr> <tr> <td data-bbox="298 1158 515 1252">2</td> <td data-bbox="515 1158 730 1252">34</td> <td data-bbox="730 1158 946 1252">100</td> <td data-bbox="946 1158 1163 1252">140</td> </tr> <tr> <td data-bbox="298 1252 515 1346">3</td> <td data-bbox="515 1252 730 1346">30</td> <td data-bbox="730 1252 946 1346">50</td> <td data-bbox="946 1252 1163 1346">260</td> </tr> <tr> <td data-bbox="298 1346 515 1440">4</td> <td data-bbox="515 1346 730 1440">300</td> <td data-bbox="730 1346 946 1440">803</td> <td data-bbox="946 1346 1163 1440">106</td> </tr> <tr> <td data-bbox="298 1440 515 1534">5</td> <td data-bbox="515 1440 730 1534">120</td> <td data-bbox="730 1440 946 1534">158</td> <td data-bbox="946 1440 1163 1534">300</td> </tr> <tr> <td data-bbox="298 1534 515 1628">6</td> <td data-bbox="515 1534 730 1628">2</td> <td data-bbox="730 1534 946 1628">7</td> <td data-bbox="946 1534 1163 1628">500</td> </tr> <tr> <td data-bbox="298 1628 515 1722">7</td> <td data-bbox="515 1628 730 1722">0</td> <td data-bbox="730 1628 946 1722">1000</td> <td data-bbox="946 1628 1163 1722">190</td> </tr> <tr> <td data-bbox="298 1722 515 1816">8</td> <td data-bbox="515 1722 730 1816">108</td> <td data-bbox="730 1722 946 1816">400</td> <td data-bbox="946 1722 1163 1816">390</td> </tr> <tr> <td data-bbox="298 1816 515 1886">9</td> <td data-bbox="515 1816 730 1886">200</td> <td data-bbox="730 1816 946 1886">208</td> <td data-bbox="946 1816 1163 1886">200</td> </tr> </tbody> </table> <p data-bbox="298 1944 715 1977">Пример решения для 9 варианта</p> <pre data-bbox="298 2002 480 2036">using System;</pre>	Номер варианта	N	M	Q	0	1	10	100	1	10	26	200	2	34	100	140	3	30	50	260	4	300	803	106	5	120	158	300	6	2	7	500	7	0	1000	190	8	108	400	390	9	200	208	200
Номер варианта	N	M	Q																																										
0	1	10	100																																										
1	10	26	200																																										
2	34	100	140																																										
3	30	50	260																																										
4	300	803	106																																										
5	120	158	300																																										
6	2	7	500																																										
7	0	1000	190																																										
8	108	400	390																																										
9	200	208	200																																										

```

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

namespace задача1
{
    class Program
    {
        private static int Fp=0;

        private static int N = 200;

        private static int M = 208;

        static void Main(string[] args)
        {
            Thread[] Threadfirst=new Thread[5];

            for (int i = 0; i < 5; i++){

                Fp = i + 1;

                Threadfirst[i] = new Thread(stream);

                Threadfirst[i].Start();

            }

        }

        static void write(int n)
        {
            for (int i = N; i <= M; i++)
            {
                Console.Write("итерация {0}, поток №
{1}",i.ToString(),n.ToString()+"\n");

```



```

    }

    }

    static void stream()
    {
        while (true)
        {
            write(Fp);
            Thread.Sleep(50);
        }
    }
}

```

ПКв-2 Способность разрабатывать и адаптировать прикладное программное обеспечение

№ задания	примеры заданий				
251.	<p>Создание приложения, которое в начале своей работы устанавливает максимальное количество рабочих потоков и потоков ввода-вывода в пуле потоков. Реализовать цикл, состоящий из K итераций, в котором происходил бы вызов метода, запрошенного из пула потоков. Данный метод выводит на экран случайное число от N до M, а также идентификатор потока.</p> <table border="1" data-bbox="300 1951 1161 2040"> <tr> <td data-bbox="300 1951 515 2040">Номер варианта</td> <td data-bbox="515 1951 730 2040">N</td> <td data-bbox="730 1951 946 2040">M</td> <td data-bbox="946 1951 1161 2040">Q</td> </tr> </table>	Номер варианта	N	M	Q
Номер варианта	N	M	Q		

0	1	10	100
1	10	26	200
2	34	100	140
3	30	50	260
4	300	803	106
5	120	158	300
6	2	7	500
7	0	1000	190
8	108	400	390
9	200	208	200

Пример решения для 9 варианта

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System;
using System.Threading;

namespace Задача2
{

class Program
{

    private static int N = 200;

```

```

private static int M = 208;

static void Main(string[] args)
{
    int Q=200;
    int nWorkerThreads;
    int nCompletionThreads;

    ThreadPool.SetMaxThreads(Q,Q);
    ThreadPool.GetMaxThreads(out nWorkerThreads, out nCompletionThreads);
    Console.WriteLine("Максимальное количество потоков: " + nWorkerThreads
        + "\nПотоков ввода-вывода доступно: " + nCompletionThreads);

    for (int i = 0; i <= Q; i++)
        ThreadPool.QueueUserWorkItem(JobForAThread);

    Thread.Sleep(3000);
    Console.ReadLine();
}

static void JobForAThread(object state)
{
    Random random=new Random();
    Console.WriteLine("число = {0}, выполнение внутри потока из пула {1}",
        random.Next(N, M),
        Thread.CurrentThread.ManagedThreadId);
    Thread.Sleep(50);
}
}
}

```

ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

№ задания	примеры заданий																																												
252.	<p>Создание приложения, реализующего одновременную параллельную работу Q таймеров, каждый из которых с периодичностью раз в (N*10) мс выводит на экран случайное число от N до M и номер текущего потока.</p> <table border="1" data-bbox="300 712 1161 1724"> <thead> <tr> <th>Номер варианта</th> <th>N</th> <th>M</th> <th>Q</th> </tr> </thead> <tbody> <tr><td>0</td><td>1</td><td>10</td><td>100</td></tr> <tr><td>1</td><td>10</td><td>26</td><td>200</td></tr> <tr><td>2</td><td>34</td><td>100</td><td>140</td></tr> <tr><td>3</td><td>30</td><td>50</td><td>260</td></tr> <tr><td>4</td><td>300</td><td>803</td><td>106</td></tr> <tr><td>5</td><td>120</td><td>158</td><td>300</td></tr> <tr><td>6</td><td>2</td><td>7</td><td>500</td></tr> <tr><td>7</td><td>0</td><td>1000</td><td>190</td></tr> <tr><td>8</td><td>108</td><td>400</td><td>390</td></tr> <tr><td>9</td><td>200</td><td>208</td><td>200</td></tr> </tbody> </table> <p>Пример решения для 9 варианта</p> <pre data-bbox="300 1904 758 2056"> using System; using System.Collections.Generic; using System.Linq; </pre>	Номер варианта	N	M	Q	0	1	10	100	1	10	26	200	2	34	100	140	3	30	50	260	4	300	803	106	5	120	158	300	6	2	7	500	7	0	1000	190	8	108	400	390	9	200	208	200
Номер варианта	N	M	Q																																										
0	1	10	100																																										
1	10	26	200																																										
2	34	100	140																																										
3	30	50	260																																										
4	300	803	106																																										
5	120	158	300																																										
6	2	7	500																																										
7	0	1000	190																																										
8	108	400	390																																										
9	200	208	200																																										

```

using System.Text;

using System.Timers;

using System.Threading;

namespace задача3
{
    class MyTimer
    {
        private System.Timers.Timer Tim;

        private int Number=0;

        private int N=0;

        private int M=0;

        System.Random r;

        public MyTimer(int num, int n, int m, System.Random R)
        {
            Number = num;

            r = R;

            N = n;

            M = m;

            Tim = new System.Timers.Timer();

            Tim.Interval=10*N;

            Tim.Elapsed += tEvent;

            Tim.AutoReset = true;

            Tim.Enabled = true;
        }
    }
}

```

```

private void tEvent(Object S, ElapsedEventArgs e)
{
    Console.WriteLine("Таймер №{0} число от {1} до {2} ={3}",
Number.ToString(), N.ToString(), M.ToString(), Convert.ToString(r.Next(N,M)));
}

}

class Program
{
    private static System.Random r;

    static void Main(string[] args)
    {
        r = new Random((int)DateTime.Now.Ticks);
        int Q = 200;
        int N = 200;
        int M = 208;

        for (int i = 0; i < Q; i++)
        {
            MyTimer myTimer = new MyTimer(i,N,M,r);
        }
        Console.ReadLine();
    }
}
}

```

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Процедуры оценивания в ходе изучения дисциплины знаний, умений и навыков, характеризующих этапы формирования компетенций, регламентируются положениями:

- П ВГУИТ 2.4.03 – 2017 Положение о курсовых, экзаменах и зачетах;
- П ВГУИТ 4.1.02 – 2018 Положение о рейтинговой оценке текущей успеваемости.

Оценка по дисциплине выставляется в экзаменационную ведомость по результатам работы в семестре после выполнения всех видов учебной работы, предусмотренных рабочей программой дисциплины и определяется как среднее арифметическое из всех оценок в течение периода изучения дисциплины

Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания для каждого результата обучения по дисциплине

Результаты обучения по этапам формирования компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач					
ЗНАТЬ: методы и алгоритмы параллельного программирования для решения задач в рамках поставленной цели.	Тест (итоговый контроль - экзамен, промежуточное тестирование)	Результат тестирования	85% и более правильных ответов	Отлично	Освоена (повышенный)
			60-85% правильных ответов	Хорошо	Освоена (повышенный)
			50-60% правильных ответов	Удовлетворительно	Освоена (базовый)
			Менее 50% правильных ответов	Неудовлетворительно	Не освоена (недостаточный)

<p>УМЕТЬ: проводить сравнительный анализ последовательных и параллельных программных средств и оценивать их эффективность, принимать конкретные решения для повышения эффективности разработки программных приложений, использовать полученные знания для создания параллельных программ в различных предметных областях исходя из имеющихся ресурсов и ограничений.</p>	<p>Собеседование (опрос на практических занятиях)</p>	<p>Результат собеседования</p>	<p>Обучающийся качественно выполнил задание практической работы. Оформил отчет в соответствии с методическими указаниями. Ответил на контрольные вопросы.</p>	<p>Зачтено</p>	<p>Освоена (повышенный, базовый)</p>
			<p>Обучающийся не выполнил задание практической работы. Не оформил отчет в соответствии с методическими указаниями. Не ответил на контрольные вопросы.</p>	<p>Не зачтено</p>	<p>Не освоена (недостаточный)</p>
<p>Владеть навыками реализации параллельных алгоритмов и их использования для решения прикладных задач;</p>	<p>Кейс-задание (промежуточное тестирование, экзамен)</p>	<p>Результат решения кейс-задания</p>	<p>Студент грамотно разобрался в задании, предложил правильное решение сложившейся задании</p>	<p>Отлично</p>	<p>Освоена (повышенный, базовый)</p>
			<p>Обучающийся разобрался в задании, неверно указал решение сложившейся задании</p>	<p>Хорошо</p>	<p>Освоена (повышенный, базовый)</p>
			<p>Обучающийся разобрался в задании, не указал решение сложившейся задании</p>	<p>Удовлетворительно</p>	<p>Освоена (базовый)</p>
			<p>Обучающийся не разобрался</p>	<p>Неудовлетв</p>	<p>Не освоена</p>

			в задании, не указал решение сложившейся задании	орительно	(недостаточный)
ПКв-2 Способность разрабатывать и адаптировать прикладное программное обеспечение					
ЗНАТЬ: подходы для проведения анализа требований к параллельным программам, архитектурные принципы реализации параллельной обработки в вычислительных машинах	Тест (итоговый контроль - экзамен, промежуточное тестирование)	Результат тестирования	85% и более правильных ответов	Отлично	Освоена (повышенный)
			60-85% правильных ответов	Хорошо	Освоена (повышенный)
			50-60% правильных ответов	Удовлетворительно	Освоена (базовый)
			Менее 50% правильных ответов	Неудовлетворительно	Не освоена (недостаточный)
УМЕТЬ: использовать программные инструменты для написания, отладки, тестирования и запуска параллельных приложений, производить трансформацию параллельных и распределенных приложений с целью повышения эффективности вычислений	Кейс-задание	Результат решения кейс-задания	Студент грамотно разобрался в задании, предложил правильное решение сложившейся задании	Отлично	Освоена (повышенный, базовый)
			Обучающийся разобрался в задании, неверно указал	Хорошо	Освоена (повышенный)

			решение сложившейся задании		й, базовый)
			Обучающийся разобрался в задании, не указал решение сложившейся задании	Удовлетворительно	Освоена (базовый)
			Обучающийся не разобрался в задании, не указал решение сложившейся задании	Неудовлетворительно	Не освоена (недостаточный)
Владеть приемами разработки параллельных программ на языках параллельного программирования для решения задач исходя из действующих правовых норм, имеющихся ресурсов и ограничений, навыками настройки высокопроизводительных вычислительных систем и комплексов	Кейс-задание (промежуточное тестирование, экзамен)	Результат решения кейс-задания	Студент грамотно разобрался в задании, предложил правильное решение сложившейся задании	Отлично	Освоена (повышенный, базовый)
			Обучающийся разобрался в задании, неверно указал решение сложившейся задании	Хорошо	Освоена (повышенный, базовый)
			Обучающийся разобрался в задании, не указал решение сложившейся задании	Удовлетворительно	Освоена (базовый)
			Обучающийся не разобрался в задании, не указал решение сложившейся задании	Неудовлетворительно	Не освоена (недостаточный)
ПКв-9 Способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.					

ЗНАТЬ: основные команды и функции языка программирования	Тест (итоговый контроль - экзамен, промежуточное тестирование)	Результат тестирования	85% и более правильных ответов	Отлично	Освоена (повышенный)
			60-85% правильных ответов	Хорошо	Освоена (повышенный)
			50-60% правильных ответов	Удовлетворительно	Освоена (базовый)
			Менее 50% правильных ответов	Неудовлетворительно	Не освоена (недостаточный)
УМЕТЬ: использовать различные прикладные средства разработки параллельного программного обеспечения	Собеседование (опрос на практических занятиях)		Обучающийся качественно выполнил задание практической работы. Оформил отчет в соответствии с методическими указаниями. Ответил на контрольные вопросы.	Зачтено	Освоена (повышенный, базовый)
			Обучающийся не выполнил задание практической работы. Не оформил отчет в соответствии с методическими указаниями.	Не зачтено	Не освоена (недостаточный)

			Не ответил на контрольные вопросы.		
Владеть прикладным инструментарием для разработки параллельных алгоритмов в соответствии с назначением объекта проектирования	Кейс-задание (промежуточное тестирование, экзамен)	Результат решения кейс-задания	Студент грамотно разобрался в задании, предложил правильное решение сложившейся задании	Отлично	Освоена (повышенный, базовый)
			Обучающийся разобрался в задании, неверно указал решение сложившейся задании	Хорошо	Освоена (повышенный, базовый)
			Обучающийся разобрался в задании, не указал решение сложившейся задании	Удовлетворительно	Освоена (базовый)
			Обучающийся не разобрался в задании, не указал решение сложившейся задании	Неудовлетворительно	Не освоена (недостаточный)

