Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный индустриальный университет»

Кафедра прикладных информационных технологий и программирования

УТВЕРЖДАЮ
Директор института
информационных технологий и
автоматизированных систем
_____ А.В. Зимин
подпись
«____» _____ 20___ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы программирования

09.03.03 «Прикладная информатика»
(направленность (профиль): «Искусственный интеллект и машинное обучение»)

Квалификация выпускника
Бакалавр

Форма обучения
Очная форма

Срок обучения: 4 года

Год начала подготовки 2025

Новокузнецк
2025

СибГИУ
Подлинник электронного документа хранится в системе электронного согласования документов СибГИУ
ДОКУМЕНТ ПОДПИСАН ЭП
Сертификат: daba56d7b892efca3bf5c04ed0251c98b19b099c
Владелец: Директор ИИТиАС Зимин Алексей Валерьевич
Действителен: с 15 января 2025 г. по 10 апреля 2026 г.
Дата и время: 27-05-2025 09:39:33 ПРОВЕРИТЬ ПОДПИСЬ

1 Цели и задачи освоения учебной дисциплины

Целями учебной дисциплины являются:
- изучение общих принципов разработки программ;
- получение первичных навыков программирования на одном из языков программирования высокого уровня.

Задачами учебной дисциплины являются:
- изучение базовых алгоритмических конструкций, используемых при разработке программ;
- изучение основных операторов языка Java;
- получение навыков создания программ простого уровня сложности.

2 Место учебной дисциплины в структуре ООП по направлению подготовки (специальности)

Учебная дисциплина относится к учебным дисциплинам обязательной части **Блока 1 «Дисциплины (модули)»** ООП по направлению подготовки (специальности) 09.03.03 «Прикладная информатика».

Учебная дисциплина опирается на базовые знания и компетенции, полученные в процессе получения предыдущего образования.

Учебная дисциплина дополняет знания и умения, получаемые по одновременно изучаемым и последующим учебным дисциплинам:
- Информатика;
- Python для анализа данных.

3 Планируемые результаты обучения по учебной дисциплине

Процесс изучения учебной дисциплины направлен на формирование следующих компетенций:

– Общепрофессиональные компетенции

Наименование категории (группы) ОПК	Код и наименование ОПК	Код и наименование индикатора достижения ОПК	Планируемые результаты обучения
	ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1 Понимает теоретические основы алгоритмизации и программирования	– знать: теоретические основы алгоритмизации и программирования. – уметь: разрабатывать алгоритмы и программировать.
		ОПК-7.2 Разрабатывает алгоритмическое обеспечение	– знать: основы алгоритмического обеспечения. – уметь: разрабатывать алгоритмическое обеспечение.

		ОПК-7.3 Разрабатывает программное обеспечение	– знать: особенности разработки программного обеспечения. – уметь: разрабатывать программное обеспечение.

4 Объем и содержание учебной дисциплины

Учебные занятия по учебной дисциплине проводятся в форме контактной работы и в форме самостоятельной работы обучающихся.

Контактная работа включает в себя занятия лекционного типа (лекции), занятия семинарского типа (лабораторные работы, коллоквиумы), промежуточную аттестацию обучающихся и иные формы взаимодействия обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации ООП на иных условиях, в том числе при проведении промежуточной аттестации обучающихся. Контактная работа может проводиться с применением электронного обучения, дистанционных образовательных технологий.

Объем учебной дисциплины

Семестр / курс		***ИТОГО***	***1 семестр***	***2 семестр***
Форма промежуточной аттестации			*экзамен*	*зачет*
Трудоёмкость	*академ. час.*	***252***	*144*	*108*
	зачетных единиц	***7***	*4*	*3*
Лекции, *академ. час.*		***16***	*16*	*0*
в форме практической подготовки		***0***	*0*	*0*
Лабораторные работы, *академ. час.*		***48***	*16*	*32*
в форме практической подготовки		***0***	*0*	*0*
Практические занятия, *академ. час.*		***0***	*0*	*0*
в форме практической подготовки		***0***	*0*	*0*
Курсовая работа / проект, *академ. час.*		***0***	*0*	*0*
в форме практической подготовки		***0***	*0*	*0*
Консультации, *академ. час.*		***0***	*0*	*0*
в форме практической подготовки		***0***	*0*	*0*
Самостоятельная работа, *академ. час.*		***143***	*76*	*67*
в форме практической подготовки		***0***	*0*	*0*
Контроль, *академ. час.*		***45***	*36*	*9*
в форме практической подготовки		***0***	*0*	*0*

Содержание учебной дисциплины

Раздел 1 Основы языка Java;

Тема 1.1 Знакомство со средой разработки Visual Studio / IntelliSence (Понятие проекта и порядок его создания. Интерфейс Visual Studio / IntelliSence. Порядок разработки программы);

Тема 1.2 Типы данных, переменные, построение выражений (Типы данных, переменные, константы, выражения и операторы, используемые при их построении);

Тема 1.3 Стандартные классы. Понятие пространства имен и значения null (Класс Object – основа всех классов. Вычисление математических функций с помощью класса Math. Преобразование типов с использованием класса Convert. Класс Random – средство для генерации случайных величин. Понятие пространства имен, структура, вложенность, использование. Понятие null и типы данных, его использующие);

Раздел 2 Разработка программ с использованием базовых конструкций языка;

Тема 2.1 Основные операторы языка (Операторы присваивания, инкремента и декремента, операторные скобки, операторы условия, выбора, условный оператор, операторы циклов, оператор new);

Тема 2.2 Структурированные типы (Одномерные и многомерные массивы, ступенчатые массивы, оператор foreach, структуры);

Тема 2.3 Строки (Базовый класс работы со строками, алгоритмы работы со строками);

Тема 2.4 Методы (Метод, как выделенный код для обработки данных. Правила описания метода);

Раздел 3 Работа с файлами;

Тема 3.1 Работа с файлами, как элементами файловой системы (Класс File – основа для манипуляции над файлами. Перечисления FileMode, FileAccess, FileShare, определяющие параметра доступа к файлу);

Тема 3.2 Работа с файлами, как с потоками (Класс FileStream – универсальный класс доступа к данным файла. Классы StreamReader и StreamWriter для работы с текстовыми файлами. Классы BinaryReader и BinaryWriter для работы с двоичными файлами).

5 Перечень тем лекций

№ раздела / темы дисциплины	**Темы лекций**	**Трудоемкость, *академ.час***	
		всего	**в форме практической подготовки**
Тема 1.2.	Типы данных, переменные, построение выражений	2	
Тема 1.3.	Стандартные классы.	1	

	Понятие пространства имен и значения null		
Тема 2.1.	Основные операторы языка	2	
Тема 2.2.	Структурированные типы	7	
Тема 2.3.	Строки	1	
Тема 2.4.	Методы	1	
Раздел 3.	Работа с файлами	2	
Итого:		**16**	**0**

6 Перечень тем практических занятий (семинаров)

№ раздела / темы дисциплины	**Темы практических занятий (семинаров)**	**Трудоемкость, *академ.час***	
		всего	**в форме практической подготовки**
	Отсутствуют		
Итого:		**0**	**0**

7 Перечень тем лабораторных работ

№ раздела / темы дисциплины	**Темы лабораторных работ**	**Трудоемкость, *академ.час***	
		всего	**в форме практической подготовки**
Тема 1.1.	Знакомство со средой разработки Visual Studio / IntelliSence	2	
Тема 1.2.	Простейшие алгоритмы	4	
Тема 1.2; Тема 1.3; Тема 2.1.	Выражения	4	
Тема 1.2; Тема 2.1.	Циклы	6	
Тема 1.2; Тема 2.1; Тема 2.2.	Одномерные массивы	4	
Тема 1.2; Тема 2.1; Тема 2.2.	Матрицы	6	
Тема 1.2; Тема 2.1; Тема 2.3.	Строки	4	
Тема 1.2; Тема 2.1; Тема 2.2; Тема 2.4.	Методы	4	
Тема 1.2; Тема 2.1; Тема 2.2; Тема 3.2.	Структуры и файлы	4	
Тема 1.2;	Стандартные алгоритмы	4	

Тема 2.1; Тема 2.2; Тема 2.4.	обработки данных		
Тема 1.2; Тема 1.3; Тема 2.1; Тема 2.2; Тема 3.2.	Комплексное задание	6	
Итого:		**48**	**0**

8 Перечень тем курсовых работ (проектов)

№ раздела / темы дисциплины	**Темы курсовых работ (проектов)**	**Трудоемкость, *академ.час***	
		всего	**в форме практической подготовки**
	Отсутствуют		
Итого:		**0**	**0**

9 Виды самостоятельной работы

№ раздела / темы дисциплины	**Виды самостоятельной работы**	**Трудоемкость, *академ.час***	
		всего	**в форме практической подготовки**
Раздел 1.	1. Изучение лекционного материала; 2. Подготовка к лабораторной работе; 3. Прохождение тестирования.	47	
Раздел 2.	1. Изучение лекционного материала; 2. Подготовка к лабораторной работе; 3. Прохождение тестирования.	47	
Раздел 3.	1. Изучение лекционного материала; 2. Подготовка к лабораторной работе; 3. Прохождение тестирования.	49	
Контроль	*Подготовка к экзамену*	36	
Контроль	*Подготовка к зачёту*	9	
Итого:		**188**	**0**

10 Матрица компетентностных задач по учебной дисциплине

Контролируемые разделы (темы) дисциплины	**Контролируемые компетенции**	**Оценочные средства**

Раздел 1.	ОПК-7	1. Наблюдение за работой на лабораторной работе; 2. Тестовые задания
Раздел 2.	ОПК-7	1. Наблюдение за работой на лабораторной работе; 2. Тестовые задания
Раздел 3.	ОПК-7	1. Наблюдение за работой на лабораторной работе; 2. Тестовые задания
Зачет.	ОПК-7	1. Зачет на основе результатов текущего контроля
Экзамен.	ОПК-7	1. Экзамен на основе результатов тестирования

11 Входной контроль

Дисциплина «Основы программирования», изучаемая в 1 семестре, не подразумевает проведение входного контроля и рассчитана на обучающегося 1 курса, поступившего в университет в соответствии с Правилами приема в СибГИУ.

12 Текущий контроль

12.1 Текущий контроль знаний используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной) обучающихся. Данный вид контроля стимулирует у обучающихся стремление к систематической самостоятельной работе по изучению учебной дисциплины и позволяет оценить степень освоения изученного материала.

Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы по индивидуальной инициативе педагогического работника, в том числе посредством использования СУО «Moodle».

Формы проведения текущего контроля – наблюдение за работой на лабораторной работе; тестовые задания.

12.2 Описание оценочных средств

Шкалы оценивания (методики оценки)

12.2.1 Рекомендации по оцениванию результатов тестирования обучающихся

Тестирование проводится по изученным разделам (темам) учебной дисциплины для выявления уровня знаний и умений обучающегося, мотивирования обучающегося к активизации самостоятельной работы по усвоению учебного материала, выявления и устранения пробелов в знаниях, повышения дисциплины и организации учебной деятельности обучающегося.

Тестирование проводится в СУО «Moodle» в рамках часов, отведенных на самостоятельную работу обучающихся. Для тестирования используется 20 вопросов из базы типовых тестовых заданий по изученным разделам (темам) учебной дисциплины. Обучающийся выбирает один ответ из предложенных вариантов ответа. В случае выбора неправильного ответа обучающийся получает за данный вопрос 0 баллов; в случае правильного ответа – 1 балл.

Оценка формируется в соответствии с критериями таблицы:

Максимальный балл	Проходной балл	Оценка
20	не менее 18	отлично
17	не менее 14	хорошо
13	не менее 10	удовлетворительно
9	–	неудовлетворительно

12.2.2 Рекомендации по оцениванию в процессе наблюдения за работой на лабораторном занятии

Наблюдение за работой на лабораторном занятии – важнейший компонент контроля деятельности обучающихся и качества их обучения.

Критерии оценки в процессе наблюдения за работой на лабораторном занятии:

– самостоятельность выполнения задания и работы с методическими указаниями (учитывается работа в течение занятия, быстрота и способность нахождения в методических указаниях нужной информации);

– правильность выполнения работы (учитывается последовательность выполняемых действий, умение работать с инструментами и/или средствами вычислительной техники);

– рациональность использования времени, отведенного на задание (не одобряется затянутость выполнения работы с учетом индивидуальных особенностей обучающихся);

– активность обучающегося при выполнении лабораторной работы (учитывается индивидуальная работа и работа в группе).

Критерии оценки по результатам наблюдения за работой на лабораторном занятии

Наименование параметра оценки	Максимальное количество баллов
Самостоятельно выполняет задания; умеет работать с методическими указаниями.	2,5
Указанные в методических указаниях действия выполняет в правильной последовательности; умеет работать с инструментами и/или средствами вычислительной техники.	1,25
Рационально использует время, отведенное на задание; проявляет активность при выполнении лабораторной работы.	1,25

Наименование параметра оценки	Максимальное количество баллов
Итого	**5**

Максимальное количество – 5 баллов соответствует оценке «отлично», 4 балла – «хорошо», 3 балла – «удовлетворительно», 2 балла – «неудовлетворительно».

Задания и вопросы для текущего контроля знаний обучающихся по разделам (темам) учебной дисциплины приведены в приложении.

13 Промежуточная аттестация

13.1 Оценочные материалы для промежуточной аттестации обучающихся по учебной дисциплине предназначены для оценки степени достижения запланированных результатов обучения по завершению изучения учебной дисциплины в установленной учебным планом форме и позволяют определить качество усвоения изученного материала.

Итоговой формой контроля сформированности компетенций у обучающихся по учебной дисциплине является: зачет; экзамен.

Оценочные материалы для промежуточной аттестации обучающихся включают: зачет на основе результатов текущего контроля; экзамен на основе результатов тестирования.

13.2 Оценивание обучающегося на зачете

Оценка	Требования к знаниям
зачтено	Оценка «зачтено» выставляется обучающемуся, если он твердо знает программный материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, обладает необходимыми умениями и приемами их выполнения, а также имеет достаточно полное представление о значимости знаний по учебной дисциплине.
не зачтено	Оценка «не зачтено» выставляется обучающемуся, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «не зачтено» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по данной учебной дисциплине.

13.3 Оценивание обучающегося на экзамене

Оценка	Требования к знаниям
отлично	Оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал,

Оценка	Требования к знаниям
	исчерпывающе, последовательно, четко и логически его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, обладает приемами выполнения практических задач по формированию компетенций.
хорошо	Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, обладает необходимыми умениями и приемами их выполнения.
удовлетворительно	Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой.
неудовлетворительно	Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по данной учебной дисциплине.

Задания и вопросы для промежуточной аттестации обучающихся приведены в приложении.

14 Учебно-методическое и информационное обеспечение учебной дисциплины

а) литература:

1 Дунаев, С. Intranet-Технологии : WebDBC. CGI. Corba 2.0. Netscape Suite. Borland IntraBuilder. Java и JavaScript. LiveWire : практическое пособие / С. Дунаев. – Москва : Диалог-МИФИ, 1996. – 265 с. : табл., схем., ил. – ISBN 5-86404-093-2. – URL: https://biblioclub.ru/index.php?page=book&id=89276 (дата обращения: 25.05.2025);

2 Соколова, В. В. Разработка мобильных приложений : учебное пособие / В. В. Соколова. — Томск : ТПУ, 2014. — 176 с. — ISBN 978-5-4387-0369-3. – URL: http://e.lanbook.com/books/element.php?pl1_id=82830 (дата обращения: 25.05.2025);

3 Пономарчук, Ю. В. Программирование на языке Java : учебное пособие / Ю. В. Пономарчук, И. В. Кузнецов. — Хабаровск : ДВГУПС,

2021. — 103 с. – URL: https://e.lanbook.com/book/259451 (дата обращения: 25.05.2025).

б) ресурсы информационно-телекоммуникационной сети «Интернет»:

Не задана.

в) лицензионное и свободно распространяемое программное обеспечение:

– 7-Zip;
– Adobe Acrobat Reader;
– Astra Linux Special Edition;
– Kaspersky Endpoint Security;
– Microsoft Office;
– Microsoft Windows;
– OnlyOffice;
– Visual Studio;
– Р7-Офис.

г) базы данных и информационно-справочные системы:

Не задана.

15 Материально-техническое обеспечение учебной дисциплины

Материально-техническое обеспечение учебной дисциплины включает учебные аудитории, оснащенные оборудованием, компьютерной техникой, и техническими средствами обучения, в том числе:

- учебную аудиторию для проведения занятий лекционного типа, оборудованную учебной доской, экраном и мультимедийным проектором;
- учебную аудиторию для проведения занятий семинарского типа (лабораторных работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенную компьютерной техникой;
- помещение для самостоятельной работы, оснащенное компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду, научно-техническую библиотеку СибГИУ.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) 09.03.03 «Прикладная информатика».

Составитель(и):

доцент Кожемяченко Вадим Иванович (кафедра прикладных информационных технологий и программирования);

доцент Мартусевич Ефим Александрович (кафедра прикладных информационных технологий и программирования).

Рабочая программа дисциплины рассмотрена и утверждена на заседании кафедры.

Согласована:

Ген. директор
ООО «Агит-плюс»  *А.Ю. Рыжих*

Приложение

Задания и вопросы для текущего контроля знаний обучающихся по разделам (темам) учебной дисциплины

1 Тестирование по результатам освоения лекционного материала и самостоятельной работы

Тестирование проводится для определения степени усвоения материалов лекций, а также выполнения самостоятельной работы в рамках часов, отведенных на самостоятельную работу обучающихся, в СУО «Moodle».

Перечень вопросов для тестирования:

Тестирование проводится для определения степени усвоения материалов лекций, а также выполнения самостоятельной работы в рамках часов, отведенных на самостоятельную работу обучающихся, в СУО «Moodle». Ниже приведены примеры вопросов по темам.

Тема 1.2 Типы данных, переменные, построение выражений

Вопрос 1:

Имеется фрагмент программы:

```
int i = 5, j = 7;
j += i*3;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 2:

Имеется фрагмент программы:

```
int i = 5;
double j = i/3;
```

Как можно изменить вторую строку, чтобы в переменную j был записан результат обычного деления?

Метод ответа – выбор одного или нескольких ответов.

- `double j = i/3.0;`
- `double j = (double)i/3;`
- `double j = (double)i/3.0;`
- `double j = double(i)/3;`
- `double j = (i as double)/3;`

Вопрос 3:

Имеется фрагмент программы:

```
int n = <...>.Parse("-5");
```

Что должно быть написано вместо <...>, чтобы было выполнено правильное преобразование из строки в целое число?

Метод ответа – ввод ответа.

Вопрос 4:

Имеется фрагмент программы:

```
int i = <...>;
```

Что должно быть написано вместо <...>, чтобы переменной i было присвоено минимально возможное значение?

Метод ответа – ввод ответа.

Тема 1.3 Стандартные классы. Понятие пространства имен и значения null

Вопрос 1:

Какое значение вернет метод Sign, реализованный в классе Math, если в качестве параметра указать целое число -5?

Метод ответа – выбор только одного ответа.

- 1
- 0
- -1
- 5
- -5

Вопрос 2:

Требуется подключить пространство имен MyNameSpace. Какая команда позволит это сделать?

Примечание: при необходимости использования пробелов, ставить не более одного пробела подряд. Перед знаками препинания пробелы не ставятся.

Метод ответа – ввод ответа.

Вопрос 3:

Имеется фрагмент программы:

```
int? i;
...
int j = <...>;
```

Какой код, поставленный вместо <...>, позволит присвоить переменной j значение переменной i, если оно отлично от null, или значение 5, если оно равно null?

Примечание: при необходимости использования пробелов, ставить не более одного пробела подряд.

Метод ответа – ввод ответа.

Тема 2.1 Основные операторы языка

Вопрос 1:

Имеется фрагмент программы:

```
int i = 5;
int j = i++;
```

Каким будет значение переменных i и j после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- i=6, j=5
- i=5, j=5
- i=6, j=6
- i=5, j=6

Вопрос 2:

Имеется фрагмент программы:

```
int i = 5, j = 7;
if (j >= 5 || i++ < 7)
  j += i;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 3:

Имеется фрагмент программы:

```
int i;
...
if (i<0 <...> i>100)
  i = 50;
```

Какой код, подставленный вместо <...>, позволит присвоить переменной i значение 50 в случае, если она не входит в диапазон от 0 до 100?

Метод ответа – ввод ответа.

Вопрос 4:

Имеется фрагмент программы:

```
int i = 5, j = 6;
int k = i < j ? i + j : i - j;
```

Каким будет значение переменной k после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 5:

Имеется фрагмент программы:

```
int x = 10, n = 6, m = 4;
switch (n % m)
{
  case 0:
    x = 5;
    break;
  case 1:
    x = 8;
    break;
  case 2:
    x = 4;
    break;
  case 3:
    x = 2;
    break;
}
int y = x;
```

Каким будет значение переменной y после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- 4
- 5
- 8
- 2
- 10

Вопрос 6:

Заголовок оператора for имеет три блока: <инициализация>, <условие>, <итерация>.

```
for (<инициализация>; <условие>; <итерация>)
  <оператор>;
```

Когда выполняется блок <инициализация>?

Метод ответа – выбор только одного ответа.

- один раз перед началом цикла
- перед каждой итерацией цикла
- после каждой итерации цикла
- по окончании цикла

Вопрос 7:

Имеется фрагмент программы:

```
int i, j=1;
for (i=5; i>2; i--)
  j += i;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 8:

Имеется фрагмент программы:

```
int i=1, j=0;
while (i<4)
  j += ++i;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Тема 2.2 Структурированные типы

Вопрос 1:

Имеется фрагмент программы:

```
int[] mas = <...>;
foreach (int i in mas)
  <...>
```

Что содержит переменная i при выполнении цикла?

Примечание: блок <...> означает подстановку вместо него некого корректного кода, не влияющего на суть вопроса.

Метод ответа – выбор только одного ответа.

- элемент массива
- весь массив
- номер элемента массива
- количество элементов массива

Вопрос 2:

Имеется фрагмент программы:

```
int sum = 0;
int[] mas = {1,2,3};
foreach (int i in mas)
  <...>
```

Какая строка, подставленная вместо <...>, позволят найти сумму элементов массива и записать ее в переменную sum?

Метод ответа – выбор только одного ответа.

- sum += i;
- sum += mas[i];
- sum += mas(i);
- sum += mas.i;

Вопрос 3:

Имеется фрагмент программы:

```
int[,,] mas = new int[2,3,4];
foreach (int i in mas)
  <...>
```

Сколько раз выполнится блок <...>?

Метод ответа – ввод ответа.

Вопрос 4:

Какие из нижеприведенных операторов позволят правильно **описать и создать одномерный массив**?

Метод ответа – выбор одного или нескольких ответов.

- int[] mas = new int[10];
- int[,] mas = new int[10];
- int[10] mas;
- int[] mas = int[10];

Вопрос 5:

Имеется фрагмент программы:

```
int[] mas = new int[???];
mas[<...>] = 10;
```

Какой код, подставленный вместо <...> позволит присвоить значение последнему элементу массива?

Примечание: считать, что вместо ??? подставлено некоторое правильное значение.

Метод ответа – ввод ответа.

Вопрос 6:

Какие из нижеприведенных строк **не содержат** синтаксических ошибок?

Метод ответа – выбор одного или нескольких ответов.

- int[] mas;
- int[,] mas;
- int[,] mas = {{1,2,3},{5,6,7}};
- int[,] mas = {1,2,3,5,6,7};
- int[] mas = {{1,2,3},{5,6,7}};

Вопрос 7:

Какие из нижеприведенных строк **не содержат** синтаксических ошибок?

Метод ответа – выбор одного или нескольких ответов.

- `int[][] mas;`
- `int[][] mas = new int[2][];`
- `int[][] mas = new int[2][2];`
- `int[2][2] mas;`
- `int[][] mas = new int[2,2];`

Вопрос 8:

Какие из нижеприведенных фрагментов **полностью** создают ступенчатый массив?

Примечание: полностью, значит задав количество элементов во всех размерностях, без синтаксических ошибок, не выходя за границы массивов

Метод ответа – выбор одного или нескольких ответов.

- ```
  int[][] mas = new int[2][];
  mas[0] = new int[2];
  mas[1] = new int[3];
  ```
- `int[][] mas = new int[2][3];`
- `int[][] mas = new int[2][];`
- ```
  int[][] mas = new int[2][];
  mas[1] = new int[2];
  ```
- ```
  int[][] mas = new int[2][];
  mas[0] = new int[2];
  mas[1] = new int[3];
  mas[2] = new int[2];
  ```

Вопрос 9:

Имеется фрагмент программы:

```
int[] mas1 = {1,2};
int[] mas2 = mas1;
mas1[0] = 3;
```

Каким будет значение mas2[0]?

Метод ответа – ввод ответа.

Вопрос 10:

Имеется фрагмент программы:

```
int[] mas1 = {1,2};
int[] mas2 = mas1;
mas1 = new int[5];
mas2[0] = 3;
```

Каким будет значение mas1[0]?
```

Метод ответа – ввод ответа.

Тема 2.3 Строки

Вопрос 1:

Какие из нижеприведенных операторов позволят присвоить строке **s** значение "**Привет**"?

Метод ответа – выбор одного или нескольких ответа.

- `string s = "Привет";`
- `string s = "При"+"вет";`
- `char[] mas = {'П','р','и','в','е','т'};`
  `string s = new string(mas);`
- `string s = 'Привет';`
- `string s = 'При'+'вет';`

Вопрос 2:

Имеется фрагмент программы:

```
string s1 = "Диск";
string s2 = "Мышь";
bool b = s1==s2;
```

Каким будет значение переменной b после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- false
- true
- фрагмент не будет выполнен, т.к. содержит синтаксическую ошибку

Вопрос 3:

Имеется фрагмент программы:

```
string s1 = "Диск";
string s2 = "Мышь";
bool b = s1>s2;
```

Каким будет значение переменной b после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- фрагмент не будет выполнен, т.к. содержит синтаксическую ошибку
- false
- true

Вопрос 4:

Имеется фрагмент программы:

```
string s = "Диск";
if (<...>)
```

```
s = "диск";
```

Какой код, подставленный вместо <...>, позволит сравнить первую букву строки со значением "Д"?

Примечание: при вводе не использовать пробелы и методы Equals, Compare, CompareTo.

Метод ответа – ввод ответа.

Вопрос 5:

Имеется фрагмент программы:

```
string s1 = "one";
string s2 = "two";
<...> x = s1.CompareTo(s2);
```

Каким будет значение переменной **x** при условии, что ее тип (<...>) описан правильно?

Метод ответа – ввод ответа.

Вопрос 6:

Имеется фрагмент программы:

```
string s1 = "one";
string s2 = "ne";
<...> x = s1.IndexOf(s2);
```

Каким будет значение переменной **x** при условии, что ее тип (<...>) описан правильно?

Метод ответа – ввод ответа.

Вопрос 7:

Имеется фрагмент программы:

```
string s1 = "one";
string s2 = "no";
string s3 = s1.Insert(2,s2);
```

Каким будет значение переменной **s3** после выполнения фрагмента?

Примечание: при необходимости используйте кавычки

Метод ответа – ввод ответа.

Тема 2.4 Методы

Вопрос 1:

Какие описания методов допустимы?

Метод ответа – выбор одного или нескольких ответа.

- `public int M() {...}`
- `int M() {...}`
- `public void M() {...}`

- public int.M {...}
- public M() : int {...}

Вопрос 2:

Какие из нижеприведенных описаний и реализаций методов **допустимы** с точки зрения синтаксиса?

Метод ответа – выбор одного или нескольких ответа.

-
```
private int M()
  {
    int n = 5;
    return n;
  }
```
-
```
double M(int n, double m)
  {
    return n*m;
  }
```
-
```
public void M()
  {
    int j = 7;
    return j;
  }
```
-
```
public int M()
  {
    string s = "Пример";
  }
```
-
```
public int M(int m, n)
  {
    return m+n;
  }
```

Вопрос 3:

Имеется описание метода:

```
public bool M(ref int i)
{
  return i>0 ? true : false;
}
```

и описание переменной:

```
int m = 5;
```

Какие из нижеприведенных команд вызова метода допустимы?

Метод ответа – выбор одного или нескольких ответа.

- M(ref m)
- M(m)
- M(out m)

- M(ref 5)
- M(5)

Вопрос 4:

Имеется описание метода:

```
public int M(int n, params int[] m)
{
  int sum = 0;
  foreach (int i in m)
    sum += i;
  sum -= n;
  return sum;
}
```

а также его вызов:

```
M(1,2,3);
```

Какое значение вернет метод?

Метод ответа – ввод ответа.

Вопрос 5:

Имеется описание метода:

```
public int M(int n1, int n2)
{
  return n1 - n2;
}
```

описание переменных:

```
int n1 = 1, n2 = 2;
```

а также его вызов

```
M(n2,n1);
```

Какое значение вернет метод?

Метод ответа – ввод ответа.

Тема 3.1 Работа с файлами, как элементами файловой системы

Вопрос 1:

Каково назначение метода Exists класса File?

Метод ответа – выбор только одного ответа.

- проверка существования файла
- удаление файла

– определение размера файла
– открытие существующего файла

Вопрос 2:

Каково назначение перечисления FileMode?

Метод ответа – выбор только одного ответа.

– список режимов работы с файлом: Create, Open и т.д.
– список доступных операций для файлом: Read, Write и т.д.
– список типов совместного использования файла

Вопрос 3:

Каково назначение метода CreateText класса File?

Метод ответа – выбор только одного ответа.

– создание нового или перезапись существующего файла в кодировке UTF-8
– создание нового файла, содержащего указанный текст
– создание текстовой переменной, содержащий содержимое текстового файла
– генерация случайного имени файла указанной длины

Тема 3.2 Работа с файлами, как с потоками

Вопрос 1:

Каково назначение метода Seek класса FileStream?

Метод ответа – выбор только одного ответа.

– перемещение текущей позиции в файле
– поиск данных в файле
– определение размера потока, связанного с файлом
– возвращение признака достижения конца файла

Вопрос 2:

Меняет ли выполнение метода Peek класса StreamReader текущую позицию в файле?

Метод ответа – выбор только одного ответа.

– нет
– да
– это определяется параметром метода

Вопрос 3:

Каково назначение метода ReadBytes класса BinaryReader?

Метод ответа – выбор только одного ответа.

– считывание из потока заданного количества байт и размещение их в массиве
– считывание из потока заданного количества байт и нахождение их суммарного значения
– считывание из потока байт в количестве, соответствующем разрядности операционной системы и размещение их в отдельной переменной

– проверка возможности считать из потока заданное количество байт

2 Выполнение лабораторных работ

Задания для лабораторных работ, теоретический и методический материал для самостоятельного изучения и вопросы для собеседования представлены в методических указаниях к лабораторным работам.

3 Оценивание обучающегося на зачете

Промежуточная аттестация производится на основе текущего контроля успеваемости при условии выполнения всех заданий в семестре, предусмотренных фондом оценочных средств по дисциплине.

4 Оценивание обучающегося на экзамене

Тестовые задания для компьютерного тестирования

Тема 1.2 Типы данных, переменные, построение выражений

Вопрос 1:

Имеется фрагмент программы:

```
int i = 5;
i += i*4;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 2:

Какие типы данных не относятся к ссылочным?

Метод ответа – выбор одного или нескольких ответа.

- числовые (целые числа)
- числовые (вещественные числа)
- перечисления
- классы
- массивы

Вопрос 3:

Какие типы данных относятся к целочисленным?

Метод ответа – выбор одного или нескольких ответа.

- short
- byte
- ulong
- double
- string

Вопрос 4:

Имеется фрагмент программы:

```
int n = 0;
```

```
bool b = int.TryParse("5,1", out n);
```

Каким будет значение переменных b и n после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- b=false; n=0
- b=false; n=5
- b=true; n=0
- b=true; n=5
- фрагмент не будет скомпилирован, т.к. содержит ошибку

Тема 1.3 Стандартные классы. Понятие пространства имен и значения null

Вопрос 1:

Известно, что класс Math реализован в пространстве имен System. Будем считать, что данное пространство имен подключено к проекту. Какое из нижеприведенных выражений позволит рассчитать синус 13 градусов?

Метод ответа – выбор только одного ответа.

- Math.Sin(13.0/180*Math.PI)
- Sin(13/180*PI)
- Sin(13)
- Math.Sin(13)
- Math.Sin(13.0/180*PI)

Вопрос 2:

Имеется фрагмент программы:

```
Random r = new Random();
int n = <...>;
```

Какой код, подставленный вместо <...>, позволит присвоить переменной n случайное значение в диапазоне от -10 до 15?

Примечания: использовать метод генерации числа **с одним параметром.** Не использовать пробелы.

Метод ответа – ввод ответа.

Вопрос 3:

Имеются фрагменты программы в двух файлах:

- первый файл:

```
namespace NS1
{
  class A {...}
  namespace NS2
  {
```

```
    class B {...}
  }
}
```

-второй файл:

```
using NS1.NS2;
```

Как можно описать переменную clA класса A во втором файле?

Примечания: при описании использовать не более одного пробела подряд.

Метод ответа – ввод ответа.

Вопрос 4:

Как описать переменную d таким образом, чтобы она могла принимать не только значения типа double, но и значение null?

Примечания: при описании использовать не более одного пробела подряд.

Метод ответа – ввод ответа.

Тема 2.1 Основные операторы языка

Вопрос 1:

Имеется фрагмент программы:

```
int i = 4;
int j = ++i;
```

Каким будет значение переменных i и j после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- i=5, j=5
- i=4, j=5
- i=5, j=4
- i=4, j=4

Вопрос 2:

Имеется фрагмент программы:

```
int i = 4, j = 6;
if (j >= 4 || i++ < 6)
  j += i;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 3:

Имеется фрагмент программы:

```
int n = 1;
switch (n)
{
  case 1:
    n = 2;
    goto case 2;
  case 2:
    n = 3;
    break;
  case 3:
    n = 4;
    break;
}
```

Каким будет значение переменной n после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 4:

Заголовок оператора for имеет три блока: <инициализация>, <условие>, <итерация>.

```
for (<инициализация>; <условие>; <итерация>)
  <оператор>;
```

Когда выполняется блок <итерация>?

Метод ответа – выбор только одного ответа.

- после каждой итерации цикла
- перед каждой итерацией цикла
- один раз перед началом цикла
- по окончании цикла

Вопрос 5:

Имеется фрагмент программы:

```
int i, j=1;
for (i=1; i<4; i++)
  j += j;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – ввод ответа.

Вопрос 6:

Имеется фрагмент программы:

```
int i=10, j=0;
```

```
while (i>7)
  j++;
  i--;
```

Каким будет значение переменной j после выполнения фрагмента?

Метод ответа – выбор только одного ответа.

- данный фрагмент образует "вечный цикл"
- 1
- 3
- 0
- фрагмент не будет скомпилирован из-за ошибки

Тема 2.2 Структурированные типы

Вопрос 1:

Какие из нижеприведенных операторов позволят правильно описать и создать одномерный массив?

Метод ответа – выбор одного или нескольких ответов.

- `double[] mas = new double[5];`
- `double[,] mas = new int[3,4];`
- `int[4] mas = {1,2,3,4};`
- `int[] mas = int[4] {1,2,3,4};`

Вопрос 2:

Имеется фрагмент программы:

```
int[] mas = new int[6];
```

Какой код позволит присвоить пятому элементу массива значение 1?

Примечание: при вводе не использовать пробелы.

Метод ответа – ввод ответа.

Вопрос 3:

Имеется фрагмент программы:

```
byte[] mas = {7,8,9};
```

Какое максимальное значение индекса допустимо при обращении к массиву mas?

Метод ответа – ввод ответа.

Вопрос 4:

Какие из нижеприведенных строк **не содержат** синтаксических ошибок?

Метод ответа – выбор одного или нескольких ответов.

- `bool[,,,] mas;`
- `int[] mas = new int[3] {-1,-2,-3};`
- `int[,] mas = new int[5];`

- int[,] mas = int[2,3];
- int[][] mas = new int[2,3] {{1,2},{3,4},{5,6}};

Вопрос 5:

Имеется фрагмент программы:

```
int[] mas1 = {1,2};
int[] mas2 = mas1;
mas1[0] = 5;
mas1 = new int[5];
mas1[0] = 3;
```

Каким будет значение mas2[0]?

Метод ответа – ввод ответа.

Тема 2.3 Строки

Вопрос 1:

Какие из нижеприведенных операторов позволят присвоить строке s значение "Привет"?

Метод ответа – выбор одного или нескольких ответов.

- string s = "П"+"р"+"и"+"в"+"е"+"т";
- char[] mas = {'П','р','и','в','е','т'};
  string s = new string(mas);
- string s = 'Привет';
- string s = {'П','р','и','в','е','т'};
- string s = {"П","р","и","в","е","т"};

Вопрос 2:

Имеется фрагмент программы:

```
string s1 = "Компьютер";
string s2;
```

Введите код программы, позволяющий записать в строку s2 слово, полученное путем **удаления** из строки s1 второй, третьей и четвертой букв (использовать метод удаления)?

Примечание: при вводе не использовать пробелы.

Метод ответа – ввод ответа.

Тема 2.4 Методы

Вопрос 1:

Какие из нижеприведенных описаний и реализаций методов допустимы с точки зрения синтаксиса?

Метод ответа – выбор одного или нескольких ответов.

- private int M()
  {
    int n = 4;

```
    return n/3;
  }
- void M(double n, double m)
  {
    n *= m;
  }
- public void M(int n)
  {
     return n*2;
  }
- public string M()
  {
    string s = "Строка";
  }
- public int M(int a, b)
  {
    return a-b;
  }
```

Вопрос 2:

Имеется описание метода:

```
public int M(int[] mas)
{
  int n = 0;
  for (int i = 1; i < mas.Length; i++)
    if (mas[n] > mas[i])
      n = i;
  return n;
}
```

Что рассчитывает данный метод?

Метод ответа – выбор только одного ответа.

- позицию минимального элемента
- значение минимального элемента
- позицию максимального элемента;
- значение максимального элемента
- количество элементов с максимальным значением;

Вопрос 3:

Имеется описание метода:

```
public void M(int a, out int b, int c)
{
  b = c - a;
}
```

описание переменных:

```
int n = 1, m = 3, k;
```

и вызов метода

```
M(<...>);
```

Какой код должен быть поставлен вместо <...>, чтобы при вызове метода в переменную k записалась разница между n и m?

Примечание: при вводе использовать пробелы можно только после модификатора параметра (при необходимости, не более одного подряд).

Метод ответа – ввод ответа.

Тема 3.1 Работа с файлами, как элементами файловой системы

Вопрос 1:

Для чего используется перечисление FileAccess?

Метод ответа – выбор только одного ответа.

- для задания операций, которые доступны при работе с файлом
- для определения, может ли конкретный пользователь получить доступ к файлу
- для определения группы пользователей, которые могут получить доступ к файлу
- для указания, возможна ли с файлом одновременная работа нескольких пользователей

Вопрос 2:

Какой из методов класса File позволяет создать новый файл, или открыть на запись и очистить существующий?

Метод ответа – выбор только одного ответа.

- Create
- Open
- OpenWrite
- AppendText

Вопрос 3:

Что выполняет метод Replace класса File?

Метод ответа – выбор только одного ответа.

- заменяет содержимое одного файла содержимым другого файла
- переименовывает файл
- заменяет в файле заданный фрагмент текста на другой

– переименовывает все файлы в папке по заданному шаблону

Тема 3.2 Работа с файлами, как с потоками

Вопрос 1:

Какое значение вернет метод ReadByte класса FileStream если считывание не произошло?

Метод ответа – выбор только одного ответа.

– -1
– 0
– 1
– 255
– возникнет исключительная ситуация

Вопрос 2:

Каково назначение метода ReadToEnd класса StreamReader?

Метод ответа – выбор только одного ответа.

– считывает все данные до конца потока
– считывает один символ и перемещает позицию в конец потока
– считывает одну строку и перемещает позицию в конец потока
– считывает один байт и перемещает позицию в конец потока
– сначала перемещается в конец потока, а потом считывает последний байт

Вопрос 3:

Каково назначение метода Seek класса FileStream?

Метод ответа – выбор только одного ответа.

– меняет текущую позицию в потоке
– ищет заданные данные в потоке
– усекает поток до заданного размера
– ищет файл с заданным именем