

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ДАГЕСТАН  
Государственное бюджетное профессиональное образовательное учреждение  
Республики Дагестан «Технический колледж имени Р.Н. Ашуралиева»

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
по междисциплинарному курсу

**МДК.01.01 Разработка программных модулей**

Специальность: 09.02.13 Интеграция решений с применением технологий  
искусственного интеллекта

Квалификация выпускника: специалист по работе с искусственным интеллектом

## ОГЛАВЛЕНИЕ

1. ОБЩИЕ ПОЛОЖЕНИЯ .....	3
2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА .....	3
3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ .....	7
4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ .....	16
5.1. Типовые задания для оценки знаний и умений .....	16
5.2. Критерии оценивания .....	54
5.2.1. Критерии оценивания устного ответа .....	54
5.2.2. Критерии оценивания выполнения заданий на лабораторных и практических занятиях .....	54
5.2.3. Критерии оценивания тестовых заданий .....	55
5.2.4. Общая классификация ошибок .....	55
6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ .....	55
ОБУЧАЮЩИХСЯ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ .....	55
6.1. Перечень заданий для подготовки к экзамену в V семестре .....	55
6.2. Перечень заданий для подготовки к экзамену в VI семестре .....	59
6.3. Перечень заданий для подготовки к экзамену в VII семестре .....	62
6.3. Критерии оценивания ответов на экзамене .....	65
7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ .....	65

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

Фонд оценочных средств (далее – ФОС) междисциплинарного курса МДК.01.01 Разработка программных модулей является частью программы подготовки специалистов среднего звена специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта.

ФОС позволяет оценить достижение, запланированных по междисциплинарному курсу, результатов обучения.

ФОС включают оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации по междисциплинарному курсу.

**Текущий контроль** успеваемости осуществляется с целью регулярного наблюдения за ходом поэтапного освоения обучающимися междисциплинарного курса, оптимизации управления образовательной деятельностью обучающихся, своевременной корректировки персональных образовательных результатов, обучающихся педагогическими средствами.

Текущему контролю успеваемости подлежат все обучающиеся, осваивающие междисциплинарный курс.

Текущий контроль проводится в пределах учебного времени, отведенного на изучение междисциплинарного курса традиционными и инновационными методами с использованием современных технологий.

Результаты текущего контроля успеваемости обучающихся в виде оценки в балльном выражении («5», «4», «3», «2») записываются в журнале учебных занятий.

Текущий контроль освоения обучающимися программного материала междисциплинарного курса может иметь следующие виды: оперативный и рубежный контроль.

**Оперативный контроль** проводится с целью объективной оценки качества освоения программы междисциплинарного курса, а также стимулирования учебной работы обучающихся, мониторинга результатов образовательной деятельности, подготовки к промежуточной аттестации и обеспечения максимальной эффективности учебно-воспитательного процесса.

**Рубежный контроль** является контрольной точкой и проводится с целью комплексной оценки уровня освоения программного материала.

**Курсовой проект** является завершающим этапом в изучении МДК. 02.01. Технология разработки программного обеспечения, в ходе, которого обучающиеся приобретают практический опыт в области проектирования и разработки программного обеспечения.

**Промежуточная аттестация** по междисциплинарному курсу проводится с целью оценки уровня освоения теоретических знаний, умений, приобретенного практического опыта.

Формы и периодичность промежуточной аттестации по междисциплинарному курсу определяются учебным планом образовательной программы: экзамен 5, 6, 7-м семестре, курсовой проект в 7 семестре.

Экзамен проводится в сроки, установленные календарным учебным графиком. Экзамен проводится в день, освобожденный от других форм учебной нагрузки.

Экзаменационные вопросы и задания составляются на основе рабочей программы междисциплинарного курса. Экзаменационные вопросы и задания должны соответствовать проверяемым результатам обучения и доводятся до сведения обучающихся в течение первых двух месяцев от начала обучения.

## 2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ МЕЖДИСЦИПЛИНАРНОГО КУРСА

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем направлен на формирование общих и профессиональных компетенций.

Освоение междисциплинарного курса должно способствовать формированию общих компетенций:

- ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
- ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.
- ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.
- ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.
- ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.
- ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения.
- ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.
- ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.
- ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

В результате освоения междисциплинарного курса обучающийся должен получить **практический опыт**:

- Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования;
- Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля;
- Разрабатывать мобильные приложения;
- Регистрации изменений исходного текста программного кода в системе управления версиями.

В результате освоения междисциплинарного курса обучающийся должен **уметь**:

Код умения	Название умения
У1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
У2	Оформлять документацию на программные средства
У3	Оценивать сложность алгоритма
У4	Создавать программу по разработанному алгоритму как отдельный модуль
У5	Использовать выбранную систему управления версиями

В результате освоения дисциплины обучающийся должен **знать**:

Код знания	Название знания
З1	Основные этапы разработки программного обеспечения
З2	Основные принципы технологии структурного и объектно-ориентированного программирования
З3	Актуальную нормативно-правовую базу в области документирования алгоритмов

34	Основные этапы разработки программного обеспечения
35	Основные принципы технологии структурного и объектно-ориентированного программирования
36	Возможности используемой системы управления версиями и вспомогательных инструментальных программных средств

### Общие компетенции:

Код компетенции	Формулировка компетенции	Знания, умения
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	<p><b>Умения:</b> распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> <p><b>Знания:</b> актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности</p>
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	<p><b>Умения:</b> определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска; применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение</p> <p><b>Знания:</b> номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации; современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности</p>
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в	<p><b>Умения:</b> определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования; выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования</p>

	различных жизненных ситуациях	<b>Знания:</b> содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования; основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде	<b>Умения:</b> организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности <b>Знания:</b> психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	<b>Умения:</b> грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе <b>Знания:</b> особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.
ОК 06	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	<b>Умения:</b> описывать значимость своей специальности <b>Знания:</b> сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности
ОК 07	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	<b>Умения:</b> соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности <b>Знания:</b> правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения
ОК 08	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.	<b>Умения:</b> использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности <b>Знания:</b> роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках	<b>Умения:</b> понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы

		<b>Знания:</b> правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности
--	--	---

### Профессиональные компетенции:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	<b>Практический опыт:</b> Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
	<b>Умения:</b> Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства. Оценивать сложность алгоритма.
	<b>Знания:</b> Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Актуальная нормативно-правовая база в области документирования алгоритмов.
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.	<b>Практический опыт:</b> Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля.
	<b>Умения:</b> Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства.
	<b>Знания:</b> Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования.

## 3. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ

Формы текущего контроля по междисциплинарному курсу:

- устный опрос (фронтальный, индивидуальный, комбинированный);
- тестирование (письменное или компьютерное);
- письменная проверка (ответы на вопросы, решение задач и примеров, составление тезисов, рефератов, выполнение схем, выполнение заданий для самостоятельной работы и др.);
- практическая проверка (при проведении практических и лабораторных занятий, выполнении и защите курсовых проектов (работ));
- самоконтроль и взаимопроверка.

Возможны и другие формы текущего контроля успеваемости, в том числе инновационные на основе информационно-коммуникационных технологий.

Преподаватель на одном учебном занятии может использовать одну или несколько форм текущего контроля.

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
<b>Раздел модуля 1. Анализ и проектирование программных решений</b>		

<p>ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием</p>	<p>Оценка «<b>отлично</b>» - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры. Указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма</p> <p>Оценка «<b>хорошо</b>» -алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры. Выполнена оценка сложности алгоритма</p> <p>Оценка «<b>удовлетворительно</b>» - алгоритм разработан и соответствует заданию.</p>	<p>Экзамен в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам Защита курсового проекта</p>
<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «отлично» - программный разработан по имеющемуся алгоритму в среде разработки Visual Studio на языке C# методами объектно- ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки Visual Studio на языке C# методами объектно- ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки Visual Studio на языке C# методами объектно- ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием Защита отчетов по практическим и лабораторным работам Защита курсового проекта</p>

<p>ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам</p>	<p>- Обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач. - Адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач</p>	<p>Экспертное наблюдение за выполнением работ</p>
<p>ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности</p>	<p>- Использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач. - Эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту</p>	
<p>ОК 03. Планировать и реализовывать собственное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по</p>	<p>- Демонстрация ответственности за принятые решения. - Обоснованность самоанализа и коррекция результатов собственной работы</p>	



финансовой грамотности в различных жизненных ситуациях		
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	- Взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик. - Обоснованность анализа работы членов команды (подчиненных)	
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста	- Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей	
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	- Соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик	
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	- Эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик. - Демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	- Эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности	
ОК 09. Использовать информационные технологии в профессиональной деятельности	- Эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке	

**4. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНОГО КУРСА**  
**МДК 01.01 Разработка программных модулей**

№	Наименование темы	Результаты обучения (освоенные умения и знаний)	ПК, ОК	Текущий контроль успеваемости	Промежуточная аттестация
	<b>2</b>		<b>4</b>		
<i>Раздел 1. Разработка программных модулей</i>					
<i>МДК. 01.01 Разработка программных модулей</i>					
	<b>Тема 1.1.1 Жизненный цикл ПО</b>	<b>34</b>	<b>ОК 1 - ОК 9, ПК 1.1</b>	<b>Тестирование</b>	экзамен
1.	Понятие ЖЦ ПО. Этапы ЖЦ ПО.	34		Тестирование	
	<b>Тема 1.1.2 Структурное программирование</b>	<b>31, 33, 34, У3</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2</b>	<b>Тестирование</b>	
2.	Технология структурного программирования.	31, 33, 34		Тестирование	
3.	Инструментальные средства оформления и документирования алгоритмов программ			Тестирование	
4.	Оценка сложности алгоритма: классификация, классы алгоритмов, неразрешимые задачи			Тестирование	
5.	Оценка сложности алгоритмов сортировки	У3		Отчет по ПЗ	
6.	Оценка сложности алгоритмов поиска			Отчет по ПЗ	
7.	Оценка сложности рекурсивных алгоритмов			Отчет по ПЗ	
8.	Оценка сложности эвристических алгоритмов			Отчет по ПЗ	
	<b>Тема 1.1.3 Объектно-ориентированное программирование</b>	<b>31, 32, 35, У1, У4, У5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2</b>	<b>Проверочные работы, тестирование, устный опрос</b>	
9.	Основные принципы объектно-ориентированного программирования	31, 32, 35		Тестирование	
10.	Классы: основные понятия			Проверочная работа	
11.	Перегрузка методов			Проверочная работа	
12.	Операции класса. Иерархия классов				
13.	Синтаксис интерфейсов. Интерфейсы и наследование				
14.	Структуры			Проверочная работа	
15.	Делегаты			Проверочная работа	
16.	Регулярные выражения			Проверочная работа	
17.	Коллекции. Параметризованные классы			Проверочная работа	
18.	Указатели. Операции со списками			Устный опрос	
19.	Работа с классами	У1, У4, У5	Отчет по ЛЗ		

20.	Перегрузка методов			Отчет по ЛЗ	
21.	Определение операций в классе			Отчет по ЛЗ	
22.	Создание наследованных классов			Отчет по ЛЗ	
23.	Работа с объектами через интерфейсы			Отчет по ЛЗ	
24.	Использование стандартных интерфейсов			Отчет по ЛЗ	
25.	Работа с типом данных структура			Отчет по ЛЗ	
26.	Коллекции. Параметризованные классы			Отчет по ЛЗ	
27.	Использование регулярных выражений			Отчет по ЛЗ	
28.	Операции со списками			Отчет по ЛЗ	
	<b>Тема 1.1.4. Паттерны проектирования</b>	<b>31, 32, 35, У1, У4, У5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2</b>	<b>Тестирование</b>	
29.	Назначение и виды паттернов. Основные шаблоны	31, 32, 35			
30.	Порождающие шаблоны				
31.	Структурные шаблоны				
32.	Поведенческие шаблоны				
33.	Использование основных шаблонов. Использование порождающих шаблонов	У1, У4, У5		Отчет по ЛЗ	
34.	Использование структурных шаблонов. Использование поведенческих шаблонов			Отчет по ЛЗ	
	<b>Тема 1.1.5. Событийно-управляемое программирование</b>	<b>31, 32, 33, 34, 35, У1, У2, У3, У4, У5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2</b>	<b>Проверочные работы, тестирование, устный опрос</b>	
35.	Событийно-управляемое программирование	31, 32, 33, 34, 35			Тестирование
36.	Инструменты и библиотеки СОБЫТИЙНО-управляемого программирования				
37.	Создание приложения WPF (WPF.NET).				
38.	Элементы управления. Диалоговые окна				
39.	Обработчики событий				Устный опрос
40.	Введение в графику. Изучение кода XAML				Устный опрос
41.	Добавление кода для события Click				Устный опрос
42.	Виды меню				Устный опрос
43.	Программирование с использованием WPF	У1, У2, У3, У4, У5			Отчет по ЛЗ
44.	Разметка и код программной части				Отчет по ЛЗ
45.	Разработка приложения с использованием текстовых компонентов		Отчет по ЛЗ		
46.	Разработка приложения с несколькими формами		Отчет по ЛЗ		

47.	Разработка приложения с не визуальными компонентами			Отчет по ЛЗ	
48.	<b>Консультация перед экзаменом</b>				
	<b>Тема 1.1.6. Программирование WPF-приложений</b>	<b>31, 32, 33, 34, 35, У1, У2, У3, У4, У5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2</b>	<b>Проверочные работы, устный опрос</b>	<b>экзамен</b>
49.	Всплывающие подсказки	31, 32, 33, 34, 35		Проверочная работа	
50.	Свойства ToolTipService и Popup			Устный опрос	
51.	Класс ScrollViewer			Устный опрос	
52.	Классы GroupBox и TabItem. Элемент Expander			Проверочная работа	
53.	Текстовые элементы управления TextBox и TextBlock			Устный опрос	
54.	Проверка правописания и класс PasswordBox			Устный опрос	
55.	Классы ListBox и ComboBox			Проверочная работа	
56.	Элементы, основанные на диапазонах значений			Устный опрос	
57.	Элементы управления датами			Проверочная работа	
58.	Класс ListView. Шаблоны ячеек ListView			Устный опрос	
59.	Элемент управления TreeView			Устный опрос	
60.	Элемент управления DataGrid. Настройка DataGrid			Проверочная работа	
61.	Форматирование и стилизация столбцов и строк DataGrid				
62.	Контейнеры управления содержимым				
63.	Создание всплывающих подсказок	У1, У4, У5		Отчет по ЛЗ	
64.	Настройка свойств ToolTipService и Popup			Отчет по ЛЗ	
65.	Работа с ScrollViewer			Отчет по ЛЗ	
66.	Работа с TextBox и TextBlock			Отчет по ЛЗ	
67.	Создание ListBox и ComboBox			Отчет по ЛЗ	
68.	Работа с диапазонами и датами			Отчет по ЛЗ	
69.	Работа с ListView			Отчет по ЛЗ	
70.	Работа с контейнерами управления содержимым			Отчет по ЛЗ	
71.	Создание каркаса приложения. Создание и использование стилей			Отчет по ЛЗ	
72.	Разработка игрового приложения			Отчет по ЛЗ	
73.	Разработка приложения с анимацией			Отчет по ЛЗ	
	<b>Тема 1.1.7. Использование систем контроля версий</b>	<b>36, У5</b>	<b>ОК 1 - ОК 9, ПК 1.7</b>	<b>Тестирование</b>	
74.	Назначение и виды систем контроля версий. Работа с системой контроля версий – разработка кода	36			

75.	Типичный порядок работы с системой контроля версий			
76.	Создание веток кода	У5		Отчет по ЛЗ
77.	Объединение изменений в ветках			Отчет по ЛЗ
	<b>Тема 1.1.8. Основы ADO.Net</b>	<b>31, 32, 33, 34, 35, 36, У1, У2, У3, У4, У5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2, ПК 1.7</b>	<b>Тестирование</b>
78.	Работа с базами данных. Архитектура ADO.NET	31, 32, 33, 34, 35, 36		Тестирование
79.	Фундаментальные классы ADO.NET. Генератор поставщиков данных .NET			
80.	Доступ к данным			Устный опрос
81.	Создание таблицы, работа с записями			Устный опрос
82.	Способы создания команд			Тестирование
83.	Entity Framework			Тестирование
84.	Работа с базой данных в приложении: чтение и добавление данных			Тестирование
85.	Работа с базой данных в приложении: редактирование и удаление			
86.	Работа с неструктурированными данными: обработка и импорт в базу данных			
87.	Создание списков (ListView). Поиск и фильтрация данных			Тестирование
88.	Разработка API			
89.	Программная работа с файловой системой с помощью пространства имен System.IO			
90.	Программная работа с таблицами Excel с помощью библиотеки Microsoft.Office.Interop.Excel			
91.	Программная работа с документами Word с помощью библиотеки Microsoft.Office.InteropWord			
92.	Реализация пользовательских элементов управления (UserControl)			
93.	Создание базы данных SQL Server через Visual Studio	У1, У2, У3, У4, У5		Отчет по ЛЗ
94.	Создание скрипта и backup базы данных. Восстановление базы данных из скрипта и backup		Отчет по ЛЗ	
95.	Чтение и добавление данных из БД в приложении		Отчет по ЛЗ	
96.	Связывание приложения Visual Studio с базой данных SQL Server		Отчет по ЛЗ	
97.	Редактирование и удаление из БД в приложении		Отчет по ЛЗ	
98.	Обработка и импорт в базу данных		Отчет по ЛЗ	
99.	Поиск и фильтрация данных		Отчет по ЛЗ	
100.	Работа с API		Отчет по ЛЗ	

101.	Работа с файловой системой			Отчет по ЛЗ	
102.	Работа с таблицами Excel			Отчет по ЛЗ	
103.	Работа с документами Word			Отчет по ЛЗ	
104.	Работа с элементами управления UserControl			Отчет по ЛЗ	
105.	Консультация перед экзаменом				
	<b>Тема 1.1.9. Оптимизация и рефакторинг кода</b>	<b>31, 32, 33, 34, 35, 36, Y1, Y2, Y3, Y4, Y5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2, ПК 1.7</b>	<b>Тестирование</b>	
106.	Методы оптимизации программного кода	31, 32, 33, 34, 35, 36		Тестирование	
107.	Цели и методы рефакторинга			Тестирование	
108.	Генерация и рефакторинг кода в Visual Studio			Тестирование	
109.	Операции рефакторинга			Тестирование	
110.	Генерация кода методом перетащить и отпустить (drag-and-drop)	Y1, Y2, Y3, Y4, Y5		Отчет по ЛЗ	
111.	Генерация кода с помощью технологии IntelliSense			Отчет по ЛЗ	
112.	Оптимизация и рефакторинг кода. Рефакторинг с помощью надстройки Class Designer			Отчет по ЛЗ	
113.	Использование операций рефакторинга		Отчет по ЛЗ		
	<b>Тема 1.1.10. Разработка пользовательского интерфейса</b>	<b>31, 32, 33, 34, 35, 36, Y1, Y2, Y3, Y4, Y5</b>	<b>ОК 1 - ОК 9, ПК 1.1, ПК 1.2, ПК 1.7</b>	<b>Тестирование, устный опрос</b>	<b>экзамен</b>
114.	Правила разработки интерфейсов пользователя	31, 32, 33, 34, 35, 36		Тестирование	
115.	Разработка интерфейса пользователя типа «вопрос - ответ»	Y1, Y2, Y3, Y4, Y5		Отчет по ЛЗ	
116.	Разработка интерфейса пользователя на основе меню и экранных форм			Отчет по ЛЗ	
	<b>Тема 1.1.11. Диаграммы классов в Visual Studio</b>	<b>31, 32, 33, 34, 35, 36, Y1, Y2, Y3, Y4, Y5</b>		Проверочная работа	
117.	Диаграммы классов в Visual Studio	31, 32, 33, 34, 35, 36		Проверочная работа	
118.	Инструментальная панель Toolbox				
119.	Окна Class Details и Properties				
120.	Создание диаграммы классов в Visual Studio	Y1, Y2, Y3, Y4, Y5		Отчет по ЛЗ	
121.	Работа с Инструментальной панелью Toolbox			Отчет по ЛЗ	
122.	Работа с окнами Class Details и Properties			Отчет по ЛЗ	
123.	Компоновка и экспорт диаграмм			Отчет по ЛЗ	
124.	Консультация перед экзаменом				
	<b>Самостоятельная работа обучающихся:</b>				
	– Решение вариантных задач на создание классов – Решение вариантных задач со структурами – Решение вариантных задач с делегатами – Решение вариантных задач с регулярными выражениями – Решение вариантных задач с коллекциями				

	<ul style="list-style-type: none"> <li>– Решение вариантных задач на создание интерфейсов</li> <li>– Решение вариантных задач на обработку баз данных</li> </ul>				
	<b>Курсовое проектирование</b>				Защита курсового проекта
	<ul style="list-style-type: none"> <li>– Анализ требований и уточнение спецификаций</li> <li>– Разработка основных компонентов и диалогов</li> <li>– Разработка алгоритмов</li> <li>– Реализация</li> <li>– Подготовка расчетно-пояснительной записки</li> </ul>				

## 4. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ МДК 01.01. Разработка программных модулей

### 5.1. Типовые задания для оценки знаний и умений

#### ТЕМА 1.1. ЖИЗНЕННЫЙ ЦИКЛ ПО

##### 1. Тест по теме 1.1.1. Жизненный цикл программного обеспечения

**Вопрос 1.** Установите соответствие между понятиями и их определениями:

1. ЖЦ ПО
  2. Спецификация
  3. Визуальное программирование
  4. Сопровождение
- 
- a. Непрерывный процесс, начинающийся с момента принятия решения о создании программного обеспечения и заканчивающийся полным изъятием его из эксплуатации
  - b. Процесс адаптации программного обеспечения к внесению изменений, вызванных возникшими проблемами или потребностями в модификации при сохранении его основных функций
  - c. Способ создания программы путем манипулирования графическими объектами вместо написания текста
  - d. Точное формализованное описание функций и ограничение разрабатываемого программного продукта

Ответ: 1 - a, 2 - d, 3 - c, 4 - b

**Вопрос 2.** К стадиям ЖЦ ПО не относится:

- a. анализ предметной области и формирование требований (концепции)
- b. проектирование
- c. планирование
- d. реализация

Ответ: c

**Вопрос 3.** ЖЦ ПО завершается:

- a. вводом в эксплуатацию
- b. выводом из эксплуатации
- c. сопровождением проекта (эксплуатация)

Ответ: b

**Вопрос 4.** Цели и задачи проекта формулируются на стадии:

- a. реализации
- b. анализа предметной области
- c. тестирования
- d. ввода в эксплуатацию

Ответ: b

**Вопрос 5.** Определение архитектуры программной системы, ее функций, внешних условий функционирования определяются на стадии:

- a. реализации
- b. тестирования
- c. проектирования
- d. ввод в эксплуатацию

Ответ: c



**Вопрос 6.** Какая стратегия основана на полном определении всех требований к разрабатываемому программному средству или системе в начале процесса разработки и представляет собой однократный проход этапов разработки?

- a. инкрементная
- b. эволюционная
- c. каскадная

Ответ: c

**Вопрос 7.** Какая стратегия представляет собой поэтапную, следующую временным графикам стратегию, в которой разные части системы разрабатываются в разное время и разными темпами, и если одна часть готова, то ее интегрируют в систему?

- a. инкрементная
- b. эволюционная
- c. каскадная

Ответ: a

**Вопрос 8.** Какая стратегия представляет собой многократный проход этапов разработки?

- a. инкрементная
- b. каскадная
- c. эволюционная

Ответ: c

**Вопрос 9.** Под моделью предметной области понимается:

- a. точное количество необходимых итераций и сложность определения критериев для процесса разработки
- b. некоторая система, имитирующая структуру или функционирование исследуемой предметной области и отвечающая основному требованию
- c. частичное определение требований к разрабатываемому программному средству или системе в начале процесса разработки

Ответ: b

**Вопрос 10.** Объектная модель отражает:

- a. состав взаимодействующих в процессах материальных и информационных объектов предметной области
- b. взаимосвязь функций (действий) по преобразованию объектов в процессах
- c. топологию расположения и способы коммуникации комплекса технических средств

Ответ: a

## ТЕМА 1.1.2. СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

### 2 – 3. Тест «Структурное программирование»

**Вопрос 1.** Укажите международный стандарт, регламентирующий ЖЦ ПО

- a. ISO/IEC 9127
- b. ISO/IEC 9126
- c. ISO/IEC 9294
- d. ISO/IEC 12207

Ответ: d

**Вопрос 2.** Укажите этап ЖЦ ПО, на котором формулируются цели и задачи проекта

- a. Проектирование
- b. Эксплуатация (сопровождение проекта)
- c. Тестирование
- d. Анализ предметной области
- e. Реализация

f. Ввод в эксплуатацию

Ответ: d

**Вопрос 3.** Укажите действия, осуществляемые на стадии анализа и формирования требований (выберите несколько вариантов)

- a. Разработка структуры данных
- b. Проектирование архитектуры программной системы
- c. Анализ существующих аналогичных программных продуктов
- d. Планирование работ
- e. Проведение обследования автоматизируемого объекта

Ответ: a, c, d, e

**Вопрос 4.** Укажите основные задачи этапа планирования работ при создании ПО (выберите несколько вариантов)

- a. Разработка предварительной версии пользовательской документации
- b. Построение плана-графика выполнения работ
- c. Предварительная экономическая оценка проекта
- d. Разработка и документирование детального проекта базы данных
- e. Определение целей разработки

Ответ: b, c, e

**Вопрос 5.** Укажите операции проведения обследования автоматизируемого объекта (выберете несколько вариантов)

- a. Построение моделей предметной области на основании результатов обследования
- b. Эксплуатационное тестирование
- c. Разработка и документирование программных интерфейсов
- d. Анализ информации, поступающей из внешних источников
- e. Анализ внутренних информационных потоков
- f. Предварительное определение требований к программному обеспечению

Ответ: a, d, e

**Вопрос 6.** Укажите модифицируемую и расширяемую рабочую модель разрабатываемого ПО, позволяющую получить представление о ключевых свойствах до полной реализации

- a. Проект
- b. Макет
- c. Прототип
- d. Конфигуратор

Ответ: c

**Вопрос 7.** Процесс обобщения, при котором внимание концентрируется на сходстве явлений и предметов, и они объединяются в группы на основе этого сходства – это:

- a. модульность
- b. абстрагирование
- c. иерархичность
- d. функциональность

Ответ: b

**Вопрос 8.** Схема, отображающая уровни отдельных модулей ПО, их функции и взаимодействие модулей разных уровней – это:

- a. блок-схема
- b. абстракция
- c. иерархия
- d. концентрирование

Ответ: c

**Вопрос 9.** Иерархический подход к проектированию – это:

- a. подход, при котором обработка исключительных ситуаций, связанных с неправильными данными, должна программироваться в первую очередь
- b. подход, при котором порядок программирования и тестирования модулей определяется их расположением в схеме иерархии
- c. подход, при котором модули разрабатываются в порядке их выполнения при запуске готовой программы

Ответ: b

**Вопрос 10.** Операционный подход к проектированию – это:

- a. подход, при котором обработка исключительных ситуаций, связанных с неправильными данными, должна программироваться в первую очередь
- b. подход, при котором порядок программирования и тестирования модулей определяется их расположением в схеме иерархии
- c. подход, при котором модули разрабатываются в порядке их выполнения при запуске готовой программы

Ответ: c

#### 4. Тест «Оценка сложности алгоритма»

**Вопрос 1.** Количественная характеристика ресурсов, необходимых алгоритму для работы – это

- a. процессорное время
- b. сложность
- c. ёмкость
- d. модульность

Ответ: b

**Вопрос 2.** Экземпляры задачи – это:

- a. различные входные данные
- b. переменные
- c. алгоритмы
- d. блок-схемы

Ответ: a

**Вопрос 3.** Фактическая сложность алгоритма зависит от ... (выберите несколько вариантов):

- a. скорости работы компьютера
- b. количества переменных и констант
- c. алгоритма
- d. количества экземпляров задачи

Ответ: a, c

**Вопрос 4.** Сложность задачи – это:

- a. сложность любого алгоритма решения данной задачи
- b. сложность самого неэффективного алгоритма решения данной задачи
- c. сложность самого эффективного алгоритма решения данной задачи

Ответ: c

**Вопрос 5.** «Существует такая алгоритмически разрешимая задача  $Z$ , для которой любой алгоритм  $A$ , решающий задачу  $Z$ , можно ускорить следующим образом: существует другой алгоритм  $A'$ , также решающий  $Z$  и такой, что  $T_{A'}(n) \leq \log T_A(n)$  для почти всех  $n$ » - это:

- a. Теорема Блюма

- b. Теорема Дейкстры
- c. Теорема структуры
- d. Теорема сложности

Ответ: a

### ТЕМА 1.1.3. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

#### 9. Тест «Основные принципы объектно-ориентированного программирования»

**Вопрос 1.** Какой из перечисленных подходов к программированию является наиболее поздним?

- a. структурный
- b. объектно-ориентированный
- c. компонентно-ориентированный

Ответ: c

**Вопрос 2.** Какие из перечисленных языков программирования основаны на объектно-ориентированном подходе?

- a. C# и SML
- b. C# и C++
- c. C# и Haskell

Ответ: b

**Вопрос 3.** Что отличает императивные языки программирования от декларативных? (выберите несколько вариантов ответов)

- a. степень зависимости от среды реализации
- b. стиль программирования
- c. структура программы

Ответ: b, c

**Вопрос 4.** Каковы основные принципы объектно-ориентированного программирования? (выберите несколько вариантов ответов)

- a. абстракция, аппликация, инкапсуляция
- b. абстракция, инкапсуляция, полиморфизм
- c. наследование, инкапсуляция, полиморфизм

Ответ: b, c

**Вопрос 5.** Что отличает объектно-ориентированный подход к программированию от компонентно-ориентированного?

- a. степень зависимости от среды реализации
- b. стиль программирования
- c. структура программы

Ответ: c

**Вопрос 6.** В чем состоит особенность языков объектно-ориентированного программирования?

- a. этот класс языков основан на сценариях
- b. этот класс языков концептуально близок к любой предметной области
- c. этот класс языков является наиболее машинно-независимым

Ответ: b

**Вопрос 7.** Что понимается в ООП под термином "абстракция"?

- a. выражение языка, являющееся идентификатором
- b. выражение языка, отличное от идентификатора
- c. выражение языка, моделирующее лямбда-абстракцию

Ответ: b

**Вопрос 8.** В чем состоит особенность языков функционального программирования?

- a. этот класс языков основан на сценариях
- b. этот класс языков концептуально близок к любой предметной области
- c. этот класс языков легко формализуем математически

Ответ: c

**Вопрос 9.** Что понимается под термином «наследование»?

- a. свойство производного объекта сохранять атрибуты базового
- b. свойство производного объекта сохранять операции базового
- c. свойство производного объекта сохранять поведение базового

Ответ: c

**Вопрос 10.** Какие виды наследования существуют в языке C#?

- a. единичное и множественное
- b. только единичное
- c. только множественное

Ответ: a

**Вопрос 11.** В каком отношении находятся базовый класс и подкласс? (выберите несколько вариантов)

- a. подкласс наследует свойства базового класса
- b. подкласс расширяет базовый класс
- c. базовый класс наследует свойства подкласса

Ответ: a, b

**Вопрос 12.** В случае единичного наследования класс может наследовать свойства:

- a. структуры
- b. множественных классов
- c. единственного класса

Ответ: c

**Вопрос 13.** Свойства какого объекта наследует подкласс с неявным базовым классом?

- a. свойства класса object
- b. свойства структуры
- c. свойства интерфейса

Ответ: a

**Вопрос 14.** Что понимается под термином "инкапсуляция"?

- a. доступность объекта лишь посредством внешних методов
- b. доступность объекта лишь посредством его свойств и методов
- c. доступность объекта лишь посредством свойств других объектов

Ответ: b

**10. Проверочная работа «Классы: основные понятия»**

## Вопрос 1

Дан следующий класс:

```
1 class Person
2 {
3     public string name = "Sam";
4     public int age;
5
6     public Person(string name, int age)
7     {
8         this.name = name;
9         this.age = age;
10    }
11 }
```

Какое значение поле `name` будет иметь при выполнении следующего кода и почему?

```
1 Person tom = new Person("Tom", 34) { name = "Bob", age = 29 };
```

*Ответ:*

*Поле `name` объекта `tom` будет иметь значение "Bob". В самом начале выполняется начальная инициализация полей класса, и полю `name` присваивается значение "Sam". Далее выполняется конструктор, который устанавливает для переменной `name` значение "Tom". И в последнюю очередь выполняется инициализатор, который устанавливает для поля `name` значение "Bob".*

## Вопрос 2

Дан следующий класс:

```
1 class Person
2 {
3     public string name = "Ben";
4     public int age = 18;
5     public string email = "ben@gmail.com";
6
7     public Person(string name)
8     {
9         this.name = name;
10    }
11    public Person(string name, int age) : this(name)
12    {
13        this.age = age;
14    }
15    public Person(string name, int age, string email) : this("Bob", age)
16    {
17        this.email = email;
18    }
19 }
```

Какие значения будут иметь поля `name`, `age` и `email` после выполнения следующего кода и почему? В каком порядке будут вызываться конструкторы класса `Person`?

```
1 Person person = new Person("Tom", 31, "tom@gmail.com");
```

*Ответ*

*Значения полей:*

- *name: Bob*
- *age: 31*
- *email: tom@gmail.com*

Порядок выполнения конструкторов:

1. Вначале вызывается конструктор с тремя параметрами: `Person(string name, int age, string email)`. Он вызывает конструктор с двумя параметрами `Person(string name, int age)`. Этому конструктору передаются значения `name: "Bob", age: 31`
2. Далее вызывается конструктор с двумя параметрами: `Person(string name, int age)`. Он вызывает конструктор с одним параметром `Person(string name)`. Этому конструктору передаются значения `name: "Bob"`
3. Далее вызывается конструктор с одним параметром: `Person(string name)`. Он получает строку `"Bob"` и устанавливает ее в качестве значения для поля `name`
4. Затем выполняется конструктор с двумя параметрами: `Person(string name, int age)`. Он устанавливает число `31` в качестве значения для поля `age`
5. Далее выполняется конструктор с тремя параметрами: `Person(string name, int age, string email)`. Он устанавливает строку `"tom@gmail.com"` в качестве значения для поля `email`

### Вопрос 3

Какие модификаторы доступа есть в C# 7?

Ответ

В C# 7 есть следующие модификаторы доступа:

- `private`
- `private protected`
- `protected`
- `internal`
- `protected internal`
- `public`

### Вопрос 4

Вам надо определить в классе переменную, которая должна быть доступна из любого места в текущем проекте. Какой модификатор (или модификаторы, если их несколько) вы будете использовать?

Ответ

- `internal`
- `protected internal`
- `public`

### Вопрос 5

В чем различие между модификаторами **protected** и **private protected**?

Ответ

Члены класса, определенные с этими модификаторами, могут быть доступны только в текущем классе или в классах-наследниках. Но если модификатор `protected` гарантирует доступ для любого класса-наследника вне зависимости от того, в каком проекте (сборке) он расположен. А модификатор **`private protected`** делает члены класса доступными только для классов-наследников, которые размещены в том же проекте, где определен исходный класс.

Ответ

Члены класса, определенные с этими модификаторами, могут быть доступны только в текущем классе или в классах-наследниках. Но если модификатор `protected` гарантирует доступ для любого класса-наследника вне зависимости от того, в каком проекте (сборке) он расположен. А модификатор **`private protected`** делает члены класса доступными только для классов-наследников, которые размещены в том же проекте, где определен исходный класс.

### Вопрос 6

Если классы и члены класса не имеют никаких модификаторов, какие модификаторы доступа к ним применяются по умолчанию?

*Ответ*

*Для классов, размещенных непосредственно в пространстве имен, модификатор доступа по умолчанию - **internal**. Для классов, размещенных внутри других классов (вложенных классов), модификатор по умолчанию **private**. А для членов класса модификатор доступа по умолчанию - **private**.*

### Вопрос 7

Что выведет на консоль следующая программа и почему?

```
1 class Person
2 {
3     int age = 26;
4     string name = "Tom";
5
6     public Person(int age, string name)
7     {
8         this.age = age;
9         this.name = name;
10    }
11 }
12 class Program
13 {
14     static void Main(string[] args)
15     {
16         Person person = new Person(19, "Bob");
17         Console.WriteLine(person.name);
18
19         Console.ReadKey();
20     }
21 }
```

*Ответ*

*Программа не скомпилируется, поскольку переменная `name`, к которой мы хотим обратиться в методе `Main`, не имеет модификаторов. Следовательно, по умолчанию она имеет модификатор доступа **private**. Поэтому вне класса `Person` к этой переменной мы обратиться не можем.*

## 11 - 13. Проверочная работа «Перегрузка методов. Операции класса. Синтаксис интерфейсов»

### Вопрос 1

Что будет выведено на консоль в результате выполнения следующей программы?

```
1 class Person
2 {
3     private int age = 15;
4
5     public int Age
6     {
7         get { return age; }
8         set { }
```



```

9     }
10    }
11    class Program
12    {
13        static void Main(string[] args)
14        {
15            Person tom = new Person();
16            tom.Age = 25;
17            Console.WriteLine(tom.Age);
18
19            Console.ReadKey();
20        }
21    }

```

Варианты ответов:

- 15
- 25
- 0
- Программа не скомпилируется

*Ответ: На консоль будет выведено число 15*

## Вопрос 2

Зачем нужны автосвойства, не проще ли их заменить обычными переменными?

*Ответ*

*При необходимости автосвойства можно развернуть в полноценные свойства и через них управлять доступом к переменным. Если бы мы использовали обычные переменные вместо автосвойств, то при необходимости ограничить доступ к переменным нам, возможно, пришлось исправить кучу имеющегося кода в программе.*

## Вопрос 3

Что будет выведено на консоль в результате выполнения следующей программы и почему?

```

1    class Person
2    {
3        internal string Name { get; set; } = "Bob";
4    }
5    class Program
6    {
7        static void Main(string[] args)
8        {
9            Person tom = new Person { Name = "Tom" };
10           Console.WriteLine(tom.Name);
11
12           Console.ReadKey();
13        }
14    }

```

*Ответ*

*На консоль будет выведена строка "Tom". Потому что для автосвойства будет создана переменная, которой будет присвоено значение "Bob". Однако во время выполнения программы при инициализации объекта это значение будет заменяться на строку "Tom".*

## Вопрос 4

Что будет выведено на консоль в результате выполнения следующей программы и почему?

```
1 class Person
2 {
3     internal string Name { internal get; set; } = "Bob";
4 }
5 class Program
6 {
7     static void Main(string[] args)
8     {
9         Person tom = new Person { Name = "Tom" };
10        Console.WriteLine(tom.Name);
11
12        Console.ReadKey();
13    }
14 }
```

*Ответ*

*Данная программа не скомпилируется, потому что блок get имеет тот же модификатор доступа, что и свойство. Для блока get в данном случае можно использовать более ограничивающие модификаторы - private.*

### **Вопрос 5**

Каким образом можно перегрузить метод?

*Ответ*

- *Определить версию метода с другим количеством параметров*
- *Определить версию метода с другими типами параметров*
- *Определить версию метода, где порядок параметров будет иным*
- *Изменить модификаторы параметров*

### **Вопрос 6**

Корректна ли следующая перегрузка методов? Если да, то почему? Если нет, то почему?

```
1 static void IncrementVal(ref int val)
2 {
3     val++;
4     Console.WriteLine(val);
5 }
6
7 static void IncrementVal(int val)
8 {
9     val++;
10    Console.WriteLine(val);
11 }
```

*Ответ*

*Перегрузка корректна, потому что параметры метода имеют разные модификаторы*

## **14. Проверочная работа «Структуры»**

### **Вопрос 1**

Что будет выведено на консоль при выполнении следующей программы и почему?

```
1 struct Person
```

```

2  {
3    public string name = "Sam";
4  }
5  class Program
6  {
7    static void Main(string[] args)
8    {
9      Person person;
10     person.name = "Bob";
11     Console.WriteLine(person.name);
12  }
13 }

```

*Ответ*

*Программа не скомпилируется, поскольку в структурах нельзя инициализировать глобальные переменные при их объявлении.*

### **Вопрос 2**

Почему не компилируется следующая программа:

```

1  struct Person
2  {
3    public string name;
4  }
5  class Program
6  {
7    static void Main(string[] args)
8    {
9      Person person;
10     Console.WriteLine(person.name);
11     person.name = "Bob";
12  }
13 }

```

*Ответ*

*Если для создания объекта структуры не используется конструктор, то перед обращением к полям и методам подобного объекта необходимо проинициализировать все поля этого объекта.*

### **Вопрос 3**

Почему не компилируется следующая программа:

```

1  struct Person
2  {
3    public string name;
4    public int age;
5    public Person(string n)
6    {
7      name = n;
8    }
9  }
10 class Program
11 {
12 static void Main(string[] args)
13 {
14     Person person;

```

```
15     person.name = "Bob";
16   }
17 }
```

*Ответ*

*Если структура определяет конструктор, то в нем должны быть инициализированы все поля этой структуры. В примере из вопроса инициализировано только поле name.*

## 15. Проверочная работа «Делегаты»

### Вопрос 1

Есть следующий делегат:

```
1 delegate int Operation (int val);
```

Какие из следующих методов соответствуют данному делегату:

```
1 static void Method1(int x)
2 {
3     Console.WriteLine(x * x);
4 }
5 static int Method2(ref int x)
6 {
7     return x * x;
8 }
9 static int Method3(int x, int y)
10 {
11     return x * y;
12 }
13 static int Method4(int x, int y = 7)
14 {
15     return x * y;
16 }
17 static int Method5(out int x)
18 {
19     x = 7;
20     return x * x;
21 }
```

*Ответ*

*Ни один из предложенных методов не соответствует делегату Operation.*

### Вопрос 2

Что будет выведено на консоль в результате выполнения следующей программы:

```
1 class Program
2 {
3     delegate int Operation(int x, int y);
4
5     static void Main(string[] args)
6     {
7         Operation del = Add;
8         del += Multiply;
9
10        int result = del(6, 5);
11        Console.WriteLine(result);
12        Console.Read();
13    }
```

```
14 private static int Add(int x, int y) { return x + y; }
15 private static int Multiply(int x, int y) { return x * y; }
16 }
```

*Ответ*

*На консоль будет выведено число 30 - результат последнего метода, добавленного в список методов делегата.*

### **Вопрос 3**

Какой будет консольный вывод при выполнении следующей программы:

```
1 class Program
2 {
3     delegate void Message();
4
5     static void Main(string[] args)
6     {
7         Message mes1 = Hello;
8         mes1 += HowAreYou;
9         mes1 += Hello;
10        mes1 += Hello;
11        mes1 -= Hello;
12        mes1();
13
14        Console.Read();
15    }
16    private static void Hello() { Console.WriteLine("Hello"); }
17    private static void HowAreYou() { Console.WriteLine("How are you?"); }
18 }
```

*Ответ*

*Консольный вывод данной программы:*

```
Hello
How are you?
Hello
```

### **Вопрос 4**

Что будет выведено на консоль в результате выполнения следующей программы:

```
1 class Program
2 {
3     delegate void Message();
4
5     static void Main(string[] args)
6     {
7         Message mes = new Message(Hello);
8         mes += Hi;
9         mes -= Hello;
10        mes -= Hi;
11        mes();
12        Console.Read();
13    }
14    private static void Hello() { Console.WriteLine("Hello"); }
15    private static void Hi() { Console.WriteLine("Hi"); }
16 }
```

*Ответ*

*При выполнении программы, точнее при выполнении делегата возникнет исключение, поскольку список вызова делегата пуст.*

### **Вопрос 5**

Что будет выведено на консоль в результате выполнения следующей программы?

```
1 class Person
2 {
3     private int age = 15;
4
5     public int Age
6     {
7         get { return age; }
8         set { }
9     }
10 }
11 class Program
12 {
13     static void Main(string[] args)
14     {
15         Person tom = new Person();
16         tom.Age = 25;
17         Console.WriteLine(tom.Age);
18
19         Console.ReadKey();
20     }
21 }
```

### **Вопрос 6**

Зачем нужны автосвойства, не проще ли их заменить обычными переменными?

### **Вопрос 7**

Что будет выведено на консоль в результате выполнения следующей программы и почему?

```
1 class Person
2 {
3     internal string Name { get; set; } = "Bob";
4 }
5 class Program
6 {
7     static void Main(string[] args)
8     {
9         Person tom = new Person { Name = "Tom" };
10        Console.WriteLine(tom.Name);
11
12        Console.ReadKey();
13    }
14 }
```

### **Вопрос 8**

Что будет выведено на консоль в результате выполнения следующей программы и почему?

```

1 class Person
2 {
3     internal string Name { internal get; set; } = "Bob";
4 }
5 class Program
6 {
7     static void Main(string[] args)
8     {
9         Person tom = new Person { Name = "Tom" };
10        Console.WriteLine(tom.Name);
11
12        Console.ReadKey();
13    }
14 }

```

### Вопрос 9

Есть следующий делегат:

```
1 delegate int Operation (int val);
```

Какие из следующих методов соответствуют данному делегату:

```

1 static void Method1(int x)
2 {
3     Console.WriteLine(x * x);
4 }
5 static int Method2(ref int x)
6 {
7     return x * x;
8 }
9 static int Method3(int x, int y)
10 {
11     return x * y;
12 }
13 static int Method4(int x, int y = 7)
14 {
15     return x * y;
16 }
17 static int Method5(out int x)
18 {
19     x = 7;
20     return x * x;
21 }

```

### Вопрос10

Что будет выведено на консоль в результате выполнения следующей программы:

```

1 class Program
2 {
3     delegate int Operation(int x, int y);
4
5     static void Main(string[] args)
6     {
7         Operation del = Add;
8         del += Multiply;
9

```

```

10     int result = del(6, 5);
11     Console.WriteLine(result);
12     Console.Read();
13 }
14 private static int Add(int x, int y) { return x + y; }
15 private static int Multiply(int x, int y) { return x * y; }
16 }

```

### Вопрос 11

Какой будет консольный вывод при выполнении следующей программы:

```

1 class Program
2 {
3     delegate void Message();
4
5     static void Main(string[] args)
6     {
7         Message mes1 = Hello;
8         mes1 += HowAreYou;
9         mes1 += Hello;
10        mes1 += Hello;
11        mes1 -= Hello;
12        mes1();
13
14        Console.Read();
15    }
16    private static void Hello() { Console.WriteLine("Hello"); }
17    private static void HowAreYou() { Console.WriteLine("How are you?"); }
18 }

```

### Вопрос 12

Что будет выведено на консоль в результате выполнения следующей программы:

```

1 class Program
2 {
3     delegate void Message();
4
5     static void Main(string[] args)
6     {
7         Message mes = new Message(Hello);
8         mes += Hi;
9         mes -= Hello;
10        mes -= Hi;
11        mes();
12        Console.Read();
13    }
14    private static void Hello() { Console.WriteLine("Hello"); }
15    private static void Hi() { Console.WriteLine("Hi"); }
16 }

```

### 16. Проверочная работа «Регулярные выражения – Операции со строками»

1. Сравнивает две строки с учетом текущей культуры (локали) пользователя -
2. Сравнивает две строки без учета локали –
3. Удаляет начальные и конечные пробелы из строки -



4. Форматирует строку -
5. Замещает в строке символ или подстроку другим символом или подстрокой -
6. Вставляет в строку подстроку -
7. Извлекает из строки подстроку, начиная с указанной позиции -
8. Переводит все символы строки в нижний регистр -
9. Соединяет элементы массива строк -
10. Определяет, совпадает ли конец строки с подстрокой -
11. Копирует часть строки, начиная с определенного индекса в массив -
12. Определяет, содержится ли подстрока в строке -
13. Соединяет строки -
14. Разделяет одну строку на массив строк -
15. Находит индекс первого вхождения символа или подстроки в строке -
16. Находит индекс последнего вхождения символа или подстроки в строке -
17. Переводит все символы строки в верхний регистр -

## 17. Проверочная работа «Коллекции»

### 1. Класс ArrayList

1. Добавить в список объект -
2. Удалить из списка все элементы -
3. Проверить, содержится ли в списке объект -
4. Копировать текущий список в массив -
5. Возвратить новый список ArrayList, который содержит определенное количество элементов текущего списка, начиная с указанного индекса -
6. Возвратить индекс элемента -
7. Вставить в список объект по заданному по индексу -
8. Вставить в список коллекцию, начиная с заданного индекса -
9. Возвратить индекс последнего вхождения в списке объекта -
10. Удалить из списка элемент по индексу -
11. Удаляет из списка заданное количество элементов, начиная с указанного индекса
12. Перевернуть список
13. Копировать в список элементы коллекции, начиная с заданного индекса -
14. Сортировать коллекцию -

### 2. Список List<T>

1. Добавить новый элемент в список -
2. Добавить в список коллекцию или массив -
3. Бинарный поиск элемента в списке -
4. Возвратить индекс первого вхождения элемента в списке -
5. Вставить элемент в список на заданную позицию -
6. Удалить элемент из списка -
7. Сортировать список -

### 3. Двухсвязный список LinkedList<T>

1. Вставить узел в список после указанного узла -
2. Вставить в список новый узел с заданным значением после указанного узла -
3. Вставить в список узел перед указанным узлом -
4. Вставить в список новый узел с заданным значением перед узлом -
5. Вставить новый узел в начало списка -
6. Вставляет новый узел с заданным значением в начало списка -
7. Вставить новый узел в конец списка -
8. Вставить новый узел с заданным значением в конец списка -
9. Удалить первый узел из списка -
10. Удалить последний узел из списка -

## ТЕМА 1.1.4. ПАТТЕРНЫ ПРОЕКТИРОВАНИЯ

### 29 – 32. Тест «Паттерны проектирования»

**Вопрос 1.** Какой из перечисленных паттернов применяется в ситуации, когда нужно произвести некоторые действия над всеми объектами структуры в зависимости от фактического типа каждого объекта?

- a) Visitor
- b) Strategy
- c) State
- d) Composite

Ответ: d

**Вопрос 2.** Для чего нужен паттерн Strategy?

- a) Для определения набора классов и стратегии их создания
- b) Для определения семейства алгоритмов и их инкапсуляции для обеспечения взаимозаменяемости
- c) Для определения порядка (стратегии) при создании объекта класса

Ответ: b

**Вопрос 3.** Назначением какого паттерна проектирования является предоставление удобного интерфейса к громоздкому и сложному API?

- a) Factory Method
- b) Iterator
- c) Decorator
- d) Strategy
- e) Façade

Ответ: e

**Вопрос 4.** Какой шаблон подменяет собой сложный объект и контролирует доступ к нему?

- a) Proxy
- b) Adapter
- c) Decorator
- d) Bridge

Ответ: a

**Вопрос 5.** Какие основные задачи решает применение паттернов проектирования?

- a) Производится унификация терминологии, названий модулей и элементов проекта
- b) Показываются отношения и взаимодействия между классами или объектами
- c) Облегчается дискуссия об абстрактных структурах данных между разработчиками
- d) Описывается решение целого класса абстрактных проблем

Ответ: d

**Вопрос 6.** Какой шаблон проектирования можно определить следующей фразой: " .. - поведенческий шаблон проектирования, обеспечивающий взаимодействие множества объектов, формируя при этом слабую связанность и избавляя объекты от необходимости явно ссылаться друг на друга " ?

- a) Mediator
- b) Low Coupling

- c) High Cohesion
- d) Information Expert

Ответ: a

**Вопрос 7.** Когда следует использовать паттерн Приспособленец (Flyweight)?

Когда большинство состояний объектов могут быть сохранены на диске или рассчитаны на время исполнения

- a) Когда приложение использует большое количество однообразных объектов, из-за чего происходит выделение большого количества памяти
- b) Когда необходимо обеспечить выполнение очереди запросов, а также их возможную отмену
- c) Когда нужно сохранить состояние объекта для возможного последующего восстановления

Ответ: a

**Вопрос 8.** Когда следует использовать паттерн Хранитель (Memento)?

- a) Когда необходимо обеспечить выполнение очереди запросов, а также их возможную отмену
- b) Когда система должна быть независимой от процесса создания новых объектов и расширяемой: в нее можно легко вводить новые классы, объекты которых система должна создавать
- c) Когда нужно сохранить состояние объекта для возможного последующего восстановления
- d) Когда необходимо, чтобы для класса существовал только один экземпляр

Ответ: c

**Вопрос 9.** Какой шаблон проектирования можно определить следующей фразой: «это паттерн, который определяет интерфейс для создания объектов некоторого класса, но непосредственное решение о том, объект какого класса создавать происходит в подклассах»:

- a) Command
- b) Decorator
- c) Flyweight
- d) Factory Method

Ответ: d

**Вопрос 10.** К порождающим шаблонам относится:

- a) Strategy
- b) Singleton
- c) Adapter
- d) Bridge

Ответ: b

**Вопрос 11.** К поведенческим шаблонам относится:

- a) Builder
- b) State
- c) Composite
- d) Adapter

Ответ: b

**Вопрос 12.** К структурным шаблонам относится:

- a) Decorator

- b) Mediator
- c) Observer
- d) Prototype

Ответ: а

## ТЕМА 1.1.5. СОБЫТИЙНО-УПРАВЛЯЕМОЕ ПРОГРАММИРОВАНИЕ

### 35 – 38. Тест «Событийно-управляемое программирование. Разработка приложений на базе WPF»

**Вопрос 1.** Язык XAML это ...

- (1) язык расширенной разметки приложений**
- (2) язык расширенной разметки
- (3) язык программирования
- (4) язык манипулирования данными

**Вопрос 2.** XAML-документ содержит ...

*Выберите несколько вариантов ответа*

- (1) разметку, описывающую внешний вид и поведение окна приложения**
- (2) разметку, описывающую внешний вид и поведение страницы приложения**
- (3) логику приложения
- (4) бизнес-логику приложения

**Вопрос 3.** Укажите правильное утверждение

- (1) WPF XAML является подмножеством языка XML**
- (2) XML является подмножеством языка WPF XAML
- (3) XAML является подмножеством языка C#
- (4) C# является подмножеством языка XAML

**Вопрос 4.** Укажите правильные утверждения в отношении синтаксиса XAML

*Выберите несколько вариантов ответа*

- (1) элемент XAML-документа отображается на некоторый экземпляр класса .NET**
- (2) элементы XAML можно вкладывать друг в друга**
- (3) свойства класса определяются с помощью атрибутов**
- (4) свойства класса определяют бизнес-логику

**Вопрос 5.** Все дескрипторы XAML-документа начинаются символом ...

- (1) >
- (2) <**
- (3) &
- (4) :

**Вопрос 6.** Текст XAML-документа должен содержать ...

- (1) один корневой элемент**
- (2) два корневых элемента
- (3) любое количество корневых элементов
- (4) только корневые элементы

**Вопрос 7.** Пространство имен в XAML-документе задается с помощью атрибута

- (1) xmlns**
- (2) http

- (3) window
- (4) page

**Вопрос 8.** Соответствие между ключевым словом XAML и его назначением

- (1) **x:Array** Представляет тип массива .NET на XAML
- x:FieldModifier** Позволяет определять видимость типа класс
- x:ClassModifier** Позволяет определять видимость члена типа
- x:Key** Позволяет установить значение ключа для элемента XAML
- (2) **x:Array** Представляет тип массива .NET на XAML
- x:ClassModifier** Позволяет определять видимость типа класс
- x:FieldModifier** Позволяет определять видимость члена типа
- x:Key** Позволяет установить значение ключа для элемента XAML
- (3) **x:Key** Представляет тип массива .NET на XAML
- x:ClassModifier** Позволяет определять видимость типа класс
- x:FieldModifier** Позволяет определять видимость члена типа
- x:Array** Позволяет установить значение ключа для элемента XAML

**Вопрос 9.** Соответствие между ключевым словом XAML и его назначением

- (1) **x:Name** Позволяет указывать сгенерированное C# имя заданного элемента XAML
- x:Null** Представляет null-ссылку
- x:Static** Позволяет ссылаться на статический член типа
- x:TypeArgument** Позволяет устанавливать элемент как обобщенный тип с определенными параметрами
- (2) **x:TypeArgument** Позволяет указывать сгенерированное C# имя заданного элемента XAML
- x:Null** Представляет null-ссылку
- x:Static** Позволяет ссылаться на статический член типа
- x:Name** Позволяет устанавливать элемент как обобщенный тип с определенными параметрами
- (3) **x:Name** Позволяет указывать сгенерированное C# имя заданного элемента XAML
- x:Null** Представляет null-ссылку
- x:TypeArgument** Позволяет ссылаться на статический член типа
- x:Static** Позволяет устанавливать элемент как обобщенный тип с определенными параметрами

**Вопрос 10.** Пространство имен: <http://schemas.microsoft.com/winfx/2006/xaml/presentation>

- (1) базовое пространство имен WPF, которое охватывает все классы WPF
- (2) пространство имен XAML, которое включает различные свойства утилит XAML
- (3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой
- (4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

**Вопрос 11.** Пространство имен: <http://schemas.microsoft.com/winfx/2006/xaml>

- (1) базовое пространство имен WPF, которое охватывает все классы WPF
- (2) пространство имен XAML, которое включает различные свойства утилит XAML
- (3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой
- (4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

**Вопрос 12.** Строка XAML-документа

<Window x:Class="MyProject.MainWindow" ...>

определяет ...

**(1) создание дочернего класса Window – MyProject.MainWindow**

(2) префикс x атрибута Class определяет то, что данный атрибут помещается в пространство имен <http://schemas.microsoft.com/expression/blend/2008>

(3) префикс x атрибута Class определяет то, что данный атрибут помещается в пространство имен <http://schemas.openxmlformats.org/markup-compatibility/2006>

**Вопрос 13.** В следующем XAML- описании

<Button Name="Exit" Content="Выход" />

фрагмент Name="Exit" определяет

**(1) простое свойство**

(2) сложное свойство

(3) присоединенное свойство

(4) расширение разметки

14. Дескриптор <StackPanel.Background> определяет

(1) простое свойство

**(2) сложное свойство**

(3) присоединенное свойство

(4) расширение разметки

**Вопрос 15.** В следующем XAML- описании

<Button Foreground="{x:Static SystemColors.ActiveCaptionBrush}" />

фрагмент Foreground="{x:Static SystemColors.ActiveCaptionBrush}" определяет

(1) простое свойство

(2) сложное свойство

(3) присоединенное свойство

**(4) расширение разметки**

**Вопрос 16.** В следующем XAML- описании

<Button Name="Exit" Content="Выход" Click="Exit\_Click" />

фрагмент Click="Exit\_Click" определяет

(1) простое свойство

(2) сложное свойство

(3) присоединенное свойство

**(4) присоединение обработчика события**

### **39. Проверочная работа «Размещение элементов управления»**

1. Для определения строк в Grid используется свойство – ... *RowDefinitions*

2. Для определения столбцов в Grid используется свойство - ... *ColumnDefinitions:*

3. Чтобы задать позицию элемента управления с привязкой к определенной ячейке Grid, в разметке элемента нужно прописать значения свойств - ... *Grid.Column* и *Grid.Row*,

4. Если мы хотим растянуть элемент управления на несколько строк или столбцов, то нужно указать свойства - ... *Grid.ColumnSpan* и *Grid.RowSpan*

5. Видимость сетки у элемента Grid задает атрибут - ... *ShowGridLines="True"*

6. Как задать автоматическую ширину столбца Grid - ... *<ColumnDefinition Width="Auto" />*

7. Как задать автоматическую высоту строки Grid - ... *<RowDefinition Height="Auto" />*

8. Как задать пропорциональные размеры столбцов Grid - ... *<ColumnDefinition Width="\*/>*

9. Контейнер, задающий строки и столбцы одинакового размера - ... *UniformGrid*

10. Элемент, который задает разделитель между столбцами или строками - ... *GridSplitter*

11. Контейнеры, которые располагают все элементы в ряд либо по горизонтали, либо по вертикали в зависимости от ориентации - ... *StackPanel* или *WrapPanel*
12. Для того, чтобы в предыдущих элементах задать горизонтальный и вертикальный стек, нужно указать свойство - ... *Orientation="Horizontal"* и *"Vertical"*
13. Этот контейнер прижимает свое содержимое к определенной стороне внешнего контейнера - ... *DockPanel*

#### **40. Проверочная работа «Элементы управления»**

1. Свойство установки имени элемента – ... *Name*
2. Свойство, которое задает модификатор доступа к объекту – ... *FieldModifier*
3. Это свойство, которое устанавливает параметры видимости элемента – ... *Visibility*
4. Значения свойства видимости:
  - а) - элемент виден и участвует в компоновке - ... *Visible*
  - б) - элемент не виден и не участвует в компоновке - ... *Collapsed*
  - в) - элемент не виден, но при этом участвует в компоновке – ... *Hidden*
5. Свойства настройки шрифтов:
  - а) определяет семейство шрифта - ... *FontFamily*
  - б) определяет высоту шрифта – ... *FontSize*
  - в) определяет наклон шрифта - ... *FontStyle*
  - г) определяет толщину шрифта – ... *FontWeight*
  - д) определяет, как будет растягивать или сжимать текст - ... *FontStretch*
6. Свойство, которое позволяет установить курсор для элемента управления - ... *Cursor*
7. Свойство, которое задает направление текста – ... *FlowDirection*
8. Свойство, которое задает цвет фона элемента управления - ... *Background*
9. Свойство, которое задает цвет текста элемента управления - ... *Foreground*
10. Выравнивание по горизонтали содержимого внутри элемента задается свойством – ... *HorizontalContentAlignment*
11. Выравнивание по вертикали содержимого внутри элемента задается свойством - ... *VerticalContentAlignment*
12. Свойство для установки отступа содержимого элемента – ... *Padding*

### **ТЕМА 1.1.6. ПРОГРАММИРОВАНИЕ WPF-ПРИЛОЖЕНИЙ**

#### **49 - 50. Проверочная работа «Всплывающие подсказки», «Настройка свойств *ToolTipService* и *Popup*»**

1. Для вывода всплывающей подсказки на нужном элементе достаточно определить свойство - ... *ToolTip*
2. Свойство, которое определяет, имеет ли окно подсказки размытую темную тень - ... *HasDropShadow*
3. Свойство, которое определяет позицию подсказки с помощью одного из значений из перечисления - ... *Placement*
4. Свойство, с помощью которого можно задавать местоположение подсказки, указав абсолютные координаты на экране или координаты относительно некоторого элемента - ... *PlacementTarget*
5. Свойство, задает смещение окна подсказки - ... *PlacementRectangle*
6. Свойство, которое позволяет динамически позиционировать окно подсказки с помощью кода - ... *CustomPopupPlacementCallback*
7. Свойство, которое предназначено для создания всплывающего окна подсказки, которое остается открытым до тех пор, пока пользователь не щелкнет еще где-нибудь - ... *StaysOpen*

8. Свойства, которые позволяют управлять поведением подсказки с помощью кода - ... *IsEnabled, IsOpen*
9. Свойство, которое задает задержку (в миллисекундах) перед выводом подсказки после наведения указателя мыши на элемент - ... *InitialShowDelay*
10. Свойство, которое задает временной интервал (в миллисекундах), в течение которого пользователь может переходить от одной всплывающей подсказки к другой без задержки, определяемой предыдущим свойством - ... *BetweenShowDelay*
11. Свойство, которое задает время (в миллисекундах), в течение которого будет отображаться подсказка, если пользователь не сдвинет указатель мыши - ... *ShowDuration*

## **52. Проверочная работа «Текстовые элементы управления»**

1. Элемент, который предназначен для вывода текстовой информации, для создания простых надписей - ... *TextBlock*
2. Поле для ввода текстовой информации - ... *TextBox*
3. Элементы, с помощью которого можно задать отдельное форматирование для отдельных кусков текста - ... *Run*
4. Свойство, которое позволяет указывать высоту строк в текстовом элементе - ... *LineHeight*
5. Свойство, которое позволяет переносить текст - ... *TextWrapping="Wrap"*.
6. Свойство для выравнивания текста - ... *TextAlignment*
7. Как подчеркнуть текст внутри текстового элемента - ... *TextDecorations="Underline"*
8. Свойство, которое задает предельное количество вводимых символов - ... *MaxLength*
9. Для отображения полос прокрутки используются свойства - ... *VerticalScrollBarVisibility* и *HorizontalScrollBarVisibility*
10. Задействовать встроенную поддержку орфографии - ... *SpellCheck.IsEnabled="True"*
11. Элемент, который предназначен для ввода парольной информации - ... *PasswordBox*
12. Свойство, которое устанавливает символ маски, отображаемый при вводе пароля - ... *PasswordChar*
13. Свойство, которое устанавливает парольную строку, отображаемую по умолчанию при загрузке окна приложения - *Password*

## **56 - 57. Проверочная работа «Работа с датами» «Элементы, использующие концепцию диапазонов»**

1. Элемент в виде календаря - ... *Calendar*
2. Текстовое поле для ввода даты с выпадающим календарем после ввода - ... *DatePicker*
3. Свойство формата отображения дат - ... *DisplayMode*
4. Свойство способа выделения дат - ... *SelectionMode*
5. Значения свойства *DisplayMode*:
  - а) (по умолчанию) отображает все дни текущего месяца - ... *Month*
  - б) отображает все года текущего десятилетия - ... *Decade*
  - в) отображает все месяцы текущего года - ... *Year*
6. Значения свойства *SelectionMode*:
  - а) (по умолчанию) выделяет только одну дату – ... *SingleDate*
  - б) запрещает выделение - ... *None*
  - в) по нажатию на **Ctrl** выделяет несколько последовательно идущих дат - ... *SingleRange*
  - г) по нажатию на **Ctrl** выделяет несколько не последовательно идущих диапазонов дат - ... *MultipleRange*
7. Свойства класса *RangeBase*:



- а) текущее значение элемента управления - ... *Value*
- б) верхний и нижний предел - ... *Maximum, Minimum*
- в) величина, на которую уменьшается или увеличивается значение свойства *Value* при "малом изменении" - ... *SmallChange*
- г) величина, на которую уменьшается или увеличивается значение свойства *Value* при "большом изменении" - ... *LargeChange*

#### 8. Свойства класса *Slider*:

- а) устанавливает вертикальную или горизонтальную ориентацию элемента - ... *Orientation*
- б) управляют скоростью перемещения ползунка вдоль линейки, когда пользователь щелкает и удерживает нажатой клавишу мыши с любой стороны ползунка - ... *Delay, Interval*
- в) определяет размещение отметок возле линейки, которые помогают визуализировать шкалу - ... *TickPlacement*
- г) задает интервал между отметками, определяя их количество - ... *TickFrequency*
- д) используется для помещения отметок в нерегулярных позициях - ... *Ticks*

### 60 - 62. Проверочная работа «*DataGrid*»

1. Установить автоматическую генерацию столбцов - ... *AutoGenerateColumns="True"*
2. Кисть, которая используется для закрашивания фона позади каждой строки - ... *RowBackground*
3. Фон чередующихся строк - ... *AlternatingRowBackground*
4. Высота (в независимых от устройства единицах) строки, отображающей заголовки столбцов в верхней части *DataGrid* - ... *ColumnHeaderHeight*
5. Ширина (в независимых от устройства единицах) столбца, содержащего заголовки строк - ... *RowHeaderWidth*
6. Режим изменения размера, который применяется для установки ширины по умолчанию для каждого столбца в виде объекта *DataGridColumnLength* - ... *ColumnWidth*
7. Высота каждой строки - ... *RowHeight*
8. Значение из перечисления *DataGridGridlines*, которое указывает, какие линии должны отображаться в сетке *GridLines* - ... *Visibility*
9. Кисть, которая используется для закрашивания линий сетки, отображаемых между столбцами - ... *VerticalGridLinesBrush*
10. Кисть, которая используется для закрашивания линий сетки, отображаемых между строками - ... *HorizontalGridLinesBrush*
11. Значение из перечисления *DataGridHeaders*, которое указывает, какие заголовки должны отображаться - ... *HeadersVisibility*
12. Типы столбцов *DataGrid*:
  - а) этот столбец является стандартным выбором для большинства типов данных. Его значение преобразуется в текст - ... *DataGridTextColumn*
  - б) этот столбец отображает флажок и чаще всего автоматически используется для булевских (или булевских, допускающих null) значений - ... *DataGridCheckBoxColumn*
  - в) этот столбец отображает ссылку, на которой можно щелкать - ... *DataGridHyperlinkColumn*
  - г) этот столбец в режиме редактирования превращается в раскрывающийся список *ComboBox* - ... *DataGridTemplateColumn*

### ТЕМА 1.1.7. Использование систем контроля версий

## 74. Тест «Системы контроля версий»

1. Как в Git установить значение «Имя пользователя»?

- а) Все команды, кроме `git user.name +`
- б) `git config --global user.name`
- в) `git config user.name`
- г) `git user.name`

2. В каком редакторе можно работать с Git?

- а) Можно во всех редакторах +
- б) Только в редакторе Atom
- в) Можно во всех редакторах, что поддерживают подключение терминала

3. Что такое Git?

- а) Это облачное хранилище
- б) Это таск-менеджер
- в) Это распределённая система управления версиями +
- г) Это сервер для ваших проектов

4. Git и GitHub одно и то же самое?

- а) Оба продукта разработаны одной компанией, но они не одно и то же +
- б) Да. Git это сокращение от GitHub
- в) Нет, это продукты разных компаний

5. Как можно установить Git?

- а) Лишь через терминал (командную строку)
- б) Лишь через официальный сайт
- в) Можно через сайт или командную строку +

6. Какая команда в терминале служит для перехода к нужной папке?

- а) `go "папка"`
- б) `open "папка"`
- в) `cd "папка" +`
- г) `to "папка"`
- д) `mk "папка"`

7. Что такое GitHub?

- а) Графический интерфейс для работы с Git
- б) Веб-сервис для хостинга IT-проектов +
- в) Технология для контроля версий проектов

8. Как в Git установить глобально Email пользователя?

- а) `git config-global user.email`
- б) `git user.email`
- в) `git config --global user.email +`
- г) `git config user.email`
- д) `git --global user.email`

9. Сколько людей могут работать через Git над одним проектом?

- а) Неограниченное количество +
- б) Не более 30
- в) Не более 10

- г) Не более 50
- д) Не более 5

10. Можно ли отследить хронологию событий через Git?

- а) Да, но лишь даты кто и что добавил в проект
- б) Да, но лишь отслеживать добавленные файлы
- в) Нет, нельзя
- г) Да, можно отследить дату, автора и изменения что были внесены +

## **ТЕМА 1.1.8. ОСНОВЫ ADO.NET**

### **78 – 80. Тест «Работа с базами данных»**

**Вопрос 1.** Управляемый API-интерфейс WPF включает

- (1) +компонент PresentationFramework
- (2) +компонент PresentationCore
- (3) +компонент WindowsBase
- (4) компонент WindowsCodecs

**Вопрос 2.**

Укажите правильные утверждения в отношении свойств зависимостей WPF

- (1) +обеспечивают возможность работы с основными средствами WPF
- (2) +обладают дополнительным набором возможностей по сравнению с обычными свойствами
- (3) +формируют уведомления об изменениях значений свойств
- (4) нерационально потребляют память

**Вопрос 3.** Укажите правильные утверждения в отношении свойств зависимостей WPF

- (1) +обеспечивают возможность работы с основными средствами WPF
- (2) обладают дополнительным набором возможностей по сравнению с обычными свойствами
- (3) +формируют уведомления об изменениях значений свойств
- (4) нерационально потребляют память

**Вопрос 4.** Укажите правильные утверждения в отношении синтаксиса XAML

- (1) +элемент XAML-документа отображается на некоторый экземпляр класса .NET
- (2) +элементы XAML можно вкладывать друг в друга
- (3) +свойства класса определяются с помощью атрибутов
- (4) свойства класса определяют бизнес-логику

**Вопрос 5.** Укажите правильные утверждения

- (1) +большинство объектов в WPF являются производными от класса DispatcherObject
- (2) +класс Visual задает базовую функциональность рисования элементов WPF
- (3) +пользовательские элементы управления являются потомками класса UIElement
- (4) класс FrameworkElement является базовым для построения геометрических форм

**Вопрос 6.** Тип столбца DataGridViewTextColumn класса DataGridView используется для отображения в ячейках столбцов

- (1) +столбцов текстового содержимого

- (2) логических данных
- (3) данных, когда имеется набор элементов для выбора
- (4) элементов Uri

**Вопрос 7.** В состав служб WCF Data Services включен набор клиентских библиотек для

- (1) +типовых клиентских приложений .NET Framework
- (2) +приложений на основе Silverlight
- (3) сервера SQL Server

**Вопрос 8.** Укажите правильные утверждения

- (1) +визуальная структура интерфейсного элемента задается в его шаблоне ControlTemplate
- (2) +визуальное поведение интерфейсного элемента задается в его шаблоне ControlTemplate
- (3) визуальная структура интерфейсного элемента задается в его шаблоне DataTemplate
- (4) визуальное поведение интерфейсного элемента задается в его шаблоне DataTemplate

**Вопрос 9.** Текст XAML-документа должен содержать ...

- (1) +один корневой элемент
- (2) два корневых элемента
- (3) любое количество корневых элементов
- (4) только корневые элементы

**Вопрос 10.** Класс DataPager

- (1) +предоставляет пользовательский интерфейс для перелистывания страниц коллекции данных
- (2) предоставляет пользовательский интерфейс для отображения даты
- (3) +для правильной работы с моделью EDM данные должны быть отсортированы
- (4) предоставляет пользовательский интерфейс для отображения календаря

## 82. Тест «Способы создания команд»

**Вопрос 1.** Укажите правильные утверждения

- (1) +окно в WPF может содержать только один элемент - контейнер
- (2) окно в WPF может содержать несколько элементов управления
- (3) +страница в WPF может содержать только один элемент - контейнер
- (4) страница в WPF может содержать несколько элементов управления

**Вопрос 2.** Тип столбца DataGridHyperlinkColumn класса DataGrid используется для отображения в ячейках столбцов

- (1) столбцов текстового содержимого
- (2) логических данных
- (3) данных, когда имеется набор элементов для выбора
- (4) +элементов Uri

**Вопрос 3.** Контейнер WrapPanel

- (1) размещают элементы в строки и колонки в соответствии с невидимой таблицей
- (2) размещает элементы в горизонтальные и вертикальные стопки
- (3) +размещает элементы управления в доступном пространстве, по одной строке или колонке
- (4) размещает элементы управления относительно одного из своих внешних краев

**Вопрос 4.** Укажите правильные утверждения в отношении обобщенного класса DataServiceQuery<T>

- (1) представляет типизированный запрос по отношению к концептуальной модели в данном контексте объекта
- (2) +представляет одиночный запрос к службе данных
- (3) +для класса не определены публичные конструкторы
- (4) представляет Linq- запрос к данным

**Вопрос 5.** Пространство имен: <http://schemas.openxmlformats.org/markup-compatibility/2006>

- (1) базовое пространство имен WPF, которое охватывает все классы WPF
- (2) пространство имен XAML, которое включает различные свойства утилит XAML
- (3) +пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой
- (4) пространство имен XAML, поддерживаемое программами Expression Blend и Visual Studio

**Вопрос 6.** Свойство зависимости объявляется как экземпляр класса

- (1) RoutedEvent
- (2) Event
- (3) +DependencyProperty
- (4) Property

**Вопрос 7.** Свойство NavigationService класса Page

- (1) +возвращает ссылку на объект, которую можно использовать для отправки пользователя на другую страницу программным путем
- (2) определяет, должен ли объект страницы оставаться действующим после перехода пользователя на другую страницу
- (3) определяет, должны ли в обслуживающем данную страницу хосте отображаться навигационные элементы управления
- (4) принимает один элемент, который отображается на странице

**Вопрос 8.** В следующем XAML- описании <Button ... Grid.Row="0" ... /> фрагмент Grid.Row="0" определяет

- (1) простое свойство
- (2) сложное свойство
- (3) +присоединенное свойство
- (4) расширение разметки

**Вопрос 9.** Элемент управления ScrollBar

- (1) +предоставляет полосу прокрутки с перемещаемым элементом, позиция которого соответствует определенному значению
- (2) показывает ход выполнения длительной задачи
- (3) используется для задания числового значения путем перемещения бегунка на линейке прокрутки
- (4) является контейнером элементов управления

**Вопрос 10.** Установите соответствие между типом маршрутизируемого события и способом его обработки

- (1) прямые события
- (2) поднимающиеся события
- (3) туннельные события
- (а) возникают в одном элементе и не передаются в другой

- (б) перемещаются вверх по иерархии элементов дерева визуализации
  - (в) перемещаются вниз по иерархии
- 1-а, 2-б, 3-в

### 83. Тест «Entity Framework»

**Вопрос 1.** Какой метод будет автоматически сгенерирован при создании EDM-модели для формирования новой записи в таблице Employee базы данных

- (1) +CreateEmployee()
- (2) AddEmployee()
- (3) NewEmployee()
- (4) Employee()

**Вопрос 2.** Для ограничения доступа к заданной таблице только зарегистрированных пользователей, необходимо метод загрузки данных для этой таблицы класса DomainService отметить атрибутом

- (1) +RequiresAuthentication
- (2) RequiresRole
- (3) RequiresApplication

**Вопрос 3.** При привязке к объекту, не являющемуся элементом, свойство Source

- (1) +указывает на объект-источник, поставляющий данные
- (2) указывает на объект-источник, позволяющий базировать ссылку на текущем элементе
- (3) определяет необходимость присвоить значение объекта-источника для текущего или более высокого элемента в дереве визуальных элементов приложения
- (4) определяет необходимость присвоить значение объекта-источника для более высокого элемента в дереве визуальных элементов приложения

**Вопрос 4.** Укажите правильные утверждения

- (1) +стили – это коллекция значений свойств, которые могут быть применены к элементу
- (2) +стили в WPF могут устанавливать любое свойство зависимостей
- (3) +стили WPF поддерживают триггеры
- (4) стили не позволяют использовать шаблоны

**Вопрос 5.** Если свойство Binding.Mode, определяющее направление привязки, равно OneWay, то это определяет, что

- (1) +целевое свойство обновляется при изменении свойства-источника
- (2) и источник и целевое свойство обновляются при изменении какого-либо свойства
- (3) целевое свойство устанавливается изначально на основе значения свойства-источника
- (4) исходное свойство-источника обновляется при изменении целевого свойства, которое никогда не обновляется

**Вопрос 6.** Шаблон данных DataTemplate применяются для

- (1) задания представления и визуального поведения элементов управления
- (2) +извлечения данных из объекта
- (3) +отображения в элементах управления содержимым
- (4) организации бизнес-логики

**Вопрос 7.** Если свойство `Binding.UpdateSourceTrigger`, которое определяет режим обновления данных между источником и целевым свойством, принимает значение `Explicit`, то

- (1) обновление источника происходит немедленно после обновления целевого свойства
- (2) обновление источника происходит, когда целевой элемент теряет фокус
- (3) +обновление источника происходит только при вызове метода `BindingExpression.UpdateSource()`
- (4) обновления определяются метаданными целевого свойства

**Вопрос 8.** XAML-документ содержит ...

- (1) +разметку, описывающую внешний вид и поведение окна приложения
- (2) +разметку, описывающую внешний вид и поведение страницы приложения
- (3) логику приложения
- (4) бизнес-логику приложения

**Вопрос 9.** Уровень медиа-интеграции WPF включает

- (1) +компонент `milcore`
- (2) +компонент `WindowsCodecs`
- (3) компонент `WindowsBase`
- (4) компонент `PresentationCore`

**Вопрос 10.** При объявлении свойства зависимости должен использоваться модификатор доступа

- (1) +`public`
- (2) `private`
- (3) `protected`
- (4) `internal`

## 84 – 88. Тест «Работа с базой данных в приложении»

**Вопрос 1.** Укажите правильные утверждения в отношении гиперссылок в WPF

- (1) +гиперссылки не являются отдельными элементами
- (2) +гиперссылки являются внутрискриптовыми потоковыми элементами
- (3) +гиперссылки должны размещаться внутри другого поддерживающего их элемента
- (4) гиперссылки являются отдельными элементами XAML-описания документа

**Вопрос 2.** Модель EDM использует следующие понятия для описания структуры данных

- (1) +тип сущности
- (2) +тип ассоциации
- (3) +свойство
- (4) абстрактный тип

**Вопрос 3.** Укажите правильные утверждения в отношении служб WCF Data Services

- (1) +входят в состав библиотеки `.NET Framework`
- (2) +состоят из набора шаблонов и библиотек
- (3) позволяют создавать службы данных и обращаться к ним только в Интернете
- (4) +позволяют создавать службы данных и обращаться к ним в приложении

**Вопрос 4.** Укажите правильные утверждения в отношении класса `BitmapImage`

- (1) +класс можно использовать для ссылок на изображения в форматах JPEG и PNG
- (2) является абстрактным классом

- (3) +предоставляет тип источника практического объекта для свойства источника изображения
- (4) является базовым классом для создания источников изображений

**Вопрос 5.** Установите соответствие между методами класса ObjectContext и их назначением

- (1) Select
  - (2) Skip
  - (3) Top
  - (4) Where
  - (а) Ограничивает результаты запроса только теми свойствами, которые определены в заданной проекции
  - (б) Сортирует результаты запроса по заданным критериям и пропускает указанное число результатов
  - (в) Ограничивает результаты запроса указанным числом элементов
  - (г) Ограничивает запрос результатами, соответствующими определенным критериям фильтрации
- 1-а, 2-б, 3-в, 4-г

**Вопрос 6.** Укажите правильные утверждения

- (1) +пользовательские элементы управления являются потомками класса FrameworkElement
- (2) +класс Shape является базовым для построения таких геометрических форм
- (3) +класс Control определяет элементы управления, которые могут взаимодействовать с пользователем
- (4) класс FrameworkElement является базовым для всех контейнеров компоновки

**Вопрос 7.** При объявлении свойства зависимости должен использоваться модификатор

- (1) +public
- (2) +static
- (3) protected
- (4) +readonly

**Вопрос 8.** Тип столбца DataGridViewCheckBoxColumn класса DataGridView используется для отображения в ячейках столбцов

- (1) столбцов текстового содержимого
- (2) +логических данных
- (3) данных, когда имеется набор элементов для выбора
- (4) элементов Uri

**Вопрос 9.** Автоматически сгенерированный в проекте файл EDM-модели имеет расширение ...

- (1) +edmx
- (2) xaml
- (3) xlsx
- (4) xml

**Вопрос 10.** Автоматически сгенерированный в проекте файл Службы данных WCF имеет расширение ...

- (1) +svc
- (2) cs
- (3) sql



(4) xml

## 86 – 92. Тест «Работа с таблицами, документами, пользовательскими элементами»

**Вопрос 1.** В визуальную группу состояний `CommonStates` для элемента управления `ListBoxItem` входят состояния

- (1) `+Normal`
- (2) `+MouseOver`
- (3) `+Disabled`
- (4) `Focused`

**Вопрос 2.** Для группировки по определенному полю данных в `DataGrid`, полученных с помощью объекта `DomainDataSource`, необходимо в XAML-описание добавить свойство

- (1) `+GroupDescriptors`
- (2) `DomainContext`
- (3) `FilterDescriptor`
- (4) `riaControls`

**Вопрос 3.** Контейнер `Grid`

- (1) размещают элементы в строки и колонки в соответствии с невидимой таблицей
- (2) размещает элементы в горизонтальные и вертикальные стопки
- (3) размещает элементы управления в доступном пространстве, по одной строке или колонке
- (4) размещает элементы управления относительно одного из своих внешних краев

**Вопрос 4.** Классы, реализующие команды в WPF, должны поддерживать интерфейс

- (1) `ICommand`
- (2) `ICollection`
- (3) `IList`
- (4) `IComparable`

**Вопрос 5.** Если `List1` является коллекцией объектов, то строка кода `DataGrid1.ItemsSource = List1`

- (1) устанавливает источник данных для элемента управления `DataGrid`
- (2) формирует элемент управления `DataGrid`
- (3) сортирует элемент управления `DataGrid`
- (4) фильтрует элемент управления `DataGrid`

**Вопрос 6.** Пространство имен: `http://schemas.microsoft.com/winfx/2006/xaml/presentation`

- (1) базовое пространство имен WPF, которое охватывает все классы WPF
- (2) пространство имен XAML, которое включает различные свойства утилит XAML
- (3) пространство имен XAML, связанное с проблемой совместимости разметки с рабочей средой
- (4) пространство имен XAML, поддерживаемое программами `Expression Blend` и `Visual Studio`

**Вопрос 7.** Маршрутизируемое событие — это сообщение о событии, которое возникло ...

- (1) в одном элементе, а генерация о нем может осуществляться в другом элементе
- (2) в разных элементах, а генерация о нем может осуществляться в другом элементе
- (3) в одном элементе, а генерация о нем осуществляется в том же элементе

(4) в разных элементах, а генерация о нем может осуществляться в одном из этих элементов

**Вопрос 8.** Укажите правильные утверждения в отношении обобщенного класса `DataServiceCollection<T>`

- (1) +представляет коллекцию динамических сущностей, обеспечивающую выдачу уведомлений при добавлении в неё элементов
- (2) +службы данных WCF используют данный класс с целью поддержки привязки извлекаемых из базы данных для элементов управления Silverlight
- (3) +представляет коллекцию динамических сущностей, обеспечивающую выдачу уведомлений при удалении из неё элементов
- (4) обеспечивает сортировку элементов списка

**Вопрос 9.** Соответствие между свойствами класса `ContentControl` и их назначением

- (1) `Background`
  - (2) `BorderBrush`
  - (3) `BorderThickness`
  - (4) `Content`
  - (5) `DataContext`
  - (а) Возвращает или задает кисть, которая описывает фон элемента управления
  - (б) Возвращает или задает кисть, которая описывает фоновый цвет границы элемента управления
  - (в) Возвращает или задает толщину границы элемента управления
  - (г) Получает или задает содержимое объекта `ContentControl`
  - (д) Возвращает или задает контекст данных для элемента, участвующего в привязке данных
- 1-а, 2-б, 3-в, 4-г, 5-д

**Вопрос 10.** Для создания многострочного представления элемента управления `TextBox` необходимо использовать свойство

- (1) `TextWrapping`
- (2) `Text`
- (3) `TextAlignment`
- (4) `Visibility`

## ТЕМА 1.1.9. ОПТИМИЗАЦИЯ И РЕФАКТОРИНГ КОДА

### 106-109. Тест «Оптимизация и рефакторинг кода»

**Вопрос 1.** Верно ли утверждение, что между эффективностью и оптимизацией программы существует связь? - ....

**Вопрос 2.** Что такое оптимизация программ:

- а) улучшение работы существующей программы
- б) создание удобного интерфейса пользователя
- с) разработка модульной конструкции программы
- д) применение методов объектно-ориентированного программирования

**Вопрос 3.** Критерии оптимизации:

- а) время выполнения или размер требуемой памяти

- b) размер программы и ее эффективность
- c) независимость модулей
- d) качество программы, ее надежность

**Вопрос 4.** Возможна ли оптимизация программ без участия программиста? - ....

**Вопрос 5.** Оптимизация программы это:

- a) модификация
- b) отладка
- c) повышение сложности программы
- d) уменьшение сложности программы

**Вопрос 6.** Отладка программы – это

- a) работы, связанные с обслуживанием программы в процессе ее эксплуатации
- b) контроль логически сложных участков программы
- c) испытание, проверка правильности работы программы в целом либо ее составных частей
- d) процесс поиска и устранения ошибок в программе, производимый по результатам ее прогона на ПК
- e) определение функциональных возможностей системы

**Вопрос 7.** Какой этап выполняется раньше:

- a) отладка
- b) тестирование

**Вопрос 8.** Процедура поиска ошибки, когда известно, что она есть это:

- a) отладка
- b) тестирование
- c) компоновка
- d) транзакция
- e) трансляция

**Вопрос 9.** Верно ли утверждение, что между отладкой и тестированием существует различие? - ....

**Вопрос 10.** Инструментальные средства отладки (НЕ правильный ответ):

- a) компиляторы
- b) отладчики
- c) трассировка

**Вопрос 11.** Как называется процесс преобразования кода программы в машинный код?

- a) трансляция
- b) отладка
- c) тестирование
- d) компиляция
- e) компоновка

## ТЕМА 1.1.10. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА

### 114. Тест «Разработка пользовательского интерфейса»

**Вопрос 1.** В виде чего задаются команды через командный интерфейс:

- а) комбинации символов +
- б) движения пользователя
- в) голосовые сообщения

**Вопрос 2.** Как называется совокупность средств и правил взаимодействия человека и компьютера:

- а) приложение
- б) пользовательский интерфейс +
- в) объектный интерфейс

**Вопрос 3.** Какое слово пропущено в переводе на русский язык аббревиатуры SILK:

Речь, образ, язык, ...:

- а) визуализация
- б) наука
- в) знание +

**Вопрос 4.** Какой вид пользовательского интерфейса сейчас наиболее распространён:

- а) WIMP-интерфейс +
- б) командный
- в) биометрический

**Вопрос 5.** Какая разновидность пользовательского интерфейса появилась позже всех:

- а) графический
- б) Командный
- в) SILK +

**Вопрос 6.** Какой тип интерфейса позволяет проводить идентификацию пользователя по отпечаткам пальцев:

- а) интерфейс командной строки
- б) на основе биометрической технологии +
- в) речевой интерфейс

**Вопрос 7.** Интерфейс, в котором команды подаются голосом путем проговаривания специальных слов, называется:

- а) простым графическим интерфейсом
- б) интерфейсом командной строки
- в) речевым интерфейсом +

**Вопрос 8.** Элемент управления, который содержит перечень команд, раскрывается при нажатии раскрывающей кнопки, называется:

- а) выпадающий список +
- б) переключатель
- в) командная кнопка

**Вопрос 9.** Совокупность инструментов и приемов общения человека и ЭВМ:

- а) базовая система ввода-вывода
- б) периферийное устройство
- в) пользовательский интерфейс +

**Вопрос 10.** Вид интерфейса, в котором взаимодействие осуществляется через набор команд, вводимых с клавиатуры:

- а) мимический интерфейс
- б) командный интерфейс +
- в) графический интерфейс

**Вопрос 11.** Использование графических элементов управления характерно для:

- а) графического интерфейса +
- б) речевого интерфейса
- в) интерфейса командной строки

**Вопрос 12.** Элемент управления, предназначенный для выполнения какого-либо действия:

- а) выпадающий список
- б) командная кнопка +
- в) флажок

**Вопрос 13.** Элемент управления, предназначенный для выбора нескольких вариантов, включение которого помечается галочкой:

- а) флажок +
- б) переключатель
- в) выпадающий список

**Вопрос 14.** Какие из перечисленных ниже объектов являются графическим элементом управления диалогового окна:

- а) ярлыки
- б) указатели
- в) флажки +

**Вопрос 15.** Основными элементами графического интерфейса являются:

- а) окна +
- б) файлы
- в) команды

## ТЕМА 1.1.11. ДИАГРАММЫ КЛАССОВ В VISUAL STUDIO

### 117 – 119. Проверочная работа «Диаграммы классов»

**Вопрос 1.** Какая утилита системы Visual Studio позволяет внедрить моделирование в интегрированную среду разработки в виде действий, которые можно выполнить в любое время на всем протяжении разработки проекта - ...

*Class Designer*

**Вопрос 2.** Сопоставьте термины и определения

1. design surface
  2. Toolbox
  3. Add New Item
- а. область проектирования
  - б. инструментальная панель
  - в. Диалоговое окно
- 1 - а, 2 - б, 3 - в

**Вопрос 3.** Элементы на панели Toolbox можно разделить на - ...

*сущности (entities) и соединители (connectors)*

**Вопрос 4.** Какую команду нужно использовать для того, чтобы удалить сущности из диаграммы - ...

*Remove From Diagram*

**Вопрос 5.** Какую команду нужно использовать для экспорта диаграммы - ...

*Export Diagram as Image menu*

## **5.2. Критерии оценивания**

### **5.2.1. Критерии оценивания устного ответа**

При оценке устного ответа, обучающегося учитывается:

- 1) полнота и правильность ответа;
- 2) степень осознанности, понимания изученного;

**Отметка «5»:** ответ правильный, полный в соответствии с изученным материалом; материал изложен в определенной логической последовательности, литературным языком.

**Отметка «4»:** ответ правильный, полный в соответствии с изученным материалом; материал изложен в определенной логической последовательности; возможны отдельные затруднения в формулировке выводов.

**Отметка «3»:** ответ, в котором в основном правильно, но схематично или с отклонениями от последовательности изложения раскрыт материал или неполный, несвязный ответ, изложенный нелогично

**Отметка «2»:** при ответе обнаружено непонимание обучающимся основного содержания учебного материала, неумение его анализировать допущены существенные ошибки, которые обучающийся не смог исправить при наводящих вопросах преподавателя, отсутствует логика в изложении материала, нет необходимых обобщений и самостоятельной оценки фактов; недостаточно сформированы навыки устной речи.

### **5.2.2. Критерии оценивания выполнения заданий на лабораторных и практических занятиях**

- **Отметка «5»:** работа выполнена полностью и правильно; сделаны правильные выводы.
- **Отметка «4»:** работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.
- **Отметка «3»:** работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки.
- **Отметка «2»:** допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя.

Задания к лабораторным занятиям представлены в методических указаниях к лабораторным занятиям по МДК.02.01. Технология разработки программного обеспечения.

Задания к практическим занятиям представлены в методических указаниях к практическим занятиям по МДК.02.01. Технология разработки программного обеспечения.

Методы оценки:

- Защита отчетов по выполненному заданию на практических занятиях.
- Интерпретация результатов наблюдений за деятельностью обучающегося на практических занятиях

### 5.2.3. Критерии оценивания тестовых заданий

Оценка в баллах	Степень выполнения задания
Неуд.	Выполнено от 0 до 49,9 % предложенных заданий
Удов.	Выполнено от 50 до 69,9% предложенных заданий
Хор.	Выполнено от 70 до 89,9% предложенных заданий
Отл.	Выполнено от 90 до 100% предложенных заданий

### 5.2.4. Общая классификация ошибок

При оценке знаний и умений учитываются ошибки и недочёты в работе.

*Грубыми считаются ошибки:*

- незнание определения основных понятий, законов, общепринятых символов обозначений величин;
- неумение выделить в ответе главное; обобщить результаты изучения;
- неумение применить знания для решения задач;
- неумение использовать полученные данные для выводов;
- неумение пользоваться первоисточниками, учебником, справочником;
- нарушение техники безопасности, небрежное отношение к оборудованию

*Негрубыми считаются ошибки:*

- неточность формулировок, определений, понятий, законов, вызванная неполнотой охвата основных признаков определяемого понятия или заменой 1-3 из этих признаков второстепенными;
- ошибки, вызванные несоблюдением условий проведения лабораторных занятий;
- недостаточно продуманный план устного ответа (нарушение логики изложения, подмена отдельных основных вопросов второстепенными);
- нерациональные методы работы со справочной литературой;
- неумение выполнять лабораторные задания в общем виде.

## 6. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО МЕЖДИСЦИПЛИНАРНОМУ КУРСУ МДК 01.01. Разработка программных модулей

### 6.1. Перечень заданий для подготовки к экзамену в V семестре

**Теоретические вопросы:**

1. Понятие ЖЦ ПО. Этапы ЖЦ ПО
2. Технология структурного программирования. Инструментальные средства оформления и документирования алгоритмов программ
3. Оценка сложности алгоритма: классификация. Оценка сложности алгоритма: классы алгоритмов, неразрешимые задачи
4. Основные принципы объектно-ориентированного программирования.
5. Классы: основные понятия
6. Перегрузка методов
7. Операции класса. Иерархия классов
8. Синтаксис интерфейсов. Интерфейсы и наследование
9. Структуры
10. Делегаты

11. Регулярные выражения
12. Коллекции. Параметризованные классы
13. Указатели. Операции со списками
14. Назначение и виды паттернов. Основные шаблоны
15. Порождающие шаблоны
16. Структурные шаблоны
17. Поведенческие шаблоны
18. Сферы применения событийно-управляемого программирования
19. Инструменты и библиотеки событийно-управляемого программирования
20. Введение в графику: Программирование с использованием WPF
21. Введение в графику: Разметка и код программной части
22. Обработчики событий. Маршрутизация событий
23. Поднимающиеся события
24. Фреймы
25. Диалоговые окна

### Практические задания по классам, структурам и регулярным выражениям:

#### Вариант 1

Создать класс Rectangle, разработав следующие элементы класса:

Поля: int a, b;

Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.

Методы, позволяющие:

- вывести длины сторон прямоугольника на экран;
- рассчитать периметр прямоугольника;
- рассчитать площадь прямоугольника.

Свойства:

- получить-установить длины сторон прямоугольника
- позволяющее установить, является ли данный прямоугольник квадратом

#### Вариант 2

Создать класс «Студент», который определяется полями

- ФИО;
- номер группы;
- название факультета;
- название специальности;
- средний балл успеваемости.

Пусть имеется массив объектов этого класса. Разработать метод выбора студентов из массива по условию (учится на конкретном факультете, имеет средний балл более заданного уровня и пр.).

Для определения, удовлетворяет ли объект условию, передать в метод параметр-делегат.

#### Вариант 3

Создайте структуру с именем train, содержащую поля:

- название пункта назначения,
- номер поезда,
- время отправления.

Ввести данные в массив из пяти элементов типа train. Добавить возможность вывода информации о поезде, номер которого введен пользователем.

#### Вариант 4

Создать класс с двумя переменными.

- добавить функцию вывода на экран и функцию изменения этих переменных;
- добавить функцию, которая находит сумму значений этих переменных;
- добавить функцию, которая находит наибольшее значение из этих двух переменных.

#### Вариант 5

Описать класс «домашняя библиотека».

Предусмотреть возможность:



- работы с произвольным числом книг;
- поиска книги по какому-либо признаку (например, по автору или по году издания); добавления книг в библиотеку;
- удаления книг из библиотеки;
- сортировки книг по разным полям.

#### Вариант 6

Создать класс Класс Покупатель со следующими полями:

- ФИО;
- Адрес;
- Номер кредитной карточки;
- Номер банковского счета.

Методы:

- установка значений атрибутов;
- получение значений атрибутов;
- вывод информации.

Создать массив объектов данного класса. Вывести список покупателей, у которых номер кредитной карточки находится в заданном диапазоне.

#### Вариант 7

Создайте структуру с именем train, содержащую поля:

- название пункта назначения,
- номер поезда,
- время отправления.

Ввести данные в массив из пяти элементов типа train. Добавить возможность вывода информации о поезде, номер которого введен пользователем.

#### Вариант 8

- Создать класс Абонент:
- Идентификационный номер;
- ФИО;
- Адрес;
- Время междугородных и городских переговоров.

Методы:

- установка значений атрибутов,
- получение значений атрибутов,
- вывод информации.

Создать массив объектов данного класса. Вывести сведения относительно абонентов, у которых время городских переговоров превышает заданное.

#### Вариант 9

Требуется разработать регулярное выражение, которым возможно проверить, может ли являться входная строка (целиком) корректным паролем.

Пароль должен содержать только латинские символы, цифры и специальные символы ^\$% @#&\*!?

- пароль должен состоять из не менее чем восьми символов
- пароль должен содержать по крайней мере один латинский символ в верхнем регистре
- пароль должен содержать по крайней мере один латинский символ в нижнем регистре
- пароль должен содержать по крайней мере одну цифру
- пароль должен содержать по крайней мере два различных специальных символа

**Примеры корректных паролей:**

- rtG3FG!Tr^e
- aA1\*!1Aa
- oF^a1D@y5eб
- enroi#&RkdeR#\$092uwedchf34tguv394h

**Примеры некорректных паролей**

- пароль
- password
- qwerty
- lOngPa\$\$W0Rd

#### Вариант 10

Требуется разработать регулярное выражение, способное определить, является ли входная строка

(целиком) датой в одном из нескольких форматов. Допускаются следующие форматы даты:

- день.месяц.год (14.09.2022, 5.02.1995, 01.4.2012)
- день/месяц/год (14/09/2022, 5/02/1995, 01/4/2012)
- день-месяц-год (14-09-2022, 5-02-1995, 01-4-2012)
- год.месяц.день (2022.09.14, 1995.02.5, 2012.4.01)
- год/месяц/день (2022/09/14, 1995/02/5, 2012/4/01)
- год-месяц-день (2022-09-14, 1995-02-5, 2012-4-01)
- день\_месяц\_год (14 сентября 2022, 5 февраля 1995, 01 апреля 2012)
- Месяц\_eng день, год (September 14, 2022, February 5, 1995, April 01, 2012)
- Мес\_eng день, год (Sep 14, 2022, Feb 5, 1995, Apr 01, 2012)

**Примеры корректных дат:**

- 20 января 1806
- 1924, July 25
- 26/09/1635
- 3.1.1506

**Примеры некорректных дат:**

- 25.08-1002
- декабря 19, 1838
- 8.20.1973
- Jun 7, -1563

**Вариант 11**

Требуется разработать регулярное выражение, которым возможно проверить, может ли являться входная строка (целиком) корректной записью цвета в одном из трёх меѢ форматов:

- **rgb:** rgb(r, g, b), где «r, g, b» - это комбинация из трёх целых чисел (от 0 до 255) или трёх процентных значений (от 0% до 100%), перечисленных через запятую.
- **hex** (шестнадцатеричный код цвета, #rrggbb) – это шестизначное представление цвета в RGB пространстве. Первые две цифры (rr) – представляют собой красное значение, следующие две – зелёное значение (gg), а последние – синее значение (bb). Перед значениями каналов предшествует символ #. Также допускается сокращённый вид записи – по одной цифре – #rgb
- **hsl** (тон, насыщенность и светлота, hsl(h, s, l)) - записывается похожим на rgb формат образом. Тон – целое число в диапазоне от 0 до 360, насыщенность и светлота - целочисленные процентные значения.

**Примеры корректных цветов:**

- #21f48D
- #888 • rgb(255, 255,255)
- rgb(10%, 20%, 0%)
- hsl(200,100%,50%)
- hsl(0, 0%, 0%)

**Примеры некорректных цветов:**

- #2345
- ffffff
- rgb(257, 50, 10)
- hsl(20, 10, 0.5)
- hsl(34%, 20%, 50%)

**Практические задания по строкам и символам:**

**Вариант 1**

Дано слово из 12 букв. Поменять местами его трети следующим образом:

- а) первую треть слова разместить на месте третьей, вторую треть — на месте первой, третью треть — на месте второй;
- б) первую треть слова разместить на месте второй, вторую треть — на месте третьей, третью треть — на месте первой.

**Вариант 2**

Дано слово. Переставить первые три и последние три буквы, сохранив порядок их следования. Задачу решить двумя способами:

- 1) без использования оператора цикла;
- 2) с использованием оператора цикла.

**Вариант 3**

Дано слово. Перенести первые k его букв в конец. Задачу решить двумя способами:

- 1) без использования оператора цикла;
- 2) с использованием оператора цикла.

**Вариант 4**

1. Дан символ. Выяснить, является ли он цифрой.
2. Дан текст. Напечатать все имеющиеся в нем цифры.
3. Дан текст. Определить количество цифр в нем.

<p>4. Дан текст, в котором имеются цифры:</p> <ul style="list-style-type: none"> <li>• найти их сумму;</li> <li>• найти максимальную цифру.</li> </ul>
<p><b>Вариант 5</b></p> <ol style="list-style-type: none"> <li>1. Дано слово. Вывести на экран его третий символ.</li> <li>2. Дано слово. Вывести на экран его последний символ.</li> <li>3. Дано слово. Вывести на экран его k-й символ.</li> <li>4. Дано слово. Определить, одинаковы ли второй и четвертый символы в нем.</li> <li>5. Дано слово. Верно ли, что оно начинается и оканчивается на одну и ту же букву?</li> </ol>
<p><b>Вариант 6</b></p> <p>Ввести свои фамилию, имя и отчество как одно данное строкового типа. Определить длину строки и количество букв «а» в них. Выполнить дополнительно задание:</p> <ul style="list-style-type: none"> <li>• вывести имя и количество букв в третьем слове;</li> <li>• определить, сколько букв «а» есть в фамилии;</li> <li>• вывести три буквы (свои инициалы) с точками;</li> <li>• вывести длины фамилии и имени;</li> <li>• изъять все буквы «а» и «о» из фамилии.</li> <li>• Вывести имя в столбик</li> </ul>
<p><b>Вариант 7</b></p> <p>Ввести свои фамилию, имя и отчество как одно данное строкового типа. Определить длину строки и количество букв «а» в них. Выполнить дополнительно задание:</p> <ul style="list-style-type: none"> <li>• вывести имя в обратном порядке;</li> <li>• вывести фамилию в столбик;</li> <li>• вывести имя, отчество и количество букв в фамилии;</li> <li>• вывести самое короткое слово;</li> <li>• вывести строку без пропусков;</li> <li>• вывести длины трех слов.</li> </ul>
<p><b>Вариант 8</b></p> <p>Ввести свои фамилию, имя и отчество как одно данное строкового типа. Определить длину строки и количество букв «о» в них. Выполнить дополнительно задание:</p> <ul style="list-style-type: none"> <li>• вывести фамилию и инициалы;</li> <li>• вывести имя и количество букв в фамилии;</li> <li>• определить сколько букв «о» есть в имени;</li> <li>• вывести самое длинное слово;</li> <li>• Все буквы «и» в имени продублировать;</li> <li>• Вывести фамилию и количество букв в имени.</li> </ul>
<p><b>Вариант 9</b></p> <p>Ввести свои фамилию, имя и отчество как одно данное строкового типа. Определить длину строки и количество букв «о» в них. Выполнить дополнительно задание:</p> <ul style="list-style-type: none"> <li>• вывести имя и количество букв в фамилии;</li> <li>• вывести имя и фамилию;</li> <li>• каждую букву имени продублировать;</li> <li>• вывести фамилию в обратном порядке;</li> <li>• определить, сколько букв «а» и «б» в фамилии;</li> <li>• вывести третье слово и количество букв в фамилии;</li> </ul>

## 6.2. Перечень заданий для подготовки к экзамену в VI семестре

### Теоретические вопросы:

1. Общие сведения о системах контроля версий. Распределенные системы управления версиями
2. Всплывающие подсказки
3. Свойства ToolTipService и Popup



**Вариант 4**

Создать WPF-приложение для вычисления площади прямоугольника по следующему макету:

Длина                      TextBox

Ширина                    TextBox

Рассчитать площадь

TextBox

**Вариант 5**

Создать WPF-приложение для вычисления следующему макету:

Первое число              TextBox

Второе число              TextBox

Вычислить сумму         TextBox

Вычислить разность     TextBox

**Вариант 6**

Создать WPF-приложение для вычисления следующему макету:

Первое число              TextBox

Второе число              TextBox

Вычислить произведение    TextBox

Вычислить частное         TextBox

**Вариант 7**

Создать WPF-приложение для вычисления следующему макету:

Первое число              TextBox

Второе число              TextBox

Сравнить

Результат вывести с помощью `MessageBox.Show(" ");`

**Вариант 8**

Создать WPF-приложение для вычисления следующему макету:

Первое число              TextBox

Второе число              TextBox

Вычислить среднее арифметическое

Результат вывести с помощью `MessageBox.Show(" ");`

**Вариант 9**

Создать WPF-приложение, которое реализует решение следующей задачи:

<p>Запросить стороны треугольника и определить, является ли он разносторонним, равносторонним или равнобедренным.  <b>!!! Учесть:</b> третья сторона не должна превышать сумму двух других, иначе треугольник не существует.</p>
<p><b>Вариант 10</b>  <b>Создать WPF-приложение, которое реализует решение следующей задачи:</b>          Разработать программу для реализации простых арифметических действий (сложение, вычитание, умножение, деление) над целыми числами.  <b>!!! Предусмотреть</b> возможную ошибку пользователя при вводе исходных данных, например - деление на 0.</p>
<p><b>Вариант 11</b>  <b>Создать WPF-приложение, которое реализует решение следующей задачи:</b>          Разработать программу для вычисления значений корней квадратного уравнения: <math>ax^2+bx+c=0</math>, где a, b, c - вводятся пользователем.  <b>Результат</b> выводится как значение x1 и x2, или значение x1, или сообщение «Решений не существует».</p>
<p><b>Вариант 12</b>  <b>Создать WPF-приложение, которое реализует решение следующей задачи:</b>          Разработать калькулятор для пересчета заданной суммы рублей в доллары (курс доллара к рублю задается вручную).  <b>!!! Предусмотреть</b> возможную ошибку пользователя при вводе исходных данных, например, отрицательные значения.</p>
<p><b>Вариант 13</b>  <b>Создать WPF-приложение, которое реализует решение следующей задачи:</b>          Разработать программу для магазина, которая предусматривает ввод стоимость покупки и суммы денег, внесенную покупателем. Результат работы программы - это сообщение «Доплатите ...», «Возьмите сдачу ...» или «Спасибо за покупку».  <b>!!! Предусмотреть</b> возможные ошибки ввода, например, отрицательные значения.</p>
<p><b>Вариант 14</b>  <b>Создать WPF-приложение, которое реализует решение следующей задачи:</b>          Разработать программу-конвертер для перевода температуры воздуха в градусах по Цельсию в Кельвины и Кельвинов в Цельсии.  <b>Формула</b> для перевода: <math>T = t + T_0</math>, где T - температура в кельвинах, t - температура в Цельсиях, <math>T_0 = 273,15</math> кельвина</p>
<p><b>Вариант 15</b>  <b>Создать WPF-приложение, которое реализует решение следующей задачи:</b>          Запросить стороны треугольника и определить, является ли он разносторонним, равносторонним или равнобедренным.  <b>!!! Учесть:</b> третья сторона не должна превышать сумму двух других, иначе треугольник не существует.</p>

### 6.3. Перечень заданий для подготовки к экзамену в VII семестре

#### Теоретические вопросы:

1. Правила разработки интерфейсов пользователя
2. Этапы проектирования пользовательского интерфейса
3. Структура диалога в пользовательском интерфейсе
4. Выбор цветов и шрифта при разработке пользовательского интерфейса
5. Разработка пользовательского интерфейса
6. Компоновка, компиляция и построение проекта
7. Отладка проекта
8. Расширение кода проекта
9. Методы оптимизации программного кода
10. Цели и методы рефакторинга

11. Конструктор классов
12. Xml-документация
13. Ошибки и исключения
14. Роль обработки исключений в Net
15. Последствия не перехвата событий
16. Пространства имен
17. Метод Main()
18. Строки и класс System.String
19. Форматирование и интерполяция строк
20. Класс StringBuilder
21. Работа с основными компонентами
22. Элемент ListBox
23. Элемент DataGridView
24. Дополнительные классы и структуры .NET
25. Создание потоков. Делегат ThreadStart

### Практические задания по Entity Framework

<p><b>Вариант 1</b> Создайте БД Студенты (паспортные данные, адрес, группа). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 2</b> Создайте БД Книги (авторы, название, год издания). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 3</b> Создайте БД Сотрудники (ФИО, паспортные данные, адрес). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 4</b> Создайте БД Поставщики (название компании, ФИО контактного лица, расчётный счёт в банке). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 5</b> Создайте БД Производственные процессы (продукты, объёмы их производства, необходимые материалы). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 6</b> Создайте БД Заказы (тип заказа (покупка, гарантийный ремонт, негарантийный ремонт), общая стоимость, скидка). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 7</b> Создайте БД Продажи (наименования, модели (марки) и серийные номера товаров). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 8</b> Создайте БД Товары (категория, модель, производитель). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 9</b></p>

<p>Создайте БД Клиенты (ФИО, заказ, общая стоимость заказа). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 10</b> Создайте БД Блюда (цена, название, вид кухни). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 11</b> Создайте БД Публикации (название, тип (газетная, книжная, web и т.п.), источник (газета, книга, сайт и т.п.)). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 12</b> Создайте БД Организация (название, тип (промышленная, финансовая, торговая, исследовательская и т.п.), контактная информация). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 13</b> Создайте БД Ошибка (bug) (краткое описание, срок поступления информации об ошибке, её источник (пользователь, тестировщик)). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 14</b> Создайте БД Учебные курсы (название, форма отчётности, количество часов). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 15</b> Создайте БД Читатели библиотеки (номер читательского билета, ФИО, названия взятых книг). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 16</b> Создайте БД Учебные заведения (Название, ФИО руководителя, юридический адрес). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 17</b> Создайте БД Домашние животные (вид, порода, возраст, кличка). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 18</b> Создайте БД Компьютеры (Производитель, Процессор, ОЗУ, HDD). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 19</b> Создайте БД Авторы (ФИО, направление, название публикации, год публикации). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.</p>
<p><b>Вариант 20</b> Создайте БД Канцелярские товары (название, производитель, цена). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия</p>



сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.

**Вариант 21**

Создайте БД Канцелярские товары (название, производитель, цена). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.

**Вариант 22**

Создайте БД Планеты Солнечной системы (название, диаметр, номер по удаленности от Солнца). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.

**Вариант 23**

Создайте БД Автомобили (марка, модель, год выпуска, цена). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.

**Вариант 24**

Создайте БД Города (название, регион, численность населения). Свяжите созданную БД с приложением WPF. В приложении сформируйте запросы на поиск и фильтрацию (условия сформулируйте самостоятельно), удаление, добавление и редактирование информации в таблице.

### 6.3. Критерии оценивания ответов на экзамене

- оценка «отлично», если студент обладает глубокими и прочными знаниями программного материала; при ответе на вопросы продемонстрировал исчерпывающее, последовательное и логически стройное изложение; правильно сформулировал понятия и закономерности по вопросам; сделал вывод по излагаемому материалу;
- оценка «хорошо», если студент обладает достаточно полным знанием программного материала; его ответ представляет грамотное изложение учебного материала; но имеются существенные неточности в формулировании понятий и закономерностей по вопросам; не полностью сделаны выводы по излагаемому материалу;
- оценка «удовлетворительно», если студент имеет общие знания основного материала без усвоения некоторых существенных положений; формулирует основные понятия с некоторой неточностью; затрудняется в приведении примеров, подтверждающих теоретические положения;
- оценка «неудовлетворительно», если студент не знает значительную часть программного материала; допустил существенные ошибки в процессе изложения; не умеет выделить главное и сделать вывод; приводит ошибочные определения; ни один вопрос не рассмотрен до конца, наводящие вопросы не помогают.

## 7. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

### 7.1. Основные печатные издания

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: учебник / Г.Н. Федорова. – М.: Академия, 2021. – 384 с

### 7.2. Основные электронные издания

1. Федорова Г.Н. Разработка модулей программного обеспечения для компьютерных систем: электронный учебно-методический комплекс / Г.Н. Федорова. – М.: Академия, 2021. – URL: <https://www.academia-moscow.ru/catalogue/5411/478674/>

### 7.3. Дополнительные источники

1. Гниденко, И. Г. Технология разработки программного обеспечения: учебное пособие для среднего профессионального образования / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2021. — 235 с. — (Профессиональное образование). — ISBN 978-5-534-05047-9. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/472502> (дата обращения: 13.12.2021).
2. Белугина С.В. Разработка программных модулей программного обеспечения для компьютерных систем, Прикладное программирование. – Санкт-Петербург: Лань, 2021 – 312 с.
3. Подбельский В. Язык C#. Базовый курс. Издание второе, переработанное и дополненное. Издательство: Финансы и статистика, 2016. – 408 с. - ISBN: 9785279035342
4. Эндрю Троелсен, Филипп Джекпикс Язык программирования C# 6.0 и платформа .NET 4.6 7-е издание. Издательская группа "Диалектика-Вильямс", 2016

### 7.4. Интернет ресурсы:

- <http://www.ict.edu.ru> - Федеральный образовательный портал
- <http://www.edu-it.ru> - ИТ-образование в России: сайт открытого е-консорциума
- <http://www.intuit.ru> - Интернет-университет информационных технологий (ИНТУИТ.ру)
- <http://www.metod-kopilka.ru> – Методическая копилка учителя информатики
- [http://real.tepkom.ru/Real\\_OM-СМ\\_A.asp](http://real.tepkom.ru/Real_OM-СМ_A.asp) От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам.
- <http://programm.ws/index.php> - Учебники по программированию
- <https://metanit.com/> Сайт по программированию
- <https://code-live.ru/> - портал о программировании
- <http://www.programbeginner.ru/> - сайт о программировании