

Рабочая программа дисциплины

**Алгоритмизация и методы программирования**

<i>Направление подготовки</i>	Информационные системы и технологии
<i>Код</i>	09.03.02
<i>Направленность (профиль)</i>	Проектирование, разработка и сопровождение информационных систем
<i>Квалификация выпускника</i>	бакалавр

## 1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Общепрофессиональные		ОПК-2
Общепрофессиональные		ОПК-6
Профессиональные		ПК-3

## 2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
<b>ОПК-2</b>	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	<p><b>ОПК-2.1.</b> Владеет широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий.</p> <p><b>ОПК-2.2.</b> Применяет на практике методы теоретического и экспериментального исследования для решения практических задач в области информационных систем и технологий</p> <p><b>ОПК-2.3.</b> Понимает основы информатики и принципы работы современных информационных технологий и применяет их для решения задач в профессиональной сфере.</p> <p><b>ОПК-2.4.</b> Выбирает и оценивает способ реализации информационных систем и устройств (программно- или программно-аппаратно-) для решения поставленной задачи.</p> <p><b>ОПК-2.5.</b> Использует принцип работы современных информационных технологий и применяет их для решения задач профессиональной деятельности</p>
<b>ОПК-6</b>	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.	<p><b>ОПК-6.1</b> Разрабатывает и реализует алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования для решения поставленной задачи.</p> <p><b>ОПК-6.2</b> Способен написать код на языке программирования или использовать прикладную программу моделирования для решения поставленной задачи</p> <p><b>ОПК-6.3</b> Применяет алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления.</p> <p><b>ОПК-6.4</b> Записывает простейшие алгоритмы на алгоритмическом языке программирования</p>

		<p>высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования.</p> <p><b>ОПК-6.5</b> Создает простейшие приложения для операционной системы Windows, иллюстрирующие технологию визуального программирования; простейшие программы в технологии объектно-ориентированного программирования.</p>
<b>ПК-3</b>	<p>Способен использовать математический аппарат, методологию программирования и современные компьютерные технологии для решения практических задач получения, хранения, обработки и передачи информации.</p>	<p><b>ПК-3.1.</b> Подбирает парадигму программирования под решение конкретной прикладной задачи;</p> <p><b>ПК-3.2.</b> Модифицирует стандартные алгоритмы обработки информации для оптимизации решения прикладных задач.</p> <p><b>ПК-3.3.</b> Комбинирует известные алгоритмы решения задач.</p> <p><b>ПК-3.4.</b> Реализует аналитические и технологические решения в области программного обеспечения и компьютерной обработки информации;</p> <p><b>ПК-3.5.</b> Применяет знания теоретической информатики, фундаментальной и прикладной математики для анализа и синтеза информационных систем и процессов.</p> <p><b>ПК-3.6.</b> Разработка и верификация кода ИС и баз данных ИС на основе архитектуры ИС</p>

### 3. Описание планируемых результатов обучения по дисциплине

#### 3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
<b>Код компетенции</b>	<b>ОПК-2</b>		
	<p>Знать ключевые концепции и технологии, лежащие в основе современных информационных систем, включая архитектуру клиент-сервер, облачные вычисления и базы данных</p> <p>знать различные типы программных средств (операционные системы, СУБД, языки программирования,</p>	<p>Уметь использовать актуальные информационные технологии и программные средства для решения профессиональных задач, таких как автоматизация процессов или анализ данных.</p> <p>Уметь внедрять и адаптировать отечественные программные средства в рамках профессиональной</p>	<p>Владение навыками самостоятельного изучения новых технологий и программных средств позволяет специалисту оставаться актуальным в быстро меняющемся мире информационных технологий.</p>

	инструменты разработки) и их функциональные возможности, включая отечественные решения.	деятельности, учитывая особенности и требования конкретной организации.	Коммуникационными навыками для взаимодействия с коллегами
<b>Код компетенции</b>	<b>ОПК-6</b>		
	- основы алгоритмизации и методы программирования;	- применять методы математического, статистического и имитационного моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий;	- применять методы математического, статистического и имитационного моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий.
<b>Код компетенции</b>	<b>ПК-3</b>		
	Основы математического аппарата: Знать основные математические концепции, используемые в оптимизации, такие как линейная алгебра, математический анализ и теория вероятностей. Знать типы задач оптимизации (линейные, нелинейные, целочисленные, стохастические) и их характеристики. Методология программирования: Знать алгоритмы и методы оптимизации, такие как градиентный спуск, метод Ньютона и	Применять математические методы: Уметь использовать математические модели для формулирования и решения задач оптимизации. Уметь анализировать и интерпретировать результаты оптимизации с использованием статистических методов. Разрабатывать алгоритмы оптимизации: Уметь разрабатывать и реализовывать алгоритмы оптимизации для решения практических задач. Уметь применять различные методы оптимизации в зависимости от типа и характеристик задачи. Работать с данными:	Владеть навыками решения задач оптимизации в различных областях, таких как экономика, логистика, инженерия и наука. Владеть навыками применения алгоритмов оптимизации для повышения эффективности процессов. Работой с инструментами разработки: Владеть умением использовать современные инструменты разработки и

	<p>генетические алгоритмы. Знать основные структуры данных и их применение в задачах оптимизации. Современные компьютерные технологии: Знать языки программирования, используемые для реализации алгоритмов оптимизации (например, Python, R, MATLAB). Знать инструменты и библиотеки для решения задач оптимизации (например, SciPy, NumPy, PuLP). Принципы работы с данными: Знать методы получения, хранения и обработки данных, необходимые для решения задач оптимизации. Знать основы работы с базами данных и системами управления данными.</p>	<p>Уметь собирать, обрабатывать и анализировать данные, необходимые для решения задач оптимизации. Уметь использовать базы данных для хранения и извлечения данных. Использовать программные инструменты: Уметь использовать программные инструменты и библиотеки для реализации алгоритмов оптимизации. Уметь проводить тестирование и отладку программного обеспечения, связанного с оптимизацией.</p>	<p>среды программирования для реализации алгоритмов оптимизации. Владеть навыками работы с системами контроля версий (например, Git) для управления проектами оптимизации.</p>
--	--	--	--

#### 4. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Алгоритмизация и методы программирования» относится к обязательной части учебного плана ОПОП.

Данная дисциплина взаимосвязана с другими дисциплинами, такими как «Математический анализ», «Линейная алгебра», «Математическая логика и дискретная математика».

В рамках освоения программы бакалавриата выпускники готовятся к решению задач профессиональной деятельности следующих типов: научно-исследовательский, производственно-технологический, организационно-управленческий, проектный.

Профиль (направленность) программы установлена путем ее ориентации на сферу профессиональной деятельности выпускников: проектирование, разработка и сопровождение информационных систем.

## 5. Объем дисциплины

Виды учебной работы	Формы обучения
	Очная
<b>Общая трудоемкость:</b> зачетные единицы/часы	7/252
<b>Контактная работа:</b>	
Занятия лекционного типа	58
Занятия семинарского типа	58
Промежуточная аттестация: зачет с оценкой	0,3
<b>Самостоятельная работа (СРС)</b>	135,7

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

### 6.1. Распределение часов по разделам/темам и видам работы

#### 6.1.1. Очная форма обучения

№ п/п	Раздел/тема	Виды учебной работы (в часах)						Самостоятельная работа
		Контактная работа						
		Занятия лекционного типа		Занятия семинарского типа				
		Лекции	Иные учебные занятия	Практические занятия	Семинары	Лабораторные работы	Иные	
1.	Понятие алгоритма и его способы записи. Базовые алгоритмические структуры.	6			6			13
2.	Инструменты программирования.	6			6			13
3.	Понятия алгоритмического программирования.	6			6			13
4.	Принципы структурного программирования.	6			6			13
5.	Программное средство как продукт технологии программирования	6			6			14
6.	Основные методологии программирования.	6			6			14
7.	Технологии создания программных систем	6			6			14
8.	Стандарты и методики, используемые при разработке	6			6			14

	программных средств							
9.	Понятия визуального программирования.	6			6			14
10.	Объектно-ориентированное программирование.	4			4			13,7
	<b>Промежуточная аттестация</b>	<b>0,3</b>						
	<b>Итого</b>	<b>58</b>			<b>58</b>			<b>135,7</b>

## **6.2 Программа дисциплины, структурированная по темам / разделам**

### **6.2.1 Содержание лекционного курса**

<b>№ п/п</b>	<b>Наименование темы (раздела) дисциплины</b>	<b>Содержание лекционного занятия</b>
1.	Понятие алгоритма и его способы записи. Базовые алгоритмические структуры.	Понятие алгоритма и его способы записи. Базовые алгоритмические структуры и принцип структурной алгоритмизации. Примеры классических алгоритмов
2.	Инструменты программирования.	Эволюция программирования как деятельности. Проектирование и внедрение программ. Инструменты программирования.
3.	Понятия алгоритмического программирования.	Понятия алгоритмического программирования. Сущность данных, их классификации для целей программирования. Базовые (простые) и сложные (структурированные) типы данных. Поясняются понятия переменной и константы, их назначение.
4.	Принципы структурного программирования.	Принципы структурного программирования. Понятия «процедуры» и «функции» как разновидности подпрограмм. Способ взаимодействия подпрограмм посредством передачи параметров при решении общей задачи. Способы описания процедур и функций в теле основной программы.
5.	Программное средство как продукт технологии программирования	Основные понятия и термины. Программа. Характеристика программ. Понятие программного обеспечения. Виды программного обеспечения. Программный продукт. Характерные особенности программного продукта. Программный комплекс. Программное средство. Определение требований к программному средству. Программная система. Сложность и сложные системы. Источники сложности. Признаки работоспособной сложной системы. Классификация программных систем по сложности. Проблемы проектирования сложных программных средств.
6.	Основные методологии программирования.	Понятие методологии. Основные методологии программирования. Атрибуты методологий. Ядра методологий: императивное программирование (процедурное), объектно-ориентированное программирование, функциональное

		программирование, логическое программирование.
7.	Технологии создания программных систем	Общая система понятий технологии программирования. Технология. Процесс. Стадия. Технологический подход. Технология создания (конструирования) ПО. Методы, средства, процедуры технологии конструирования ПО. Требования, предъявляемые к технологиям. Критерии оценки технологий. Выбор технологий.
8.	Стандарты и методики, используемые при разработке программных средств	Виды стандартов. Методики проектирования. Стандартизация жизненного цикла программного средства в системе государственных стандартов и стандартов ISO. Стандарт пользовательского интерфейса. Стандарт проектирования. Стандарт оформления документации. Стандартизация процесса разработки программ и программной документации.
9.	Понятия визуального программирования.	Понятия визуального программирования. Событийное программирование. Конструирование экранных форм, определение значений свойств элементов управления экранной формы, создание программного кода как совокупность методов обработки событий.
10.	Объектно-ориентированное программирование.	Объектно-ориентированное программирование. Базовые понятия объекта и класса, их интерпретация как структурированного типа данных. Сравнение с другими структурированными типами данных.

### 6.2.2 Содержание практических занятий

№ п/п	Наименование темы (раздела) дисциплины	Содержание практического занятия
1.	Понятие алгоритма и его способы записи. Базовые алгоритмические структуры.	Разработка и запись стандартными средствами алгоритмов обработки базовых типов данных (практическая работа).
2.	Инструменты программирования.	Запись разработанных алгоритмов на алгоритмическом языке высокого уровня.
3.	Понятия алгоритмического программирования.	Исследование механизма вызова подпрограмм из основной программы; описание и использование процедур и функций как разновидностей подпрограмм.
4.	Принципы структурного программирования.	Понятия «процедуры» и «функции» как разновидности подпрограмм. Способы описания процедур и функций в основной программе.
5.	Программное средство как продукт технологии программирования	Характерные особенности программного продукта. Программный комплекс. Программное средство. Определение требований к программному средству. Признаки работоспособной сложной системы. Проблемы проектирования сложных программных средств.
6.	Основные методологии программирования.	Основные методологии программирования: императивное программирование (процедурное),



		объектно-ориентированное программирование, функциональное программирование, логическое программирование.
7.	Технологии создания программных систем	Технология создания (конструирования) ПО. Методы, средства, процедуры технологии конструирования.
8.	Стандарты и методики, используемые при разработке программных средств	Методики проектирования. Стандартизация процесса разработки программ и программной документации.
9.	Понятия визуального программирования.	Овладение навыками работы в визуальной среде программирования при создании Windows – приложения.
10.	Объектно-ориентированное программирование.	Проверка области действия внутренних и внешних свойств и методов объектов.

### **6.2.3 Содержание самостоятельной работы**

<b>№ п/п</b>	<b>Наименование темы (раздела) дисциплины</b>	<b>Содержание самостоятельной работы</b>
1.	Понятие алгоритма и его способы записи. Базовые алгоритмические структуры.	Примеры классических алгоритмов
2.	Инструменты программирования.	Проектирование и внедрение программ.
3.	Понятия алгоритмического программирования.	Базовые (простые) и сложные (структурированные) типы данных.
4.	Принципы структурного программирования.	Способ взаимодействия подпрограмм посредством передачи параметров при решении общей задачи. Способы описания процедур и функций в теле основной программы.
5.	Программное средство как продукт технологии программирования	Определение требований к программному средству. Программная система. Источники сложности. Признаки работоспособной сложной системы.
6.	Основные методологии программирования.	Ядра методологий: императивное программирование (процедурное), объектно-ориентированное программирование, функциональное программирование, логическое программирование.
7.	Технологии создания программных систем	Требования, предъявляемые к технологиям. Критерии оценки технологий. Выбор технологий.
8.	Стандарты и методики, используемые при разработке программных средств	Стандартизация жизненного цикла программного средства в системе государственных стандартов и стандартов ISO. Стандарт пользовательского интерфейса. Стандарт проектирования. Стандарт оформления документации. Стандартизация процесса разработки программ и программной документации.
9.	Понятия визуального программирования.	Понятия визуального программирования. Событийное программирование.

10.	Объектно-ориентированное программирование.	Базовые понятия объекта и класса, их интерпретация как структурированного типа данных.
-----	--	--

## 7. Текущий контроль по дисциплине (модулю) в рамках учебных занятий

В рамках текущего контроля преподаватель самостоятельно может проводить следующие мероприятия:

№ п/п	Контролируемые разделы (темы)	Наименование оценочного средства
1.	Понятие алгоритма и его способы записи. Базовые алгоритмические структуры.	Опрос, проблемно-аналитическое задание, тестирование.
2.	Инструменты программирования.	Опрос, проблемно-аналитическое задание, тестирование.
3.	Понятия алгоритмического программирования.	Опрос, проблемно-аналитическое задание, тестирование.
4.	Принципы структурного программирования.	Опрос, проблемно-аналитическое задание, тестирование.
5.	Программное средство как продукт технологии программирования	Опрос, проблемно-аналитическое задание, тестирование.
6.	Основные методологии программирования.	Опрос, проблемно-аналитическое задание, тестирование.
7.	Технологии создания программных систем	Опрос, проблемно-аналитическое задание, тестирование.
8.	Стандарты и методики, используемые при разработке программных средств	Опрос, проблемно-аналитическое задание, тестирование.
9.	Понятия визуального программирования.	Опрос, проблемно-аналитическое задание, тестирование.
10.	Объектно-ориентированное программирование.	Опрос, проблемно-аналитическое задание, тестирование.

## 8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

### 8.1. Основная учебная литература

1. Дроботун Н.В. Алгоритмизация и программирование. Язык Python: учебное пособие / Дроботун Н.В., Рудков Е.О., Баев Н.А. — Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020. — 119 с. — ISBN

978-5-7937-1829-5. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/102400.html>

2. Тюльпинова Н.В. Алгоритмизация и программирование: учебное пособие / Тюльпинова Н.В. — Саратов: Вузовское образование, 2019. — 200 с. — ISBN 978-5-4487-0470-3. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/80539.html>

### **8.2. Дополнительная учебная литература:**

1. Дорохова Т.Ю. Алгоритмизация и программирование [Электронный ресурс]: учебное пособие/ Дорохова Т.Ю., Ильина И.Е.— Электрон. текстовые данные.— М.: Ай Пи Ар Медиа, 2022.— 136 с.— Режим доступа: <https://ipr-smart.ru/122425>. — IPR SMART, по паролю. - DOI: <https://doi.org/10.23682/122425>

### **8.3. Периодические издания**

1. Журнал РАН «Информатика и её применения». <http://www.ipiran.ru/journal/issues>
2. Журнал «Программные продукты и системы». <http://swsys.ru>
3. Журнал «Образование и Информатика». <http://infojournal.ru>

## **9. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины (модуля)**

1. Федеральный портал «Российское образование» <http://www.edu.ru/>
2. Электронно-библиотечная система «Научная электронная библиотека eLIBRARY.RU» <https://www.elibrary.ru> /
3. Электронно-библиотечная система ЛАНЬ <https://e.lanbook.com/>
4. Электронно-библиотечная система IPR BOOKS <https://www.iprbookshop.ru/>
5. <https://www.rsl.ru> - Российская Государственная Библиотека (ресурсы открытого доступа)
6. <https://link.springer.com> - Международная реферативная база данных научных изданий Springerlink (ресурсы открытого доступа)
7. <https://zbmath.org> - Международная реферативная база данных научных изданий zbMATH (ресурсы открытого доступа)
8. <https://openedu.ru> - «Национальная платформа открытого образования» (ресурсы открытого доступа)

## **10. Методические указания для обучающихся по освоению дисциплины (модуля)**

Успешное освоение данного курса базируется на рациональном сочетании нескольких видов учебной деятельности – лекций, семинарских занятий, самостоятельной работы. При этом самостоятельную работу следует рассматривать одним из главных звеньев полноценного высшего образования, на которую отводится значительная часть учебного времени.

Самостоятельная работа студентов складывается из следующих составляющих:

1. работа с основной и дополнительной литературой, с материалами интернета и конспектами лекций;
2. внеаудиторная подготовка к контрольным работам, выполнение докладов, рефератов и курсовых работ;
3. выполнение самостоятельных практических работ;
4. подготовка к экзаменам (зачетам) непосредственно перед ними.

Для правильной организации работы необходимо учитывать порядок изучения разделов курса, находящихся в строгой логической последовательности. Поэтому хорошее усвоение одной части дисциплины является предпосылкой для успешного перехода к следующей. Задания, проблемные вопросы, предложенные для изучения дисциплины, в том числе и для

самостоятельного выполнения, носят междисциплинарный характер и базируются, прежде всего, на причинно-следственных связях между компонентами окружающего нас мира. В течение семестра, необходимо подготовить рефераты (проекты) с использованием рекомендуемой основной и дополнительной литературы и сдать рефераты для проверки преподавателю. Важным составляющим в изучении данного курса является решение ситуационных задач и работа над проблемно-аналитическими заданиями, что предполагает знание соответствующей научной терминологии и т.д.

Для лучшего запоминания материала целесообразно использовать индивидуальные особенности и разные виды памяти: зрительную, слуховую, ассоциативную. Успешному запоминанию также способствует приведение ярких свидетельств и наглядных примеров. Учебный материал должен постоянно повторяться и закрепляться.

При выполнении докладов, творческих, информационных, исследовательских проектов особое внимание следует обращать на подбор источников информации и методику работы с ними.

Для успешной сдачи экзамена (зачета) рекомендуется соблюдать следующие правила:

1. Подготовка к экзамену (зачету) должна проводиться систематически, в течение всего семестра.

2. Интенсивная подготовка должна начаться не позднее, чем за месяц до экзамена.

3. Время непосредственно перед экзаменом (зачетом) лучше использовать таким образом, чтобы оставить последний день свободным для повторения курса в целом, для систематизации материала и доработки отдельных вопросов.

На экзамене высокую оценку получают студенты, использующие данные, полученные в процессе выполнения самостоятельных работ, а также использующие собственные выводы на основе изученного материала.

Учитывая значительный объем теоретического материала, студентам рекомендуется регулярное посещение и подробное конспектирование лекций.

#### **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

1. Microsoft Windows Server;

2. Семейство ОС Microsoft Windows;

3. Libre Office свободно распространяемый офисный пакет с открытым исходным кодом;

4. Информационно-справочная система: Система КонсультантПлюс (КонсультантПлюс);

5. Информационно-правовое обеспечение Гарант: Электронный периодический справочник «Система ГАРАНТ» (Система ГАРАНТ);

#### **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

12.1. Учебная аудитория для проведения учебных занятий, предусмотренных программой бакалавриата, оснащенная оборудованием и техническими средствами обучения.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя, колонки, проектор, экран.

Перечень лицензионного программного обеспечения, в том числе отечественного производства: Windows 10, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Yandex Browser, пакет LibreOffice, МТС Линк, Gimp, FreeCAD.

- 1) IDE Visual Studio Community (нагрузка «Разработка классических приложений на C++» с компонентом «Поддержка C++/CLI»; поддержка MFC)
- 2) СУБД MySQL (клиент-серверная)
- 3) Ramus Modelio
- 4) Cisco Packet Tracer (версии 7.x и 8.x)
- 5) Oracle Virtual Box
- 6) Adobe Reader

Подключение к сети «Интернет» и обеспечение доступа в электронную информационно-образовательную среду ММУ.

12.2. Помещение для самостоятельной работы обучающихся.

Специализированная мебель:

Комплект учебной мебели (стол, стул) по количеству обучающихся; комплект мебели для преподавателя; доска (маркерная).

Технические средства обучения:

Компьютер в сборе для преподавателя; компьютеры в сборе для обучающихся; колонки; проектор, экран.

Перечень лицензионного программного обеспечения, в том числе отечественного производства: Windows 10, КонсультантПлюс, Kaspersky Endpoint Security.

Перечень свободно распространяемого программного обеспечения:

Adobe Reader, Yandex Browser, пакет LibreOffice, МТС Линк, Gimp, FreeCAD.

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду ММУ.

### **13. Образовательные технологии, используемые при освоении дисциплины**

Для освоения дисциплины используются как традиционные формы занятий – лекции (типы лекций – установочная, вводная, текущая, заключительная, обзорная; виды лекций – проблемная, визуальная, лекция конференция, лекция консультация); и семинарские (практические) занятия, так и активные и интерактивные формы занятий - деловые и ролевые игры, решение ситуационных задач и разбор конкретных ситуаций.

На учебных занятиях используются технические средства обучения мультимедийной аудитории: компьютер, монитор, колонки, настенный экран, проектор, микрофон, пакет программ Microsoft Office для демонстрации презентаций и медиафайлов, видеопроектор для демонстрации слайдов, видеосюжетов и др. Тестирование обучаемых может осуществляться с использованием компьютерного оборудования университета.

#### **13.1. В освоении учебной дисциплины используются следующие традиционные образовательные технологии:**

- чтение проблемно-информационных лекций с использованием доски и видеоматериалов;
- семинарские занятия для обсуждения, дискуссий и обмена мнениями;
- контрольные опросы;
- консультации;
- самостоятельная работа студентов с учебной литературой и первоисточниками;
- подготовка и обсуждение рефератов (проектов), презентаций (научно-исследовательская работа);
- тестирование по основным темам дисциплины.

#### **13.2. Активные и интерактивные методы и формы обучения**

Из перечня видов: («мозговой штурм», анализ НПА, анализ проблемных ситуаций, анализ конкретных ситуаций, инциденты, имитация коллективной профессиональной деятельности, разыгрывание ролей, творческая работа, связанная с освоением дисциплины, ролевая игра, круглый стол, диспут, беседа, дискуссия, мини-конференция и др.) используются следующие:

- диспут
- анализ проблемных, творческих заданий, ситуационных задач
- ролевая игра;
- круглый стол;
- мини-конференция
- дискуссия
- беседа.

### **13.3. Особенности обучения инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ)**

При организации обучения по дисциплине учитываются особенности организации взаимодействия с инвалидами и лицами с ограниченными возможностями здоровья (далее – инвалиды и лица с ОВЗ) с целью обеспечения их прав. При обучении учитываются особенности их психофизического развития, индивидуальные возможности и при необходимости обеспечивается коррекция нарушений развития и социальная адаптация указанных лиц.

Выбор методов обучения определяется содержанием обучения, уровнем методического и материально-технического обеспечения, особенностями восприятия учебной информации студентов-инвалидов и студентов с ограниченными возможностями здоровья и т.д. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.

При обучении лиц с ограниченными возможностями здоровья электронное обучение и дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Обучающиеся из числа лиц с ограниченными возможностями здоровья обеспечены печатными и электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

**Автономная некоммерческая организация высшего образования  
«МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ УНИВЕРСИТЕТ»**

---

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ  
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ  
ПО ДИСЦИПЛИНЕ**

**Алгоритмизация и методы программирования**

<i>Направление подготовки</i>	Информационные системы и технологии
<i>Код</i>	09.03.02
<i>Направленность (профиль)</i>	Проектирование, разработка и сопровождение информационных систем
<i>Квалификация выпускника</i>	бакалавр

## 1. Перечень кодов компетенций, формируемых дисциплиной в процессе освоения образовательной программы

Группа компетенций	Категория компетенций	Код
Общепрофессиональные		ОПК-2
Общепрофессиональные		ОПК-6
Профессиональные		ПК-3

## 2. Компетенции и индикаторы их достижения

Код компетенции	Формулировка компетенции	Индикаторы достижения компетенции
<b>ОПК-2</b>	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	<p><b>ОПК-2.1.</b> Владеет широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий.</p> <p><b>ОПК-2.2.</b> Применяет на практике методы теоретического и экспериментального исследования для решения практических задач в области информационных систем и технологий</p> <p><b>ОПК-2.3.</b> Понимает основы информатики и принципы работы современных информационных технологий и применяет их для решения задач в профессиональной сфере.</p> <p><b>ОПК-2.4.</b> Выбирает и оценивает способ реализации информационных систем и устройств (программно- или программно-аппаратно-) для решения поставленной задачи.</p> <p><b>ОПК-2.5.</b> Использует принцип работы современных информационных технологий и применяет их для решения задач профессиональной деятельности</p>
<b>ОПК-6</b>	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.	<p><b>ОПК-6.1</b> Разрабатывает и реализует алгоритмы математических моделей на базе языков и пакетов прикладных программ моделирования для решения поставленной задачи.</p> <p><b>ОПК-6.2</b> Способен написать код на языке программирования или использовать прикладную программу моделирования для решения поставленной задачи</p> <p><b>ОПК-6.3</b> Применяет алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления.</p> <p><b>ОПК-6.4</b> Записывает простейшие алгоритмы на алгоритмическом языке программирования</p>



		<p>высокого уровня, редактировать и отлаживать тексты программ в инструментальной среде программирования.</p> <p><b>ОПК-6.5</b> Создает простейшие приложения для операционной системы Windows, иллюстрирующие технологию визуального программирования; простейшие программы в технологии объектно-ориентированного программирования.</p>
<b>ПК-3</b>	<p>Способен использовать математический аппарат, методологию программирования и современные компьютерные технологии для решения практических задач получения, хранения, обработки и передачи информации.</p>	<p><b>ПК-3.1.</b> Подбирает парадигму программирования под решение конкретной прикладной задачи;</p> <p><b>ПК-3.2.</b> Модифицирует стандартные алгоритмы обработки информации для оптимизации решения прикладных задач.</p> <p><b>ПК-3.3.</b> Комбинирует известные алгоритмы решения задач.</p> <p><b>ПК-3.4.</b> Реализует аналитические и технологические решения в области программного обеспечения и компьютерной обработки информации;</p> <p><b>ПК-3.5.</b> Применяет знания теоретической информатики, фундаментальной и прикладной математики для анализа и синтеза информационных систем и процессов.</p> <p><b>ПК-3.6.</b> Разработка и верификация кода ИС и баз данных ИС на основе архитектуры ИС</p>

### 3. Описание планируемых результатов обучения по дисциплине

#### 3.1. Описание планируемых результатов обучения по дисциплине

Планируемые результаты обучения по дисциплине представлены дескрипторами (знания, умения, навыки).

Дескрипторы по дисциплине	Знать	Уметь	Владеть
<b>Код компетенции</b>	<b>ОПК-2</b>		
	<p>Знать ключевые концепции и технологии, лежащие в основе современных информационных систем, включая архитектуру клиент-сервер, облачные вычисления и базы данных</p> <p>знать различные типы программных средств (операционные системы, СУБД, языки программирования,</p>	<p>Уметь использовать актуальные информационные технологии и программные средства для решения профессиональных задач, таких как автоматизация процессов или анализ данных.</p> <p>Уметь внедрять и адаптировать отечественные программные средства в рамках профессиональной деятельности, учитывая</p>	<p>Владение навыками самостоятельного изучения новых технологий и программных средств позволяет специалисту оставаться актуальным в быстро меняющемся мире информационных технологий.</p> <p>Коммуникационн</p>

	инструменты разработки) и их функциональные возможности, включая отечественные решения.	особенности и требования конкретной организации.	ыми навыками для взаимодействия с коллегами
<b>Код компетенции</b>	<b>ОПК-6</b>		
	- основы алгоритмизации и методы программирования;	- применять методы математического, статистического и имитационного моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий;	- применять методы математического, статистического и имитационного моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности информационных систем и технологий.
<b>Код компетенции</b>	<b>ПК-3</b>		
	Основы математического аппарата: Знать основные математические концепции, используемые в оптимизации, такие как линейная алгебра, математический анализ и теория вероятностей. Знать типы задач оптимизации (линейные, нелинейные, целочисленные, стохастические) и их характеристики. Методология программирования: Знать алгоритмы и методы оптимизации, такие как градиентный спуск, метод Ньютона и	Применять математические методы: Уметь использовать математические модели для формулирования и решения задач оптимизации. Уметь анализировать и интерпретировать результаты оптимизации с использованием статистических методов. Разрабатывать алгоритмы оптимизации: Уметь разрабатывать и реализовывать алгоритмы оптимизации для решения практических задач. Уметь применять различные методы оптимизации в зависимости от типа и характеристик задачи. Работать с данными:	Владеть навыками решения задач оптимизации в различных областях, таких как экономика, логистика, инженерия и наука. Владеть навыками применения алгоритмов оптимизации для повышения эффективности процессов. Работой с инструментами разработки: Владеть умением использовать современные инструменты и

	<p>генетические алгоритмы.</p> <p>Знать основные структуры данных и их применение в задачах оптимизации.</p> <p>Современные компьютерные технологии:</p> <p>Знать языки программирования, используемые для реализации алгоритмов оптимизации (например, Python, R, MATLAB).</p> <p>Знать инструменты и библиотеки для решения задач оптимизации (например, SciPy, NumPy, PuLP).</p> <p>Принципы работы с данными:</p> <p>Знать методы получения, хранения и обработки данных, необходимые для решения задач оптимизации.</p> <p>Знать основы работы с базами данных и системами управления данными.</p>	<p>Уметь собирать, обрабатывать и анализировать данные, необходимые для решения задач оптимизации.</p> <p>Уметь использовать базы данных для хранения и извлечения данных.</p> <p>Использовать программные инструменты:</p> <p>Уметь использовать программные инструменты и библиотеки для реализации алгоритмов оптимизации.</p> <p>Уметь проводить тестирование и отладку программного обеспечения, связанного с оптимизацией.</p>	<p>среды программирования для реализации алгоритмов оптимизации.</p> <p>Владеть навыками работы с системами контроля версий (например, Git) для управления проектами оптимизации.</p>
--	---	--	---

### 3.2. Критерии оценки результатов обучения по дисциплине

Шкала оценивания	Индикаторы достижения	Показатели оценивания результатов обучения
<b>ОТЛИЧНО/ЗАЧТЕНО</b>	Знает:	<ul style="list-style-type: none"> <li>- студент глубоко и всесторонне усвоил материал, уверенно, логично, последовательно и грамотно его излагает, опираясь на знания основной и дополнительной литературы,</li> <li>- на основе системных научных знаний делает квалифицированные выводы и обобщения, свободно оперирует категориями и понятиями.</li> </ul>
	Умеет:	<ul style="list-style-type: none"> <li>- студент умеет самостоятельно и правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, используя научные понятия, ссылаясь на нормативную базу.</li> </ul>

	Владеет:	<ul style="list-style-type: none"> <li>- студент владеет рациональными методами (с использованием рациональных методик) решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.;</li> <li>При решении продемонстрировал навыки</li> <li>- выделения главного,</li> <li>- связкой теоретических положений с требованиями руководящих документов,</li> <li>- изложения мыслей в логической последовательности,</li> <li>- самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.</li> </ul>
<b>ХОРОШО/ЗАЧТЕНО</b>	Знает:	<ul style="list-style-type: none"> <li>- студент твердо усвоил материал, достаточно грамотно его излагает, опираясь на знания основной и дополнительной литературы,</li> <li>- затрудняется в формулировании квалифицированных выводов и обобщений, оперирует категориями и понятиями, но не всегда правильно их верифицирует.</li> </ul>
	Умеет:	<ul style="list-style-type: none"> <li>- студент умеет самостоятельно и в основном правильно решать учебно-профессиональные задачи или задания, уверенно, логично, последовательно и аргументировано излагать свое решение, не в полной мере используя научные понятия и ссылки на нормативную базу.</li> </ul>
	Владеет:	<ul style="list-style-type: none"> <li>- студент в целом владеет рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.;</li> <li>При решении смог продемонстрировать достаточность, но не глубинность навыков</li> <li>- выделения главного,</li> <li>- изложения мыслей в логической последовательности.</li> <li>- связки теоретических положений с требованиями руководящих документов,</li> <li>- самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.</li> </ul>
<b>УДОВЛЕТВИТЕЛЬНО/ЗАЧТЕНО</b>	Знает:	<ul style="list-style-type: none"> <li>- студент ориентируется в материале, однако затрудняется в его изложении;</li> <li>- показывает недостаточность знаний основной и дополнительной литературы;</li> <li>- слабо аргументирует научные положения;</li> <li>- практически не способен сформулировать выводы и обобщения;</li> <li>- частично владеет системой понятий.</li> </ul>
	Умеет:	<ul style="list-style-type: none"> <li>- студент в основном умеет решить учебно-профессиональную задачу или задание, но допускает ошибки, слабо аргументирует свое решение, недостаточно использует научные понятия и руководящие документы.</li> </ul>
	Владеет:	<ul style="list-style-type: none"> <li>- студент владеет некоторыми рациональными методами решения сложных профессиональных задач, представленных деловыми играми, кейсами и т.д.;</li> <li>При решении продемонстрировал недостаточность навыков</li> <li>- выделения главного,</li> <li>- изложения мыслей в логической последовательности.</li> <li>- связки теоретических положений с требованиями руководящих</li> </ul>

		документов, - самостоятельного анализа факты, событий, явлений, процессов в их взаимосвязи и диалектическом развитии.
<b>Компетенция не достигнута</b>		
<b>НЕУДОВЛЕТВОРИТЕЛЬНО/ НЕЗАЧТЕНО</b>	Знает:	- студент не усвоил значительной части материала; - не может аргументировать научные положения; - не формулирует квалифицированных выводов и обобщений; - не владеет системой понятий.
	Умеет:	студент не показал умение решать учебно-профессиональную задачу или задание.
	Владеет:	не выполнены требования, предъявляемые к навыкам, оцениваемым “удовлетворительно”.

*При ответе на вопросы в рамках прохождения промежуточной аттестации (зачет/зачет с оценкой/ экзамен) допускается вольная формулировка ответа, по смыслу раскрывающая содержание ответа, указанного в фонде оценочных средств, в качестве верного ответа.*

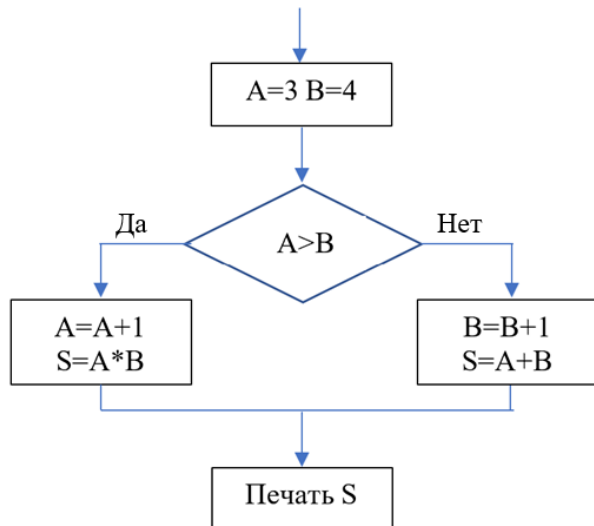
*При подготовке ответа в рамках прохождения промежуточной аттестации (зачет/зачет с оценкой/ экзамен) обучающимся разрешается использовать калькулятор и справочные таблицы.*

**4. Типовые контрольные задания (закрытого, открытого и иного типа) для проведения промежуточной аттестации, необходимые для оценки достижения компетенции, соотнесенной с результатами обучения по дисциплине**

## 1 СЕМЕСТР ОПК-2

1. Укажите значение переменной  $S$ , которое будет получено в результате выполнения

алгоритма, который представлен в виде блок-схемы.

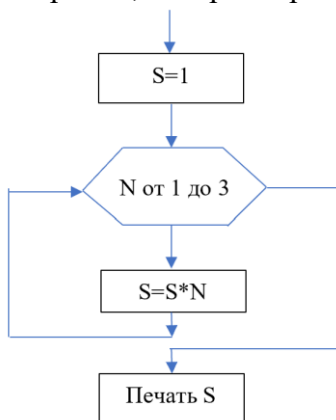


- А) 8
- Б) 15
- В) 7
- Г) 16

**Ответ: А) 8**

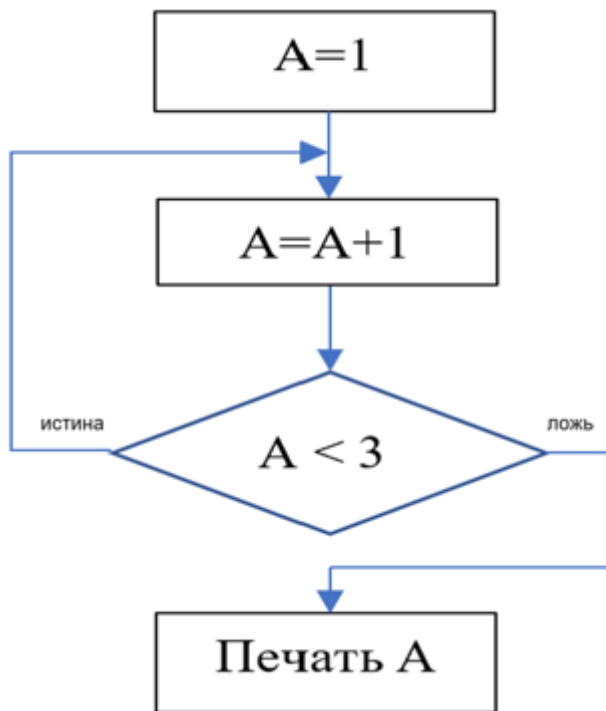
**Обоснование:** по условию будет выполнена правая ветка команд, т.е.  $B$  станет равным 5 и сумма  $A + B$  станет равной 8.

2. Вычислите какое значение переменной  $S$ , которое будет получено в результате выполнения алгоритма, который представлен в виде блок-схемы.



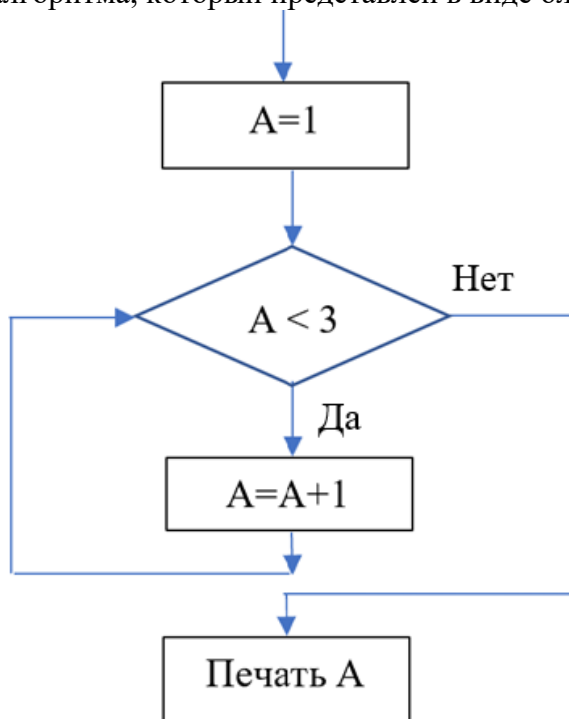
**Ответ: 6**

3. Вычислите значение переменной  $A$ , которое будет получено в результате выполнения алгоритма, который представлен в виде блок-схемы.



**Ответ: 3**

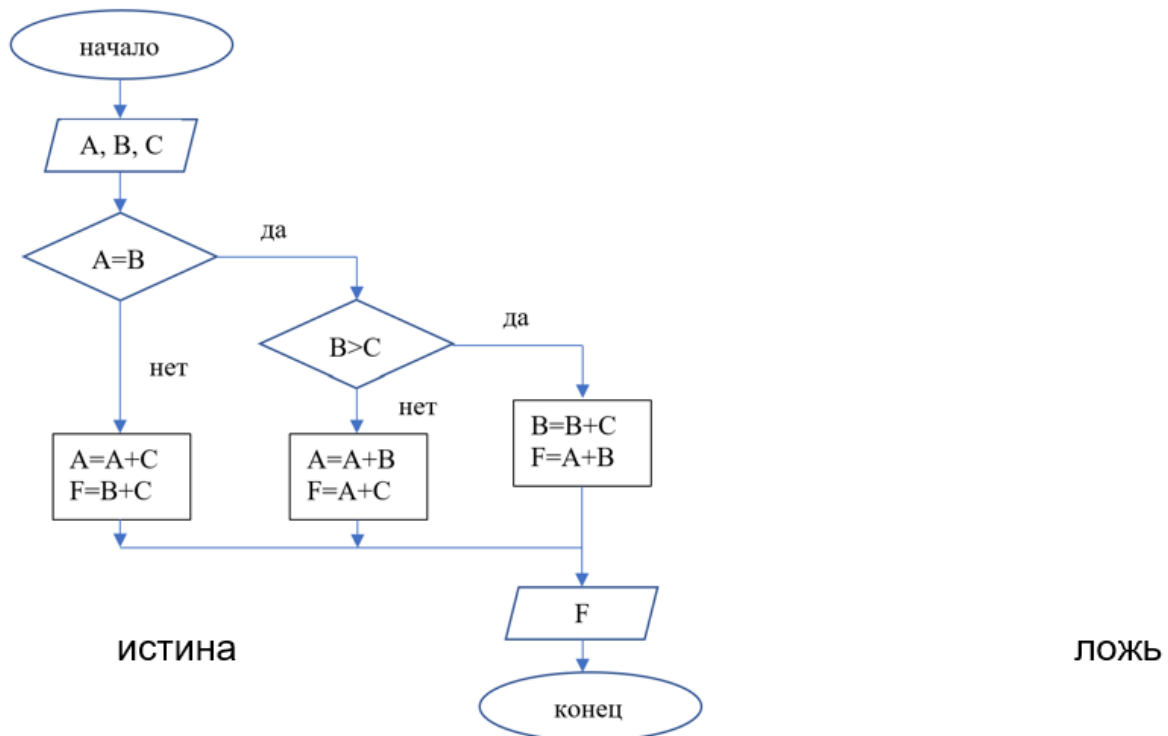
4. Вычислите значение переменной A, которое будет получено в результате выполнения алгоритма, который представлен в виде блок-схемы.



**Ответ: 3**

5. Вычислите по блок-схеме значение переменной F для входных данных A, B, C, равных

значениям 1, 1, 3:



**Ответ: 5**

6. Дайте определение переменной в программе.

**Ответ: Переменная – это именованная ячейка памяти, способная хранить значение заданного типа.**

7. Укажите значение переменной S, которое будет получено после выполнения фрагмента программы на псевдоязыке.

S=1

Для N от 1 до 3 с шагом 1 выполняется действие  $S=S*N$

Печать S

**Ответ: 6**

8. Укажите значение переменной S, которое будет получено после выполнения фрагмента программы на псевдоязыке.

S=1

Для N от 1 до 3 с шагом 1 выполняется действие  $S=S+N$

Печать S

**Ответ: 7**

9. Укажите значение переменной S, какое будет получено после выполнения фрагмента программы на псевдоязыке.

A=2

B=3

Если  $A>B$  то выполняется действие  $S=A*B$  иначе выполняется действие  $S=A+B$

Печать S

**Ответ: 5**

10. Укажите отличие между break и continue. Выберите один вариант ответа.

A) Continue пропускает итерацию тела цикла, break досрочно прерывает работу



**оператора «цикл»**

Б) Нет отличий

В) Break используется в Switch case, а continue в циклах

Г) Continue работает только в циклах, break дополнительно в методах

**Ответ: А) Continue пропускает итерацию тела цикла, break досрочно прерывает работу оператора «цикл»**

11. Сопоставьте логические выражения и их результаты при заданных начальных данных: int a = 1, b = 5; bool some = false;

1	(b > 5 && a <= 3)    (!some    a < 1)	А	true
2	(b != 5    a == 3)    (!some && a > 1)	Б	false
3	(some && a != 2)    b > 5		
4	(b <= 5    a == 3) && !some		

**Ответ: 1А, 2Б, 3Б, 4А**

12. Сопоставьте термины и определения в объектно-ориентированном программировании.

1	Пространство имен	А	способ доступа к внутреннему состоянию объекта, имитирующий переменную некоторого типа
2	Свойство	Б	Последовательность действий при работе с объектом
3	Метод	В	абстрактное хранилище, созданное для логической группировки уникальных идентификаторов.

**Ответ: 1В, 2А, 3Б**

13. Сопоставьте типы сортировки и их описание:

1	Сортировка вставками	А	Максимальное значение меняется местами со значением первого неотсортированного элемента. Этот шаг повторяется до тех пор, пока в массиве не закончится ещё не упорядоченная его часть.
2	Сортировка выбором	Б	Каждый последующий элемент размещается так, чтобы он оказался между ближайшими элементами с меньшим и большим значениями.
3	Сортировка пузырьком	В	Массив разбивается на две примерно равные части, и каждая из них сортируется по отдельности. Затем два отсортированных подмассива сливаются в один.
4	Сортировка слиянием	Г	Последовательно сравниваются значения соседних элементов и меняются местами, если предыдущее оказывается больше последующего.

**Ответ: 1Б, 2А, 3Г, 4В**

14. Укажите основные принципы объектно-ориентированного программирования. Выберите несколько вариантов ответа.

А) Аппликация

- Б) Наследование
- В) Инкапсуляция
- Г) Полиморфизм

Ответ: Б) Наследование В) Инкапсуляция Г) Полиморфизм

15. Что не является составной частью арифметических выражений?

- А) литералы
- Б) функции
- В) машинные команды
- Г) переменные
- Д) знаки операций

Ответ: В) машинные команды

Обоснование: Машинная команда – это результат трансляции арифметического выражения, машинная команда не может быть его частью.

## 1 СЕМЕСТР ОПК-6

1. Задание на соответствие. Проведите соответствие символов блок-схемы и их действий:

1	Эллипс	А	Ввод или вывод
2	Прямоугольник	Б	Начало или конец
3	Ромб	В	Условие
4	Параллелограмм	Г	Обработка данных

Ответ: 1Б, 2Г, 3В, 4А

2. Задание на соответствие. Проведите соответствие типов данных, приведенных на разных языках программирования, и их значений:

1	int, integer	А	Действительные числа
2	float, real	Б	Логические константы
3	bool, Boolean	В	Целые числа

Ответ: 1В, 2А, 3Б

3. Укажите нелинейные базовые конструкции алгоритмов:

- А) Ветвление
- Б) Цикл
- Г) Модульный
- Д) Объектно-ориентированный

Ответ: А) Ветвление Б) Цикл

Обоснование: базовыми алгоритмическими конструкциями являются простое следование (линейная), ветвление и цикл. Модульный – это про способ организации кода в программе, объектно-ориентированный – это методология программирования.

4. Приведите неформальное определение алгоритма.

Ответ: это набор инструкций, описывающих порядок действий исполнителя для решения определённой задачи.

5. Назовите программный инструмент разработки, который переводит написанный на языке высокого уровня текст в набор машинных команд.

**Ответ: Транслятор**

**Возможный вариант ответа: Компилятор**

6. Назовите структурный тип алгоритма, содержащий набор команд, выполняемых последовательно во времени друг за другом.

**Ответ: Линейный**

**Возможный вариант ответа: Простое следование**

7. Выберите тип цикла, при котором тело цикла всегда выполнится хотя бы один раз:

А) С предусловием

**Б) С постусловием.**

В) С параметром.

Г) Условный

**Ответ: Б) С постусловием**

**Обоснование:** В цикле с постусловием сначала выполняется тело цикла, затем проверяется условие продолжения, т.е. тело цикла хотя бы один раз выполнится всегда.

8. Назовите тип алгоритма, предусматривающий многократное повторение одного и того же действия или последовательности действий.

**Ответ: Циклический**

9. Назовите сообщение программного обеспечения либо его части, которое указывает, что произошло.

**Ответ: Событие**

**Возможный вариант ответа: event**

10. Назовите абстрактный тип данных в объектно-ориентированном программировании, задающий общее поведение для группы объектов или представляющий модель объекта.

**Ответ: Класс**

**Возможный вариант ответа: class**

11. Назовите простейший интерфейс ввода-вывода, относящийся к классу символьных устройств, позволяющий вводить данные с клавиатуры и выводить их на экран или в другое устройство отображения.

**Ответ: Консоль**

**Возможный вариант ответа: console**

12. Дайте характеристику переменной, доступ к которой обеспечивается из любой части программы.

**Ответ: Глобальная**

13. Дайте характеристику переменной, доступ к которой обеспечивается только внутри подпрограммы, где она была объявлена.

**Ответ: Локальная**

14. Количество индексов в массиве для доступа к значению определяет ... этого массива. Укажите пропущенную характеристику.

**Ответ: Размерность**

15. Назовите специальный блок инструкций, который вызывается при создании объекта

(экземпляра класса).

**Ответ: Конструктор**

**Возможный вариант ответа: Метод-конструктор**

## 1 СЕМЕСТР

### ПК-3

1. Задание на соответствие. Проведите соответствие типов данных и их возможных значений:

1	int	А	true
2	double	Б	123
3	bool	В	1.23

**Ответ: 1Б, 2В, 3А**

2. Задание на соответствие. Проведите соответствие типов данных и их возможных максимальных значений:

1	short	А	65535
2	byte	Б	32767
3	sbyte	В	127
4	ushort	Г	255

**Ответ: 1Б, 2Г, 3В, 4А**

3. Задание на соответствие. Проведите соответствие типов данных и их возможных минимальных значений:

1	int	А	0
2	uint	Б	-9 223 372 036 854 775 808
3	long	В	-2 147 483 648

**Ответ: 1В, 2А, 3Б**

4. Задание на соответствие. Проведите соответствие типов данных и требуемого для их значений объема памяти:

1	double	А	16 байт
2	float	Б	8 байт
3	decimal	В	4 байта

**Ответ: 1Б, 2В, 3А**

5. Укажите пример неявного преобразования значения к целочисленному типу данных:

А) int.Parse(S)

Б) Convert.ToInt32(S)

**В) int S = 3.4**

Г) (int) S

**Ответ: В) int S = 3.4**

**Обоснование:** Присвоение целочисленной переменной значения с плавающей точкой вынуждает компилятор неявно изменить тип этого значения.

6. Укажите методы преобразования значения переменной N к типу данных с плавающей

точкой в языке программирования C#:

А) Convert.ToFloat(N)

**Б) Convert.ToDouble(N)**

В) (float) N

Г) (double) N

**Ответ: Б) Convert.ToDouble(N) В) (float) N Г) (double) N**

7. Укажите возможные значения, которые может принимать переменная типа «bool» в языке программирования C#:

А) true

**Б) false**

В) 0

Г) 1

Д) null

**Ответ: А) true Б) false**

**Обоснование:** тип bool предназначен для работы с логическими значениями, множество которых ограничено двумя – истина (true), ложь (false).

8. Назовите стандартный тип данных, позволяющий хранить в переменных логические значения.

**Ответ: bool**

9. Назовите стандартный тип данных, позволяющий хранить в переменных действительные числа с ячейкой памяти наименьшего размера.

**Ответ: float**

10. Назовите стандартный тип данных, позволяющий хранить в переменных символьные значения.

**Ответ: char**

11. Укажите ключевое слово, с которого начинается цикл с параметром (счётчиком).

**Ответ: for**

12. Укажите ключевое слово, с которого начинается цикл с предусловием.

**Ответ: while**

13. Укажите ключевое слово, которое ставится перед телом цикла с постусловием в языке программирования C#.

**Ответ: do**

14. Дайте определение термина «идентификатор» в компьютерной программе.

**Ответ: Идентификатор - имя любого объекта языка программирования.**

15. Укажите совокупность символов, с которой начинается необрабатываемый компилятором однострочный комментарий в тексте программы в языке программирования C#.

**Ответ: //**

## **2 СЕМЕСТР**

### **ОПК-2**

1. Назовите прием программирования, который использует одно имя для разных методов.

**Ответ: Перегрузка**

**Возможный вариант ответа: Перегрузка метода**  
**Возможный вариант ответа: Перегрузка методов**

2. Что такое тип данных в языке программирования?

**Ответ: Тип данных – это множество допустимых значений и набор допустимых операций над этими значениями.**

3. Назовите набор правил языка программирования, который позволяет формализовать виды и формы конструкций языка программирования.

**Ответ: Синтаксис**

**Возможный вариант ответа: синтаксические правила**

4. Назовите как называется виртуальная сущность, которая характеризуется определённым состоянием, заданным значениями его полей, и определённым поведением, заданным в его методах?

**Ответ: Объект**

5. Назовите модификатор доступа в языке программирования C#, с помощью которого компонент доступен только в рамках своего класса или структуры.

**Ответ: private**

6. Назовите модификатор доступа в языке программирования C#, с помощью которого компонент доступен из любого места в коде, а также из других программ и сборок.

**Ответ: public**

7. Назовите модификатор доступа в языке программирования C#, с помощью которого компонент класса доступен из любого места в своем классе или в производных классах. При этом производные классы могут располагаться в других сборках.

**Ответ: protected**

8. Назовите модификатор доступа в языке программирования C#, с помощью которого компонент класса или структуры доступен из любого места кода в той же сборке, однако он недоступен для других программ и сборок.

**Ответ: internal**

9. Сопоставьте конструкции языка программирования C# и соответствующие им ключевые слова.

1	Циклы	А	<b>if</b>
2	Обработка исключения	Б	<b>for, while</b>
3	Условия	В	<b>try, catch</b>

**Ответ: 1Б, 2В, 3А**

10. Сопоставьте классы пространства имен System языка программирования C# и их методы.

1	Console	А	<b>Split, Concat</b>
2	String	Б	<b>ToString, FromHexString</b>
3	Convert	В	<b>Beep, Clear</b>

**Ответ: 1В, 2А, 3Б**

11. Сопоставьте типы переменных на языке программирования C# и их инициализации.

1	char c =	А	"3.14f"
2	String s =	Б	3.14f
3	float f =	В	314
4	double d =	Г	3.14
5	int i =	Д	'3'

**Ответ: 1Д, 2А, 3Б, 4Г, 5В**

12. Укажите вариант, где правильно создан массив (язык программирования C#). Выберите один вариант ответа.

А) `int arr[] = {2, 5};`

Б) `int arr = {2, 5};`

**В) `int[] arr = new int [2] {2, 5};`**

Г) `int arr = [2, 5];`

Д) `int[] arr = new Array [2, 5];`

**Ответ: В) `int[] arr = new int [2] {2, 5};`**

13. Укажите вариант, где корректно записана команда присваивания:

А) `X+Y=X`

Б) `F==G`

**В) `X=X+Y`**

Г) `A > D`

**Ответ: В) `X=X+Y`**

14. Укажите варианты, где записаны логические выражения:

А) `X+7`

**Б) `X+7 >= 0`**

В) `X=7`

Г) `X+7 > 0`

**Ответ: Б) `X+7 >= 0` Г) `X+7 > 0`**

15. Укажите назначение использования языка программирования C#. Выберите несколько вариантов ответа.

А) Для написания игр

Б) Для создания веб сайтов

**В) Для создания программ под ПК**

Г) Для создания программ для микроконтроллеров

Д) Всё перечисленное

**Ответ: А) Для написания игр, Б) Для создания веб сайтов, В) Для создания программ под ПК**

## 2 СЕМЕСТР ОПК-6

1. Задание на соответствие. Проведите соответствие между терминами и их определениями, характеризующими алгоритмические языки.

1	Алфавит	А	система правил, по которым образуются правильные конструкции языка
---	---------	---	--

2	Синтаксис	Б	набор используемых символов
3	Семантика	В	система правил, строго определяющая смысл и способ употребления конструкций языка

**Ответ: 1Б, 2А, 3В**

2. Задание на соответствие. Проведите соответствие между этапами компиляции и их определениями.

1	Лексический анализ	А	процесс аналитического разбора входной последовательности символов на распознанные группы
2	Синтаксический анализ	Б	процесс полного соответствующего анализа программы на основании данных в таблице
3	Семантический анализ	В	процесс сопоставления линейной последовательности лексем языка с его формальной грамматикой

**Ответ: 1А, 2В, 3Б**

3. Задание на соответствие. Проведите соответствие свойств алгоритма и определений этих свойств:

1	Понятность	А	завершение алгоритма определёнными результатами
2	Массовость	Б	алгоритм должен включать только те команды, которые доступны исполнителю и входят в его систему команд
3	Результативность	В	алгоритм должен быть применим к разным наборам начальных данных

**Ответ: 1Б, 2В, 3А**

4. Укажите неправильное утверждение о свойствах операции присваивания:

- А) пока переменной не присвоено значение, она остаётся неопределённой
- Б) пока переменной не присвоено значение, она определена минимальным значением своего типа**
- В) значение, присвоенное переменной, сохраняется в ней вплоть до выполнения следующего присваивания этой переменной нового значения
- Г) если переменной присвоить очередное значение, то предыдущее её значение теряется безвозвратно

**Ответ: Б) пока переменной не присвоено значение, она определена минимальным значением своего типа**

**Обоснование:** Пока переменной не присвоено значение, значение переменной до присваивания остаётся неопределённым.

5. Укажите предписания, которые должны содержать в себе алгоритмы

- А) направленные на непосредственное преобразование информации**
- Б) направленные на выбор платформы формирования конечного результата
- В) определяющие области памяти для выполнения действий
- Г) определяющие дальнейшее направление действий**



**Ответ: А) направленные на непосредственное преобразование информации Г) определяющие дальнейшее направление действий**

6. Укажите способы записи алгоритмов:

**А) операторный**

**Б) графический**

В) условный

Г) предикторный

**Ответ: А) операторный, Б) графический**

**Обоснование:** Среди приведённых ответов имеются два способа: операторный, когда программа – это последовательность операторов, и графический – с помощью визуальных средств. Иные варианты ответов не имеют отношения к способам записи алгоритмов.

**Возможный вариант обоснования:** Иные ответы не имеют отношения к записям алгоритмов.

7. Назовите свойство алгоритма, означающее, что он завершает работу и приводит к результату через конечное число шагов.

**Ответ: Конечность**

8. Назовите свойство алгоритма, означающее, что путь решения задачи разделён на отдельные части.

**Ответ: Дискретность**

9. Назовите свойство алгоритма, означающее, что он задан с помощью таких предписаний, которые исполнитель может воспринимать и по которым может выполнять требуемые действия.

**Ответ: Понятность**

10. Назовите свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач.

**Ответ: Массовость**

11. Назовите свойство алгоритма, означающее, что путь решения задачи определен вполне однозначно, а сам алгоритм выдает один и тот же результат для одних и тех же исходных данных.

**Ответ: Определенность**

**Возможный вариант ответа: Детерминированность**

12. Назовите свойство алгоритма, означающее, что он завершается определенными результатами.

**Ответ: Результативность**

13. Назовите способ записи алгоритма, который использует комбинацию естественного языка и элементов языка программирования.

**Ответ: Псевдокод**

14. Назовите построчный способ выполнения исходного кода программы (запроса).

**Ответ: Интерпретация**

15. Назовите оператор, который задает список констант в языке программирования C#.

**Ответ: enum**

**2 СЕМЕСТР  
ПК-3**

1. Задание на соответствие. Проведите соответствие частей программного кода (получите последовательность двух команд) таким образом, чтобы соединенные части приводили значение переменной «а» к одинаковому результату:

1	int a = 0;	A	a += 2;
2	int a = 5;	Б	for (int i=0; i<10; i++) {a++;}
3	int a = 8;	B	a = a + 5;

**Ответ: 1Б, 2В, 3А**

2. Задание на соответствие. Проведите соответствие частей программного кода (получите последовательность двух команд) таким образом, чтобы соединенные части приводили значение переменной «s» к одинаковому результату:

1	string s = "hello";	A	s = s[1] + s[0];
2	string s = "old";	Б	s = s[3] + s[4];
3	string s = "world";	B	s = s[3] + s[1];

**Ответ: 1Б, 2А, 3В**

3. Задание на соответствие. Проведите соответствие частей программного кода (получите последовательность двух команд) таким образом, чтобы соединенные части приводили значение переменной «b» к одинаковому результату:

1	int b = 10;	A	while (b>10) {b--;}
2	int b = 11;	Б	do {b+=10;} while (b>10);
3	int b = 0;	B	for (int i=10; i>0; i-=2) {int c = 1;}

**Ответ: 1В, 2А, 3Б**

4. Укажите операцию, которая не пытается обратиться к области памяти за границами массива int[] m = {1,2,3,4,5} и не вызовет ошибку во время выполнения программы:

- A) int i = 0; m[i] = m[m.Length];
- Б) int i = m.Length; m[i] = int.MaxValue;
- В) int i = m.Length; m[i-1] = 0;**
- Г) int i = m.Length; m[i] = 5;

**Ответ: В) int i = m.Length; m[i-1] = 0;**

**Обоснование:** Максимально допустимый индекс элемента в массиве на 1 меньше длины массива (при нумерации с нуля).

5. Укажите операции, которые пытаются обратиться к области памяти за границами массива int[,] x = {{1,2,3},{4,5,6}} и будут вызывать ошибку во время выполнения программы:

- A) int i = -1; int j = -1; x[i+1,j+1] = 0;
- Б) int i = x.GetLength(0); int j = x.GetLength(1); x[i-1,j-1] = 0;

**В) int i = 2; int j = 2; x[i,j] = 0;**

**Г) int i = x.GetLength(1)-1; int j = x.GetLength(1)-1; x[i,j] = 0;**

**Ответ: В) int i = 2; int j = 2; x[i,j] = 0; Г) int i = x.GetLength(1)-1; int j = x.GetLength(1)-1; x[i,j] = 0;**

**Обоснование:** Индексы массива начинаются с нуля, а последний индекс равен 1, поэтому индекс равный 2 выходит за границы массива.

6. Укажите логические операции, результатом которых будет значение true:

**А) (Math.PI) == (3.14)**

**Б) (float)(Math.PI) > (Math.PI)**

**В) (ushort.MinValue) == (uint.MinValue)**

**Г) (float.MinValue) < (double.MinValue)**

**Ответ: Б) (float)(Math.PI) > (Math.PI) В) (ushort.MinValue) == (uint.MinValue)**

7. Напишите значение переменной «i», которое она получит в результате работы программы:

```
int i=0; i++;
```

**Ответ: 1**

8. Напишите значение переменной «k», которое она получит в результате работы программы:

```
int k=10; for (int i=1; i<10; i++) {k++;}
```

**Ответ: 19**

9. Напишите значение переменной «s», которое она получит в результате работы программы (ответ заключите в кавычки): string s1="3"; string s2="4"; string s=s1+s2;

**Ответ: "34"**

10. Напишите значение переменной «c», которое она получит в результате работы программы:

```
int c1=3; int c2=4; int c=c1+c2;
```

**Ответ: 7**

11. Дайте определение, что такое массив?

**Ответ: Массив – это именованная конечная упорядоченная последовательность однотипных значений в памяти компьютера.**

12. Укажите индекс элемента массива int[] m={-3,-2,-1,0} со значением 0.

**Ответ: 3**

13. Напишите значение переменной «k», которое она получит в результате работы программы:

```
int k = 0; int[] m = { 1,-1,2,-2,3,3 }; foreach(int e in m) {k+=e;}
```

**Ответ: 6**

14. Напишите значение переменной «a», которое она получит в результате работы программы с переменной Q типа «очередь»: var Q = new Queue<int>(); Q.Enqueue(5); Q.Enqueue(6); Q.Enqueue(7); int a = Q.Peek();

**Ответ: 5**

15. Напишите значение переменной «b», которое она получит в результате работы программы с переменной S типа «стек»: var S = new Stack<int>(); S.Push(5); S.Push(6); S.Push(7); int b = S.Peek();

**Ответ: 7**