

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
функционального анализа
и операторных уравнений

 Каменский М.И.

11.04.2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.04.01 Языки программирования

- 1. Код и наименование направления специальности:** 10.05.04 информационно-аналитические системы безопасности
- 2. Профиль специализации:** Информационная безопасность финансовых и экономических структур, Автоматизация информационно-аналитической деятельности
- 3. Квалификация выпускника:** специалист по защите информации
- 4. Форма обучения:** очная
- 5. Кафедра, отвечающая за реализацию дисциплины:** функционального анализа и операторных уравнений
- 6. Составители программы:** Болдырева Елена Сергеевна преподаватель, математический факультет, кафедра функционального анализа и операторных уравнений.
- 7. Рекомендована:** научно-методическим советом математического факультета, 28.03.2024 протокол № 0500-03.
- 8. Учебный год:** 2024-2025 **Семестр:** 1,2

9. Цели и задачи учебной дисциплины:

Целями освоения учебной дисциплины являются:

- изучение основ технологии разработки программных средств;
- знакомство с основными этапами разработки программ и применяемыми при этом инструментальными средствами.

Задачи учебной дисциплины:

- изучить общие принципы построения и использования современных языков программирования высокого уровня (на примере языка C++), основные сведения о базовых структурах данных, основные комбинаторные и теоретико-графовые алгоритмы, а также способы их эффективной реализации и оценки сложности.

10. Место учебной дисциплины в структуре ООП:

Учебная дисциплина «Языки программирования» относится к обязательной части блока Б1, группе учебных дисциплин «Информационные технологии». Предшествующие дисциплины – Информатика.

Дисциплина необходима для изучения дисциплины Технология и методы программирования.

Специальные требования к входным знаниям, умениям и компетенциям не предусматриваются.

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Коды	Индикаторы	Планируемые результаты обучения
ОПК-7	Способен создавать программы на языках высокого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования	ОПК-7.1	Способен выбирать и применять средства разработки программного обеспечения, а также разрабатывать программное обеспечение на языках программирования высокого уровня	<p>Знать: языки программирования высокого уровня, средства разработки программного обеспечения;</p> <p>Уметь: выбирать и применять средства разработки программного обеспечения, а также разрабатывать программное обеспечение на языках программирования высокого уровня;</p> <p>Владеть: навыками применения технологий и методов программирования для разработки программного обеспечения.</p>

12. Объем дисциплины в зачетных единицах/час.— 7/252**Форма промежуточной аттестации:** зачет, экзамен.**13. Трудоемкость по видам учебной работы**

Вид учебной работы		Трудоемкость		
		Всего	По семестрам	
			1 семестр	2 семестр
Аудиторные занятия		136	68	68
в том числе:	лекции	68	34	34
	практические			
	лабораторные	68	34	34
Самостоятельная работа		80	40	40
в том числе: курсовая работа (проект)				
Форма промежуточной аттестации (экзамен – 36 час.)		36		36
Итого:		252	108	144

13.1. Содержание дисциплины:

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Введение в С++ и С.	Общая характеристика языков, технология разработки программ. Сходство и различие языков С и С++.	https://edu.vsu.ru/enrol/index.php?id=5679
1.2	Типы данных и выражения	Алфавит, идентификаторы, операции, выражения, операторы, классификация типов данных, переменные и константы	https://edu.vsu.ru/enrol/index.php?id=5679
1.3	Управляющие структуры	Организация алгоритмов ветвления и циклов, выбор циклов.	https://edu.vsu.ru/enrol/index.php?id=5679
1.4	Массивы и указатели	Понятие массива, инициализация массивов, ссылки и указатели, указатели и массивы, динамические	https://edu.vsu.ru/enrol/index.php?id=5679

		массивы. Двумерные массивы.	d=5679
1.5	Функции	Общие сведения о функциях, функции с переменным числом параметров, рекурсивные и подставляемые функции, области действия переменных, массивы в качестве параметров функций.	https://edu.vsu.ru/enrol/index.php?id=5679
1.6	Сортировки	Сортировки массивов: пузырьком, вставками, выбором, быстрая сортировка, пирамидальная сортировка.	https://edu.vsu.ru/enrol/index.php?id=5679
2. Практические занятия			
3. Лабораторные работы			
3.1	Типы данных и выражения	Алфавит, идентификаторы, операции, выражения, операторы, классификация типов данных, переменные и константы	https://edu.vsu.ru/enrol/index.php?id=5679
3.2	Управляющие структуры	Организация алгоритмов ветвления и циклов, выбор циклов.	https://edu.vsu.ru/enrol/index.php?id=5679
3.3	Массивы и указатели	Понятие массива, инициализация массивов, ссылки и указатели, указатели и массивы, динамические массивы. Двумерные массивы.	https://edu.vsu.ru/enrol/index.php?id=5679
3.4	Функции	Функции с переменным числом параметров, рекурсивные и подставляемые функции, массивы в качестве параметров функций.	https://edu.vsu.ru/enrol/index.php?id=5679
3.5	Сортировки	Сортировки массивов: пузырьком, вставками, выбором, быстрая сортировка, пирамидальная сортировка.	https://edu.vsu.ru/enrol/index.php?id=5679

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение в С++ и С	12		12	14	38
2	Типы данных и выражения	12		12	14	38
3	Управляющие структуры	12		12	14	38
4	Массивы и указатели	12		12	14	38
5	Функции	10		10	12	32
6	Сортировки	10		10	12	32

	Итого	68		68	80	216
--	-------	----	--	----	----	-----

14. Методические указания для обучающихся по освоению дисциплины

В процессе преподавания дисциплины используются такие виды учебной работы, как лекции, лабораторные занятия, а также различные виды самостоятельной работы обучающихся. На лекциях рассказывается теоретический материал, на лабораторных занятиях решаются примеры по теоретическому материалу, прочитанному на лекциях.

При изучении курса «Языки программирования» обучающимся следует внимательно слушать и конспектировать материал, излагаемый на аудиторных занятиях. Для его понимания и качественного усвоения рекомендуется следующая последовательность действий.

1. После каждой лекции студентам рекомендуется подробно разобрать прочитанный теоретический материал, разобрать примеры, решенные на лекции. Перед следующей лекцией обязательно повторить материал предыдущей лекции.

2. Перед лабораторным занятием обязательно повторить лекционный материал. После лабораторного занятия еще раз разобрать решенные на этом занятии примеры, после чего приступить к выполнению домашнего задания. Если при решении примеров, заданных на дом, возникнут вопросы, обязательно задать на следующем практическом занятии или в часы консультаций преподавателю.

3. При подготовке к лабораторным занятиям повторить основные понятия по темам, изучить примеры. Решая задачи, предварительно понять, какой теоретический материал нужно использовать. Наметить план решения, попробовать на его основе решить практические задачи.

4. Выбрать время для работы с литературой по дисциплине в библиотеке

Освоение дисциплины предполагает не только обязательное посещение обучающимся аудиторных занятий (лекций и лабораторных занятий) и активную работу на них, но и самостоятельную учебную деятельность, на которую отводится 80 часов.

Самостоятельная учебная деятельность студентов по дисциплине «Языки программирования» предполагает изучение рекомендуемой преподавателем литературы по вопросам лекционных и лабораторных занятий (приведены выше), самостоятельное освоение понятийного аппарата и подготовку к текущим аттестациям (коллоквиумам и выполнению практических заданий) (примеры см. ниже).

Вопросы лекционных и практических занятий обсуждаются на занятиях в виде устного опроса – индивидуального и фронтального. При подготовке к лекционным и практическим занятиям, обучающимся важно помнить, что их задача, отвечая на основные вопросы плана занятия и дополнительные вопросы преподавателя, показать свои знания и кругозор, умение логически построить ответ, владение математическим аппаратом и иные коммуникативные навыки, умение отстаивать свою профессиональную позицию. В ходе устного опроса выявляются детали, которые по каким-то причинам оказались недостаточно осмысленными студентами в ходе учебных занятий. Тем самым опрос выполняет важнейшие обучающую, развивающую и корректирующую функции, позволяет

студентам учесть недоработки и избежать их при подготовке к промежуточным аттестациям.

Все выполняемые студентами самостоятельно задания (выполнение контрольной работы и практических заданий) подлежат последующей проверке преподавателем. Результаты текущих аттестаций учитываются преподавателем при проведении промежуточной аттестации (1 семестр – зачет, 2 семестр – экзамен).

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	<i>Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня : учебник для студ. вузов, обуч. по направлению "Информатика и вычислительная техника" / Т.А. Павловская .— СПб. [и др.] : Питер, 2008 .— 460 с. : ил .— (Учебник для вузов) .— Предм. указ.: с.450-460 .— ISBN 978-5-94723-568-5.</i>
2	<i>Прата, Стивен. Язык программирования С++ : [учебник] : лекции и упражнения / Стивен Прата ; [пер. с англ.Ю.И. Корниенко, А.А. Моргунова ; под ред. Ю.Н. Артеменко] .— 6-е изд. — Москва ; Санкт-Петербург ; Киев : Вильямс, 2016 .— 1244 с. : ил .— Парал. тит. л. англ. — Предм. указ.: с.1240-1244.</i>
3	<i>Страуструп, Бьерн. Язык программирования С++. Специальное издание = The С++ programming language. Special edition. / Бьерн Страуструп ; пер. с англ. под ред. Н.Н. Мартынова .— Москва : Бином, 2015 .— 1135 с. : ил .— Предм. указ.: с.1117-1135 .— ISBN 978-5-7989-0425-9 .— ISBN 0-201-70073-5.</i>

б) дополнительная литература:

№ п/п	Источник
4	<i>Лафоре, Р. Объектно-ориентированное программирование в С+ / Р. Лафоре; Пер. с англ. А.Кузнецов и др. — 4-е изд. — СПб. : Питер, 2003 .— 923 с..</i>

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
5	<i>Белоцерковская, И. Е. Алгоритмизация. Введение в язык программирования С++ / Белоцерковская И. Е. , Галина Н. В. , Катаева Л. Ю. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. -</i>

	<i>Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : https://www.studentlibrary.ru/book/intuit026.html</i>
6	<i>Груздев, Денис Владиславович. Программирование С++ (1 курс) [Электронный ресурс] : учебное пособие / Д.В. Груздев .— Воронеж : Издательский дом ВГУ, 2017 .— <URL:http://www.lib.vsu.ru/elib/texts/method/vsu/m17-19.pdf</i>
7	<i>Седжвик, Р. Алгоритмы на С++ / Седжвик Р. - Москва : Национальный Открытый Университет "ИНТУИТ", 2016. - Текст : электронный // ЭБС "Консультант студента" : [сайт]. - URL : https://www.studentlibrary.ru/book/intuit029.html</i>

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	<i>Груздев, Денис Владиславович. Программирование С++ (1 курс) [Электронный ресурс] : учебное пособие / Д.В. Груздев .— Воронеж : Издательский дом ВГУ, 2017 .— <URL:http://www.lib.vsu.ru/elib/texts/method/vsu/m17-19.pdf</i>
2	<i>Тюгашев, Андрей Александрович. Языки программирования : [учебное пособие для студ. вузов, обуч. по специальности 10.05.03 (090303) "Информ. безопасность автоматизир. систем"] : [для бакалавров и специалистов] / А.А. Тюгашев .— Санкт-Петербург [и др.] : Питер , 2015 .— 333 с</i>

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):

При реализации дисциплины используются следующие образовательные технологии: логическое построение дисциплины, установление межпредметных связей, обозначение теоретического и практического компонентов в учебном материале, актуализация личного и учебно-профессионального опыта обучающихся, включение элементов дистанционных образовательных технологий.

В практической части курса используется стандартное современное программное обеспечение персонального компьютера.

В части освоения материала лекционных и лабораторных занятий, самостоятельной работы по отдельным разделам дисциплины, прохождения текущей и промежуточной аттестации может применяться электронное обучение и дистанционные образовательные технологии, в частности, электронный курс «Языки программирования» (URL: <https://edu.vsu.ru/enrol/index.php?id=5679>) на портале «Электронный университет ВГУ».

18. Материально-техническое обеспечение дисциплины:

Для проведения лекционных и лабораторных занятий используются учебные аудитории. Для самостоятельной работы используется класс с компьютерной техникой, оснащенный необходимым программным обеспечением, электронными учебными пособиями и законодательно - правовой и нормативной поисковой системой, имеющий выход в глобальную сеть.

Компьютерный класс: специализированная мебель, маркерная доска, персональные компьютеры

Ubuntu (бесплатное и/или свободное ПО, лицензия:
<https://ubuntu.com/download/desktop>)

Visual Studio Community (бесплатное и/или свободное ПО, лицензия
<https://visualstudio.microsoft.com/ru/vs/community/>)

LibreOffice (GNU Lesser General Public License (LGPL), бесплатное и/или свободное ПО, лицензия: <https://ru.libreoffice.org/about-us/license/>)

В самостоятельной работе обучающиеся используют ресурсы Зональной научной библиотеки ВГУ (электронный каталог: <http://www.lib.vsu.ru>)

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Введение в C++ и C	ОПК-7	ОПК-7.1	контрольная работа
2.	Типы данных и выражения	ОПК-7	ОПК-7.1	контрольная работа
3.	Управляющие структуры	ОПК-7	ОПК-7.1	контрольная работа
4.	Массивы и указатели	ОПК-7	ОПК-7.1	контрольная работа

5.	Функции	ОПК-7	ОПК-7.1	контрольная работа
6.	Сортировки	ОПК-7	ОПК-7.1	контрольная работа
Промежуточная аттестация, форма контроля – зачет, экзамен				Перечень вопросов к зачету, экзамену

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств: контрольная работа.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета.

Текущая аттестация проводится в форме контрольных работ, содержание которых приведено ниже. В ходе контрольной работы обучающемуся выдается КИМ с практическим перечнем заданий и предлагается решить данные задания. В ходе выполнения заданий нельзя пользоваться средствами связи (включая сеть Интернет) и любыми печатными материалами, ограничение по времени — 90 астрономических минут.

При текущем контроле уровень освоения учебной дисциплины и степень сформированности компетенции определяются оценками «зачтено», «не зачтено».

Перечень заданий для контрольных работ

Комплект заданий для контрольной работы № 1

по дисциплине *Языки программирования*

Тема: Методы представления данных. Управляющие структуры

Задание. Составьте алгоритм и напишите программу для вычисления ниже следующих сумм с заданной точностью ϵ . Проведите вычисления при трёх различных значениях параметра ϵ .

Вариант 1 $\sum_{n=1}^{\infty} \frac{\sin n^2}{n\sqrt{n+1}}$;

Вариант 2 $\sum_{n=1}^{\infty} \frac{1}{n \ln(\sin \frac{1}{n})}$

Комплект заданий для контрольной работы № 2

по дисциплине *Языки программирования*

Тема: Массивы и указатели. Функции

Задание. Составьте алгоритм и напишите программу для условий перечисленных в варианте.

Вариант 1 С клавиатуры вводится число элементов массива n , и сами элементы массива. Вычислить и вывести на экран произведение элементов массива, сумма цифр которых кратна 7. Проверку элемента массива производить при помощи логической функции.

Вариант 2 С клавиатуры вводится число элементов массива n , и сами элементы массива. Вычислить и вывести на экран сумму элементов массива, произведение цифр которых делится на 5. Проверку элементов массива производить при помощи логической функции.

Критерии оценивания результатов обучения при текущей аттестации.

Для оценивания результатов контрольной работы используется **шкала:** «зачтено», «не зачтено».

Соотношение показателей, критериев и шкалы оценивания результатов обучения показаны в следующей таблице:

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
При выполнении контрольной работы студент продемонстрировал в достаточной мере: знание основ составления компьютерных программ для решения типовых математических задач, имеющихся ресурсов для решения прикладных математических задач, умение использовать стандартные пакеты программного обеспечения для решения типовых математических задач, владение навыками хранения, поиска, сбора, систематизации, обработки и использования информации.	Достаточный уровень	Зачтено
При выполнении контрольной работы студент не	–	Не

<p>продемонстрировал в достаточной мере: знание основ составления компьютерных программ для решения типовых математических задач, имеющихся ресурсов для решения прикладных математических задач, умение использовать стандартные пакеты программного обеспечения для решения типовых математических задач, владение навыками хранения, поиска, сбора, систематизации, обработки и использования информации.</p>		зачтено
--	--	---------

20.2 Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств:

Собеседование по билетам к зачету

Собеседование по экзаменационным билетам

Промежуточная аттестация предназначена для определения уровня освоения всего объема учебной дисциплины. Промежуточная аттестация по дисциплине «Языки программирования» проводится в форме зачета (1 семестр), экзамена (2 семестр).

Промежуточная аттестация, как правило, осуществляется в конце семестра. Результаты текущей аттестации обучающегося по решению кафедры могут быть учтены при проведении промежуточной аттестации. При несогласии студента, ему дается возможность пройти промежуточную аттестацию (без учета его текущих аттестаций) на общих основаниях.

В ходе промежуточной аттестации обучающемуся выдается КИМ. В ходе выполнения заданий нельзя пользоваться ресурсами глобальной сети Интернет.

Промежуточная аттестация в 1 семестре по дисциплине заключается в собеседовании по вопросам билета к зачету.

Задания к зачету:

Задание. Составьте алгоритм и напишите программу для условий перечисленных в варианте.

Вариант 1 С клавиатуры вводится число элементов массива n , и сами элементы массива. Вычислить и вывести на экран произведение элементов массива, сумма цифр которых кратна 7. Проверку элемента массива производить при помощи логической функции.

Вариант 2 С клавиатуры вводится число элементов массива n , и сами элементы массива. Вычислить и вывести на экран сумму элементов массива, произведение цифр которых делится на 5. Проверку элементов массива производить при помощи логической функции.

Для оценивания результатов обучения на зачете используются следующие **показатели**:

- 1) знание теоретических основ;
- 2) умение решать задачи;
- 3) успешное прохождение текущей аттестации.

Для оценивания результатов обучения на зачете используется **шкала**: «зачтено», «не зачтено».

Соотношение показателей, критериев и шкалы оценивания результатов обучения:

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Полное соответствие ответа обучающегося всем перечисленным показателям по каждому из вопросов контрольно-измерительного материала. Умение применять на практике методы и средства для решения типовых задач, эффективного использования ресурсов современных глобальных сетей в исследованиях.	<i>Повышенный уровень</i>	Зачтено
Несоответствие ответа обучающегося одному из перечисленных показателей (к одному из вопросов контрольно-измерительного материала) и правильный ответ на дополнительный вопрос в пределах программы.	<i>Базовый уровень</i>	Зачтено
Несоответствие ответа обучающегося любым двум из перечисленных показателей и неправильный ответ на дополнительный вопрос в пределах программы.	<i>Пороговый уровень</i>	Зачтено
Несоответствие ответа обучающегося любым из перечисленных показателей (в различных комбинациях по отношению к вопросам контрольно-измерительного материала). В ответе на основные вопросы содержатся отрывочные знания основ, способствующих решению задач профессиональной деятельности, допускаются грубые ошибки при демонстрации умений применять на практике методы для решения типовых задач.	–	Не зачтено

Промежуточная аттестация во 2 семестре по дисциплине осуществляется в форме собеседования по экзаменационным билетам с помощью ниже приведенных оценочных средств (перечень вопросов к экзамену).

Перечень вопросов к экзамену

1. Понятие массива, инициализация массива.
2. Быстрая сортировка
3. Вывести все члены последовательности Фибоначчи, меньшие числа, введённого с клавиатуры.
4. Сортировки пузырьком. Оценка эффективности работы этого алгоритма, его программная реализация

5. Сортировки вставками. Оценка эффективности работы этого алгоритма, его программная реализация
6. Сортировки массива вставками. Оценка эффективности работы этого алгоритма, его программная реализация
7. Быстрая сортировка. Оценка эффективности работы этого алгоритма, его программная реализация
8. Пирамидальная сортировка. Оценка эффективности работы этого алгоритма, его программная реализация

Для оценивания результатов обучения на экзамене используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

<i>Критерии оценивания компетенций</i>	<i>Уровень сформированности компетенций</i>	<i>Шкала оценок</i>
<p><i>Обучающийся знает принципы работы в интегрированной среде разработки программ (например, Microsoft Visual C++), основные инструменты отладки, компиляции в интегрированной среде разработки программ (например, Microsoft Visual C++); умеет реализовывать алгоритм в виде программы на языке C++ в различных интегрированных средах разработки программ (например, Microsoft Visual C++ знает основы организации алгоритмов ветвления, циклов, принципы использования функций в языке C++; умеет самостоятельно строить алгоритм решения задачи и реализовывать его в современных программных комплексах;</i></p>	<p><i>Повышенный уровень</i></p>	<p><i>Отлично</i></p>
<p><i>Сформированные, но содержащие отдельные пробелы представления о принципах работы с инструментами для поиска и обработки больших объемов информации; успешное, но содержащее отдельные пробелы писать доклады и делать презентации на заданную тему с использованием современных информационных технологий для поиска и обработки;</i></p> <p><i>Сформированные, но содержащие отдельные пробелы представления о принципах работы в интегрированной среде разработки программ (например, Microsoft Visual C++), об основных инструментах отладки, компиляции в интегрированной среде разработки программ (например,</i></p>	<p><i>Достаточный уровень</i></p>	<p><i>Хорошо</i></p>

<i>Microsoft Visual C++); успешное, но содержащее отдельные пробелы умение самостоятельно строить алгоритм решения задачи и реализовывать его в современных программных комплексах</i>		
<i>Успешное, но не системное умение реализовывать алгоритм в виде программы на языке C++ в различных интегрированных средах разработки программ (например, Microsoft Visual C++); Успешное, но не системное умение самостоятельно строить алгоритм решения задачи и реализовывать его в современных программных комплексах</i>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
<i>Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.</i>	<i>–</i>	<i>Неудовлетворительно</i>

20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ

Задания открытого типа (тестовые, повышенный уровень сложности) :

1. Определите значение переменной *a* после выполнения фрагмента программы:

```
int a = 10, b = 5;
if (a > 5 && a < b)
    a -= 5;
```

Ответ: 10.

Решение. Поскольку условие *if* не выполнено, значение переменной *a* остается равным 10.

2. Определите значение переменной *a* после выполнения фрагмента программы:

```
int a = 10, b = 5;
if (a > 1 || a < b)
    a -= 5;
if (a > 1 && a == b)
    a -= 5;
```

Ответ: 0.

Решение. Поскольку условие в первом *if* выполнено, значение переменной *a* становится равным 5. Тогда и условие во втором *if* выполнено, и значение переменной *a* становится равным 0.

3. Какую логическую операцию нужно добавить в программу вместо многоточия, чтобы значение переменной *a* после выполнения фрагмента программы стало равно 17?

```
int a = 10, b = 5;
if (a > 1 ... a < b)
    a -= 7;
else
    a += 7;
```

Ответ: &&.

Решение. Нужно добавить такую логическую операцию, чтобы условие в if не выполнялось, а это операция &&.

4. Чему будет равно значение целой переменной a после выполнения этого фрагмента программы?

```
int a = 10;
for (int i = 1; i <= 3; i++)
    a -= 2;
```

Ответ: 4.

Решение. Тело цикла for выполнится ровно 3 раза, следовательно, значение переменной a уменьшится на 6 и станет равным 4.

5. Укажите тип возвращаемого значения следующей функции:

```
int func(char x, float v, double t);
```

Ответ: int.

Решение. Тип возвращаемого значения указывается в самом начале объявления функции.

6. Какую логическую операцию нужно добавить в программу вместо многоточия, чтобы значение переменной a после выполнения фрагмента программы стало равно 3?

```
int a = 10, b = 5;
if (a < 1 ... a > b)
    a = a - 7;
else
    a = a + 7;
```

Ответ: ||.

Решение. Нужно добавить такую логическую операцию, чтобы условие в if выполнялось, а это операция ||.

7. Какое число нужно написать вместо многоточия, чтобы цикл выполнился ровно 2 раза?

```
int i = 0;
while (i <= ...) {
    cout << "Привет!\n";
    i++;
}
```

Ответ: 1.

Решение. Вместо многоточия нужно написать число 1, т.к. начальное значение $i = 0$, а в условии цикла while стоит нестрогое неравенство.

8. Сколько раз будет выполнен этот цикл?

```
int i = 2;
while (i < 5) {
    cout << "Привет!\n";
    i++;
}
```

Ответ: 3.

Решение. Цикл будет выполнен 3 раза, т.к. до его завершения переменная i принимает последовательно значения 2, 3 и 4.

9. Какое число будет выведено на экран в результате выполнения этого цикла?

```
int i = 1;
cout << "9";
while ( i < 3 ) {
    cout << i;
    i++;
}
```

Ответ: 912.

Решение. Сначала будет напечатана цифра 9, затем в цикле будут напечатаны значения i 1 и 2.

10. Чему будет равно значение целой переменной a после выполнения этого фрагмента программы?

```
int a = 10;
for( int i=1; i<=3; i++ )
    a -= 2;
```

Ответ: 4.

Решение. Оператор $a -= 2$, стоящий в теле цикла `for`, будет выполнен 3 раза, значит исходное значение переменной a уменьшится на 6 и станет равным 4.

11. Чему будет равно значение целой переменной a после выполнения этого фрагмента программы?

```
int a = 10;
for ( int i=0; i<=2; i++ )
    a -= i;
```

Ответ: 7.

Решение. Оператор $a -= i$, стоящий в теле цикла `for`, будет выполнен 3 раза при i равной 0, 1 и 2, значит исходное значение переменной a уменьшится на 3 и станет равным 7.

12. Чему будет равно значение целой переменной a после выполнения этого фрагмента программы?

```
int a = 3;
for ( int i=1; i<=2; i++ )
    a += a;
```

Ответ: 12.

Решение. Оператор $a += a$, стоящий в теле цикла `for`, будет выполнен 2 раза, значит исходное значение переменной a увеличится в 4 раза и станет равным 12.

13. Какое число будет выведено на экран после выполнения этого фрагмента программы?

```
cout << 4;
for ( int i=1; i>=3; i-- )
    cout << 2*i;
```

Ответ: 4.

Решение. Поскольку цикл не выполнится ни разу, на экран будет выведено число 4.

14. Чему будет равно значение переменной a после выполнения этого цикла?

```
int i = 2;
```

```
int a = 12;
while ( i < 4 ) {
    a += i;
    i++;
}
```

Ответ: 17.

Решение. Оператор `a += i`, стоящий в теле цикла `for`, будет выполнен 2 раза при `i` равной 2 и 3, значит исходное значение переменной `a` увеличится на 5 и станет равным 17.

15. Какой индекс имеет последний элемент массива `a`?

```
int a[6];
```

Ответ: 5.

Решение. Поскольку в массиве `a` 6 элементов, и индексация начинается с 0, последний элемент имеет индекс 5.

Задания закрытого типа (тестовые, средний уровень сложности):

1. Какой из ниже перечисленных операторов, не является циклом в C++?

- a) `for`
- b) `do while`
- c) `while`
- d) `repeat until`

Ответ: d) `repeat until`.

2. До каких пор будут выполняться операторы в теле цикла `while (x < 100)`?

- a) Пока `x` равен `стам`.
- b) Пока `x` меньше или равен `стам`.
- c) Пока `x` строго меньше `ста`.
- d) Пока `x` больше `ста`.

Ответ: c) Пока `x` строго меньше `ста`.

3. Какой из следующих операторов – оператор сравнения двух переменных?

- a) `equal`
- b) `:=`
- c) `=`
- d) `==`

Ответ: d) `==`.

4. Укажите правильный вызов функции. Предполагается, что функция была объявлена ранее.

- a) `int func();`
- b) `func x, y;`
- c) `func;`
- d) `func();`

Ответ: d) `func();`

5. Укажите правильное объявление массива.

- a) `array an array[10];`
- b) `int anarray;`
- c) `int anarray[10];`

- d) `anarray{10};`
Ответ: c) `int anarray[10];`
6. Язык программирования C++ разработал
- a) Бьерн Страуструп;
 - b) Кен Томпсон;
 - c) Дональд Кнут;
 - d) Никлаус Вирт;
- Ответ: a) Бьерн Страуструп;
7. Тело любого цикла выполняется до тех пор, пока его условие ...
- a) истинно;
 - b) ложно;
 - c) у цикла нет условия;
- Ответ: a) истинно;
8. Какие служебные символы используются для обозначения начала и конца блока кода?
- a) `< >`;
 - b) `()`;
 - c) `{ }`;
 - d) `begin end`;
- Ответ: c) `{ }`;
9. Цикл с предусловием?
- a) `while`;
 - b) `do while`;
 - c) `for`;
- Ответ: a) `while`;
10. Цикл с постусловием?
- a) `while`;
 - b) `do while`;
 - c) `for`;
- Ответ: b) `do while`;
11. Простые типы данных в C++.
- a) целые – `int`, вещественные – `float` или `real`, символьные – `char`;
 - b) целые – `bool`, вещественные – `float` или `double`, символьные – `string`;
 - c) целые – `int`, вещественные – `float` или `double`, символьные – `char`;
 - d) целые – `int`, вещественные – `float` или `double`, символьные – `string`;
- Ответ: c) целые – `int`, вещественные – `float` или `double`, символьные – `char`;
12. Какой из перечисленных типов данных не является типом данных в C++?
- a) `double`;
 - b) `int`;
 - c) `float`;
 - d) `real`;
- Ответ: d) `real`;
13. Объявлена переменная `char a`; Какое из следующих выражений не верно?
- a) `a = 3`;
 - b) `a = "3"`;
 - c) `a = '3'`;
- Ответ: b) `a = "3"`;

14. Укажите зарезервированное ключевое слово для динамического выделения памяти.

- a) new;
- b) delete;
- c) default;
- d) break;

Ответ: a) new;

15. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти.

- a) new;
- b) delete;
- c) default;
- d) break;

Ответ: b) delete;

Критерии и шкалы оценивания заданий ФОС:

1) Задания закрытого типа (выбор одного варианта ответа, верно/неверно):

- 1 балл – указан верный ответ;
- 0 баллов – указан неверный ответ.

2) Задания закрытого типа (множественный выбор):

- 2 балла – указаны все верные ответы;
- 0 баллов – указан хотя бы один неверный ответ.

3) Задания закрытого типа (на соответствие):

- 2 балла – все соответствия определены верно;
- 0 баллов – хотя бы одно сопоставление определено неверно.

4) Задания открытого типа (короткий текст):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

5) Задания открытого типа (число):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ.

Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).

