



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ  
Директор ИЭиАС  
В.Р. Храмшин

10.02.2023 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

***МНОГОПОТОЧНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ JAVA***

Направление подготовки (специальность)  
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы  
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения  
заочная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Вычислительной техники и программирования
Курс	3

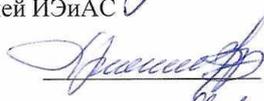
Магнитогорск  
2023 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры Вычислительной техники и программирования  
08.02.2023, протокол № 5  
Зав. кафедрой

  
\_\_\_\_\_ О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС  
10.02.2023 г. протокол № 7  
Председатель

  
\_\_\_\_\_ В.Р. Храпшин

Рабочая программа составлена:  
доцент кафедры ВТиП, канд. техн. наук

  
\_\_\_\_\_ А.Н. Калитаев

Рецензент:  
директор НИИ «Промбезопасность», канд. техн. наук

  
\_\_\_\_\_ М.Ю. Наркевич

## Лист актуализации рабочей программы

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2026 - 2027 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от \_\_\_\_\_ 20\_\_ г. № \_\_\_\_  
Зав. кафедрой \_\_\_\_\_ О.С. Логунова

### **1 Цели освоения дисциплины (модуля)**

Целями освоения дисциплины (модуля) «Многопоточное программирование на языке Java» является ознакомление студентов с концептуальными основами и особенностями реализации технологии многопоточного программирования на языке Java с использованием библиотеки стандартных классов JRE и применение полученных в процессе освоения дисциплины знаний, умений и навыков на практике.

Для достижения поставленной цели в курсе «Многопоточное программирование на языке Java» решаются задачи:

- изучение концепции параллельных вычислений;
- изучение встроенных средств языка Java для организации многопоточных вычислений;
- изучение стандартных классов библиотеки JRE, используемых в многопоточном программировании.

### **2 Место дисциплины (модуля) в структуре образовательной программы**

Дисциплина Многопоточное программирование на языке Java входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Численные методы

Программирование

Прикладная математика

Структуры и модели данных

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Теория вычислительных процессов

Выполнение и защита выпускной квалификационной работы

Проектирование программных средств

### **3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения**

В результате освоения дисциплины (модуля) «Многопоточное программирование на языке Java» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ПК-7	Владеет способами разработки процедур интеграции программных модулей, компонент и верификации выпусков программного продукта, включая базы данных
ПК-7.1	Оценивает выбор программных средств для разработки и верификации интеграционного слоя автоматизированных систем



<p>2.1 Класс Thread и интерфейс Runnable. Жизненный цикл потока. Управление приоритетами и группы потоков. Управление потоками. Потоки–демоны. Потоки и исключения. Атомарные типы и модификатор volatile.</p>		1	1/ИИ		24	<p>1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.</p>	<p>1. Устный опрос (собеседование). 2. Лабораторные работы.</p>	ПК-7.1
<p>2.2 Методы синхронизации потоков. Метод synchronized. Инструкция synchronized. Монитор. Методы wait(), notify() и notifyAll().</p>	3		1/ИИ		24	<p>1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.</p>	<p>1. Устный опрос (собеседование). 2. Лабораторные работы.</p>	ПК-7.1
<p>2.3 Новые способы управления потоками. Перечисление TimeUnit. Блокирующие очереди. Семафоры. Барьеры. "Щеколда". Обмен блокировками. Альтернатива synchronized. ExecutorService и Callable. Phaser.</p>			1/ИИ		24	<p>1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.</p>	<p>1. Устный опрос (собеседование). 2. Лабораторные работы.</p>	ПК-7.1
Итого по разделу		1	3/ИИ		72			
3. Практические аспекты многопоточного программирования и проблемы, возникающие при организации многопоточных вычислений.								

3.1	Архитектура многопоточного приложения. Разработка схемы взаимодействия потоков в многопоточном приложении.		1/ИИ		3	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-7.1
3.2	Отладка, тестирование и мониторинг многопоточных приложений.		1		12	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-7.1
3.3	Блокировки (deadlock), способы их диагностики и устранения.	3	1		3	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-7.1
3.4	Повышение производительности многопоточных вычислений				10,7	1. Самостоятельное изучение учебной и научной литературы. 2. Работа с электронными библиотеками. 3. Подготовка к семинарскому, практическому, лабораторно-практическому занятию.	1. Устный опрос (собеседование). 2. Лабораторные работы.	ПК-7.1
Итого по разделу			3/ИИ		28,7			
Итого за семестр		2	6/4И		124,7		экзамен	
Итого по дисциплине		2	6/4И		124,7		экзамен	

## **5 Образовательные технологии**

1. Традиционные образовательные технологии, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к студенту.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности аспирантов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе личностно-значимого для них образовательного результата.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-пресс-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении программных сред и технических средств работы с знаниями в различных предметных областях.

## **6 Учебно-методическое обеспечение самостоятельной работы обучающихся**

Представлено в приложении 1.

## **7 Оценочные средства для проведения промежуточной аттестации**

Представлены в приложении 2.

## **8 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **а) Основная литература:**

1. Кареева, Е. Д. Основы многопоточного и параллельного программирования: Учебное пособие / Кареева Е.Д. - Краснояр.:СФУ, 2016. - 356 с.: ISBN 978-5-7638-3385-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/966962> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1020593> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

### **б) Дополнительная литература:**

1. Васюткина, И. А. Технология разработки объектно-ориентированных программ на JAVA / Васюткина И.А. - Новосибирск :НГТУ, 2012. - 152 с.: ISBN 978-5-7782-1973-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/557111> (дата обращения: 29.10.2020). – Режим доступа: по подписке.

2. Теория вычислительных процессов. Практикум : практикум / А. Н. Калитаев,

Ю. В. Кочержинская, В. Д. Тугарова, Д. Н. Мазнин ; МГТУ. - Магнитогорск : [МГТУ], 2017. - 83 с. : ил., табл., схемы. - URL: <https://magtu.informsistema.ru/uploader/fileUpload?name=3271.pdf&show=dcatalogues/1/1137340/3271.pdf&view=true> (дата обращения: 23.10.2020). - Макрообъект. - Текст : электронный. - Имеется печатный аналог.

**в) Методические указания:**

**г) Программное обеспечение и Интернет-ресурсы:**

**Программное обеспечение**

Наименование ПО	№ договора	Срок действия лицензии
Oracle Open JDK	свободно распространяемое	бессрочно
NetBeans	свободно распространяемое	бессрочно
JetBrains IDEA Community Edition	свободно распространяемое ПО	бессрочно
Eclipse	свободно распространяемое	бессрочно

**Профессиональные базы данных и информационные справочные системы**

Название курса	Ссылка
Федеральное государственное бюджетное учреждение «Федеральный институт промышленной собственности»	URL: <a href="http://www1.fips.ru/">http://www1.fips.ru/</a>
Информационная система - Единое окно доступа к информационным ресурсам	URL: <a href="http://window.edu.ru/">http://window.edu.ru/</a>
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: <a href="https://elibrary.ru/project_risc.asp">https://elibrary.ru/project_risc.asp</a>

**9 Материально-техническое обеспечение дисциплины (модуля)**

Материально-техническое обеспечение дисциплины включает:

1. Лекционная аудитория ауд. 282. Мультимедийные средства хранения, передачи и представления информации.
2. Компьютерные классы Центра информационных технологий ФГБОУ ВО «МГТУ». Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники.
3. Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки. Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета.
4. Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Ауд. 282 и классы УИТ и АСУ.
5. Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации. Классы УИТ и АСУ.
6. Помещения для хранения и профилактического обслуживания учебного оборудования. Центр информационных технологий – ауд. 372.

**Учебно-методическое обеспечение самостоятельной работы обучающихся**

По дисциплине «Многопоточное программирование на языке Java» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает выполнение работ на лабораторных занятиях. Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала при подготовке к выполнению лабораторных работ и сдаче экзамена по данной дисциплине.

Примерные задания к лабораторным занятиям:

**Работа №1.** Многопоточное программирование на языке Java. Класс Thread и интерфейс Runnable

Написать многопоточную программу для решения следующей задачи: *составить алгоритм и программу для нахождения количества четных и нечетных N-значных чисел, состоящих из цифр, которые попарно являются соседними в натуральном ряду. N задается пользователем (например, N=8, N=10).* И провести сравнительный анализ быстродействия работы алгоритмы при разном количестве потоков k, согласно нижеприведенной таблице.

*Сравнительный анализ результатов*

Компьютер	Количество найденных чисел (четных / нечетных)	Количество потоков k	Время выполнения*, сек
<i>N-значность числа N=8</i>			
Класс Thread		1	
		2	
		4	
		8	
		64	
Интерфейс Runnable		1	
		2	
		4	
		8	
		64	
<i>N-значность числа N=10</i>			
Класс Thread		1	
		2	
		4	
		8	
		64	
Интерфейс Runnable		1	
		2	
		4	
		8	
		64	

**Работа №2.** Многопоточное программирование в решении задач численного интегрирования

Написать многопоточную программу для вычисления приближенного значения интеграла методами прямоугольников, трапеций и парабол (метод Симпсона). Программа запускает  $k$  потоков, которые начинают вычислять значение определенного интеграла по заданной формуле, главный поток ожидает завершения вычислений всеми потоками и затем выводит значение вычислений на экран. Потоки, работая параллельно, используют объекты синхронизации для изменения значения общей переменной.

Проверить результат вычисления с помощью подпрограммы, состоящей из одного цикла.

*Сравнительный анализ результатов*

Метод вычисления значения интеграла функции	Значение и время вычисления интеграла функции $f(x)$ ( $a \leq x \leq b$ )				
	значение	Время расчета при $k$ -потоках, секунд			
		$k = 1$	$k = 2$	$k = 4$	$k = 8$
Метод прямоугольников					
Метод трапеций					
Метод парабол (Симпсона)					

Варианты заданий:

Вариант	Интеграл
1.	$f(x) = \int_1^{3,5} \frac{\ln(x)}{x\sqrt{1+\ln(x)}} dx$
2.	$f(x) = \int_{\pi/6}^{\pi/3} (tg^2(x) + ctg^2(x)) dx$
3.	$f(x) = \int_2^3 \frac{1}{x \cdot \ln(x)} dx$
4.	$f(x) = \int_1^4 \frac{\ln^2(x)}{x} dx$
5.	$f(x) = \int_1^{\ln(2)} \sqrt{e^x - 1} dx$
6.	$f(x) = \int_0^1 x \cdot e^x \cdot \sin(x) dx$
7.	$f(x) = \int_1^4 \frac{dx}{\sqrt{9+x^2}}$
8.	$f(x) = \int_0^{\sqrt{3}} x \cdot arctg(x) dx$
9.	$f(x) = \int_1^e (x \cdot \ln(x))^2 dx$

10.	$f(x) = \int_0^3 arcsin \sqrt{\frac{x}{1+x}} dx$
11.	$f(x) = \int_0^1 \frac{ x^2 - 1 }{(x^2 + 1)\sqrt{x^4 + 1}} dx$
12.	$f(x) = \int_0^{1,5} \frac{e^x \cdot (1 + \sin(x))}{1 + \cos(x)} dx$
13.	$f(x) = \int_0^{3/4} \frac{dx}{(1+x)\sqrt{x^2 + 1}}$

**Работа №3.** Многопоточное программирование на языке Java. Синхронизация доступа к общему ресурсу нескольких потоков

Нередко возникает ситуация, когда несколько потоков имеют доступ к некоторому объекту, проще говоря, пытаются использовать общий ресурс и начинают мешать друг другу. Более того, они могут повредить этот общий ресурс. Например, когда два потока записывают информацию в файл/объект/поток.

Выполнить вариант синхронизации доступа к файлу для одновременной записи информации N различными потоками.

1) **Задание 1:** В результате в файл должно быть выведено (например, для N = 2 – количество потоков, M = 5 – количество сообщений):

```
First0->0
Second0->0
First1->1
Second1->1
First2->2
Second2->2
First3->3
Second3->3
First4->4
Second4->4
```

2) **Задание 2:** В результате в файл должно быть выведено (например, для N = 3 – количество потоков, M = 4 – количество сообщений):

```
A
AA
AAA
AAAA
AAAAB
AAAABV
AAAABVV
AAAABVVV
AAAABVVVC
AAAABVVVCC
AAAABVVVCCC
AAAABVVVCCCC
```

**Работа №4.** Многопоточное программирование на языке Java. Синхронизация доступа к общему ресурсу нескольких потоков с использованием блокирующей очереди (BlockingQueue, BlockingDeque)

Реализации интерфейсов BlockingQueue и BlockingDeque предлагают методы по добавлению/извлечению элементов с задержками. Максимальный размер очереди должен быть задан при ее создании, а именно, все конструкторы класса ArrayBlockingQueue принимают в качестве параметра capacity длину очереди. Пусть объявлена очередь из пяти элементов. Изначально в ней размещены три элемента. В первом потоке производится попытка добавления трех элементов. Два добавятся успешно, а при попытке добавления третьего поток будет остановлен до появления свободного места в очереди. Только когда

второй поток извлечет один элемент и освободит место, первый поток получит возможность добавить свой элемент.

Для изучения принципов работы с блокирующей очередью необходимо решить задачу:

Имеется центр проведения технического осмотра машин с  $N$  стендами для проведения осмотра. На осмотр и выявление дефектов каждой машины затрачивается в среднем  $t$  часов. На осмотр поступает в среднем  $\lambda$  машин в час. В определенный момент времени работает только один из  $N$  стендов проведения осмотра. Нахождение машины уже на стенде, гарантирует проведение технического осмотра.

Определить относительную и абсолютную пропускные способности центра проведения технического осмотра машин при следующих условиях:

1) если машина, прибывшая в центр осмотра, не застает ни одного стенда для проведения осмотра свободным, она покидает центр проведения технического осмотра необслуженной.

2) если машина, прибывшая в центр осмотра, не застает ни одного стенда для проведения осмотра свободным, она покидает центр проведения технического осмотра необслуженной через  $w$  минут ожидания.

Рекомендации к решению задачи:  $N = 3$ ;  $\lambda = 14$  ед. в час.;  $t = 0,4$  часа, ;  $w = 5$  мин.

**Работа №5.** Многопоточное программирование на языке Java. Синхронизация доступа к общему ресурсу нескольких потоков с использованием семафоров (Semaphore)

Для изучения принципов работы с семафорами необходимо решить задачу о пуле ресурсов с ограниченным числом, в данном случае телефонных линий, и значительно бóльшим числом клиентов, желающих воспользоваться одной из линий. Каждый клиент получает доступ к линии, причем пользоваться можно только одной линией. Если все линии заняты, то клиент ждет в течение заданного интервала времени. Если лимит ожидания превышен, генерируется исключение и клиент уходит, так и не воспользовавшись услугами пула.

Пример многоканальной системы массового обслуживания с отказами в обслуживании:

Коммерческая фирма занимается посреднической деятельностью по продаже автомобилей и осуществляет часть переговоров по  $N$  телефонным линиям. В среднем поступает  $\lambda$  звонков в час. Среднее время предварительных переговоров справочного характера составляет  $t$  мин. Среднее время ожидания разговора составляет  $w$  мин.

Определить:

1) относительную и абсолютную пропускные способности линий связи коммерческой фирмы;

2) определить оптимальное число линий связи, достаточное для того, чтобы вероятность отказа не превышала 20%.

Рекомендации к решению задачи:  $N = 3$ ;  $\lambda = 75$  ед. в час.;  $t = 2$  мин.;  $w = 0,5$  мин.

**Работа №6.** Многопоточное программирование на языке Java. Пакет `java.util.concurrent`

Задание:

Разработать многопоточное приложение с использованием возможностей, предоставляемых пакетом `java.util.concurrent`. Не использовать слово `synchronized`. Все сущности, желающие получить доступ к ресурсу, должны быть потоками. Использовать возможности ООП. Не использовать графический интерфейс. Приложение должно быть консольным.

Вариант задания определяется в соответствии с номером компьютера.

Варианты заданий:

1. **Порт.** Корабли заходят в порт для разгрузки/загрузки контейнеров. Число контейнеров, находящихся в текущий момент в порту и на корабле, должно быть неотрицательным и превышающим заданную грузоподъемность судна и вместимость порта. В порту работает несколько причалов. У одного причала может стоять один корабль. Корабль может загружаться у причала, разгружаться или выполнять оба действия.

2. **Маленькая библиотека.** Доступны для чтения несколько книг. Одинаковых книг в библиотеке нет. Некоторые выдаются на руки, некоторые только в читальный зал. Читатель может брать на руки и в читальный зал несколько книг.

3. **Автостоянка.** Доступно несколько машиномест. На одном месте может находиться только один автомобиль. Если все места заняты, то автомобиль не станет ждать больше определенного времени и уедет на другую стоянку.

4. **CallCenter.** В организации работает несколько операторов. Оператор может обслуживать только одного клиента, остальные должны ждать своей очереди. Клиент может положить трубку и перезвонить еще раз через некоторое время.

5. **Автобусные остановки.** На маршруте несколько остановок. На одной остановке может останавливаться несколько автобусов одновременно, но не более заданного числа.

6. **Свободная касса.** В ресторане быстрого обслуживания есть несколько касс. Посетители стоят в очереди в конкретную кассу, но могут перейти в другую очередь при уменьшении или исчезновении там очереди.

7. **Тоннель.** В горах существует два железнодорожных тоннеля, по которым поезда могут двигаться в обоих направлениях. По обоим концам тоннеля собралось много поездов. Обеспечить безопасное прохождение тоннелей в обоих направлениях. Поезд можно перенаправить из одного тоннеля в другой при превышении заданного времени ожидания на проезд.

8. **Банк.** Имеется банк с кассирами, клиентами и их счетами. Клиент может снимать/пополнять/переводить/оплачивать/обменивать денежные средства. Кассир последовательно обслуживает клиентов. Поток-наблюдатель следит, чтобы в кассах всегда были наличные, при скоплении денег более определенной суммы, часть их переводится в хранилище, при истощении запасов наличных происходит пополнение из хранилища.

9. **Аукцион.** На торги выставляется несколько лотов. Участники аукциона делают заявки. Заявку можно корректировать в сторону увеличения несколько раз за торги одного лота. Аукцион определяет победителя и переходит к следующему лоту. Участник, не заплативший за лот в заданный промежуток времени, отстраняется на несколько лотов от торгов.

10. **Биржа.** На торгах брокеры предлагают акции нескольких фирм. На бирже совершаются действия по купле-продаже акций. В зависимости от количества проданных и купленных акций их цена изменяется. Брокеры предлагают к продаже некоторую часть акций. От активности и роста падения котировок акций изменяется индекс биржи. Биржа может приостановить торги при резком падении индекса.

11. **Аэропорт.** Посадка/высадка пассажиров может осуществляться через конечное число терминалов и наземным способом через конечное число трапов. Самолеты бывают разной вместимости и дальности полета. Организовать функционирование аэропорта, если пунктов назначения 4–6, и зон дальности 2–3.

Оценочные средства для проведения промежуточной аттестации

а) Оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
<b>ПК-7: Владеет способами разработки процедур интеграции программных модулей, компонент и верификации выпусков программного продукта, включая базы данных</b>		
ПК-7.1	Оценивает выбор программных средств для разработки и верификации интеграционного слоя автоматизированных систем	<p><i>Перечень теоретических вопросов:</i></p> <ol style="list-style-type: none"> <li>1. Способ создания и запуска потока на основе расширения класса Thread. Пример.</li> <li>1. Способ создания и запуска потока на основе реализации интерфейса Runnable. Пример.</li> <li>2. Пакет java.util.concurrent. Перечисление TimeUnit.</li> <li>3. Объекты синхронизации. Блокирующие очереди. Пример.</li> <li>4. Объекты синхронизации. Семафоры. Пример.</li> <li>5. Объекты синхронизации. Барьеры. Пример.</li> <li>6. Объекты синхронизации. «Щеколда». Пример.</li> <li>7. Объекты синхронизации. Обмен блокировками. Пример.</li> <li>8. Объекты синхронизации. Альтернатива synchronized. Интерфейсы Lock. Пример.</li> <li>9. Объекты синхронизации. Класс ExecutorService и интерфейс Callable. Пример.</li> <li>10. Объекты синхронизации. Класс Phaser. Пример.</li> <li>11. Способы создания и запуска потока.</li> <li>12. Управление потоками. Жизненный цикл потока.</li> <li>13. Управление приоритетами и группами потоков.</li> <li>14. Потоки-демоны.</li> <li>15. Методы и инструкции (блок кода) synchronized.</li> <li>16. Контроль за доступом к объекту-ресурсу (монитор). Методы wait(), notify() и notifyAll().</li> <li>17. Пакет java.util.concurrent. Способы управления потоками.</li> </ol> <p><i>Практические задания:</i></p> <ol style="list-style-type: none"> <li>1. Написать многопоточную программу для вычисления приближенного значения интеграла методом прямоугольников.</li> <li>2. Написать многопоточную программу для вычисления приближенного значения интеграла методом трапеций.</li> <li>3. Написать многопоточную программу для вычисления приближенного значения</li> </ol>



Код индикатора	Индикатор достижения компетенции	Оценочные средства
		<p>a) join()  б) run()  в) sleep()  <b>г) start()</b></p> <p>2. Базовое свойство каких потоков заключается в возможности основного потока приложения завершить его выполнение с окончанием кода метода main(), не обращая внимания на то, что поток еще работает?  а) поток, созданный на основе расширения класса Thread  <b>б) поток-демон</b>  в) поток, созданный на основе реализации интерфейса Runnable</p> <p>3. В каком состоянии находится поток, созданный на основе реализации интерфейса Runnable?  а) BLOCKED  б) RUNNABLE  <b>в) NEW</b>  г) WAITING  д) TIMED_WAITING  е) TERMINATED</p>

**б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине «Многопоточное программирование на языке Java» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 2 теоретических вопроса и одно практическое задание.

**Показатели и критерии оценивания экзамена:**

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.