

**Автономная образовательная некоммерческая организация
высшего образования
«Институт Бизнеса и Информационных Систем»
(АОНО ВО «ИБИС»)**

Факультет Бизнеса и информационных систем
Кафедра Информационных технологий



ТВЕРЖДАЮ

Проректор по учебно-воспитательной
и информационной работе

М.В. Доможирова

« 12 » 05 2023 г.

**РАБОЧАЯ ПРОГРАММА
И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

дисциплины

Б1.О.16 «Объектно-ориентированное программирование»

Уровень образования:	<u>Высшее образование – бакалавриат</u>
Направление подготовки:	<u>09.03.02 Информационные системы и технологии</u>
Направленность (профиль):	<u>Информационные системы и сетевые технологии</u>
Форма обучения:	<u>Очная, заочная</u>
Составитель:	<u>к.т.н. Кольцов А.С.</u>

Воронеж 2023 г.

Разработчик рабочей программы дисциплины: к.т.н. Кольцов Андрей Сергеевич

Рабочая программа дисциплины рассмотрена и утверждена на заседаниях:
кафедры «Информационных технологий», протокол №2 от «25» апреля 2023 года.

Ученого совета АОНО «Институт Бизнеса и Информационных Систем», протокол
№ 3 от «11» мая 2023 года.

1. Цели и задачи учебной дисциплины

Цель освоения дисциплины «Объектно-ориентированное программирование»: ознакомление обучающихся с парадигмой и технологией объектно-ориентированного программирования (ООП) и сопутствующих технологий программирования, а также - обучение основам ООП на языке программирования высокого уровня C++.

Задачи дисциплины:

- ознакомление с тенденцией развития программного обеспечения;
- обучение разработке алгоритмов на основе объектно-ориентированного подхода;
- освоение языка программирования C++, а также стандартов кодирования, спецификаций и последующих решений на их основе;
- работа с современными программными инструментальными средствами разработки
- изучение особенностей разработки программного обеспечения под имеющиеся операционные системы, в том числе и мобильные платформы.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» относится к обязательной части дисциплин, базируется на знаниях, полученных обучающимися в процессе освоения дисциплин «Математика», «Информатика». Знания и умения, полученные в ходе изучения дисциплины «Объектно-ориентированное программирование», будут необходимы при изучении дисциплин: Инструментальные средства информационных систем, Современные системы управления базами данных, Технологии тестирования информационных систем, Интеллектуальные информационные системы и технологии

3. Перечень планируемых результатов обучения по дисциплине, соотнесенные с установленными в ОП ВО индикаторами достижения компетенций

Задача профессиональной деятельности	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
Применение современных информационно-коммуникационных технологий процессе осуществления профессиональных функций	ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решения задач профессиональной деятельности	ОПК-2.1 Знать: современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	Знает: - основные концепции объектно-ориентированного программирования, лексические и синтаксические основы объектно-ориентированного языка программирования C++
		ОПК-2.2 Уметь: выбирать современные информационные технологии и программные	Умеет: создавать объектно-ориентированные программы с применением классов

		средства, в том числе отечественного производства, при решении задач профессиональной деятельности.	на языке С++
		ОПК-2.3 Иметь навыки: применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности.	Владеет: - навыками по объектно-ориентированному проектированию и разработке объектно-ориентированного программного кода в современных операционных системах
Разработка и программная реализация прикладных алгоритмов	ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ОПК-6.1 Знать: методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий.	Знает алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения
		ОПК-6.2 Уметь: применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий.	Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули
		ОПК-6.3 Иметь навыки: программирования, отладки и тестирования прототипов программно-технических комплексов задач.	Имеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

4. Объем и структура дисциплины

Общая трудоемкость дисциплины составляет 6 з.е., 216 час.

Вид учебной работы	Формы обучения					
	Очная			Заочная		
	Всего часов	из них в семестре		Всего часов	курс	
2						
Общая трудоемкость дисциплины	216	216		216	216	
Контактная работа обучающихся с преподавателем, всего	90	90		22	22	
в том числе:						
Лекции	36	36		6	6	
Лабораторные работы						
Практические занятия	54	54		16	16	
Самостоятельная работа	90	90		185	185	
Промежуточная аттестация (подготовка и сдача)	36	36		9	9	
Курсовая работа/проект	+	+		+	+	
Контрольная работа	-	-		-	-	
Промежуточная аттестация: экзамен/зачет/зачет с оценкой	экзамен	экзамен		экзамен	экзамен	

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

Содержание тем дисциплины, структурированное по темам с указанием дидактического материала по каждой изучаемой теме

№ п/п	Наименование темы	Содержание темы
1	Тема 1. Линейные вычисления	1) Базовый синтаксис языка (литералы, переменные, константы, идентификаторы, разделители, комментарии) 2) Примитивные типы данных 3) Арифметические операторы 4) Логические операторы (отличие операторов & и от && и) 5) Поразрядные операторы 6) Составные операторы присваивания
2	Тема 2. Управляющие конструкции	1) Блоки кода и время жизни переменных 2) Операторы разветвления вычислений 3) Оператор выбора 4) Операторы организации повторяющихся вычислений
3	Тема 3. Введение в классы	1) Ссылочные типы 2) Правила определения классов 3) Атрибуты класса 4) Методы класса 5) Конструкторы 6) Ключевое слово this 7) Создание экземпляров класса

4	Тема 4. Массивы, строки	<ol style="list-style-type: none"> 1) Объявление и инициализация одномерных массивов 2) Объявление и инициализация многомерных массивов 3) Обработка массивов 4) Определение размеров измерений многомерных массивов 5) Цикл <code>foreach</code> 6) Создание строк 7) Постоянство строк
5	Тема 5. Углубленное изучение классов	<ol style="list-style-type: none"> 1) Перегрузка методов, конструкторов 2) Статические элементы класса 3) Модификаторы доступа 4) Композиция 5) Делегирование 6) Параметры запуска приложения
6	Тема 6. Наследование, интерфейсы	<ol style="list-style-type: none"> 1) Синтаксис наследования 2) Переопределение методов и подмена атрибутов 3) Ключевое слово <code>super</code> 4) Порядок вызовов конструкторов иерархии 5) Особенности интерфейсов 6) Динамическая диспетчеризация типов 7) Класс <code>Object</code> 8) Абстрактные классы
7	Тема 7. Исключительные ситуации	<ol style="list-style-type: none"> 1) Назначение исключительных ситуаций 2) Порядок обработки 3) Объявляемые и не объявляемые исключения 4) оператор <code>try – catch – finally</code> 5) <code>try-resources</code>
8	Тема 8. Коллекции, паттерн итератор	<ol style="list-style-type: none"> 1) Назначение Collections Framework 2) Виды коллекций (и основополагающие интерфейсы) 3) Утилитные классы Collections и Arrays 4) Назначение итераторов 5) Реализация шаблона проектирования «итератор»
9	Тема 9. IO, паттерны Декоратор, Фабрика	<ol style="list-style-type: none"> 1) Принципы работы с вводом-выводом. Понятие потока. 2) Байтовые потоки 3) Символьные потоки 4) Форматный вывод 5) Стандартные потоки ввода, вывода, вывода ошибок
10	Тема 10. Многопоточное программирование	<ol style="list-style-type: none"> 1) Создание потока 2) Синхронизация доступа к объекту различными потоками 3) Примитивы управления потоками 4) Методы <code>wait</code> и <code>notify</code> класса <code>Object</code>

Тематический план (очная форма обучения)

№ п/п	Наименование тем	Всего часов по учебному плану	Контактная работа с преподавателем:					Самостоятельная работа
			Всего часов	Лекции	Занятия семинарского типа			
					Семинарские занятия	Практические занятия	Другие виды занятий	
2 семестр								
1	Тема 1. Линейные вычисления	14	6	2		4		8
2	Тема 2. Управляющие конструкции	18	8	4		4		10
3	Тема 3. Введение в классы	20	10	4		6		10
4	Тема 4. Массивы, строки	20	10	4		6		10
5	Тема 5. Углубленное изучение классов	16	8	4		4		8
6	Тема 6. Наследование, интерфейсы	20	10	4		6		10
7	Тема 7. Исключительные ситуации	18	10	4		6		8
8	Тема 8. Коллекции, паттерн итератор	20	10	4		6		10
9	Тема 9. IO, паттерны Декоратор, Фабрика	18	10	4		6		8
10	Тема 10. Многопоточное программирование	16	8	2		6		8
Итого за семестр		180	90	36		54		90
Форма контроля: Экзамен		36						36
Всего		216	72	36		54		126

Тематический план (заочная форма обучения)

№ п/п	Наименование тем	Всего часов по учебному плану	Контактная работа с преподавателем:					Самостоятельная работа
			Всего часов	Лекции	Занятия семинарского типа			
					Семинарские занятия	Практические занятия	Другие виды занятий	
2 семестр								
1	Тема 1. Линейные вычисления	16	0					16
2	Тема 2. Управляющие конструкции	20	2	2				18
3	Тема 3. Введение в классы	20	2			2		18

4	Тема 4. Массивы, строки	20	2			2		18
5	Тема 5. Углубленное изучение классов	24	4	2		2		20
6	Тема 6. Наследование, интерфейсы	20	2			2		18
7	Тема 7. Исключительные ситуации	22	2			2		20
8	Тема 8. Коллекции, паттерн итератор	24	4	2		2		20
9	Тема 9. IO, паттерны Декоратор, Фабрика	21	2			2		19
10	Тема 10. Многопоточное программирование	20	2			2		18
Форма контроля: Экзамен		9						9
Итого за семестр		216	22	6		16		194

6. Самостоятельная работа обучающихся в ходе освоения дисциплины

№ п/п	Вид самостоятельной работы	Наименование работы и содержание
1	Освоение учебного материала по конспекту лекций и дополнительной литературе	Доработать конспект, желательно в тот же день. Прочитать записи, восстановить текст в памяти, а также исправить опечатки, расшифровать не принятые ранее сокращения, заполнить пропущенные места, понять текст, вникнуть в его смысл. Изучить материал, используя рекомендуемую литературу, разрешая в ходе чтения, возникшие ранее затруднения, находя ответы на вопросы, а также дополняя и исправляя свои записи. Записи должны быть наглядными, для чего следует применять различные способы выделений. Подготовленный конспект и рекомендуемая литература используются при подготовке к практическому занятию.
2	Подготовка к практическим занятиям	Подготовка к практическому занятию включает следующие элементы самостоятельной деятельности: четкое представление цели и задач его проведения; выделение навыков умственной, аналитической деятельности, которые станут результатом предстоящей работы. Выработка навыков осуществляется с помощью получения новой информации об изучаемых процессах и с помощью знания о том, в какой степени в данное время студент владеет методами исследовательской деятельности, которыми он станет пользоваться на практическом занятии.
3	Изучение основной и дополнительной литературы	Самостоятельная работа с учебниками и книгами (а также самостоятельное теоретическое исследование проблем, обозначенных преподавателем на лекциях) – это важнейшее условие познания. В самостоятельной работе рекомендуется прибегать к таким видам систематизированной записи прочитанного как аннотирование, тезирование, цитирование, конспектирование. Причем конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

4	Курсовая работа	<p>Курсовая работа является одним из видов самостоятельной творческой работы обучающихся, выполняемой с целью закрепления полученных теоретических и практических знаний. В ходе подготовки курсовой работы обучающийся самостоятельно знакомится с различными актуальными проблемами теории и практики исследуемой проблематики.</p> <p>Для этого следует изучить монографическую, учебную и методическую литературу, а также нормативную базу по теме исследования. В результате этого систематизируются, закрепляются и расширяются познания обучающегося по избранной теме исследования.</p>
5	Подготовка к экзамену	<p>Необходимо перечитать лекции, вспомнить то, что говорилось преподавателем на семинарах и практических занятиях, а также самостоятельно полученную информацию при подготовке к ним. важно сформировать целостное представление о содержании ответа на каждый вопрос, что предполагает знание разных научных трактовок сущности того или иного явления, процесса, умение раскрывать факторы, определяющие их противоречивость, знание имен ученых, изучавших обсуждаемую проблему. необходимо также привести информацию о материалах эмпирических исследований, что указывает на всестороннюю подготовку обучающегося к экзамену ответ, в котором присутствуют все указанные блоки информации, наверняка будет отмечен высокими баллами. для их получения требуется ответить и на дополнительные вопросы, если экзамен проходит в устной форме.</p> <p>Рекомендуется подготовку к экзамену осуществлять в два этапа. На первом, в течение 2–3 дней, подбирается из разных источников весь материал, необходимый для развернутых ответов на все вопросы. ответы можно записать в виде краткого конспекта. На втором этапе по памяти восстанавливается содержание того, что записано в ответах на каждый вопрос.</p>

7. Фонд оценочных средств для текущей и промежуточной аттестации по дисциплине

Оценочные средства для проведения текущей и промежуточной аттестации

Код контролируемого индикатора освоения компетенции	Наименование оценочного средства для проведения текущей аттестации	Наименование оценочного средства для проведения промежуточной аттестации
ОПК-2.1, ОПК-2.2, ОПК-2.3 ОПК-6.1, ОПК-6.2, ОПК-6.3	Опрос, тестовые задания, курсовая работа, практические работы	экзамен

Критерии оценивания результата обучения по дисциплине и шкала оценивания

Код контролируемой компетенции	Критерии оценивания результата обучения по дисциплине и шкала оценивания			
	неудовлетворительно	удовлетворительно	хорошо	отлично
	Не зачтено	Зачтено		
ОПК-2	обучающийся обнаруживает незнание ответа на соответствующее задание, допускает принципиальные ошибки в формулировке определений и правил, в течение семестра не сформировал необходимых умений и навыков	обучающийся демонстрирует удовлетворительное, но не систематизированное владение принципами работы современных информационных технологий и программных средств, в том числе отечественного производства, и использование их при решения задач профессиональной деятельности	обучающийся демонстрирует достаточно полное, с небольшими неточностями, владение принципами работы современных информационных технологий и программных средств, в том числе отечественного производства, и использование их при решения задач профессиональной деятельности	обучающийся демонстрирует полное, систематизированное владение принципами работы современных информационных технологий и программных средств, в том числе отечественного производства, и использование их при решения задач профессиональной деятельности
ОПК-6	обучающийся обнаруживает незнание ответа на соответствующее задание, допускает принципиальные ошибки в формулировке определений и правил, в течение семестра не сформировал необходимых умений и навыков	обучающийся демонстрирует удовлетворительное, но не систематизированное владение способностями разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	обучающийся демонстрирует достаточно полное, с небольшими неточностями, владение способностями разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	обучающийся демонстрирует полное, систематизированное владение способностями разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

8. Ресурсное обеспечение учебной дисциплины

Основная литература:

1. Объектно-ориентированное программирование на C++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. - Красноярск : Сиб. федер. ун-т, 2019. - 288 с. - ISBN 978-5-7638-4034-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1819676> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

2. Ашарина, И.В. Объектно-ориентированное программирование в C++: лекции и упражнения : учеб. пособие для вузов / И.В. Ашарина. - 2-е изд., перераб. и доп. - Москва : Горячая линия - Телеком, 2017. - 336 с. - ISBN 978-5-9912-0423-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1040247> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

Дополнительная литература:

3. Игнаткин, А. А. Объектно-ориентированное программирование : курс лекций / А. А. Игнаткин. - Москва : ИД МИСиС, 2005. - 149 с. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1232338> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

4. Романенко, В. В. Объектно-ориентированное программирование : учебное пособие / В. В. Романенко. - Томск : Томский государственный университет систем управления и радиоэлектроники, 2014. - 475 с. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1845888> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

5. Шакин, В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio .NET : учебное пособие / В. Н. Шакин, А. В. Загвоздкина, Г. К. Сосновиков. — Москва : ФОРУМ : ИНФРА-М, 2019. — 398 с. — (Высшее образование. Бакалавриат). - ISBN 978-5-00091-048-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1010028> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

6. Гуськова, О.И. Объектно ориентированное программирование в Java : учебное пособие / О. И. Гуськова. - Москва : МПГУ, 2018. - 240 с. - ISBN 978-5-4263-0648-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1020593> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

7. Шакин, В. Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio.NET : учебное пособие / В.Н. Шакин, А.В. Загвоздкина, Г.К. Сосновиков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 398 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-551-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1082462> (дата обращения: 04.12.2021). – Режим доступа: по подписке.

Электронные ресурсы:

1. Система вопросов и ответов о программировании. – Режим доступа: <https://stackoverflow.com>

2. Программирование на языке C++ - онлайн-курс – Режим доступа: <https://stepik.org/course/7/promo>

3. Программирование на языке C++ (продолжение) - онлайн-курс – Режим доступа: <https://stepik.org/course/3206/promo>

4. Российское образование. Федеральный портал.– Режим доступа: <http://www.edu.ru>

5. Электронно-библиотечная система «eLibrary»– Режим доступа: <http://www.elibrary.ru>

9. Материально-техническое обеспечение дисциплины

<p>Учебная аудитория № 315</p> <ul style="list-style-type: none">-учебная аудитория для проведения занятий лекционного типа;-учебная аудитория для проведения занятий семинарского тип и практических занятий;-учебная аудитория групповых и индивидуальных консультаций;-учебная аудитория для проведения текущего контроля и промежуточной аттестации. <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none">- комплект учебной мебели для обучающихся;- рабочее место преподавателя;- доска меловая;- стационарное видеопроекторное оборудование для мультимедиа презентации, средства звуковоспроизведения (проектор, персональный компьютер, колонки, Web-камера). <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none">- MS Windows 10;- Microsoft Office Standard 2007. <p>2) отечественного производства:</p> <ul style="list-style-type: none">- Kaspersky EndPoint Security для Windows. <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	<p>394026, Воронежская область, г. Воронеж, ул. Дружинников, д.8 Кабинет № 315 (3 этаж № 12)</p>
<p>Учебная аудитория № 314</p> <ul style="list-style-type: none">-учебная аудитория для проведения занятий лекционного типа;-учебная аудитория для проведения занятий семинарского тип и практических занятий;-учебная аудитория групповых и индивидуальных консультаций;-учебная аудитория для проведения текущего контроля и промежуточной аттестации;-учебная аудитория для курсового проектирования (выполнения курсовых работ);-помещение для самостоятельной работы обучающихся;- компьютерный класс. <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none">-автоматизированное рабочее место обучающегося;-автоматизированное рабочее место преподавателя;-доска двусторонняя (маркерно - меловая);-наушники;-принтер;-телевизор. <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none">- MS Windows 8.1 Корпоративная;- Microsoft Office Standard 2007;- iSpring suite 8;- MS Visio;- MS Access 2016;- MS Project;- Microsoft SQL Server 2014;	<p>394026, Воронежская область, г. Воронеж, ул. Дружинников, д.8 Кабинет № 314 (3 этаж № 48)</p>

<p>- Visual Studio 2017.</p> <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - Kaspersky EndPoint Security для Windows; - 1С: Предприятия 8. Комплект для обучения в высших и средних учебных заведениях. <p>Свободно распространяемое программное обеспечение иностранного производства:</p> <ul style="list-style-type: none"> - PascalABC.NET; - FreePascal IDE; - Eclipse; - IntelliJ IDEA; - GIMP; - Blender; - Firefox; - Vuze; - FileZilla; - Denver, Maxima + WxMaxima; - iTest; - Inkscape; - QCad. <p>Информационная справочная правовая система «Консультант Плюс».</p> <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	
<p>Учебная аудитория № 318</p> <ul style="list-style-type: none"> -учебная аудитория для проведения занятий лекционного типа; -учебная аудитория для проведения занятий семинарского типа и практических занятий; -учебная аудитория групповых и индивидуальных консультаций; -учебная аудитория для проведения текущего контроля и промежуточной аттестации; -учебная аудитория для курсового проектирования (выполнения курсовых работ); -компьютерный класс; -помещение для самостоятельной работы обучающихся. <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none"> -автоматизированное рабочее место обучающегося; -автоматизированное рабочее место преподавателя; -доска двусторонняя (маркерно-меловая). - переносное видеопроекторное оборудование для мультимедиа презентации (ноутбук, проектор, экран, колонки). <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - MS Windows 7; - Microsoft Office Standard 2007; - MS Visio 2007; - MS Project 2010; - Microsoft SQL Server 2012; - Microsoft Visual Studio. <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - Kaspersky EndPoint Security для Windows; 	<p>394026, Воронежская область, г. Воронеж, ул. Дружинников, д.8 Кабинет № 318 (3 этаж № 50)</p>

<p>- Автоматизированная банковская система «Управление кредитной организацией» для ВУЗов.</p> <p>Свободно распространяемое программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - PascalABC.NET; - FreePascal IDE; - GIMP; - Blender; - Firefox; - Vuze; - FileZilla; - Denver; - Maxima + WxMaxima; - iTest; - Inkscape; - QCad; <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - программа Фоторобот. <p>Российская информационная справочная правовая система «Консультант Плюс».</p> <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	
<p>Помещение для самостоятельной работы обучающихся № 102</p> <ul style="list-style-type: none"> - помещение для самостоятельной работы обучающихся с доступом к сети «Интернет» и электронной информационно-образовательной среде организации; - читальный зал библиотеки - учебная аудитория для курсового проектирования (выполнения курсовых работ); - учебная аудитория для выполнения и защиты выпускной квалификационной работы. <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none"> - автоматизированное рабочее место обучающегося; - ноутбуки; - телевизор; - столы для чтения; - стулья; - шкафы для документов; - стол офисный; - стеллажи для книг; - стойка выдачи литературы; - тумба напольная; - информационная стойка. <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - MS Windows 7 pro; - Microsoft Office Standard 2007; - MS Access 2016. <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - Kaspersky EndPoint Security для Windows; Свободно распространяемое программное обеспечение: 	<p>394026, Воронежская область, г. Воронеж, ул. Дружинников, д.8 Кабинет № 102 (1 этаж № 84)</p>

<p>- 7-Zip; - Интернет цензор. Российская информационная справочная правовая система «Консультант Плюс». Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации</p>	
<p>Учебная аудитория № 314 - помещение для самостоятельной работы обучающихся с доступом к сети «Интернет» и электронной информационно-образовательной среде организации; -учебная аудитория для курсового проектирования (выполнения курсовых работ); -учебная аудитория для выполнения выпускной квалификационной работы; - компьютерный класс. Оснащение оборудованием и техническими средствами обучения: -автоматизированное рабочее место обучающегося; - автоматизированное рабочее место преподавателя; -доска двусторонняя (маркерно - меловая); -наушники; -принтер; -телевизор. Лицензионное программное обеспечение: 1) иностранного производства: - MS Windows 8.1 Корпоративная; - Microsoft Office Standard 2007; - iSpring suite 8; - MS Visio; - MS Access 2016; - MS Project; - Microsoft SQL Server 2014; - Visual Studio 2017. 2) отечественного производства: - Kaspersky EndPoint Security для Windows; -1С: Предприятия 8. Комплект для обучения в высших и средних учебных заведениях. Свободно распространяемое программное обеспечение иностранного производства: - PascalABC.NET; - FreePascal IDE; - Eclipse; - IntelliJ IDEA; - GIMP; - Blender; - Firefox; - Vuze; - FileZilla; - Denver, Maxima + WxMaxima; - iTest; - Inkscape; - QCad.</p>	<p>394026, Воронежская область, г. Воронеж, ул. Дружинников, д.8 Кабинет № 314 (3 этаж № 48)</p>

<p>Информационная справочная правовая система «Консультант Плюс».</p> <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	
<p>Учебная аудитория № 318</p> <ul style="list-style-type: none"> - помещение для самостоятельной работы обучающихся с доступом к сети «Интернет» и электронной информационно-образовательной среде организации; - учебная аудитория для курсового проектирования (выполнения курсовых работ); - учебная аудитория для выполнения выпускной квалификационной работы; <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none"> - автоматизированное рабочее место обучающегося; - автоматизированное рабочее место преподавателя; - доска двусторонняя (маркерно-меловая); - переносное видеопроекционное оборудование для мультимедиа презентации (ноутбук, проектор, экран, колонки). <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - MS Windows 7; - Microsoft Office Standard 2007; - MS Visio 2007; - MS Project 2010; - Microsoft SQL Server 2012; - Microsoft Visual Studio. <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - Kaspersky EndPoint Security для Windows; - Автоматизированная банковская система «Управление кредитной организацией» для ВУЗов. <p>Свободно распространяемое программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - PascalABC.NET; - FreePascal IDE; - GIMP; - Blender; - Firefox; - Vuze; - FileZilla; - Denver; - Maxima + WxMaxima; - iTest; - Inkscape; - QCad; <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - программа Фоторобот. <p>Российская информационная справочная правовая система «Консультант Плюс».</p> <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	<p>394026, Воронежская область, г. Воронеж, ул. Дружинников, д.8 Кабинет № 318 (3 этаж № 50)</p>

<p>Учебная аудитория № 313</p> <ul style="list-style-type: none"> - помещение для самостоятельной работы обучающихся с доступом к сети «Интернет» и электронной информационно-образовательной среде организации; -учебная аудитория для курсового проектирования (выполнения курсовых работ); -учебная аудитория для выполнения выпускной квалификационной работы; - компьютерный класс. <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none"> -автоматизированное рабочее место обучающегося; -автоматизированное рабочее место преподавателя; -доска маркерная; - стационарное видеопроекторное оборудование для мультимедиа презентации, средства звуковоспроизведения (экран, проектор, колонки). <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - MS Windows 10; - Microsoft Office Standard 2007; - MS Visio; - MS Access 2016; - MS Project; - Microsoft SQL Server 2019; - Visual Studio 2010; <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - Kaspersky EndPoint Security для Windows. <p>Свободно распространяемое программное обеспечение иностранного производства:</p> <ul style="list-style-type: none"> - PascalABC.NET; - FreePascal IDE; - Eclipse; - IntelliJ IDEA; - GIMP; - Blender; - Firefox; - Vuze; - FileZilla; - Denver; - Maxima + WxMaxima, iTest; - Inkscape; - QCad. <p>Российская информационная справочная правовая система «Консультант Плюс».</p> <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	<p>394036, город Воронеж, ул. Карла Маркса, д.67 Кабинет № 313 (3 этаж № 62)</p>
<p>Учебная аудитория № 314</p> <ul style="list-style-type: none"> - помещение для самостоятельной работы обучающихся с доступом к сети «Интернет» и электронной информационно-образовательной среде организации; -учебная аудитория для курсового проектирования (выполнения 	<p>394036, город Воронеж, ул. Карла Маркса, д.67 Кабинет № 314</p>

<p>курсовых работ);</p> <ul style="list-style-type: none"> - учебная аудитория для выполнения выпускной квалификационной работы; - компьютерный класс. <p>Оснащение оборудованием и техническими средствами обучения:</p> <ul style="list-style-type: none"> - автоматизированное рабочее место обучающегося; - автоматизированное рабочее место преподавателя; - доска двусторонняя (маркерно - меловая); - наушники; - принтер; - телевизор. <p>Лицензионное программное обеспечение:</p> <p>1) иностранного производства:</p> <ul style="list-style-type: none"> - MS Windows 8.1 Корпоративная; - Microsoft Office Standard 2007; - iSpring suite 8; - MS Visio; - MS Access 2016; - MS Project; - Microsoft SQL Server 2014; - Visual Studio 2017. <p>2) отечественного производства:</p> <ul style="list-style-type: none"> - Kaspersky EndPoint Security для Windows; - 1С: Предприятия 8. Комплект для обучения в высших и средних учебных заведениях. <p>Свободно распространяемое программное обеспечение иностранного производства:</p> <ul style="list-style-type: none"> - PascalABC.NET; - FreePascal IDE; - Eclipse; - IntelliJ IDEA; - GIMP; - Blender; - Firefox; - Vuze; - FileZilla; - Denver, Maxima + WxMaxima; - iTest; - Inkscape; - QCad. <p>Информационная справочная правовая система «Консультант Плюс».</p> <p>Доступ к информационно-телекоммуникационной сети «Интернет» и к электронной информационно-образовательной среде организации.</p>	<p>(3 этаж № 61)</p>
--	----------------------

10. Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

10.1 Материалы для текущего контроля освоения дисциплины

Тема 1. Линейные вычисления

Средства текущего контроля для проведения опроса:

1. Основные концепции языков программирования.
2. Классификация языков программирования.
3. Императивные языки.
4. Языки декларативного программирования.
5. Объектно – ориентированные языки.
6. Критерии языков программирования.

Практическая работа 1. Программирование с использованием функций

Цель работы: освоить технологию программирования с использованием функций

Вопросы и задания по практической работе 1

1. Ввести и обработать:
 - 1) два двумерных массива, содержащие соответственно 3×5 и 4×8 вещественных элементов;
 - 2) три массива, содержащие соответственно 3, 6 и 8 целых элементов без знака;
 - 3) четыре массива, содержащие соответственно 4, 6, 3 и 5 целых элементов со знаком;
 - 4) два массива, содержащие соответственно 4 и 6 вещественных элементов;
 - 5) три массива, содержащие соответственно 5, 10 и 4 целых элементов без знака;
 - 6) четыре массива, содержащие соответственно 3, 5, 8 и 6 вещественных элементов;
 - 7) два трехмерных массива, содержащие соответственно $2 \times 3 \times 2$ и $3 \times 4 \times 2$ вещественных элементов;
 - 8) четыре массива, содержащие соответственно 4, 7, 3 и 5 вещественных элементов;
 - 9) три двумерных массива, содержащие соответственно 2×5 , 3×6 и 3×4 целых элементов без знака;
 - 10) два двумерных массива, содержащие соответственно 6×2 и 3×2 вещественных элементов.
2. Для чего предназначены функции?
3. Чем отличаются формальные и фактические параметры?
4. Чем отличаются глобальные и локальные переменные?

Тема 2. Управляющие конструкции

Средства текущего контроля для проведения опроса:

1. Императивные языки.
2. Структурное программирование.
3. Функции языка.
4. Передача параметров в функцию.
5. Побочный эффект.

Практическая работа 2. Программирование с использованием шаблонов

Цель работы: Освоить технологию программирования с использованием шаблонов

Вопросы и задания по практической работе 2

1. В чем смысл использования шаблонов?
2. Синтаксис/семантика шаблонов функций?
3. Синтаксис/семантика шаблонов классов?
4. Определить параметризованную функцию сортировки массива методом обмена.
5. Определить шаблон класса “вектор” одномерный массив.
6. Что такое параметры шаблона функции?
7. Перечислите основные свойства параметров шаблона функции.

9. Можно ли перегружать параметризованные функции?
10. Перечислите основные свойства параметризованных классов.
11. Может ли быть пустым список параметров шаблона? Объясните.
12. Как вызвать параметризованную функцию без параметров?
13. Все ли компонентные функции параметризованного класса являются параметризованными?
14. Являются ли дружественные функции, описанные в параметризованном классе, параметризованными?
15. Могут ли шаблоны классов содержать виртуальные компонентные функции?
16. Как определяются компонентные функции параметризованных классов вне определения шаблона класса?
17. Каковы синтаксис/семантика “операции-функции”?
18. Когда нужно перегружать операцию присваивания для определенного пользователя типа данных, например класса?
19. Можно ли изменить приоритет перегруженной операции?
20. Можно ли изменить количество операндов перегруженной операции?
22. Можно ли, используя дружественную функцию, перегрузить оператор присваивания?
23. Все ли операции языка могут быть перегружены?
25. Все ли операции можно перегрузить с помощью глобальной дружественной функции?
26. В каких случаях операцию можно перегрузить только глобальной функцией?
27. В каких случаях глобальная операция-функция должна быть дружественной?

Задания: При тестировании созданных шаблонов классов необходимо создавать объекты с различными допустимыми значениями параметров шаблона (например, компоненты вектора могут быть целыми, действительными или комплексными числами).

1. Создать шаблон класса для работы со стеком.
2. Создать шаблон класса для работы с одномерным массивом. Выполнить тестирование путем создания и обработки массивов, содержащих элементы различных типов (например, сортировка элементов массивов различными методами).
3. Создать шаблон класса `Vector` размерности n
4. Создать шаблон класса «Квадратная матрица» – `Matrix` размерности $n \times n$
5. Создать шаблон класса `Polynom` степени n
6. Создать шаблон класса для работы с односвязным списком.
7. Создать шаблон класса для работы с односвязным списком.
8. Создать шаблон класса для работы с бинарным деревом.
9. Создать шаблон класса `Set` (множество) мощности n
10. Создать шаблон класса для работы с односвязным списком.
11. Создать шаблон класса, обеспечивающего описание матрицы заданного размера $n \times m$ и любого минора в ней
12. Создать шаблон класса для работы с бинарным деревом.
13. Создать шаблон класса для работы с бинарным деревом.
14. Создать шаблон класса для работы с двусвязным списком.

Тема 3. Введение в классы

Средства текущего контроля для проведения опроса:

1. Классы и объекты.
2. Конструкторы.
3. Деструкторы.
4. Статические поля и методы.
5. Наследование.
6. Дружественные классы и функции.
7. Наследование.

8. Виртуальные методы.
9. Перегрузка операций.

Практическая работа 3. Композиция классов и объектов.

Цель работы: Получить практические навыки реализации классов

Вопросы по практической работе 3

1. Что такое класс?
2. Что такое объект?
3. Как связаны между собой классы и объекты в программе?
4. Что такое инкапсуляция?
5. За счет чего реализуется защита от несанкционированного доступа к данным?
6. Чем отличаются поля от переменных?
7. Что такое свойство?
8. Какие методы есть у свойства?
9. Что делают эти методы?
10. Что такое точка входа?

Тема 4. Формальные грамматики и языки.

Средства текущего контроля для проведения опроса:

1. Языки и цепочки символов.
2. Операции над цепочками.
3. Понятие языка.
4. Формальное определение языка.
5. Грамматики.
6. Классификация грамматик.
7. Вывод.
8. Сентенциальная форма.
9. Дерево вывода.
10. Проблемы однозначности и эквивалентности грамматик.
11. Распознаватели и их классификация.

Практическая работа 4. Массивы, строки

Цель работы: изучить возможности программирования классов на языке и получить основные навыки программирования

Вопросы и задания по практической работе 4

1. Какие классы и функции называются дружественными?
2. Как осуществляется перегрузка операций?
3. Сколько аргументов требуется для определения перегруженной унарной (бинарной) операции?

Тема 5. Углубленное изучение классов

Средства текущего контроля для проведения опроса:

1. Линейные списки.
2. Стеки.
3. Использование стеков для проверки правильности расстановки скобок в выражении.
4. Очереди.
5. Применение бинарных деревьев для сортировки и поиска.
6. Циклические списки.

7. Задача Джозефуса.
8. Бинарные деревья

Практическая работа 5. Наследование, виртуальные методы.

Цель: Получить практические навыки создания иерархии классов и использования статических компонентов класса.

Вопросы и задания по практической работе 5

1. Создать базовый класс Car(машина), характеризуемый торговой маркой (строка), числом цилиндров, мощностью. Определить методы переназначения и изменения мощности. Создать производный класс Loggy (грузовик), характеризуемый также грузоподъемностью кузова. Определить функции переназначения марки и изменения грузоподъемности.
2. Создать класс Pair(пара чисел); определить методы изменения полей и сравнения пар: $\text{par.p1} > \text{par.p2}$, если $(\text{first.p1} > \text{first.p2})$ или $(\text{first.p1} = \text{first.p2})$ и $(\text{second.p1} > \text{second.p2})$. Определить класс -наследник Fractionc полями: целая часть числа и дробная часть числа. Определить полный набор методов сравнения.
3. Создать класс Liquid (жидкость), имеющий поля названия и плотности. Определить методы переназначения и изменения плотности. Создать производный класс Alcohol(спирт), имеющий крепость. Определить методы переназначения и изменения крепости.
4. Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс Rectangle (прямоугольник) с полями сторонами. Определить методы вычисления периметра и площади прямоугольника.
5. Создать класс Man (человек), с полями: имя, возраст, пол и вес. Определить методы переназначения имени, изменения возраста и изменения веса. Создать производный класс Student, имеющий поле года обучения. Определить методы переназначения и увеличения года обучения.
6. Создать класс Triad (тройка чисел); определить методы изменения полей и вычисления суммы чисел. Определить производный класс Triangle с полями-сторонами. Определить методы вычисления углов и площади треугольника.
7. Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс Equilateral (равносторонний), имеющий поле площади. Определите метод вычисления площади.
8. Создать класс Triangle спольями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс RightAngled (прямоугольный), имеющий поле площади. Определить метод вычисления площади.
9. Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс RightAngled с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника.
10. Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Date с полями: год, месяц и день. Определить полный набор методов сравнения дат.
11. Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Time с полями: час, минута и секунда. Определить полный набор методов сравнения моментов времени.
12. Реализовать класс-оболочку Number для числового типа float. Реализовать методы сложения и деления. Создать производный класс Real, в котором реализовать метод возведения в произвольную степень, и метод для вычисления логарифма числа.
13. Создать класс Triad (тройка чисел); определить методы увеличения полей на 1. Определить производный класс Date с полями: год, месяц и день. Переопределить методы увеличения полей на 1 и определить метод увеличения даты на n дней.

14. Реализовать класс-оболочку Number для числового типа double. Реализовать методы умножения и вычитания. Создать производный класс Real, в котором реализовать метод, вычисляющий корень произвольной степени, и метод для вычисления числа π в данной степени.

15. Создать класс Triad (тропка чисел); определить методы увеличения полей на 1. Определить класс-наследник Time с полями: час, минута, секунда. Переопределить методы увеличения полей на 1 и определить методы увеличения на n секунд и минут.

1. Определить иерархию классов (в соответствии с вариантом).

2. Определить в классе статическую компоненту указатель на начало связанного списка объектов и статическую функцию для просмотра списка.

3. Реализовать классы.

4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается. 5. Сделать соответствующие методы не виртуальными и посмотреть, что будет.

6. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

Тема 6. Наследование, интерфейсы

Средства текущего контроля для проведения опроса:

1. Расскажите об известных Вам последних разработках языков программирования. Приведите основные характеристики названных языков.

2. Приведите конкретные примеры использования методов трансляции в областях, не связанных с языками программирования.

3. Приведите конкретные примеры компилируемых языков программирования.

4. Приведите конкретные примеры интерпретируемых языков программирования.

5. Приведите конкретные примеры языков программирования, для которых имеются как компиляторы, так и интерпретаторы.

6. Основные достоинства и недостатки компиляторов.

7. Основные достоинства и недостатки интерпретаторов.

8. Опишите основные различия в синтаксисе двух известных Вам языков программирования.

9. Опишите основные различия в семантике двух известных Вам языков программирования.

10. Назовите основные фазы процесса трансляции и их назначение.

11. Назовите специфические особенности однопроходной трансляции.

12. Назовите специфические особенности многопроходной трансляции.

13. Приведите примеры возможных комбинаций однопроходной и многопроходной трансляции. Расскажите о практическом использовании этих схем.

Практическая работа 6. Динамические структуры данных.

Цель: освоить понятия динамических структур данных

Вопросы по практической работе 6

1. Манипулирование адресами области памяти с помощью указателей

2. Разработка и реализация программ с файлами данных

3. Общая схема работы транслятора.

4. Интерпретаторы.

5. Ассемблеры.

6. Понятие прохода.

7. Однопроходные и многопроходные компиляторы. Этапы трансляции.

Тема 7. Исключительные ситуации

Средства текущего контроля для проведения опроса:

1. Принципы организации таблицы идентификаторов.

2. Простейшие способы построения таблицы идентификаторов: простой и упорядоченный список, бинарное дерево.
3. Хэш – функция и хэш – адресация.
4. Рехэширование.
5. Метод цепочек.
6. Комбинированные методы построения таблицы идентификаторов.

Практическая работа 7. Проектирование таблицы идентификаторов.

Цель: Освоить технологию проектирование таблиц идентификаторов

Вопросы по практической работе 7

1. Динамическое программирование
2. Разработка рекурсивных алгоритмов

Тема 8. Коллекции, паттерн итератор

Средства текущего контроля для проведения опроса:

1. Регулярные языки и грамматики.
2. Конечные автоматы.
3. Детерминированные и недетерминированные конечные автоматы.
4. Минимизация конечных автоматов.
5. Регулярные множества и регулярные выражения.
6. Эквивалентность трех способов задания регулярных языков
7. Чем отличаются таблица лексем и таблица идентификаторов?
8. В какую из этих таблиц лексический анализатор не должен помещать ключевые слова, разделители и знаки операций?

Практическая работа 8. Лексический анализатор.

Цель: Научиться строить и создавать лексический анализатор

Вопросы по практической работе 8

1. Реализация рекурсивных структур.
2. Применение рекурсивных алгоритмов для различных сортировок
3. Какую роль выполняет лексический анализ в процессе компиляции?
4. Как связаны лексический и синтаксический анализ?
5. Какие проблемы необходимо решить при построении лексического анализатора на основе конечного автомата?

Тема 9. Ю, паттерны Декоратор, Фабрика

Средства текущего контроля для проведения опроса:

1. Контекстно свободные грамматики и языки.
2. Автоматы с магазинной памятью.
3. Приведенные грамматики.
4. Синтаксические распознаватели с возвратом.
5. Нисходящие распознаватели.
6. Восходящие распознаватели
7. Синтаксические распознаватели на основе грамматик предшествования.
8. Грамматики операторного предшествования.

Практическая работа 9. Синтаксический анализатор.

Цель: Рассмотреть и научиться работать с синтаксическим анализатором

Вопросы по практической работе 9

1. Модульная структура программы.
2. Отладка модулей
3. Какие существуют методы задания языков? Сравните их по эффективности.

4. Что такое грамматика языка?
5. Какие существуют формы описания грамматик?
6. Какие типы языков выделяются по классификации Хомского? Как они соотносятся между собой?
7. Что такое сентенциальная форма грамматики?
8. Что является предложением языка?

Тема 10. Многопоточное программирование

Средства текущего контроля для проведения опроса:

1. Распределение памяти, его принципы.
2. Виды переменных и областей памяти.
3. Статическое и динамическое связывание.
4. Генерация кода, ее общие принципы.
5. Синтаксически управляемый перевод.
6. Способы внутреннего представления программы.
7. Многоадресный код с явно именуемым результатом –тетрады.
8. Многоадресный код с неявно именуемым результатом – триады.
9. Обратная польская инверсная запись операций.

Практическая работа 10. Оптимизирующий генератор кода.

Цель: Рассмотреть и освоить построение оптимизирующего генератора кода

Вопросы по практической работе 10

1. Что такое транслятор, компилятор и интерпретатор?
2. Расскажите об общей структуре компилятора.
3. Как строится дерево вывода (синтаксического разбора)? Какие исходные данные необходимы для его построения?
4. Объясните работу алгоритма генерации объектного кода по дереву синтаксического разбора.
5. За счет чего обеспечивается возможность генерации кода на разных объектных языках по одному и тому же дереву?
6. Какую роль выполняет генерация объектного кода в процессе компиляции?
7. Какие данные необходимы компилятору для генерации объектного кода? Какие действия выполняет компилятор перед генерацией?
8. Дайте определение понятию оптимизации программы. Для чего используется оптимизация?
9. Какие существуют методы оптимизации объектного кода?
10. Что такое триады и для чего они используются? Какие еще существуют методы для представления объектных команд?
11. Объясните работу алгоритма свертки.
12. Что такое лишняя операция? Что такое «число зависимости»?
13. Объясните работу алгоритма исключения лишних операций.

Тестовые задания

1. С помощью чего реализуется принцип полиморфизма в C++?
наличия множественного наследования.
- наличия виртуальных методов. +
-использование виртуального наследования.
- наличия абстрактных классов.
2. В программе описано класс и объект `class A {public: int a, b, c; }; A * obj;` Как обратиться к атрибуту `c`?
`obj.c`
`obj->c`
`obj.A->c`
`obj->A.c`
3. Какая из перечисленных функций не может быть конструктором?
`void String ()` +
`String ();`
`String (String & s)`
`String (const int a)`
4. Отметьте правильное утверждение для абстрактного класса для языка C++.
Класс, у которого все методы чисто виртуальные, называется абстрактным.
Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.
Невозможно создать объект абстрактного класса. +
В абстрактном классе не описываются методы вообще.
5. Если в программе на языке C++ в производном классе переопределена операция `new` то ...
 - все объекты этого класса и все объекты классов, выведенных из него, будут использовать эту операцию независимо от зоны видимости, в которой она переопределена.
 - производные от этого класса могут использовать глобальную операцию применив операцию базовый_класс :: new. +
 - операцию `new` нельзя переопределить.
 - в любом случае эта операция будет доступна только в пределах классапотомка.
6. Какой из перечисленных методов может быть конструктором для класса `String` в языке C++?
`String * String ();`
`void String ();`
`String (String & s);` +
`const String (int a);`
7. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам?
Шаблонная.
Полиморфная.
Дружеская. +
Статическая.
8. Вызовет данный код ошибку компиляции? `class Rectangle public: int a, b; int sum (); int square (); ~Rect ();`};
Ошибки нет, все записано верно.
Ошибка является: имя деструктора должно совпадать с именем класса. +
Ошибка является: имя деструктора не может начинаться с маленькой буквы.
Ошибка является: никакой идентификатор в C++ не может начинаться со знака «~».

9 Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void.

```
virtual void SomeFunction (int x); +  
void SomeFunction (int x) virtual;  
virtual SomeFunction (int x);  
virtual void SomeFunction (int * x);
```

10. Укажите правильное использование оператора friend.

```
class A {int_friend CountPass (); private: short i;};  
class A {public: friend int H :: CountPass (); private: short i;}; +  
class A {public: int A1 :: CountPass (); friend: short i;};  
class A {public: friend int H :: q; short i;};
```

11. Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

Наследование.

Сочетание.

Инициализация.

Инкапсуляция. +

12 В программе описано абстрактный класс A и производный от этого класса класс A1. Какой из записей заведомо неверный?

```
A * a = new A; +
```

```
A1 a1;
```

```
A1 a1; A & a = a1;
```

```
A1 a1; A1 a2;
```

13. Выберите верное утверждение о деструктор класса в C ++.

Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.

Деструктор принимает в качестве параметра указатель this.

Деструктор не содержит параметров. +

Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

14. Укажите правильный вариант доступа к членам объектов (язык C ++), описанных следующим образом: class my { char s;public: double Z; int f (int c, int d)

```
{return c + d;}; } T1, T2;
```

```
T1.Z = 23.1; +
```

```
T2-> f (2,1);
```

```
T1.s = '#';
```

```
my.T2-> s = 'L';
```

15 Какой из вариантов записи абстрактного класса в C ++ является правильным?

```
abstract class A {virtual int f () = 0;};
```

```
class A {virtual int f () = 0;}; +
```

```
class A {virtual int f () = 0;} abstract;
```

```
class A {virtual int f ();};
```

16. Какое из свойств скрывает внутренние данные объекта?

1) Инкапсуляция

2) Полиморфизм

3) Наследование

4) Объектно-Ориентированные

17. Какое из свойств использует виртуальные или перегружаемые элементы?

1) Инкапсуляция

- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

18. Какое из свойств строит иерархию объектов?

- 1) Инкапсуляция
- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

19. Какое из свойств предназначено для улучшения интерфейса работы с объектами?

- 1) Инкапсуляция
- 2) Полиморфизм
- 3) Наследование
- 4) Объектно-Ориентированные

20. Какая из операций используется для обозначения указателя?

- 1) DIV
- 2) *
- 3) {\$
- 4) ^

21. Какое из названий обозначает завершение работы динамических методов объекта?

- 1) свойство
- 2) виртуальная функция
- 3) конструктор
- 4) деструктор

22. Какое из названий обозначает общедоступные элементы объекта?

- 1) public
- 2) published
- 3) protected
- 4) private

23. Какое из названий обозначает доступные только в модуле элементы класса?

- 1) public
- 2) published
- 3) protected
- 4) private

24. Какая из операций обозначает получение адреса?

- 1) @
- 2) &
- 3) ^
- 4) *

25. В какой из областей класса элементы недоступны для потомков вне данного модуля?

- 1) public
- 2) published
- 3) protected
- 4) private

26. Какой из терминов обозначает события объекта?

- 1) Canvas
- 2) Caption
- 3) Events
- 4) Enabled

27. Какое из свойств связано с соединением полей, методов и свойств в одном объекте?

- 1) Инкапсуляция
- 2) Наследование
- 3) Полиморфизм
- 4) Визуальность

28. Какое из названий обозначает создание объекта данного класса?

- 1) переопределение функции
- 2) виртуальная функция
- 3) конструктор
- 4) деструктор

29. Динамическая структура, которая имеет две основные операции: добавление в «хвост» и извлечение из «головы» является

- 1) очередью
- 2) стеком
- 3) списком
- 4) файлом

30. Динамическая структура, которая имеет одну точку доступа к его элементам («голова»), называется

- 1) очередью
- 2) стеком
- 3) списком
- 4) файлом

31. Упорядоченная динамическая структура, каждый элемент которой содержит ссылку, связывающую его со следующим элементом, называется

- 1) очередью
- 2) стеком
- 3) списком
- 4) файлом

32. Что будет выведено в результате выполнения программы?

- 1 class Program
- 2 {
- 3 static void Main(string[] args)

```

4.     {
5.     try
6.     {
7.         var a1 = new A1();
8.         Console.Write("1");
9.         var a2 = new A2();
10.    }
11.    catch (Exception)
12.    {
13.        Console.Write("3");
14.    }
15.    Console.ReadLine();
16.    }
17. }
18.
19. class A1
20. {
21.     public int B1;
22. }
23.
24. class A2
25. {
26.     public int B2;
27.
28.     public A2(int b2)
29.     {
30.         B2 = b2;
31.     }
32. }
33. 1) 1
34. 2) 3
35. 3) 13
36. 4) Возникнет ошибка на этапе компиляции

```

33. Что будет выведено в результате выполнения программы?

```

1.     class Program
2.     {
3.         static void Main(string[] args)
4.         {
5.             Console.Write(A.B);
6.             var a1 = new A();
7.             Console.Write(A.B);
8.             a1.Write();
9.             Console.ReadLine();
10.        }
11.
12.    public class A
13.    {
14.        public static int B;
15.
16.        public A()
17.        {
18.            B = 3;
19.        }

```

```
20.     }
21.
22.     public void Write()
23.     {
24.         Console.Write(B);
25.     }
26.
27.     static A()
28.     {
29.         B = 5;
30.     }
31. }
```

- 1) 033
- 2) 533
- 3) 553
- 4) Возникнет ошибка на этапе компиляции

34. Что будет выведено в результате выполнения программы?

```
1.     class Program
2.     {
3.         static void Main(string[] args)
4.         {
5.             var a = new A(2) {B = 3};
6.             Console.Write(a.B);
7.             Console.ReadLine();
8.         }
9.     }
10.
11.     public class A
12.     {
13.         public int B { get; set; }
14.
15.         public A(int b)
16.         {
17.             Console.Write("1");
18.             B = b;
19.         }
20.     }
```

- 1) 3
- 2) 12
- 3) 13
- 4) Возникнет ошибка на этапе компиляции

35. Какие утверждения относительно языка C# верны?

- 1) Допустимо множественное наследование
- 2) Класс может реализовать несколько интерфейсов
- 3) Интерфейс может наследоваться от множества других интерфейсов
- 4) Нельзя наследовать от класса, помеченного ключевым словом sealed

36 Что будет выведено в результате выполнения программы?

```
1. class Program
2. {
3.     static void Main(string[] args)
4.     {
5.         var b = new B();
6.         var c = new C();
7.
8.         Console.Write(b.Sum(2, 3));
9.         Console.Write(c.Sum(2, 3));
10.        Console.ReadLine();
11.    }
12. }
13.
14. class A
15. {
16.     public virtual int Sum(int a, int b)
17.     {
18.         return a + b;
19.     }
20. }
21.
22. class B : A
23. {
24.
27. class C : A
28. {
29.     public override int Sum(int a, int b)
30.     {
31.         return a + b + 1;
32.     }
33. }
1. 55
2. 56
3. 66
4. Возникнет ошибка на этапе компиляции
```

37 Что будет выведено в результате выполнения программы?

```
1. class Program
2. {
3.     static void Main(string[] args)
4.     {
5.         try
6.         {
7.             var a1 = new A1();
8.             Console.Write("1");
9.             var a2 = new A2();
10.        }
11.        catch (Exception)
```



```

12.     {
13.         Console.WriteLine("3");
14.     }
15.     Console.ReadLine();
16. }
17. }
18.
19. class A1
20. {
21.     public int B1;
22. }
23.
24. class A2
25. {
26.     public int B2;
27.
28.     public A2(int b2)
29.     {
31.     }
32. }
1.     1
2.     3
3.     13
4.     Возникнет ошибка на этапе компиляции

```

38. Будет ли компилироваться следующий фрагмент кода?

```

try
{
FileStream F = new FileStream("myfile.txt");
string s = F.ReadLine();
}
catch (IOException) { }
finally
{
F.Close();
}

```

Да;
Нет.

39. Что делает оператор %?

Переводит дробное число в проценты
Возвращает остаток от деления
Возвращает процентное соотношение двух операндов
Форматирует значения разных типов в строку

40. Сколько родительских классов может иметь производный класс?
Не больше одного

Всегда один
Не больше двух
Любое количество

41. Какой класс является базовым для всех классов в C#? Ответ:

42. Первый сценарий (компиляция без ключа /checked):

```
byte mask = 255;  
mask++;
```

Второй сценарий:

```
checked  
{  
mask++;  
}
```

Третий сценарий (компиляция с ключом /checked):

```
byte mask = 255;  
mask++;
```

Четвертый сценарий: `byte mask = 255; unchecked`

```
{  
mask++;  
}
```

Перечислите чему равно `mask` во всех 4 случаях.

-255, 0, 1, 0

-0, -1, -1, 0

-1, вызовется исключение `OverflowException`, 0, 256

-0, вызовется исключение `OverflowException`, вызовется исключение
`OverflowException`, 0

-256, -1, -1, 0

43.

Выберите допустимые прототипы метода `Main`.

-static int Main(string[] args) { }

-static int main() { }

-unsafe static void Main(int argc, char* argv[]) { }

-static void main() { }

-static void Main(string[] args) { }

static void Main() { }

static int Main() { }

44. Выберите элементы, которые нельзя пометить атрибутом.

-Интерфейсы

-Все перечисленное можно пометить атрибутом

- Возвращаемые значения
- Классы
- Методы
- Структуры

45 Выберите конструкцию, которая наиболее схожа по функциональности со следующим кодом:

```
using (StreamWriter writer = new StreamWriter("example.txt"))
{
}
-StreamWriter writer = new StreamWriter("example.txt"); writer.Close();
-StreamWriter writer; try { writer = new StreamWriter("example.txt"); } catch
(Exception e) { -Trace.WriteLine(e.ToString()); } finally { writer.Dispose(); }
-Оператору using() { } нет альтернативы
-StreamWriter writer; try { writer = new StreamWriter("example.txt"); } finally {
writer.Dispose(); }
```

46. Что произойдет при выполнении следующего кода?

```
int i = 5;
object o = i;
long j = (long)o;
-Значение переменной j предсказать нельзя
-Средой исполнения будет вызвано исключение InvalidCastException
-Произойдет ошибка времени компиляции
-Ошибок не произойдет. Переменная j будет иметь значение 5
```

47. Что произойдет при компиляции проекта, где используется класс, структура, интерфейс или перечисление, помеченное атрибутом Obsolete?

- Что произойдет при компиляции проекта, где используется класс, структура, интерфейс или перечисление, помеченное атрибутом Obsolete?
- Будет выведено предупреждение о том, что данный тип устарел, но сборка будет создана
- Сборка будет создана, но при запуске произойдет ошибка времени выполнения
- Атрибут Obsolete никак не влияет на компиляцию
- Нет нужного варианта ответа

48. Выберите правильные объявления метода с корректным использованием ключевого слова params:

```
-public void Foo(params object[] objectArray, params int[] intArray) { }
-public void Foo(params double[] array) { }
-public void Foo(params int[] intArray, params double[] doubleArray) { }
-public void Foo(params double[] array, double volume) { }
-public void Foo(double volume, params double[] array) { }
```

49. Чем опасны подобные конструкции?

```
public class Widget
{
private int width;
public int IncrementWidth
```

```

{
get { return ++width; }
}
}
class Program
{
static void Main(string[] args)
{
Widget widget = new Widget();
int width = widget.IncrementWidth;
}
}

```

- Неправильно сделан инкремент поля width, нужно ++width заменить на width++
- Ничем, код полностью корректен
- При отладке кода в режиме Debug при просмотре свойства IncrementWidth объекта widget будет увеличиваться значение width, что повлечет сложности при отладке
- Нет блока set {} в свойстве IncrementWidth

50. Какая утилита из .NET Framework служит для генерации XMLдокументации из комментариев /// и /** */ комментариев в коде?

- ilasm.exe с ключом /doc:FileName.xml
- tlbexp.exe
- xsd.exe с ключом /doc:FileName.xml
- csc.exe с ключом /doc:FileName.xml
- Нет правильного ответа

51. С помощью какой утилиты .NET Framework можно сгенерировать C# код класса по имеющейся XML схеме?

- xsd.exe с ключом /c
- xml.exe с ключом /c
- tlbimp.exe с ключом /c
- Автоматизировать это нельзя – код придется писать самому
- Используя утилиту ildasm.exe с ключом /c

52. Стоит задача обработать большой XML-файл. Какой класс для этого следует выбрать, чтобы обработка прошла наиболее быстро?

- XmlTextReader
- XmlDocument
- XmlPathDocument
- XmlSerializer

53. Чтобы использовать unsafe код в приложении, необходимо ...

- Пометить методы, где используется небезопасный код атрибутом Unsafe
- Пометить методы, где используется небезопасный код с помощью ключевого слова unsafe
- Пометить методы, где используется небезопасный код с помощью ключевого слова fixed
- Компилировать код приложения с ключом /unsafe

54. Каким образом нужно применить приведенный атрибут, чтобы одновременно присвоить значения свойствам Description и Price?

```
public class Product : Attribute {
    public Product(string description) { Description = description;
    }
    private string description;
    public string Description {
        get { return description; }
        set { description = value; }
    }
    private float price;
    public float Price {
        get { return price; }
        set { price = value; }
    }
}
```

-Это невозможно сделать

-[Product(Price = 149.95f, "CPU")]

-[Product("CPU", Price = 149.95f)]

-[Product("CPU", 149.95f)]

-[Product(Description = "CPU", Price = 149.95f)]

55. Что произойдет при Binary сериализации объекта widget?

```
public interface IWidget
{
    int Index
    {
        get;
        set;
    }
}
[Serializable]
public class Widget : IWidget
{
    private int index;
    public int Index
    {
        get { return index; }
        set { index = value; }
    }
}
// Метод Main()
IWidget widget = new Widget();
widget.Index = 10;
using (FileStream writer = new FileStream("data.dat", FileMode.Truncate))
{
    BinaryFormatter serializer = new BinaryFormatter ();
    serializer.Serialize(writer, widget);
}
```

- }
 - Ошибка времени компиляции: для сериализации нужно использовать не StreamWriter, а только BinaryWriter
 - Ошибка времени исполнения: интерфейс IWidget должен быть помечен атрибутом Serializable
 - Ошибка времени исполнения: класс Widget должен быть помечен атрибутом BinarySerializable, а не Serializable
 - Будет вызвано исключение NotSupportedException с сообщением, что нельзя сериализовать интерфейс Iwidget
 - Объект widget будет успешно сериализован

56. Что произойдет при XML сериализации объекта widget?

```
public interface IWidget
{
    int Index
    {
        get;
        set; }
}
[Serializable]
public class Widget : IWidget
{
    private int index;
    public int Index
    {
        get { return index; }
        set { index = value; }
    }
}
// Метод Main()
IWidget widget = new Widget();
widget.Index = 10;
using (StreamWriter writer = new StreamWriter("data.xml"))
{
    XmlSerializer serializer = new XmlSerializer(typeof(IWidget));
    serializer.Serialize(writer, widget);
}
```

- Ошибка времени исполнения: класс Widget должен быть помечен атрибутом XmlSerializable, а не Serializable
- Ошибка времени компиляции: для сериализации нужно использовать не StreamWriter, а только XmlWriter
- Объект widget будет успешно сериализован
- Ошибка времени исполнения: интерфейс IWidget должен быть помечен атрибутом Serializable
 - Будет вызвано исключение NotSupportedException с сообщением, что нельзя сериализовать интерфейс Iwidget

57. Как называется технология, благодаря которой возможно взаимодействие управляемого кода (managed code) с Win32 API функциями и COM-объектами?

- Remoting
- WebServices
- CodeDOM

- Interop
- Reflection

58. Какие операторы нужно переопределить у класса Digit, чтобы были возможны преобразования:

```
Digit digit = new Digit(100); decimal decimalDigit = digit; byte byteDigit = (byte)digit;
-public static implicit operator byte(Digit digit) { }
-public static explicit operator byte(Digit digit) { }
-Нужные операторы не приведены
-public static implicit operator decimal(Digit digit) { }
```

59. Какие ошибки присутствуют в следующем коде?

```
public interface IWidget {
void Draw();
}
public interface IControl : IWidget {
void Move(int x, int y);
}
public class Widget : IWidget {
public void Draw() { }
}
public class Control : Widget, IControl {
public void Move(int x, int y) { }
```

- Тела методов Move() и Draw() не должны оставаться пустыми
- Нельзя одновременно наследоваться от Widget, который реализует IWidget, и реализовывать IControl, который наследуется от IWidget
- В Control не реализован метод Draw()
- Ошибок нет

60. Выберите средства, которые предоставляет C# для условной компиляции.

- Директива #endif
- Директива #typedef
- Директива #else
- Директива #elseif
- Атрибут Conditional
- Директива #define
- Директива #if

61. Выберите правильный вариант переопределенного метода GetHashCode() у класса Point (будем считать, что Equals() уже переопределено правильно)

```
public class Point
{
public int x;
public int y;
}
-public override int GetHashCode() { return x | y; }
```

```
-public override int GetHashCode() { return x * y; }  
-Нет правильного ответа  
-public override int GetHashCode() { return x & y; }  
-public override int GetHashCode() { return x ^ y; }
```

62. Выберите операторы, которые необходимо переопределять попарно.

```
-> и <  
-!= и ==  
-++ и --  
-* и /  
->= и <=  
-+ и -
```

63. Чем отличаются константы и доступные только для чтения поля?

```
-Ничем не отличаются  
-Доступные только для чтения поля инициализируются во время компиляции, кон-  
станты во время выполнения  
-Константы можно изменять, а доступные только для чтения поля нет  
-Константы инициализируются во время компиляции, доступные только для чтения  
поля во время выполнения
```

64. Когда вызываются статические конструкторы классов в C#?

```
-После каждого обращения к статическим полям, методам и свойствам  
-Строгий порядок вызова не определен  
-Статических конструкторов в C# нет  
-Один раз при первом создании экземпляра класса или при первом обращении к  
статическим членам класса C#?
```

65. Реализацией какого паттерна (шаблона проектирования) являются события в

```
-Декоратор (Decorator)  
-Шаблонный метод (Template Method)  
-Посетитель (Visitor)  
-Издатель-подписчик (Publisher-Subscriber)
```

66. Чтобы вывести на экран число типа float 10.56f с количеством цифр после запятой равным 3 и шириной равной 10 символам нужно написать:

```
-Console.WriteLine("{3:10F0}", 10.56f);  
-Console.WriteLine("{10:0:3F}", 10.56f);  
-Console.WriteLine("{0, 3F:10}", 10.56f);  
-Console.WriteLine("{0, 10:F3}", 10.56f);
```

67. По какой причине данный код не скомпилируется?

```
public delegate void EventHandler();  
public class Control  
{  
public event EventHandler Invalidate;
```



```

}
public class Button : Control
{
public void Draw()
{
if (Invalidate != null)
{
Invalidate();
}
}
}
}

```

- Вызывать событие напрямую может только класс, в котором они объявлены
- Нельзя сравнивать событие с null
- Событие всегда равно null
- У события отсутствуют обязательные части add { } и remove { }

68. Чем отличаются перечисление State от перечисления StateFlags, помеченного атрибутом Flags, и что будет выведено на консоль при выполнении следующего кода:

```

public enum State
{
} [Flags]
ToolBox, Menu, StatusBar
public enum StateFlags
{
ToolBox, Menu, StatusBar
}
class Program
{
static void Main(string[] args)
{
Console.WriteLine(State.Menu | State.StatusBar); Console.WriteLine(StateFlags.Menu |
StateFlags.StatusBar);
}
}

```

-Будет выведено «Menu, StatusBar» и «3». StateFlags нельзя использовать как битовое поле, в отличие от State

-Код не откомпилируется

-Будет выведено «Menu, StatusBar» и «Menu, StatusBar». Как State, так и StateFlags можно использовать как битовые поля-Будет выведено «3» и «Menu, StatusBar». Только перечисления, помеченные атрибутом Flags, можно использовать как битовые поля

-Будет выведено «3» и «3». StateFlags и State в любом случае нельзя использовать как битовые поля

69. Какие ошибки допущены в классе?

```

private class Money
{
    public virtual abstract void Transfer(); public abstract Money Exchange(); public abstract
decimal Value
    { get; set;
    }
}

```

- Нельзя использовать private для классов
- Свойство Value не может быть помечено abstract
- В объявлении класса пропущено ключевое слово abstract
- Метод Transfer() не может быть одновременно помечен как виртуальный и абстрактный

70. Есть ли ошибки в следующем коде, и если да то какие?

```

public class Description : Attribute
{
    public Description(decimal money)
    {
        this.money = money;
    }
    private decimal money;
    public decimal Money
    {
        get { return money; }
    }
}
[Description(10)]
public class Widget
{
}

```

- В конструктор нужно передавать 10.0M, а не 10
- Недостает set { } для свойства Money
- Ошибок нет, код полностью корректен
- Тип decimal не разрешено использовать в атрибутах

71. Как правильно объявить implicit оператор, преобразующий объект класса Widget в int?

- public static implicit int(Widget widget) { }
- public static operator int(Widget widget) { }
- public static implicit operator int(Widget widget) { }
- public implicit operator int(Widget widget) { }

72. Что будет выведено на экран после выполнения кода и чему будет равно значение поля Game.player.score?

```

public class Player

```

```

{
public Player(int score)
{
this.score = score; Console.WriteLine(score);
}
public int score;
}
public class Game {
public static Player player = new Player(10);

```

```

static Game() {
player = new Player(20);
}
}

```

// ...Метод Main()

```
Game game = new Game();
```

-Будет выведено 20, затем 10, а поле score равно 20

-Будет выведено 10, затем 20, а поле score равно 10

-Будет выведено 10, затем 20, а поле score равно 20

-Произойдет ошибка времени выполнения

-Нельзя сказать – строго не определен приоритет вызова

73. Что будет выведено на консоль при выполнении следующего кода?

```

int i = 0;
while (i < 10)
{
Console.WriteLine(i);
i++;
}
while (i > 10);

```

-Это бесконечный цикл: будут выведены цифры 0, 1, 2, 3... до Int32.MaxValue

-Это бесконечный цикл: будут выведены цифры от 0 до 10, а далее будет повторяться цифра 10

-Код не скомпилируется

-Будут выведены цифры от 0 до 9

74. Выберите правильные варианты, в которых пространство имен System содержит пространство имен Customizer.

-namespace System::Customizer { }

-namespace System.Customizer { }

-Нельзя создавать собственные пространства имен в пространстве имен System

-namespace System { namespace Customizer { } }

75. Каковы будут результаты выполнения кода (консольное приложение)?

```

public class Author : Attribute
{
public Author(string name)
{
Console.WriteLine(name);
}
}
[Author("Daniel Pols")]
class Program
{
static void Main(string[] args)
{
}
}

```

- На консоль будет выведено “Daniel Pols”
- Ошибка при компиляции кода: строка Console.WriteLine(name);
- На консоль ничего не будет выведено
- Ошибка при компиляции кода: строка [Author(

76. Каким образом можно перехватить добавление и удаление делегата из события?

- Использовать ключевые слова get и set
- Переопределить операторы + и – для делегата
- Такая возможность не предусмотрена
- Для этого существуют специальные ключевые слова add и remove

77.Какая технология позволяет производить компиляцию и генерировать IL код во время выполнения приложения?

- CodeDOM
- ADO .NET
- Remoting
- Такой технологии нет
- WebServices

78.Перечислите возможности структуры (value-типов).

- Определение индекатора
- Переопределение конструктора по умолчанию
- Наследование от классов
- Использование событий
- Определение статического конструктора
- Множественное наследование
- Реализация интерфейсов
- Boxing и Unboxing

79. Есть ли ошибка в коде, и если да, то какая?

```

public class Base
{
public const string Id = "123-982-232";
public static readonly string Date = "12/03/2010";
}
public class Derive : Base

```

```

{
public void ShowId()
{
Console.WriteLine(Id);
}
public void ShowDate()
{
Console.WriteLine(Date);
}
}

```

Derive

- Константные поля не наследуются, то есть поле Id не доступно из класса
- Ошибок нет, Id и Date доступны из класса Derive
- Статические, доступные только для чтения поля не наследуются, то есть поле Date не доступно из класса Derive
- Доступные только для чтения уже статичны, поэтому слово static лишнее при описании поля Date

80.Что будет выведено на экран после выполнения следующего кода?

```

class Widget { }
class Program {
static void Main(string[] args) {
Widget widget = new Widget();
WeakReference reference = new WeakReference(widget); Con-
sole.WriteLine(reference.Target == null);
widget = null; GC.Collect();
Console.WriteLine(reference.Target == null);
}
}

```

- False False. До и после вызова на объект Widget будет ссылаться reference.Target
- False True. На объект Widget ссылается как слабая ссылка, так и widget. При вызове GC.Collect() сборщик проверяет, что на объект Widget ссылается только слабая ссылка и удаляет его
- True False. После вызова GC.Collect() сборщик проверяет, что на объект Widget ничто не ссылается и присваивает reference.Target ссылку на этот объект
- True True. Независимо от вызова сборщика мусора слабая ссылка не будет ссылаться на объект Widget

Тематика курсовых работ

Вариант 1

1. Основные сведения о ООП. Характеристика ОО-языков программирования.

2. Разработка программы учета больных вирусом COVID-19

Программа предоставляет сведения о больных COVID-19, которые находятся на карантине: ФИО больного; количество дней, находящихся на карантине; назначенное лечение; дата заболевания.

Вывести в алфавитном порядке список больных COVID-19, находящиеся на больничном более x дней (x вводятся с клавиатуры).

Вариант 2

1. Классы и объекты. Ссылки.

2. Разработка программы расселения детей в лагере

Для получения места формируется список детей, который включает: Ф.И.О. ребенка; номер группы (шесть цифр); средний балл; участие в общественной деятельности; доход на одного члена семьи.

Вывести список детей, проживающих в определённой комнате (номер комнаты вводится с клавиатуры).

Вариант 3

1. Средства абстракции. Структура класса. Статические члены.

2. Разработка программы расписания движения самолетов

В аэропорту хранится расписание самолётов. Для каждого рейса указаны: номер рейса; пункт назначения; время отправления; время прибытия на конечный пункт.

Вывести информацию о всех рейсах, которыми можно воспользоваться для прибытия в пункт назначения не позднее чем за 12 часов до заданного времени (интересующее время прибытия вводится с клавиатуры).

Вариант 4

1. Средства инкапсуляции. Инкапсуляция и наследование. Друзья.

2. Разработка программы продажи билетов на самолёт

Аэропорт осуществляет продажу билетов на различные рейсы. Для каждого рейса указаны: номер рейса; пункт назначения; дату отправления; время отправления; время прибытия на конечный пункт; стоимость одного билета; количество оставшихся для продажи билетов; количество проданных билетов.

Обеспечить функциональную возможность покупки билетов на конкретный рейс в требуемом количестве (при этом количество оставшихся для продажи билетов и количество проданных билетов обновляется).

Вариант 5

1. Синтаксис объявления класса. Режимы доступа к элементам класса.

2. Разработка программы учета переговоров работников такси

Оператор сотовой связи хранит информацию о разговорах своих абонентов: номер абонента; ФИО абонента; указание принадлежности вызова к исходящему или входящему; номер исходящего или входящего вызова; дата звонка; время звонка; продолжительность разговора.

Вывести по каждому сотруднику такси за требуемый период времени: перечень входящих и исходящих вызовов, общее время входящих вызовов, общее время исходящих вызовов, общую сумму на исходящие вызовы (требуемый период времени вводится с клавиатуры).

Вариант 6

1. Синтаксис описания методов класса.

2. Разработка программы расчета заработной платы сотрудников Яндекс Такси

Сведения о сотрудниках Яндекс Такси содержат: Ф.И.О. сотрудника; табельный номер; год; месяц; количество проработанных часов за неделю; количество выполненных заказов.

Вычеты складываются из:

Комиссии сервиса – 23% от суммы заказа.

Комиссии таксопарка – варьирует в пределах 3-30%.

Определить объем выплат конкретному сотруднику за требуемый период времени (требуемый период времени вводится с клавиатуры).

Вариант 7

1. Создание экземпляров класса. Вызов методов.

2. Разработка программы учета проживающих в общежитии

Для проживающих в общежитии, задаются: ФИО; год заселения; комната; замечания; год выселения; причина выселения, отметка о проживании текущий момент (если еще проживает в общежитии, если не проживает в общежитии указать, что выселен в таком-то году).

Вывести список комнат с ФИО в алфавитном порядке, заселенных после заданного года (год вводится с клавиатуры). Вывести список проживающих, в текущий момент.

Вариант 8

1. Массивы объектов. Создание. Доступ к элементам. Вызов методов.

2. Разработка программы учета выполненных заказов Яндекс такси

Сведения о выполненных заказах включают: дату; номер заказа; точка отправления; точка прибытия; количество выполненных заказов; ФИО водителя в данный день.

Для заданного водителя Яндекс такси необходимо вывести количество выполненных заказов за требуемый период времени (требуемый период времени вводится с клавиатуры).

Вариант 9

1. Указатели на объекты. Ссылки.

2. Разработка программы учета научной деятельности сотрудников

Информация о сотрудниках содержит: Ф.И.О. сотрудника; ученую степень; название кафедры; должность; дату начала работы; количество публикаций.

Вывести список сотрудников занимающиеся активной научной деятельностью. Вывести список сотрудников в порядке убывания количества публикаций.

Вариант 10

1. Конструкторы и деструкторы.

2. Разработка программы учета выплат заработной платы работникам приюта

Информация о сотрудниках содержит: Ф.И.О. сотрудника, должность, размер заработной платы за месяц.

Вычислить общую сумму выплат за месяц по каждому отделу, а также среднемесячный заработок сотрудников по каждому отделу. Вывести список сотрудников, у которых зарплата ниже введенной с клавиатуры.

Вариант 11

1. Модульность, отдельная компиляция, пространства имен, using директива.

2. Разработка программы учета клиентов салона красоты

Информация о клиентах: Ф.И.О. клиента, дата посещения, номер телефона, мастер Ф.И.О., дата следующей записи, процедура.

Вывести список и подсчитать общее количество постоянных клиентов определенного мастера, Ф.И.О., которого будет введена с клавиатуры.

Вариант 12

1. Представление иерархических отношений. Наследование.

2. Разработка программы ассортимента машин в таксопарке

Сведения об ассортименте в таксопарке включают: название марки машины, цена, изготовитель, количество, год выпуска, пробег машины.

Вывести список машин, которые подходят по цене, где x цена (x вводится с клавиатуры).

Вариант 13

1. Методы доступа к закрытым членам класса (getter и setter –методы).

2. Разработка программы подбора материалов для ремонта квартиры

Сведения включают: артикул, наименование, цена, изготовитель, размер, количество.

Вывести список материалов артикула x и размера y (x , y вводятся с клавиатуры). Осуществить расчет приобретения необходимого количества материалов для ремонта.

Вариант 14

1. Представление иерархических отношений. Агрегация. Зависимость по времени жизни.

2. Разработка программы учета выполнения работы лаборанта

На кафедре хранится информация обо всех заявках: наименование ремонтируемого изделия (компьютеры и т.д.), инвентарный номер изделия, кабинет, Ф.И.О. заявителя, дата написания заявки, дата выполнения, статус (выполнен или нет).

Вывести список заявок, невыполненных на текущий момент (сначала – просроченные, затем – ожидающие выполнения в плановом порядке). Вывести общий объем выполненных заявок от даты x до даты y (x , y вводятся с клавиатуры).

Вариант 15

1. Правила преобразования типов. Параметрический и виртуальный полиморфизм.

2. Разработка программы учета слушателей переподготовки

Сведения об успеваемости слушателя содержат следующую информацию: номер группы (шесть цифр), Ф.И.О. слушателя, сведения о зачетах (зачет/незачет), оценки по экзаменам, рефератам, ОКР.

Вывести всех слушателей в порядке убывания количества задолженностей (количество задолженностей по каждому слушателю необходимо указать); средний балл, полученный каждым слушателем группы x (вводится с клавиатуры), и всей группой в целом.

Вариант 16

1. Классы. Отношения между классами. Ассоциация и агрегация.

2. Разработка программы учета сведений о музыкальном конкурсе

Информация о конкурсе включает: Ф.И.О участника; год рождения; название страны; наименование музыкального инструмента (гитара, фортепиано, скрипка, виолончель и др.); занятое место по результатам конкурса.

По каждому классу музыкальных инструментов вывести первые три места с указанием возраста победителей. Вывести список самых молодых (до 12 лет) победителей конкурса в порядке увеличения возраста.

Вариант 17

1. Основные принципы объектного подхода. Абстрагирование.

2. Разработка программы учета сведений о пациентах медицинского центра

Сведения о пациентах медицинского центра содержат: Ф.И.О. пациента, пол, дату рождения, место проживания (город), контактный телефон, диагноз.

Вывести иногородних пациентов. Вывести список пациентов старше x лет, у которых диагноз y (x , y вводятся с клавиатуры).

Вариант 18

1. Основные принципы объектного подхода. Инкапсуляция. Модульность.

2. Разработка программы продажи железнодорожных билетов

Железнодорожный вокзал осуществляет продажу билетов на поезда дальнего следования. Для каждого поезда указаны: номер поезда; пункт назначения; дату отправления;

время отправления; время прибытия на конечный пункт; стоимость одного билета; количество оставшихся для продажи билетов; количество проданных билетов.

Обеспечить функциональную возможность покупки билетов на конкретный рейс в требуемом количестве (при этом количество оставшихся для продажи билетов и количество проданных билетов обновляется). Вывести номер, время отправления и наличие билетов для поездов, прибывающих в город x в интервале от a до b (x , a , b вводятся с клавиатуры).

Вариант 19

1. Основные принципы объектного подхода. Иерархия. Типизация.

2. Разработка программы планирования факультативных учебных дисциплин для студентов

Для формирования факультативных занятий необходимо обработать информацию следующего вида: Ф.И.О. студента, номер группы, средний балл успеваемости, пять возможных для факультативного посещения дисциплин. Выбираемая дисциплина помечается 1, не выбираемая – 0. Перечень предлагаемых дисциплин: математика, физика, программирование, английский язык, базы данных.

Вывести список и общее количество студентов, желающих прослушать дисциплину x . Если число желающих больше 15, то отобрать 15 студентов с более высоким баллом успеваемости. Вывести предлагаемые дисциплины в порядке убывания популярности с указанием общего числа записавшихся на каждую из них.

Вариант 20

1. Основные принципы объектного подхода. Параллелизм. Сохраняемость.

2. Разработка программы учета сведений об игроках хоккейной команды

Сведения об игроках хоккейной команды включают: Ф.И.О. игрока, дату рождения, количество сыгранных матчей, число заброшенных шайб, количество голевых передач, количество штрафных минут.

Вывести 6 лучших игроков (голы+передачи) с указанием их результативности.

Вариант 21

1. Классы. Природа классов. Мета модель. Инстанцирование.

2. Разработка программы продажи авиабилетов

Авиакомпания осуществляет продажу билетов на самолеты с указанием: номера рейса; типа самолета; пункта назначения; дату вылета; время вылета; время прилета; вместимости самолета; количества оставшихся билетов бизнес-класса; стоимости билета бизнес-класса; количества оставшихся билетов эконом-класса; стоимости билета эконом-класса.

Обеспечить функциональную возможность покупки билетов на конкретный рейс в требуемом количестве (при этом количество оставшихся для продажи билетов обновляется). Если на интересующий рейс нет билетов требуемого класса, то при наличии билетов другого класса на этот рейс – вывести соответствующее информационное сообщение (например, «Билетов эконом-класса на данный рейс нет в наличии, но имеются билеты бизнес-класса в количестве 10.»).

Вариант 22

1. Передача массива в функцию.

2. Разработка программы учета автомобилей автопарка

Таксопарк содержит информацию об имеющихся автомобилях: марка автомобиля, год выпуска, расход топлива, стоимость автомобиля, количество автомобилей данной марки.

Подсчитать общую стоимость таксопарка. Подсчитать общую стоимость автомобилей каждого вида.

Вариант 23

1. Инкапсуляция - один из основных принципов ООП. Примеры

2. Разработка программы расчета стипендии

Для расчета стипендии необходимо обработать информацию следующего вида: номер группы, Ф.И.О. студента, форма обучения (платная/бюджетная), количество пропусков (их должно быть не больше 10 часов), зачеты по пяти предметам (зачет/незачет), оценки по 4 предметам по 10 бальной системе.

Рассчитать стипендию для студентов-бюджетников. При этом:

1500 руб. – если средний балл успеваемости от 5,0 по 5,9;

1700 руб. – если средний балл успеваемости от 6,0 по 7,9;

2000 руб. – если средний балл успеваемости от 8,0 по 8,9;

2500 руб. – если средний балл успеваемости от 9,0 по 10,0.

При вводе номера группы появляется вся информация о стипендиях студентов и о студентах.

Вариант 24

1. Конструкторы и деструкторы класса. Перегрузка конструкторов.

2. Разработка программы учета выполнения работы мастера по работе с волосами

Необходимо обработать следующую информацию: ФИО мастера, тип работы, номер клиента, стоимость, дата.

Вывести лучшего сотрудника за рабочий день, рабочий день вводится с клавиатуры.

Вариант 25

1. Наследование. Базовый и производный класс. Примеры использования. Использование спецификаторов доступа при наследовании.

2. Разработка программы планирования бюджета при ремонте машины

Необходимо обработать информацию следующего вида: номер машины, марка машины, год машины, Ф.И.О. сотрудника, перечень работ, количество часов, стоимость одного часа.

Для каждого мастера вывести его итоговую стоимость, перечень видов работ с указанием общего количества задействованных специалистов и стоимости данного этапа.

Вариант 26

1. Полиморфизм. Формы полиморфизма. Интерфейсы классов.

2. Разработка программы расчета оплаты лечения

Сведения о клиентах имеют следующий вид: Ф.И.О.; год; месяц; наименование услуги.

Для месяца x года y вывести список клиентов. Вывести сумму, которая была заработана, интересующий месяц X , y вводятся с клавиатуры.

Вариант 27

1. Исключения. Обработка исключений.

2. Разработка программы подбора санатория

Компания содержит сведения о предлагаемых путевках: тип путевки (отдых, лечение), количество дней, питание (завтраки/все включено), стоимость.

Вывести все путевки требуемого типа (вводится с клавиатуры) стоимостью менее x в порядке убывания стоимости.

Вариант 28

1. Классы. Структура класса. Абстрактные классы и интерфейсы.

2. Разработка программы учета продаж билетов в аквапарк

Сведения о продажах содержат следующую информацию: дата, наименование аквапарка, уровень доступа, количество проданных билетов.

Вывести список самых популярных аквапарков в порядке убывания в указанный месяц. Вывести список самых популярных аквапарков в порядке убывания в указанный месяц. Месяц вводится с клавиатуры.

Вариант 29

1. Основы STL. Структура и назначение. Контейнеры.

2. Разработка программы учета продаж квартир

Сведения о продажах следующую информацию: год продажи, месяц продажи, количество проданных квартир, стоимость квартиры; характеристика объекта; номер объекта.

Определить общий доход от продажи за определенный месяц. Вывести список самых востребованных объектов в порядке убывания в указанный месяц. Месяц вводится с клавиатуры.

Вариант 30

1. Шаблоны классов и шаблоны функций. Специализация.

2. Разработка программы автоматизации учета кадров на предприятии

Сведения о сотрудниках компании содержат следующую информацию: Ф.И.О. сотрудника; год, когда был принят на работу; месяц, когда был принят на работу; когда заканчивается контракт; должность.

Вывести список сотрудников контракт которых заканчивается (с месяца x по месяц y).

10.2 Критерии оценки результатов текущего контроля освоения дисциплины

Критерии оценки устного опроса

Оценка, уровень достижения компетенций	Описание критериев
Отлично, высокий	Обучающийся демонстрирует уверенное знание материала, полно излагает материал, дает правильное определение основных понятий; обнаруживает понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только из учебника, но и самостоятельно составленные; излагает материал последовательно и правильно с точки зрения норм литературного языка
Хорошо, продвинутый	Обучающийся демонстрирует уверенное знание материала, но допускает 1–2 ошибки, которые сам же исправляет, и 1–2 недочета в последовательности и языковом оформлении излагаемого.
Удовлетворительно, пороговый	Обучающийся обнаруживает знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил; не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; излагает материал непоследовательно и допускает ошибки в языковом оформлении излагаемого.
Неудовлетворительно, компетенция не освоена	Обучающийся демонстрирует незнание большей части соответствующего вопроса, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал.

Критерии оценки практической работы

Оценка «отлично» – ставится, если обучающийся демонстрирует знание теоретического и практического материала по теме практической работы, определяет взаимосвязи между показателями задачи, даёт правильный алгоритм решения, определяет междисциплинарные связи по условию задания. А также, если обучающийся имеет глубокие знания учебного материала по теме практической работы, показывает усвоение взаимосвязи основных понятий используемых в работе, смог ответить на все уточняющие и дополнительные вопросы.

Оценка «хорошо» – ставится, если обучающийся демонстрирует знание теоретического и практического материала по теме практической работы, допуская незначительные неточности при решении задач, имея неполное понимание междисциплинарных связей при правильном выборе алгоритма решения задания. А также, если обучающийся показал знание учебного материала, усвоил основную литературу, смог ответить почти полно на все заданные дополнительные и уточняющие вопросы.

Оценка «удовлетворительно» – ставится, если обучающийся затрудняется с правильной оценкой предложенной задачи, даёт неполный ответ, требующий наводящих вопросов преподавателя, выбор алгоритма решения задачи возможен при наводящих вопросах преподавателя. А также, если обучающийся в целом освоил материал практической работы, ответил не на все уточняющие и дополнительные вопросы.

Оценка «неудовлетворительно» – ставится, если обучающийся даёт неверную оценку ситуации, неправильно выбирает алгоритм действий. А также, если он имеет существенные пробелы в знаниях основного учебного материала практической работы, который полностью не раскрыл содержание вопросов, не смог ответить на уточняющие и дополнительные вопросы.

Критерии оценки тестовых заданий

Оценка, уровень достижения компетенций	Описание критериев
Отлично, высокий	Содержание правильных ответов в тесте не менее 90%
Хорошо, продвинутый	Содержание правильных ответов в тесте не менее 75%
Удовлетворительно, пороговый	Содержание правильных ответов в тесте не менее 50%
Неудовлетворительно, компетенция не освоена	Содержание правильных ответов в тесте менее 50%

Критерии оценки уровня сформированности компетенций при выполнении курсовой работы:

- оценка «отлично»: продемонстрировано блестящее владение проблемой исследования, материал выстроен логично, последовательно, обучающийся аргументированно отстаивает свою точку зрения. Во введении приводится обоснование выбора конкретной темы, чётко определены цель и задачи работы. Использован достаточный перечень источников и литературы для методологической базы исследования. Обучающийся грамотно использует профессиональные термины, актуальные исходные данные. Проведен самостоятельный анализ (исследование) объекта. По результатам работы сделаны логичные выводы. Оформление работы соответствует методическим рекомендациям. Объем и содержание работы соответствует требованиям. На защите обучающийся исчерпывающе отвечает на все дополнительные вопросы;

- оценка «хорошо»: обучающийся демонстрирует повышенный уровень владения проблемой исследования, логично, последовательно и аргументированно отстаивает ее концептуальное содержание. Во введении содержатся небольшие неточности в формули-

ровках цели, задач. В основной части допущены незначительные погрешности в расчетах (в исследовании). Выводы обоснованы, аргументированы. Оформление работы соответствует методическим рекомендациям. Объем работы соответствует требованиям. На защите обучающийся отвечает на все дополнительные вопросы;

- оценка «удовлетворительно»: обучающийся демонстрирует базовый уровень владения проблемой исследования. Во введении указаны цель и задачи исследования, но отсутствуют их четкие формулировки. Работа является компиляцией чужих исследований с попыткой формулировки собственных выводов в конце работы. Изложение материала логично и аргументировано. Наблюдается отступление от требований в оформлении и объеме работы. При ответе на вопросы обучающийся испытывает затруднения;

- оценка «неудовлетворительно»: обнаруживается несамостоятельность выполнения курсовой работы, некомпетентность в исследуемой проблеме. Нарушена логика изложения. Работа не соответствует требованиям, предъявляемым к оформлению и содержанию. На защите курсовой работы обучающийся не отвечает на вопросы.

10.3. Оценочные материалы для промежуточной аттестации по дисциплине

Вопросы для подготовки к экзамену

1. Основные конструкции алгоритмического языка
2. Жизненный цикл программы
3. Особенности языка программирования
4. Структура программы в языке
5. Понятие переменной
6. Объявление переменных.
7. Типы данных, определяемые пользователем.
8. Инициализация переменных.
9. Основные типы переменных
10. Синтаксис объявления имени переменной
11. Понятие константы. Объявление.
12. Понятие константы. Инициализация.
13. Простейшие арифметические операции
14. Операция %. Ее особенности.
15. Основные математические функции
16. Принципы структурного программирования
17. Оператор множественного ветвления switch
18. Условный оператор if-else
19. Вложенный оператор if.
20. Классификация циклов
21. Операции инкремента и декремента.
22. Оператор while()
23. Оператор For.
24. Оператор do...while()
25. Принципы модульного программирования
26. Описание функции
27. Вызов функции.
28. Прототипы функций.
29. Передача параметров по имени.
30. Передача параметров по адресу.
31. Данные типа «указатель».
32. Операции для работы с указателями

33. Объявление и инициализация одномерного массива.
34. Объявление и инициализация двумерного массива.
35. Доступ к элементам одномерного массива по индексу.
36. Доступ к элементам одномерного массива по адресу.
37. Доступ к элементам двумерного массива по индексу.
38. Доступ к элементам двумерного массива по адресу.
39. Принципы объектно-ориентированного программирования. Инкапсуляция.
40. Принципы объектно-ориентированного программирования. Полиморфизм.
41. Принципы объектно-ориентированного программирования. Наследование.
42. Структуры. Объявление.
43. Структуры. Инициализация.
44. Прямой доступ к элементам структуры.
45. Косвенный доступ к элементам структуры.
46. Массивы структур. Объявление.
47. Массивы структур. Инициализация.
48. Классы. Объявление открытых переменных.
49. Классы. Инициализация открытых переменных.
50. Классы. Объявление закрытых переменных.
51. Классы. Инициализация закрытых переменных.
52. Спецификаторы доступа.
53. Функции-члены класса.
54. Связывание функций.
55. Встроенные функции.
56. Конструкторы . Основные понятия.
57. Конструкторы по умолчанию.
58. Конструкторы с параметрами.
59. Стандартный класс string. Основные понятия.
60. Основные операции класса string.

10.4 Показатели, критерии и шкала оценивания ответов на экзамене

Оценка, уровень достижения компетенций	Описание критериев
Отлично, высокий	Обучающийся показал полные и глубокие знания программного материала, логично и аргументировано ответил на все вопросы экзаменационного билета, а также на дополнительные вопросы, способен самостоятельно решать сложные задачи дисциплины, практическое задание выполнено без ошибок
Хорошо, продвинутый	Обучающийся твердо знает программный материал, грамотно его излагает, не допускает существенных неточностей в ответе, достаточно полно ответил на вопросы экзаменационного билета и дополнительные вопросы, способен самостоятельно решать стандартные задачи дисциплины, практическое задание выполнено без ошибок
Удовлетворительно, пороговый	Обучающийся показал знание только основ программного материала, усвоил его поверхностно, но не допускал грубых ошибок или неточностей, требует наводящих вопросов для правильного ответа, не ответил на дополнительные вопросы, способен решать стандартные задачи дисциплины с помощью преподавателя, практическое задание в целом выполнено, но содержит незначительные недостатки

Неудовлетворительно, компетенция не освоена	Обучающийся не знает основ программного материала, допускает грубые ошибки в ответе, не способен решать стандартные задачи дисциплины даже с помощью преподавателя, практическое задание выполнено частично или не выполнено
---	--