

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Силин Яков Петрович
Должность: Ректор
Дата подписания: 04.06.2026 13:16:22
Уникальный программный ключ:
24f866be2aca16484036a8cbb3c509a9531e605f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Уральский государственный экономический университет»

Утверждена

Советом по учебно-методическим
вопросам и качеству образования

16 декабря 2025 г.

протокол № 4

Председатель Карх Д.А.

(подпись)



09.12.2025 г.

протокол № 12

И.о. зав. кафедрой Кольева Н.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

| | |
|-------------------------|--|
| Наименование дисциплины | Современные технологии разработки программного обеспечения |
| Направление подготовки | 09.04.03 Прикладная информатика |
| Профиль | Корпоративные информационные системы |
| Форма обучения | очно-заочная |
| Год набора | 2026 |

Разработана:
Профессор, д.п.н.
Плещев В.В.

Екатеринбург
2025 г.

СОДЕРЖАНИЕ

| | |
|--|-----------|
| ВВЕДЕНИЕ | 3 |
| 1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 3 |
| 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП | 3 |
| 3. ОБЪЕМ ДИСЦИПЛИНЫ | 3 |
| 4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП | 3 |
| 5. ТЕМАТИЧЕСКИЙ ПЛАН | 5 |
| 6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ | 5 |
| 7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ | 7 |
| 8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ | 9 |
| 9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ | 9 |
| 10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 10 |
| 11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ | 11 |

ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы магистратуры, разработанной в соответствии с ФГОС ВО

| | |
|---------|--|
| ФГОС ВО | Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 916) |
|---------|--|

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

исследование и разработка эффективных методов реализации информационных процессов и построения ИС в прикладных областях на основе использования современных ИКТ, моделирование прикладных и информационных процессов, разработка требований к созданию и развитию ИС и ее компонентов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части учебного плана.

3. ОБЪЕМ ДИСЦИПЛИНЫ

| Промежуточная аттестация | Часов | | | | | 3.е. |
|--------------------------|------------------|---------------------------------|--------|--------------|--|------|
| | Всего за семестр | Контактная работа .(по уч.зан.) | | | Самостоятельная работа в том числе подготовка контрольных и курсовых | |
| | | Всего | Лекции | Лабораторные | | |
| Семестр 2 | | | | | | |
| Зачет | 180 | 24 | 8 | 16 | 156 | 5 |

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

| Шифр и наименование компетенции | Индикаторы достижения компетенций |
|--|--|
| УК-2 Способен управлять проектом на всех этапах его жизненного цикла | ИД-1.УК-2 Знать: принципы формирования проектной задачи в рамках обозначенной проблемы; основные требования, предъявляемые к проектной работе и критерии оценки результатов проектной деятельности |
| | ИД-2.УК-2 Уметь: разрабатывать план реализации проекта с учетом возможных рисков реализации и возможностей их устранения; планировать необходимые ресурсы |
| | ИД-3.УК-2 Иметь практический опыт осуществления мониторинга хода реализации проекта; корректировки отклонений; внесения изменений в план реализации проекта |

Общепрофессиональные компетенции (ОПК)

| Шифр и наименование компетенции | Индикаторы достижения компетенций |
|--|--|
| <p>ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;</p> | <p>ИД-1.ОПК-2 Знать: современные интеллектуальные технологии для решения профессиональных задач</p> |
| | <p>ИД-2.ОПК-2 Уметь: обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач</p> |
| | <p>ИД-3.ОПК-2 Иметь практический опыт: разработки программного обеспечения, интеллектуальных информационных систем, алгоритмов машинного обучения</p> |
| <p>ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;</p> | <p>ИД-1.ОПК-5 Знать: знать современное программное и аппаратное обеспечение информационных и автоматизированных систем</p> |
| | <p>ИД-2.ОПК-5 Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач</p> |

| | |
|---|---|
| ОПК-5 Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем; | ИД-3.ОПК-5 Иметь практический опыт: проектирования, разработки и модернизации корпоративных информационных систем |
|---|---|

5. ТЕМАТИЧЕСКИЙ ПЛАН

| Тема | Часов | | | | | | |
|-----------|--|-------------|--------------------------------|--------------|----------------------|----------------|---------------------------------|
| | Наименование темы | Всего часов | Контактная работа (по уч.зан.) | | | Самост. работа | Контроль самостоятельной работы |
| | | | Лекции | Лабораторные | Практические занятия | | |
| Семестр 2 | | 180 | | | | | |
| Тема 1. | Организация процесса разработки программного обеспечения. (УК-2, ОПК-2, ОПК-5) | 46 | 2 | 4 | | 40 | |
| Тема 2. | Объектно-ориентированное проектирование и реализация программных систем.(УК-2, ОПК-2, ОПК-5) | 42 | 2 | 4 | | 36 | |
| Тема 3. | Разработка интернет-приложений.(УК-2, ОПК-2, ОПК-5) | 46 | 2 | 4 | | 40 | |
| Тема 4. | Организация процесса тестирования и оценки качества программных систем. (УК-2, ОПК-2, ОПК-5) | 46 | 2 | 4 | | 40 | |

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

| Раздел/Тема | Вид оценочного средства | Описание оценочного средства | Критерии оценивания |
|---|------------------------------------|--|---------------------|
| Текущий контроль (Приложение 4) | | | |
| Тема 1 | Тест (приложение 4) | Тест состоит из 25-ти вопросов | 10 баллов |
| Тема 2 | Практическая работа (приложение 4) | Практическая работа состоит из 4 заданий на программирование | 10 баллов |
| Темы 3-4 | Практическая работа (приложение 4) | Практическая работа состоит из задания на программирование | 10 баллов |
| Промежуточная аттестация (Приложение 5) | | | |
| 2 семестр (За) | Билеты к зачету (Приложение 5) | Билет состоит из 1 теоретического и 1 практического задания. | 100 |

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показатель оценки освоения ОПОП формируется на основе объединения текущего контроля и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущий контроль. Используется 100-балльная система оценивания. Оценка работы студента в течение семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущего контроля, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончании дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончании формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

| Показатель оценки | По 5-балльной системе | Характеристика показателя |
|-------------------|-----------------------|--|
| 100% - 85% | отлично | обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне |
| 84% - 70% | хорошо | обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответа и т.д.) |
| 69% - 50% | удовлетворительно | обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно. |
| 49 % и менее | неудовлетворительно | обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для решения профессиональных задач |
| 100% - 50% | зачтено | характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно» |
| 49 % и менее | не зачтено | характеристика показателя соответствует «неудовлетворительно» |

7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7.1. Содержание лекций

Тема 1. Организация процесса разработки программного обеспечения. (УК-2, ОПК-2, ОПК-5)
Процесс разработки заказного ПО. Процесс разработки инвестиционного ПО. Процесс разработки встроенного ПО. Процесс разработки игр. Инструментарий разработчика. Этапы разработки

Тема 2. Объектно-ориентированное проектирование и реализация программных систем.(УК-2, ОПК-2, ОПК-5)

Основы унифицированного языка визуального моделирования (UML). Инструменты UML для создания статических и динамических объектно-ориентированных программных средств.

Статические модели объектно-ориентированных программных систем. Динамические модели объектно - ориентированных программных систем. Унифицированный процесс разработки объектно-ориентированных программных средств. Эволюционно-инкрементная организация жизненного цикла разработки. Управление риском. Разработка в стиле экстремального программирования

Тема 3. Разработка интернет-приложений.(УК-2, ОПК-2, ОПК-5)

Основные веб-технологии и фреймворки. Корпоративный портал. CRM. ERP. Системы электронной коммерции. Преимущества web-приложений

Тема 4. Организация процесса тестирования и оценки качества программных систем.(УК-2, ОПК-2, ОПК-5)

Основные понятия. Тестирование надежности программного обеспечения. Тестирование программного обеспечения. Виды тестирования программного обеспечения.

7.2 Содержание практических занятий и лабораторных работ

Тема 2. Объектно-ориентированное проектирование и реализация программных систем.(УК-2, ОПК-2, ОПК-5)

Проектирование ПО с использованием UML. Разработка ПО на объектно-ориентированных языках программирования (Java, C#)

Тема 3. Разработка интернет-приложений.(УК-2, ОПК-2, ОПК-5)

Современные Web-технологии

Тема 4. Организация процесса тестирования и оценки качества программных систем.(УК-2, ОПК-2, ОПК-5)

Тестирование программного обеспечения. Использование специального ПО. Автотестирование.

7.3. Содержание самостоятельной работы

Тема 2. Объектно-ориентированное проектирование и реализация программных систем.(УК-2, ОПК-2, ОПК-5)

Изучение основной и дополнительной литературы по теме, интернет-источников. Работа над проектом.

Тема 3. Разработка интернет-приложений.(УК-2, ОПК-2, ОПК-5)

Изучение основной и дополнительной литературы по теме, интернет-источников. Работа над проектом.

Тема 4. Организация процесса тестирования и оценки качества программных систем.(УК-2, ОПК-2, ОПК-5)

Изучение основной и дополнительной литературы по теме, интернет-источников. Работа над проектом.

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену
Приложение 2.

7.3.3. Перечень курсовых работ
Не предусмотрено.

7.4. Электронное портфолио обучающегося
Материалы не размещаются

7.5. Методические рекомендации по выполнению контрольной работы
Не предусмотрено.

7.6 Методические рекомендации по выполнению курсовой работы
Не предусмотрено.

8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

По заявлению студента

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Сайт библиотеки УрГЭУ

<http://lib.usue.ru/>

Основная литература:

2. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2023. - 235 – Режим доступа: <https://urait.ru/bcode/511891>

3. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 248 – Режим доступа: <https://urait.ru/bcode/536903>

4. Чернышев С. А. Принципы, паттерны и методологии разработки программного обеспечения [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 176 – Режим доступа: <https://urait.ru/bcode/544319>

5. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования [Электронный ресурс]: учебник для вузов. - Москва: Юрайт, 2025. - 241 – Режим доступа: <https://urait.ru/bcode/581329>

Дополнительная литература:

2. Плещев В. В., Шишков Е. И. Основы программирования на языках С++ и С# с примерами и упражнениями [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2018. - 286 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf>

3. Гагарина Л.Г., Гаращенко А.В., Акимова Е.Н., Капитанов А.И., Капитанова И.И., Тихонов М.Р. Конструирование программного обеспечения [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2024. - 319 – Режим доступа: <https://znanium.com/catalog/product/1893880>

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Перечень лицензионного программного обеспечения:

Astra Linux Common Edition. Договор №0417-ПО/2019 от 08.05.2019, Акт №Sk000343 от 24.05.2019 и Контракт № 35-У/2018 от 13.06.2018, Акт № УТ213 от 17.12.2018. Срок действия лицензии - без ограничения срока.

МойОфис стандартный. Соглашение № СК-281 от 7 июня 2017. Дата заключения - 07.06.2017. Срок действия лицензии - без ограничения срока.

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

MySQL Community Server. Стандартная общественная лицензия GNU (GPL). Срок действия лицензии - без ограничения срока.

Язык программирования Python. Python Software Foundation License (PSFL). Срок действия лицензии - без ограничения срока.

Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:

Справочно-правовая система Гарант. Договор № 58419 от 22 декабря 2015. Срок действия лицензии - без ограничения срока

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду УрГЭУ.

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия, обеспечивающие тематические иллюстрации.

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

К зачету

1. Основные принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, наследование.
2. Среда .NET Framework и ее две составляющие.
3. Обычные типы данных в языке C#. Способы форматирования вывода.
4. Литералы. Символьные escape-последовательности.
5. Понятие «переменная». Области видимости и время жизни переменных.
6. Арифметические, логические операторы и операторы сравнения в C#.
7. Операторы присваивания. Понятие «выражение». Правила автоматического преобразования типов в выражениях.
8. Условный оператор и его модификации. Оператор множественного условия.
9. Циклы в C#. Циклы с предусловием, постусловием и параметром.
10. Операторы break, continue, goto.
11. Понятия «класс» и «объект». Общий синтаксис и определения класса. Переменные ссылочного типа.
12. Методы.
13. Конструкторы. Оператор new.
14. Деструкторы. Ключевое слово this.
15. Одномерные и многомерные массивы в C#.
16. Цикл foreach.
17. Операции со строками в C#.
18. Побитовые операторы в C#.
19. Модификаторы доступа в C#.
20. Передача объектов методу. Модификаторы ref и out.

- 21.Использование переменного количества аргументов в методах.
Возвращение объектов.
- 22.Перегрузка методов и конструкторов.
- 23.Метод Main().
- 24.Рекурсия. Ключевое слово static.
- 25.Перегрузка операторов. Основные положения и ограничения при перегрузке операторов.
- 26.Индексаторы и свойства. Ограничения в использовании свойств.
- 27.Наследование. Модификатор protected. Ключевое слово super.
- 28.Виртуальные методы и переопределение.
- 29.Абстрактные классы.
- 30.Класс Object. Упаковка и распаковка.
- 31.Интерфейсы.
- 32.Структуры и перечисления.
- 33.Обработка исключений. Блоки try – catch – finally.
- 34.Генерирование исключений. Оператор throw.
- 35.Ключевые слова checked и unchecked.
- 36.Потоки ввода/вывода в C#.
- 37.Работа с файлами в C#.
- 38.Делегаты.
- 39.События.
- 40.Пространства имен.
- 41.Препроцессор. Директивы препроцессора.
- 42.Атрибуты.
- 43.Работа с формами в C#. Свойства окна. Toolbox.
- 44.Многопоточное программирование в C#.
- 45.Язык разметки XML.
- 46.Основные элементы управления в Visual Studio.

47.Работа с базами данных в Visual Studio.

48.Технология ADO.NET.

49.Элементы управления Visual Studio для работы с печатью и изображениями

50.Работа с графическими объектами в C#.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену

Примерные вопросы закрытого типа (УК-2, ОПК-2, ОПК-5)

1. Что такое Agile методология разработки ПО?
 - A. Методология, ориентированная на гибкость и итеративность
 - B. Методология, ориентированная на жесткое планирование и документацию
 - C. Методология, ориентированная на индивидуальные таланты разработчиков
 - D. Методология, ориентированная на формальные процессы и стандарты
2. Что такое DevOps?
 - A. Методология, объединяющая разработку и тестирование ПО
 - B. Подход, объединяющий разработку и операционную деятельность
 - C. Методология, ориентированная на разработку дизайна ПО
 - D. Подход, ориентированный на разработку аппаратного обеспечения
3. Что такое Continuous Integration (непрерывная интеграция)?
 - A. Процесс объединения кода разработчиков в общий репозиторий
 - B. Процесс автоматической сборки и тестирования кода после каждого коммита
 - C. Процесс ручного объединения кода разработчиков
 - D. Процесс разработки без использования систем контроля версий
4. Что такое Scrum?
 - A. Методология разработки ПО, основанная на итерациях и ролевом распределении
 - B. Методология разработки ПО, основанная на формальных документах и процессах
 - C. Методология разработки ПО, основанная на индивидуальных усилиях разработчиков
 - D. Методология разработки ПО, основанная на строгом планировании и долгосрочных целях
5. Что такое Test-Driven Development (TDD)?
 - A. Методология разработки, при которой тесты пишутся до написания кода
 - B. Методология разработки, при которой код пишется до написания тестов
 - C. Методология разработки, при которой тесты пишутся после написания кода
 - D. Методология разработки, при которой тесты не используются
6. Что такое контейнеризация?
 - A. Технология, позволяющая упаковывать приложения и их зависимости в изолированные контейнеры
 - B. Технология, позволяющая объединять несколько приложений в одно
 - C. Технология, позволяющая управлять сетевыми контейнерами
 - D. Технология, позволяющая управлять базами данных
7. Что такое микросервисная архитектура?
 - A. Архитектурный подход, при котором приложение разбивается на независимые микросервисы
 - B. Архитектурный подход, при котором приложение разрабатывается как единое целое
 - C. Архитектурный подход, при котором приложение разрабатывается без использования сервисов
 - D. Архитектурный подход, при котором приложение разрабатывается на одной платформе
8. Что такое версионирование кода?
 - A. Процесс отслеживания изменений в коде и сохранения их в системе контроля версий
 - B. Процесс создания разных версий программного обеспечения
 - C. Процесс создания резервных копий кода
 - D. Процесс создания документации к коду
9. Что такое Continuous Deployment (непрерывное развертывание)?

- A. Процесс автоматического развертывания изменений в продакшн среду после успешного прохождения тестов
 - B. Процесс ручного развертывания изменений в продакшн среду
 - C. Процесс развертывания изменений только после ручного подтверждения
 - D. Процесс развертывания изменений только по расписанию
10. Что такое рефакторинг?
- A. Процесс улучшения структуры и читаемости кода без изменения его функциональности
 - B. Процесс добавления новых функций в код
 - C. Процесс удаления устаревших функций из кода
 - D. Процесс изменения функциональности кода без улучшения его структуры

Примерные практические задания к зачету(УК-2, ОПК-2, ОПК-5)

1. Разработать Windows-приложение «Конвертер», которое пересчитывает цену из долларов в рубли. Обеспечить безопасный ввод: пользователь может ввести в поля редактирования только числа и одну запятую.

2. Разработать Windows-приложение «Автозаправка», которое вычисляет количество литров бензина, которое можно купить на заданную сумму. Выбор бензина реализовать с помощью выплывающего списка. В поле «Сумма» можно ввести только числовые данные.

3. Разработать Windows-приложение «Кафе», которое рассчитывает сумму заказа, исходя из выбранных пунктов меню. Выбор блюд реализовать с помощью элементов CheckBox, а выбор альтернативных продуктов – с помощью элементов RadioButton.

4. Разработать Windows-приложение «Инженерный калькулятор». Ввод данных обеспечить как по нажатию на кнопки формы, так и по нажатию на числа на клавиатуре. Помимо стандартных операций, реализовать операции возведения в степень и вычисления квадратного корня.

5. Разработать Windows-приложение «Таймер», используя компоненты Timer и NumericUpDown. Программа должна позволять задать время (минуты и секунды) и по нажатию кнопки Пуск начинать отсчет заданного времени.

6. Разработать Windows-приложение, строящее диаграмму или график по заданным в файле данным. Создать меню с командами Input Data, Choose, Line, Bar, Quit. Команды Line и Bar недоступны. При выборе команды

Input Data из файла читаются исходные данные. По команде Choose открывается диалоговое окно, содержащее: список для выбора цвета графика типа TListBox; группу из двух переключателей (Line, Bar) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода цвета и выбора режима: график или диаграмма. После указания параметров становится доступной соответствующая команда меню. По команде Line или Bar в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градуацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.

7. Описать класс для работы с одномерным массивом целых чисел. Обеспечить следующие возможности: 1. задание произвольных целых границ; 2. Обращение к отдельному элементу массива с контролем выхода за пределы массива; 3. Выполнение операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов; 4. Выполнение операций умножения и деления всех элементов массива на скаляр; 5. Вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую все разработанные элементы класса.

8. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение. Написать программу, демонстрирующую все разработанные элементы класса.

9. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую все разработанные элементы класса.

10. Составить описание класса для вектора, заданного координатами его концов в трехмерном пространстве. Обеспечить операции сложения и вы-

читания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую все разработанные элементы класса.

11. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую все разработанные элементы класса.

12. Составить описание класса для представления времени. Предусмотреть возможности установки времени и изменения его отдельных полей (час, минута, секунда) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются исключения. Создать методы изменения времени на заданное количество часов, минут и секунд. Написать программу, демонстрирующую все разработанные элементы класса.

13. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, вычисления площади, периметра и точки пересечения медиан. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение. Написать программу, демонстрирующую все разработанные элементы класса.

14. Описать класс, представляющий круг. Предусмотреть методы для создания объектов, вычисления площади круга, длины окружности и проверки попадания заданной точки внутрь круга. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.

15. Описать класс дробей — рациональных чисел, являющихся отношением двух целых чисел. Предусмотреть методы сложения, вычитания, умножения и деления дробей. Написать программу, демонстрирующую все разработанные элементы класса.

16. Описать класс «комната», содержащий сведения о метраже, высоте потолков количестве окон. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать методы вычисления площади и объема комнаты и свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.

17. Описать класс, представляющий квадратное уравнение вида $ax^2 + bx + c = 0$. Описать метод, вычисляющий решение этого уравнения и выбрасывающий исключение в случае отсутствия корней. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.

18. Описать класс «процессор», содержащий сведения о марке, тактовой частоте, объеме кэша и стоимости. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать свойства для получения состояния объекта. Описать класс «материнская плата», включающий класс «процессор» и объем установленной оперативной памяти. Предусмотреть инициализацию с проверкой допустимости значений поля объема памяти. В случае недопустимых значений поля выбрасывается исключение. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы классов.

19. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по фамилии и доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.

20. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и

удаления записей, сортировки по разным полям, доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.

21. Описать класс «колода карт», включающий закрытый массив элементов класса «карта». В карте хранятся масть и номер. Обеспечить возможность вывода карты по номеру, вывода всех карт, перемешивания колоды и выдачи всех карт из колоды поодиночке и по 6 штук в случайном порядке. Написать программу, демонстрирующую все разработанные элементы классов.

22. Описать класс «поезд», содержащий следующие закрытые поля: название пункта назначения; номер поезда (может содержать буквы и цифры); время отправления. Предусмотреть свойства для получения состояния объекта. Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности: вывод информации о поезде по номеру с помощью индекса; вывод информации о поездах, отправляющихся после введенного с клавиатуры времени; перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов; вывод информации о поездах, отправляющихся в заданный пункт назначения. Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

23. Описать класс «товар», содержащий следующие закрытые поля: название товара; название магазина, в котором продается товар; стоимость товара в рублях. Предусмотреть свойства для получения состояния объекта. Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности: вывод информации о товаре по номеру с помощью индекса; вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение; сортировку товаров по названию магазина, по наименованию и по цене; перегруженную операцию сложения товаров, выполняющую сложение их цен. Написать программу, демонстрирующую все разработанные элементы классов.

24. Описать класс «самолет», содержащий следующие закрытые поля: название пункта назначения; шестизначный номер рейса; время отправления. Предусмотреть свойства для получения состояния объекта. Описать класс «аэропорт», содержащий закрытый массив самолетов. Обеспечить следующие возможности: вывод информации о самолете по номеру рейса с помощью индекса; вывод информации о самолетах, отправляющихся в течение часа после введенного с клавиатуры времени; вывод информации о самолетах, отправляющихся в заданный пункт назначения; перегруженную операцию сравнения, выполняющую сравнение времени отправления двух самолетов. Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.

25. Описать класс «запись», содержащий следующие закрытые поля: фамилия, имя; номер телефона; дата рождения (массив из трех чисел). Предусмотреть свойства для получения состояния объекта. Описать класс «записная книжка», содержащий закрытый массив записей. Обеспечить следующие возможности: вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение; поиск людей, день рождения которых сегодня или в заданный день; поиск людей, день рождения которых будет на следующей неделе; поиск людей, номер телефона которых начинается на три заданных цифры. Написать программу, демонстрирующую все разработанные элементы классов.

26. Описать базовый класс «Строка». Обязательные поля класса: поле для хранения символов строки; значение типа `word` для хранения длины строки в байтах. Реализовать обязательные методы следующего назначения: конструктор без параметров; конструктор, принимающий в качестве параметра строковый литерал; конструктор, принимающий в качестве параметра символ; метод получения длины строки; метод очистки строки. Описать производный от «Строка» класс «Комплексное число». Строки данного класса состоят из двух полей, разделенных символом `i`. Первое поле задает значение действительной части числа, второе – значение мнимой. Каждое из полей мо-

жет содержать только символы десятичных цифр и символы + и -, задающие знак числа. Символы + и - могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Для класса «Комплексное_число» определить следующие методы: проверка на равенство; сложение чисел; умножение чисел.

27. Создать меню с командами Begin, Work, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее: два поля ввода; группу из двух переключателей (First letter, All letter) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода предложения и выбора режима его преобразования: либо начинать с прописной буквы каждое слово (First letter), либо перевести все буквы в верхний регистр (All letter). При выборе команды Work открывается диалоговое окно с результатами преобразования.

28. Написать Windows-приложение – графическую иллюстрацию сортировки методом выбора. Создать меню с командами File, Animate, About, Exit. Команда Animate недоступна. Команда Exit завершает работу приложения. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда File) использовать объект класса OpenFileDialog. Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа). После чтения данных становится доступной команда Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.