

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»**
(МИ ВлГУ)

Кафедра *ПИИ*

«УТВЕРЖДАЮ»
Заместитель директора по УР
_____ Д.Е. Андрианов
_____ 17.05.2022

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы алгоритмизации и программирования

Направление подготовки

09.03.04 Программная инженерия

Профиль подготовки

*Методы и средства разработки
программного обеспечения*

Семестр	Трудоем- кость, час./зач. ед.	Лек- ции, час.	Прак- тические занятия, час.	Лабора- торные работы, час.	Консультация, час.	Конт- роль, час.	Всего (контакт- ная работа), час.	СРС, час.	Форма промежу- точного контроля (экз., зач., зач. с оц.)
1	216 / 6	32		48	5,2	0,35	85,55	94,8	Экз.(35,65)
Итого	216 / 6	32		48	5,2	0,35	85,55	94,8	35,65

Муром, 2022 г.

1. Цель освоения дисциплины

Целью дисциплины является изучение и освоение базовых понятий и приемов программирования, применяемых на всех основных этапах разработки программ; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования.

Задачами дисциплины являются формирование целостного представления о предмете, освоение теоретических знаний и практических навыков, позволяющих ориентироваться в области разработки алгоритмов решения задач и написания программных кодов на языке высокого уровня; освоение теоретических основ структурного программирования; изучение конкретных языков программирования; использования языков программирования для решения практических задач.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина базируется на школьном курсе информатики и математики, и является базовой для всех специальных дисциплин направления подготовки.

3. Планируемые результаты обучения по дисциплине

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции	Результаты обучения по дисциплине	
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;	ОПК-6.1 Использует современные языки, утилиты и среды программирования	Знать основные алгоритмические конструкции процедурного программирования (ОПК-6.1) Уметь составлять алгоритмы решения задач (ОПК-6.1) Владеть навыками разработки алгоритмов решения практических задач обработки данных (ОПК-6.1)	тест

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

4.1. Форма обучения: очная

Уровень базового образования: среднее общее.

Срок обучения 4г.

4.1.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Основы алгоритмизации	1	2		4					59	тестирование
2	Основы программирования	1	18		32					19	тестирование
3	Решение практических задач	1	12		12					16,8	тестирование
Всего за семестр		216	32		48			5,2	0,35	94,8	Экз.(35,65)
Итого		216	32		48			5,2	0,35	94,8	35,65

4.1.2. Содержание дисциплины

4.1.2.1. Перечень лекций

Семестр 1

Раздел 1. Основы алгоритмизации

Лекция 1.

Понятие алгоритма (2 часа).

Раздел 2. Основы программирования

Лекция 2.

Введение в язык программирования Си (2 часа).

Лекция 3.

Условный оператор и логическое выражение на языке Си (2 часа).

Лекция 4.

Операторы цикла в языке Си (2 часа).

Лекция 5.

Одномерные массивы (2 часа).

Лекция 6.

Функции (2 часа).

Лекция 7.

Рекурсия (2 часа).

Лекция 8.

Строки в языке Си (2 часа).

Лекция 9.

Двумерные массивы на языке Си (2 часа).

Лекция 10.

Структуры в языке Си (2 часа).

Раздел 3. Решение практических задач**Лекция 11.**

Основные алгоритмы обработки массивов (2 часа).

Лекция 12.

Основные алгоритмы обработки двумерных массивов (2 часа).

Лекция 13.

Обработка ошибок в Си (2 часа).

Лекция 14.

Преобразование данных в языке Си. Оператор "?". Оператор "." (2 часа).

Лекция 15.

Работа с файлами на языке Си (2 часа).

Лекция 16.

Многомодульные программы (2 часа).

4.1.2.2. Перечень практических занятий

Не планируется.

4.1.2.3. Перечень лабораторных работ**Семестр 1****Раздел 1. Основы алгоритмизации****Лабораторная 1.**

Составление алгоритмов решения задачи (4 часа).

Раздел 2. Основы программирования**Лабораторная 2.**

Линейные программы на языке Си (4 часа).

Лабораторная 3.

Условный оператор в языке Си (4 часа).

Лабораторная 4.

Операторы цикла в языке Си (4 часа).

Лабораторная 5.

Написание функций на языке Си (4 часа).

Лабораторная 6.

Случайные числа и метод Монте-Карло (4 часа).

Лабораторная 7.

Работа со структурами на языке Си (4 часа).

Лабораторная 8.

Чтение и запись файлов (4 часа).

Лабораторная 9.

Многомодульные программы (4 часа).

Раздел 3. Решение практических задач**Лабораторная 10.**

Одномерные статические массивы (4 часа).

Лабораторная 11.

Обработка строк в языке Си (4 часа).

Лабораторная 12.

Обработка двумерных массивов на языке Си (4 часа).

4.1.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Языки программирования.
2. Эволюция языков программирования.
3. Классификация языков программирования.
4. Интегрированная среда программирования.
5. Методы программирования: структурный, модульный, объектно-ориентированный.
6. Достоинства и недостатки методов программирования.
7. Общие принципы разработки программного обеспечения.
8. Жизненный цикл программного обеспечения.
9. Типы приложений.
10. Консольные приложения.
11. Файлы.
12. Файловые переменные.
13. Стандартные (текстовые и бестиповые) и типизированные файлы.
14. Технологические цепочки обработки файлов различных типов.
15. Подпрограммы для работы с файлами различных типов.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.1.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Не планируется.

4.1.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

4.2 Форма обучения: заочная

Уровень базового образования: среднее общее.

Срок обучения 5л.

Семестр	Трудоем- кость, час./ зач. ед.	Лек- ции, час.	Практи- ческие занятия, час.	Лабора- торные работы, час.	Консультация, час.	Конт- роль, час.	Всего (контак- тная работа), час.	СРС, час.	Форма промежуточного контроля (экс., зач., зач. с оп.)
1	103 / 2,86	12		12	6	0,5	30,5	68,75	Зач.(3,75)
2	113 / 3,14	12		12	6	0,6	30,6	73,75	Экс.(8,65)
Итого	216 / 6	24		24	12	1,1	61,1	142,5	12,4

4.2.1. Структура дисциплины

№ п\п	Раздел (тема) дисциплины	Семестр	Контактная работа обучающихся с педагогическим работником							Самостоятельная работа	Форма текущего контроля успеваемости (по неделям семестра), форма промежуточной аттестации(по семестрам)
			Лекции	Практические занятия	Лабораторные работы	Контрольные работы	КП / КР	Консультация	Контроль		
1	Основы алгоритмизации	1	2							7	тестирование
2	Основы программирования	1	10		12					61,75	тестирование
Всего за семестр		103	12		12	+		6	0,5	68,75	Зач.(3,75)
3	Основы программирования	2	10		4					7,25	тестирование
4	Решение практических задач	2	2		8					66,5	тестирование
Всего за семестр		113	12		12	+		6	0,6	73,75	Экс.(8,65)
Итого		216	24		24			12	1,1	142,5	12,4

4.2.2. Содержание дисциплины

4.2.2.1. Перечень лекций

Семестр 1

Раздел 1. Основы алгоритмизации

Лекция 1.

Понятие алгоритма. Схемы алгоритмов. Основные алгоритмические структуры (2 часа).

Раздел 3. Основы программирования

Лекция 2.

Введение в язык Си (2 часа).

Лекция 3.

Условный оператор и логические выражения (2 часа).

Лекция 4.

Операторы цикла (2 часа).

Лекция 5.

Преобразование данных (2 часа).

Лекция 6.

Массивы (2 часа).

Семестр 2

Раздел 3. Основы программирования

Лекция 7.

Двумерные массивы (2 часа).

Лекция 8.

Функции (2 часа).

Лекция 9.

Строки (2 часа).

Лекция 10.

Работа с файлами (2 часа).

Лекция 11.

Структуры (2 часа).

Раздел 4. Решение практических задач

Лекция 12.

Основные алгоритмы обработки массивов (2 часа).

4.2.2.2. Перечень практических занятий

Не планируется.

4.2.2.3. Перечень лабораторных работ

Семестр 1

Раздел 1. Основы программирования

Лабораторная 1.

Написание простейшей программы на языках высокого уровня (4 часа).

Лабораторная 2.

Операторы цикла (4 часа).

Лабораторная 3.

Написание простейших функций (4 часа).

Семестр 2

Раздел 2. Основы программирования

Лабораторная 4.

Работа со строками (4 часа).

Раздел 3. Решение практических задач

Лабораторная 5.

Одномерные статические массивы (4 часа).

Лабораторная 6.

Решение задач по обработке массивов (4 часа).

4.2.2.4. Перечень тем и учебно-методическое обеспечение самостоятельной работы

Перечень тем, вынесенных на самостоятельное изучение:

1. Языки программирования.
2. Эволюция языков программирования.
3. Классификация языков программирования.
4. Интегрированная среда программирования.
5. Методы программирования: структурный, модульный, объектно-ориентированный.
6. Достоинства и недостатки методов программирования.
7. Общие принципы разработки программного обеспечения.
8. Жизненный цикл программного обеспечения.
9. Типы приложений.
10. Консольные приложения.
11. Файлы.
12. Файловые переменные.
13. Стандартные (текстовые и бестиповые) и типизированные файлы.
14. Технологические цепочки обработки файлов различных типов.
15. Подпрограммы для работы с файлами различных типов.

Для самостоятельной работы используются методические указания по освоению дисциплины и издания из списка приведенной ниже основной и дополнительной литературы.

4.2.2.5. Перечень тем контрольных работ, рефератов, ТР, РГР, РПР

Примерные задания на контрольную работу:

1. Напишите программу на С, которая находит максимальный и минимальный элемент в массиве.
2. Напишите программу на С для разделения нечетных и четных целых чисел в отдельные массивы.
3. Напишите программу на С для сортировки элементов массива в порядке возрастания.
4. Напишите программу на С для вычитания двух матриц.
5. Напишите программу на С, чтобы найти транспонирование заданной матрицы.
6. Напишите программу на С для вычисления определителя матрицы 3×3 .
7. Напишите программу на С, чтобы найти самый большой подмассив с равным количеством нулей и единиц.
8. Напишите программу на С для проверки четности или нечетности заданного числа с помощью функции.
9. Напишите программу на С, чтобы проверить, является ли число простым или нет, используя функцию.
10. Напишите программу на С, чтобы найти сумму ряда $1!/1+2!/2+3!/3+4!/4+5!/5$ с помощью функции.
11. Напишите программу на С, чтобы проверить, является ли число простым или нет, используя функцию.
12. Напишите программу на С для проверки армстронговских и совершенных чисел с помощью функции.
13. Напишите на С программу, которая считывает с клавиатуры 10 чисел и находит их сумму и среднее значение.
14. Напишите программу на С, которая отображает куб числа до заданного целого числа.
15. Напишите на С программу, которая считывает с клавиатуры 10 чисел и находит их сумму и среднее значение.
16. Напишите программу на С, которая отображает куб числа до заданного целого числа.
17. Напишите программу на С для отображения таблицы умножения заданного целого числа.

18. Напишите программу на С, отображающую таблицу умножения по вертикали от 1 до n.

4.2.2.6. Примерный перечень тем курсовых работ (проектов)

Не планируется.

5. Образовательные технологии

Программой дисциплины предусмотрено чтение лекций, проведение лабораторных занятий. В течение изучения дисциплины студенты изучают на лекционных занятиях теоретический материал. На лабораторных занятиях под руководством преподавателя, решают практические задачи по реализации базовых алгоритмов обработки данных.

Для студентов в качестве самостоятельной работы предполагается подготовка докладов и сообщений, выполнения домашних заданий.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Фонды оценочных материалов (средств) приведены в приложении.

7. Учебно-методическое и информационное обеспечение дисциплины.

7.1. Основная учебно-методическая литература по дисциплине

1. Журавлева, М. Г. Основы программирования. Введение в язык Си. Ч.1 : учебное пособие по курсам «Программирование», «Основы алгоритмизации и программирования» / М. Г. Журавлева, В. А. Алексеев, П. А. Домашнев. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2019. — 99 с. — ISBN 978-5-00175-001-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/101463.html> (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/101463.html>

2. Двойнишников, С. В. Основы программирования (язык С) : учебное пособие / С. В. Двойнишников, К. Ф. Лысаков. — Новосибирск : Новосибирский государственный университет, 2018. — 157 с. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/93475.html> (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей - <https://www.iprbookshop.ru/93475.html>

7.2. Дополнительная учебно-методическая литература по дисциплине

1. Шульга, Т. Э. Основы программирования на языке С : учебное пособие / Т. Э. Шульга. — Саратов : Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2012. — 83 с. — ISBN 978-5-7433-2662-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/76494.html> (дата обращения: 18.10.2021). — Режим доступа: для авторизир. пользователей. — DOI: <https://doi.org/10.23682/76494> - <https://www.iprbookshop.ru/76494.html>

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

В образовательном процессе используются информационные технологии, реализованные на основе информационно-образовательного портала института (www.mivlgu.ru/iop), и инфокоммуникационной сети института:

- предоставление учебно-методических материалов в электронном виде;
- взаимодействие участников образовательного процесса через локальную сеть института и Интернет;

- предоставление сведений о результатах учебной деятельности в электронном личном кабинете обучающегося.

Информационные справочные системы:

- электронная библиотечная системы "IPRBooks" (<http://www.iprbookshop.ru/>);

Программное обеспечение:

LibreOffice (Mozilla Public License v2.0)

Notepad++ (GNU GPL 3)

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

iprbookshop.ru

mivlgu.ru/iop

8. Материально-техническое обеспечение дисциплины

Лаборатория программного обеспечения и сопровождения компьютерных систем

Сервер «Ай Тек» на базе 2 процессоров Intel Xeon; 12 шт. компьютеров Intel Core i5-2400 3,10 GHz; 4гб, DVD-R/ Philips 19'; интерактивная доска SMART Board 480 со встроенным проектором V25; маршрутизатор Gigabit Switch TEG-S16S. Маркерная доска. Доступ к сети Интернет.

9. Методические указания по освоению дисциплины

Для успешного освоения теоретического материала обучающийся: знакомится со списком рекомендуемой основной и дополнительной литературы; уточняет у преподавателя, каким дополнительным пособиям следует отдать предпочтение; ведет конспект лекций и прорабатывает лекционный материал, пользуясь как конспектом, так и учебными пособиями.

До выполнения лабораторных работ обучающийся изучает соответствующий раздел теории. Перед занятием студент знакомится с описанием заданий для выполнения работы, внимательно изучает содержание и порядок проведения лабораторной работы. Лабораторная работа проводится в компьютерном классе. Обучающиеся выполняют индивидуальную задачу компьютерного моделирования в соответствии с заданием на лабораторную работу. Полученные результаты исследований сводятся в отчет и защищаются по традиционной методике в классе на следующем лабораторном занятии. Необходимый теоретический материал, индивидуальное задание, шаги выполнения лабораторной работы и требование к отчету приведены в методических указаниях, размещенных на информационно-образовательном портале института.

Самостоятельная работа оказывает важное влияние на формирование личности будущего специалиста, она планируется обучающимся самостоятельно. Каждый обучающийся самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием дисциплины. Он выполняет внеаудиторную работу и изучение разделов, выносимых на самостоятельную работу, по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Форма заключительного контроля при промежуточной аттестации – экзамен. Для проведения промежуточной аттестации по дисциплине разработаны фонд оценочных средств и балльно-рейтинговая система оценки учебной деятельности студентов. Оценка по дисциплине выставляется в информационной системе и носит интегрированный характер, учитывающий результаты оценивания участия студентов в аудиторных занятиях, качества и своевременности выполнения заданий в ходе изучения дисциплины и промежуточной аттестации.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению *09.03.04 Программная инженерия* и профилю подготовки *Методы и средства разработки программного обеспечения*
Рабочую программу составил *к.т.н., доцент каф. ПИН Привезенцев Д.Г.*_____

Программа рассмотрена и одобрена на заседании кафедры *ПИН* протокол №11 от 05.05.2022 года.
Заведующий кафедрой *ПИН* _____ *Жизняков А.Л.*
(Подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии факультета ИТР протокол №4 от 12.05.2022 года.

Председатель комиссии ФИТР _____ Рыжкова М.Н.
(Подпись)

Лист актуализации рабочей программы дисциплины

Программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ 20__ года.

Заведующий кафедрой _____
(Подпись) _____ (Ф.И.О.)

Программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ 20__ года.

Заведующий кафедрой _____
(Подпись) _____ (Ф.И.О.)

Программа одобрена на _____ учебный год.

Протокол заседания кафедры № _____ от _____ 20__ года.

Заведующий кафедрой _____
(Подпись) _____ (Ф.И.О.)

**Фонд оценочных материалов (средств) по дисциплине
Основы алгоритмизации и программирования**

**1. Оценочные материалы для проведения текущего контроля успеваемости
по дисциплине**

1. Что такое функция?
 - a) Некоторая часть программы, содержащая описание переменных и констант основной программы
 - b) Некоторая часть программы, имеющая собственное имя и которая может вызываться из основной программы
 - c) Некоторая часть программы, содержащая вредоносный код, и блокирует определенные действия системы
 - d) Некоторая часть программы, в которой происходит начальная инициализация всех полей структур, массивов, переменных.
2. Что такое массив?
 - a) Именованный набор переменных имеющих различные типы данных, и располагающихся в одной памяти
 - b) Именованный набор переменных и функций, которые располагаются в одной области памяти
 - c) Именованный набор переменных имеющий один тип данных, и располагающихся в одной области памяти
 - d) Именованный набор переменных имеющих символьный тип данных, и располагающихся в одной области памяти
3. Как написать следующее выражение на языке С «Переменной a присвоено значение b»?
 - a) `a==b`
 - b) `a=b`
 - c) `b=a`
 - d) `a:=b`
4. Как написать следующее выражение «Второму элементу массива Myarray присвоено значение пяти»?
 - a) `int [1] Myarray=«пять»`
 - b) `int Myarray [1] = 5`
 - c) `int Myarray [2] = «пять»`
 - d) `int Myarray [2] = 5`
5. Как написать следующее выражение «Если переменная index больше size то мы инкрементируем переменную count»?
 - a) `if (index>size) { count++; }`
 - b) `if (index<size) { count--; }`
 - c) `if (index>=size) { ++count; }`
 - d) `if (index<size) { --count; }`
6. Какой диапазон значений имеет тип int для 32-разрядных вычислительных систем:
 - a) от 0 до 255
 - b) от -32768 до 32767
 - c) от 0 до 65535
 - d) от 0 до 4 294 967 295

7. Какой размер в байтах имеет переменная вещественного типа float
- 2
 - 4
 - 8
 - 10
8. Дан массив `int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } }`; Чему будет равно значение элемента этого массива `L[1][2]`
- 2
 - 3
 - 4
 - 8
9. Объявление `char *buf`; соответствует
- созданию символьной переменной `buf`
 - созданию строковой переменной `buf`
 - созданию указателя `buf` на символьное значение
 - созданию указателя `buf` на строку
10. Что называется прототипом функции?
- описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров
 - описание функции, включая ее имя, тип возвращаемого значения, типы параметров
 - имя функции и тип возвращаемого значения
 - описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров, тело функции
11. Как обозначается в языке C (C++) следующий режим работы с потоком - создание нового файла для записи и чтения?
- a+
 - wb
 - w+
 - w+b
12. Какая функция, описанная в заголовочном файле читает строку символов из файла?
- `gets()`
 - `fputs()`
 - `fgets()`
 - `fscanf()`
13. Какой размер массива `M` будет после выполнения кода:
`char M[] = "\nGoodlive" ?`
- 10
 - 8
 - 9
 - Не определен
14. В каких случаях необходимо использовать оператор `return` в теле функции?
- Всегда
 - если необходимо, чтобы функция вернула значение
 - если необходимо обеспечить выход из функции в произвольном месте
 - если указан тип возвращаемого значения, в том числе и `void`

15. При открытии файла выполняется следующее действие:
- физический файл связывается с логическим (файловой переменной)
 - устанавливается тип файла (текстовый или бинарный)
 - устанавливается вид (режим) использования файла
 - функцией открытия файла возвращается результат (ошибка)
16. Какое ключевое слово языка C++ используется для описания структурированного типа данных, все элементы которого в памяти начинаются с одного байта?
- struct
 - union
 - enum
 - template
17. Каким способом можно задать многострочный комментарий в языке C++
- /*комментарии к программе*/
 - //комментарии к программе//
 - //комментарии к программе
 - {комментарии к программе}
18. Логическое выражение может возвращать результат типа
- integer
 - boolean
 - char
 - logical
19. Выберите правильный вариант записи на языке C формулы $0 < x < 10$
- $x > 0, x \leq 10$
 - $0 < x \leq 10$
 - $x > 0 \text{ AND } x \leq 10$
 - $(x > 0) \text{ AND } (x < 10)$
20. Укажите правильный вариант записи условного оператора в языке C
- IF $x > 0$ Do $y := \text{sqrt}(x)$
 - IF $y := \text{sqrt}(x)$ then $x > 0$
 - IF $x > 0$ then $y := \text{sqrt}(x)$
 - IF $(x > 0)$ { $y := \text{sqrt}(x)$ }
21. Выберите правильный вариант записи на языке C следующего условия: « x принадлежит диапазону $[0; 10)$ »
- $x \geq 0; x < 10$
 - $0 \leq x < 10$
 - $(x > 0 \text{ AND } (x \leq 10))$
 - $(x \geq 0) \text{ AND } (x < 10)$
22. Укажите группу, содержащую последовательность правильно записанных на языке C знаков операций отношений
- $\sim, >, <, =, ?$
 - $=, <>, ><, >$
 - $=, >=, <=, !=$
 - $\sim, =, >, <, =, <$

23. Тело какого цикла всегда будет выполнено хотя бы один раз, независимо от истинности условия:

- a) While
- b) Do While
- c) For
- d) Нет такого цикла в языке C

24. В результате выполнения кода

```
int i=2;
```

```
switch (i)
```

```
{
```

```
case 1: i += 2;
```

```
case 2: i *= 3;
```

```
case 6: i /= 2;
```

```
default: ;
```

```
}
```

- a) переменная i примет значение 6
- b) переменная i примет значение 3
- c) переменная i примет значение 2
- d) тело оператора switch не поменяет значение переменной i

25. Укажите директиву препроцессора, которую необходимо подключить для организации форматированного ввода-вывода данных:

- a) #include <iostream.h>
- b) #include <stdio.h>
- c) #include <math.h>
- d) #include <conio.h>

Общее распределение баллов текущего контроля по видам учебных работ для студентов

Рейтинг-контроль 1	2 лабораторных работы	10
Рейтинг-контроль 2	4 лабораторных работы	20
Рейтинг-контроль 3	6 лабораторных работ	30
Посещение занятий студентом	-	-
Дополнительные баллы (бонусы)	-	-
Выполнение семестрового плана самостоятельной работы	-	-

2. Промежуточная аттестация по дисциплине

Перечень вопросов к экзамену / зачету / зачету с оценкой.

Перечень практических задач / заданий к экзамену / зачету / зачету с оценкой (при наличии)

ОПК-6.1

1. Почему понижение типа может привести к затруднениям?

- о нарушается баланс скобок в выражении
- причина этого проста: все число целиком может не поместиться в элементе данных низшего типа
- о сложно явно переводить типы к более низкому типу

- o нарушается структура программы
- 2. Какой тип наиболее подойдет для хранения и обработки данных о количестве жителей Москвы?
 - unsigned long
 - o unsigned double
 - o float
- 3. Как обратиться к элементу структуры при помощи указателя?
 - o new_student<id
 - new_student->id
 - o new_student-id
 - o new_student*id
- 4. Какая из операций написана неверно?
 - o >=
 - o <=
 - =>
 - o =
- 5. Как передать в функцию f указатель на структуру healt_record?
 - o обратиться к функции как к записи структуры: healt_record.f
 - o с помощью взятия адреса функции с передачей непосредственно структуры: &f(healt_record)
 - передать указатель на структуру: f(&healt_record)
 - o передать непосредственно структуру (но только с поверхностным копированием): f(healt_record)
- 6. Можно ли написать оператор switch без оператора break?
 - o нет
 - да, если нужно выполнить и последующие альтернативы
 - o да, если нужно выполнить оператор default
 - o да, если после выполнения блока операторов следует выйти из функции
- 7. Как можно представить данные при вычислении с двойной точностью?
 - o описать данные типа float
 - o описать данные типа long
 - описать данные типа double
 - o описать данные типа unsigned double float
- 8. Что называется составным оператором?
 - o составной оператор представляет собой четное число операторов , объединенных с помощью фигурных скобок
 - o составной оператор представляет собой два или более операторов, объединенных с помощью квадратных скобок
 - o составной оператор представляет собой нечетное число операторов , объединенных с помощью фигурных скобок
 - составной оператор представляет собой два или более операторов, объединенных с помощью фигурных скобок
- 9. Какие циклы называются вложенными циклами?
 - o Вложенным называется цикл, тело которого является составным оператором
 - Вложенным называется цикл, находящийся внутри другого цикла
 - o Вложенным называется цикл чаще всего использующийся
 - o Вложенным называется цикл не содержащий ключевых слов
- 10. Что означает функция malloc?
 - функция возвращает указатель на созданный динамический объект
 - o функция возвращает размер созданного динамического объекта
 - o функция уничтожает динамический объект
 - o функция подсчитывает объем памяти для хранения выражения
- 11. Каково назначение потоковой функции fwrite()?
 - записывает блок данных в поток

- о читает блок данных из потока
 - о закрывает поток
 - о записывает форматированные данные в поток
12. Чем определяется мобильность языка Си?
- тем, что программа, написанная на Си для одной вычислительной системы, может быть перенесена с небольшими изменениями или вообще без них на другую
 - о мобильность языка Си определяется его эффективностью
 - о тем, что программа, написанная на Си для одной вычислительной системы, может быть перенесена без изменений на другую систему
13. Какой из приведенных циклов не выполнится ни разу?
- о `unsigned char s = '0'; while(!('9' — s < 0)) printf(«%c\n», s++)`
 - о `for(short i=0; ++i > 0; printf(«%d\n», i))`
 - `signed char c = 255; while (c > 200) { printf(«%d\n», c--); }`
 - о `char c = 255; do { printf(«%d\n», c--); } while (c > 200)`
14. Как получить адрес переменной?
- о использовать операцию #
 - о использовать операцию ?
 - использовать операцию &
 - о использовать операцию *
15. Какое значение будет иметь переменная i после выполнения следующего цикла:
`char i=0; while (i < 255) printf("Current value: %d\n", i++)`
- о 255
 - о 256
 - это бесконечный цикл, значение i не может быть определено
 - о тело цикла не выполнится ни разу, следовательно значение i будет равно нулю
16. Как создать строковую константу?
- о используя апостроф
 - о используя восклицательный знак
 - используя кавычки
 - о используя вопросительный знак
17. На что проверяется условие в цикле `while(ch!=EOF)`?
- на конец файла
 - о на начало файла
 - о на начало строки
 - о на конец строки
18. Где используется default?
- о default используется в операции условия ?:
 - default используется в операторе switch
 - о default используется в операторе if
 - о default используется в операторе while
19. Что понимается под основным циклом?
- о самый маленький цикл программы
 - о самый большой цикл программы
 - самый внешний цикл программы
 - о самый внутренний цикл программы
20. Что является указателем на хранящуюся в памяти строковую константу?
- вся фраза в кавычках является указателем на место в памяти
 - о первая литера строковой константы является указателем на место в памяти
 - о последняя литера строковой константы является указателем на место в памяти
 - о первая четная литера строковой константы является указателем на место в памяти
21. Дано описание `int i`; Верен ли синтаксически фрагмент выражения `(char)i`, и что означает запись?
- о фрагмент выражения записан синтаксически неверно

- о фрагмент выражения записан верно и означает примечание
 - фрагмент выражения записан синтаксически верно. Такая запись означает, что результат вычисления переменной `i` будет приведен к типу `char`
 - о фрагмент записан синтаксически верно, и означает, что переменная целого типа неявно преобразуется к переменной символьного типа
22. Что понимается под вводом-выводом?
- о операция пересылки данных между различными файлами
 - о операция пересылки данных между различными частями оперативной памяти
 - о операция пересылки данных между различными внешними устройствами
 - операция пересылки данных между оперативной памятью и внешними устройствами
23. Если значение превышает наибольшее машинное целое со знаком, то оно представляется
- о как строковое
 - о как символьное
 - о как десятичное с плавающей точкой
 - как длинное целое
24. Зачем применяют отступы от поля в строках?
- о чтобы сделать структуру программы не наглядной
 - о чтобы быстрее прошла компиляция
 - чтобы сделать структуру программы наглядной
 - о чтобы быстрее писать программу
25. Может ли функция входить в структуру?
- о да
 - о да, но только типа `void`
 - о да, но только типа `int`
 - нет
26. Укажите правильный вариант функции, возвращающей сумму двух чисел:
- `int sum(int j,int k)`

```
{
return(j+k);
}
```
 - о `int sum()`

```
{
return(j+k);
int j, k;
}
```
 - о `int sum(int j; int k)`

```
{
return(j+k);
}
```
27. Каков порядок вычисления операндов операции сравнения?
- о результат не зависит от порядка вычисления
 - о справа налево
 - ✓ слева направо
28. Каким циклом является цикл `do`?
- о Циклом со счетчиком
 - Циклом с постусловием
 - о Простым циклом
 - о Циклом с предусловием
29. Каково значение `int x` после выполнения выражения `x=(int)3.8+3.3;?`
- `x=6`
 - о `x=3`
 - о `x=7`

- o x=7.1
- 30. Отметьте верное утверждение:
 - в языке Си функции разделены на категории — функции, возвращающие значения в вызывающую программу и функции, не возвращающие значения в вызывающую программу
 - o только функции с параметрами возвращают значения в вызывающую программу
 - o функции по типу результата определяют, нужно ли вернуть значение в вызывающую программу
- 31. Входят ли функции printf() и scanf() в стандартную библиотеку языка Си?
 - o функция printf() входит в описание языка Си, а функция scanf() не входит
 - o нет
 - o функция scanf() входит в описание языка Си, а функция printf() не входит
 - входят
- 32. Зачем используют оператор return()?
 - o чтобы организовать цикл
 - o чтобы перейти по произвольному адресу в программе
 - чтобы вернуть произвольное значение, указанное в качестве операнда return, в вызывающую функцию
 - o чтобы приостановить работу программы
- 33. Какие данные объединяет структура?
 - o структура объединяет логически связанные данные только неотрицательных целых типов
 - o структура объединяет логически связанные данные только целых типов
 - o структура объединяет логически связанные данные только одинаковых типов
 - структура объединяет логически связанные данные разных типов
- 34. Как можно напечатать данные таблицей?
 - o это обеспечивается анализом и отбором одинаковых по размеру данных
 - o это обеспечивается печатью в столбик
 - o это обеспечивается заданием не фиксированной ширины полей
 - это обеспечивается заданием фиксированной ширины полей
- 35. Для чего используется унарная операция минус?
 - для изменения знака переменной на противоположный знак (+ на -, а - на +)
 - o для получения разности некоторой величины
 - o для уменьшения исходной величины
- 36. Как получить значение, ссылаясь на указатель?
 - o использовать операцию ?
 - o использовать операцию #
 - использовать операцию *
 - o использовать операцию &
- 37. Каково назначение препроцессора?
 - o назначение препроцессора — для сокращения времени работы компилятора
 - назначение препроцессора — обработка исходного текста программы до ее компиляции
 - o назначение препроцессора — подготовка программиста к серьезной работе
 - o назначение препроцессора — облегчить работу программиста по созданию описания программы
- 38. Что делает функция getchar()?
 - резервирует место в куче (heap) для хранения одного символа
 - читает один символ из указанного в качестве аргумента файла
 - выводит в stdout символ, указанный в качестве аргумента
 - o читает один символ из stdin программы и возвращает его код
- 39. Что было бы напечатано, если бы данные операторы являлись частью полной программы?
int n;

- ```
n=5;
printf("%d+%d=%d", n, n, n+n);
```
- o 5
  - o 5+5
  - 5+5=10
  - o 10
40. Что произойдет в результате выполнения оператора putchar('007');?
- o переход на новую строку на устройстве вывода
  - сработает громкоговоритель
  - o печать цифры 7
  - o шаг назад на одну позицию
41. Зачем используют оператор return?
- o чтобы ввести в программу новые значения
  - o чтобы задержать работу программы
  - функция, в которой он содержится, завершает свое выполнение и управление возвращается в вызывающую функцию
- o чтобы организовать цикл
42. Пусть есть двумерный массив. Какой адрес записан верно?
- o \*grid[0]
  - grid[0]
  - o &grid[ ][ ]
43. Может ли функция содержать более одного оператора return?
- o функция может содержать более одного оператора return, только если тип ее void
  - o функция может содержать более одного оператора return, если она не возвращает значения
  - o функция может содержать более одного оператора return, только если тип ее short
  - функция может содержать произвольное количество операторов return независимо от типа возвращаемого значения
44. Какой функцией можно выделить из строки лексемы, разделенные любым из множества символов, входящих во вторую заданную строку?
- o cat
  - strtok
  - o lengst
  - o pos
45. Какие типы могут быть явно преобразованы в тип void?
- o нет, явно в тип void преобразовывать нельзя
  - o в тип void могут быть явно преобразованы только типы long и int
  - все типы могут быть явно преобразованы в тип void
  - o в тип void может быть преобразован только тип unsigned
46. Какова последовательность выполнения операций: \* (умножение), /, %?
- o первой выполняется операция %
  - o первой выполняется операция \*
  - операции последовательно выполняются слева направо в порядке расположения их в выражении
  - o первой выполняется операция /
47. Что понимается под модульным программированием?
- 33,33%организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия, простейшим модулем является функция
  - 33,33%организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия
  - o -100%организация программы в виде совокупности строк с соблюдением правил их взаимодействия

- 33,33%организация программы в виде совокупности модулей со строгим соблюдением правил их взаимодействия, описания интерфейса и описания реализации
48. Что не произойдет в результате выполнения оператора putchar('H');?
- о -100%печать буквы H
  - 33,33%вывод символа '007', в результате чего сработает громкоговоритель
  - 33,33%переход на новую строку на устройстве вывода
  - 33,33%шаг назад на одну позицию
49. Что такое функция?
- о Некоторая часть программы, содержащая описание переменных и констант основной программы
  - Некоторая часть программы, имеющая собственное имя и которая может вызываться из основной программы
    - о Некоторая часть программы, содержащая вредоносный код, и блокирует определенные действия системы
    - о Некоторая часть программы, в которой происходит начальная инициализация всех полей структур, массивов, переменных.
50. Что такое массив?
- о Именованный набор переменных имеющих различные типы данных, и располагающихся в одной памяти
  - о Именованный набор переменных и функций, которые располагаются в одной области памяти
  - Именованный набор переменных имеющих один тип данных, и располагающихся в одной области памяти
    - о Именованный набор переменных имеющих символьный тип данных, и располагающихся в одной области памяти
51. Как написать следующее выражение на языке C «Переменной a присвоено значение b»?
- о a==b
  - a=b
  - о b=a
  - о a:=b
52. Как написать следующее выражение «Второму элементу массива Myarray присвоено значение пяти »?
- о int [1] Myarray=«пять»
  - int Myarray [1] = 5
  - о int Myarray [2] = «пять»
  - о int Myarray [2] = 5
53. Как написать следующее выражение «Если переменная index больше size то мы инкрементируем переменную count »?
- if (index>size) { count++; }
  - о if (index<size) { count--; }
  - о if (index>=size) { ++count; }
  - о if (index<size) { --count; }
54. Какой диапазон значений имеет тип int для 32-разрядных вычислительных систем:
- о от 0 до 255
  - о от -32768 до 32767
  - о от 0 до 65535
  - от 0 до 4 294 967 295
55. Какой размер в байтах имеет переменная вещественного типа float
- о 2
  - 4
  - о 8
  - о 10

56. Дан массив `int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } }`; Чему будет равно значение элемента этого массива `L[1][2]`
- o 2
  - o 3
  - o 4
  - 8
57. Объявление `char *buf;` соответствует
- o созданию символьной переменной `buf`
  - o созданию строковой переменной `buf`
  - созданию указателя `buf` на символьное значение
  - o созданию указателя `buf` на строку
58. Что называется прототипом функции?
- o описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров
  - o описание функции, включая ее имя, тип возвращаемого значения, типы параметров
  - имя функции и тип возвращаемого значения
  - o описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров, тело функции
59. Как обозначается в языке C (C++) следующий режим работы с потоком - создание нового файла для записи и чтения?
- o a+
  - o wb
  - w+
  - o w+b
60. Какая функция, описанная в заголовочном файле читает строку символов из файла?
- o gets()
  - o fputs()
  - fgets()
  - o fscanf()
61. Какой размер массива `M` будет после выполнения кода: `char M[ ] = "\nGoodlive" ?`
- o 10
  - o 8
  - 9
  - o Не определен
62. В каких случаях необходимо использовать оператор `return` в теле функции?
- o Всегда
  - если необходимо, чтобы функция вернула значение
  - o если необходимо обеспечить выход из функции в произвольном месте
  - o если указан тип возвращаемого значения, в том числе и `void`
63. При открытии файла выполняется следующее действие:
- o физический файл связывается с логическим (файловой переменной)
  - o устанавливается тип файла (текстовый или бинарный)
  - устанавливается вид (режим) использования файла
  - o функцией открытия файла возвращается результат (ошибка)
64. Каким способом можно задать многострочный комментарий в языке C
- `/*комментарии к программе*/`
  - o `//комментарии к программе//`
  - o `//комментарии к программе`
  - o `{комментарии к программе}`
65. Логическое выражение может возвращать результат типа
- o integer
  - boolean

- o char
- o logical
- 66. Выберите правильный вариант записи на языке C формулы  $0 < x < 10$ 
  - o  $x > 0, x \leq 10$
  - o  $0 < x \leq 10$
  - o  $x > 0 \ \& \ x \leq 10$
  - $(x > 0) \ \& \ (x < 10)$
- 67. Укажите правильный вариант записи условного оператора в языке C
  - o IF  $x > 0$  Do  $y := \text{sqrt}(x)$
  - o IF  $y := \text{sqrt}(x)$  then  $x > 0$
  - o IF  $x > 0$  then  $y := \text{sqrt}(x)$
  - IF  $(x > 0) \{ y := \text{sqrt}(x) \}$
- 68. Выберите правильный вариант записи на языке C следующего условия: « x принадлежит диапазону  $[0; 10)$  »
  - o  $x \geq 0; x < 10$
  - o  $0 \leq x < 10$
  - o  $(x > 0 \ \& \ (x \leq 10))$
  - $(x \geq 0) \ \& \ (x < 10)$
- 69. Укажите группу, содержащую последовательность правильно записанных на языке C знаков операций отношений
  - o  $\sim, >, <, =, ?$
  - o  $=, <>, ><, >$
  - $=, \geq, \leq, !=$
  - o  $\sim \Rightarrow, \Rightarrow, =, <$
- 70. Тело какого цикла всегда будет выполнено хотя бы один раз, независимо от истинности условия:
  - o While
  - Do While
  - o For
  - o Нет такого цикла в языке C
- 71. В результате выполнения кода `int i=2;`  
`switch (i)`  
`{`  
`case 1: i += 2;`  
`case 2: i *= 3;`  
`case 6: i /= 2;`  
`default;;`  
`}`
  - o переменная i примет значение 6
  - переменная i примет значение 3
  - o переменная i примет значение 2
  - o тело оператора switch не поменяет значение переменной i
- 72. Укажите директиву препроцессора, которую необходимо подключить для организации форматированного ввода-вывода данных:
  - o `#include <iostream.h>`
  - `#include <stdio.h>`
  - o `#include <math.h>`
  - o `#include <conio.h>`
- 73. Где, когда и кем был создан язык Си?
  - o язык Си был создан Н. Виртом
  - o язык Си был создан С. Бьерном
  - язык Си был создан в США в 1972 году сотрудником фирмы Bell Labs Денисом Ритчи



о язык Си был создан в Японии в 1972 году группой разработчиков фирмы Panasonic под руководством Кена Томпсона

74. Есть ли у языка Си собственный редактор?

о да

• нет

о только в ОС UNIX

о только в ОС WINDOWS

75. Чем определяется мобильность языка Си?

о тем, что программа, написанная на Си для одной вычислительной системы, может быть перенесена без изменений на

о другую систему

• тем, что программа, написанная на Си для одной вычислительной системы, может быть перенесена с небольшими

о изменениями или вообще без них на другую

о мобильность языка Си определяется его эффективностью

76. Что будет напечатано? printf("Что ?\n мешает/n вам работать \n");

• Что ?

мешает/n вам работать

о Что ?

мешает

вам работать

о Что ? мешает /n вам работать

о Что? мешает вам работать

77. Каким символом должен заканчиваться оператор?

• символом "точка с запятой" - ';'

о символом "точка" - '.'

о символом "запятая" - ','

о символом "двоеточие" - ':'

78. Какого типа язык Си?

• компилируемого типа

о интерпретируемого типа

о компилируемо-интерпретируемого типа

о интерпретируемого-компилируемого типа

79. Будет ли работать программа и если нет - в чем ошибки?

```
#include
```

```
int main()
```

```
printf("Hello, World!\n");
```

```
}
```

• программа скомпилируется и отработает, но код завершения будет не определен

о программа не скомпилируется из-за синтаксической ошибки

о программа не скомпилируется, т.к. не указан возвращаемый функцией тип

80. Чем обеспечивается надежность программ, написанных на языке Си?

о гибкостью языка Си

о переносимостью языка Си

о мобильностью языка Си

• сильной типизацией языка Си

81. Укажите ошибки и несоответствия стандартам ANSI C в следующем фрагменте:

```
include studio.h
```

```
void main ()
```

- ```
{
printf("Hello, World!");
}
```
- o -100%ошибок нет
 - 50%ошибка в первой строке - должно быть #include
 - 50%функция main() объявлена как void, что не соответствует стандарту ANSI C
82. Укажите строки, не содержащие ошибок синтаксиса:
- o printf("Hello, World!\n");
 - printf("Hello, World!\n");
 - o printf(%d Hello, \n Hello, World!?);
 - o \printf("Hello,\n World!\n");
83. Какие символы могут использоваться для комментариев?
- /* comment */
 - o { comment }
 - o (* comment *)
84. Можно ли на языке Си написать компилятор с языка Си?
- да
 - o нет
 - o только при совместном использовании с C#
 - o только при совместном использовании с C++

Методические материалы, характеризующих процедуры оценивания

На основе перечня вопросов формируются индивидуальные задания для студентов: 4 вопроса из блока 1, 3 вопроса из блока 2, 3 вопроса из блока 3. Результатом итогового контрольного теста является балл, рассчитанный на основе количества правильных ответов. С учетом индивидуального семестрового рейтинга студента формируется итоговый балл по курсу.

Максимальная сумма баллов, набираемая студентом по дисциплине равна 100.

Оценка в баллах	Оценка по шкале	Обоснование	Уровень сформированности компетенций
Более 80	«Отлично»	Содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному	Высокий уровень
66-80	«Хорошо»	Содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено	Продвинутый уровень

		минимальным числом баллов, некоторые виды заданий выполнены с ошибками	
50-65	«Удовлетворительно»	Содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий, возможно, содержат ошибки	Пороговый уровень
Менее 50	«Неудовлетворительно»	Содержание курса не освоено, необходимые практические навыки работы не сформированы, выполненные учебные задания содержат грубые ошибки	Компетенции не сформированы

3. Задания в тестовой форме по дисциплине

Примеры заданий:

1. В каких случаях необходимо использовать оператор return в теле функции?

всегда

если необходимо, чтобы функция вернула значение

если необходимо обеспечить выход из функции в произвольном месте

если указан тип возвращаемого значения, в том числе и void

2. Как передать в функцию f указатель на структуру health_record?

обратиться к функции как к записи структуры: health_record.f

с помощью взятия адреса функции с передачей непосредственно структуры: &f(health_record)

передать указатель на структуру: f(&health_record)

передать непосредственно структуру (но только с поверхностным копированием): f(health_record)

3. Отметьте верное утверждение:

в языке Си функции разделены на категории - функции, возвращающие значения в вызывающую программу и функции, не возвращающие значения в вызывающую программу
только функции с параметрами возвращают значения в вызывающую программу
функции по типу результата определяют, нужно ли вернуть значение в вызывающую программу

4. Укажите правильный вариант функции, возвращающей сумму двух чисел:

int sum(int j,int k) { return(j+k); }

int sum() { return(j+k); int j, k; }

int sum(int j; int k) { return(j+k); }

5. Что делает функция getchar()?

резервирует место в куче (heap) для хранения одного символа
читает один символ из указанного в качестве аргумента файла
выводит в stdout символ, указанный в качестве аргумента
читает один символ из stdin программы и возвращает его код

6. Что понимается под вводом-выводом?

операция пересылки данных между различными файлами

операция пересылки данных между различными частями оперативной памяти

операция пересылки данных между различными внешними устройствами

операция пересылки данных между оперативной памятью и внешними устройствами

7. Каким символом должен заканчиваться оператор?

символом "точка с запятой" - ';'

символом "точка" - '.'

символом "запятая" - ','

символом "двоеточие" - ':'

8. Как написать следующее выражение "Если переменная index больше size то мы инкрементируем переменную count "?

```
if (index>size) { count++; }
```

```
if (index<size) { count--; }
```

```
if (index>=size) { ++count; }
```

```
if (index<size) { --count; }
```

9. Укажите группу, содержащую последовательность правильно записанных на языке C знаков операций отношений

~ >, <, =, ?

=, <>, ><, >

=, >=, <=, !=

~ =>, =<, =, <

10. Есть ли у языка Си собственный редактор?

да

нет

только в ОС UNIX

только в ОС WINDOWS

Полный перечень тестовых заданий с указанием правильных ответов, размещен в банке вопросов на информационно-образовательном портале института по ссылке <https://www.mivlgu.ru/iop/question/edit.php?courseid=2082>

Оценка рассчитывается как процент правильно выполненных тестовых заданий из их общего числа.