

Кафедра _____ прикладной информатики и высшей математики

Б1.О.21

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА)

Учебная дисциплина	<i>Алгоритмизация и программирование</i>
По направлению подготовки	<i>09.03.03</i>
	<i>«Прикладная информатика»</i>
Профиль (программа бакалавриата)	<i>«Прикладная информатика в цифровой экономике»</i>
Форма обучения	<i>Очная</i>

Оценочные материалы (средства) дисциплины рассмотрены (актуализированы) и утверждены на заседании кафедры прикладной информатики и высшей математики

Протокол заседания № 10 от «25» мая 2026 г.

Заведующий кафедрой Стрекалова Наталья Борисовна

1. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ И ЭТАПЫ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Оценочные материалы (средства) сформированы в соответствии с ФГОС ВО - бакалавриат по направлению подготовки 09.03.03 «Прикладная информатика», утвержденный приказом Министерства образования и науки Российской Федерации от 19.09.2017 № 922 (с изменениями и дополнениями от 26.11.2020, 08.02.2021). В соответствии с матрицей компетенций основной профессиональной образовательной программы «Прикладная информатика в цифровой экономике» (уровень бакалавриата) в процессе обучения по дисциплине «Алгоритмизация и программирование» происходит формирование закрепленных за дисциплиной компетенций обучающихся. Оценка сформированности компетенций на каждом этапе обучения происходит через оценку планируемых результатов обучения по дисциплине (знаний, умений, навыков).

Результаты освоения образовательной программы (компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения по дисциплине	Этапы формирования компетенции (семестры и темы)	Оценочные средства
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-7.1. Составляет алгоритмы для решения стандартных профессиональных задач	Знать: - основные понятия теории алгоритмов (алгоритм, исполнитель, виды и типы алгоритмов), правила описания алгоритмов; Уметь: - формализовано описывать поставленные задачи; - разрабатывать оптимальные алгоритмы для решения поставленных задач; Владеть: - навыками оформления алгоритмов в соответствии с требованиями ГОСТ	Семестр 1 Тема 2. Технологии и средства программирования. Алгоритмические языки и алгоритмы Тема 3. Программная модель. Синтаксис языка C#. Тема 4. Типовая структура языка C# Тема 5. Операторы языка C# Семестр 2 Тема 7. Обработка массивов и строк	- устный опрос по теме 3 - практические работы по темам 2; - проверочная работа по теме 5 - устный ответ на экзамене; - выполнение практического задания на экзамене;
	ОПК-7.2. Разрабатывает программы, производит их отладку и проверку работоспособности	Знать: - этапы проектирования программ в соответствии с жизненным циклом информационных систем; - существующие технологии и средства программирования; - синтаксис высокоуровневого языка C#, структуру программы, операторы и управляющие конструкции, типовую структуру данных; Уметь: - определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм	Семестр 1 Тема 1. Жизненный цикл программного продукта. Технология командной разработки программного продукта Тема 2. Технологии и средства программирования. Алгоритмические языки и алгоритмы Тема 3. Программная модель. Синтаксис языка C#. Тема 4. Типовая структура языка C# Тема 5. Операторы	- устный опрос по теме 1 - устный опрос по теме 3 - тестирование по темам 3-4; - диктант по теме 5 - практические работы по темам 2; - проверочная работа по теме 5 - практические работы по теме 6 - результаты командной разработки по теме 6; - диктант по темам 7

		(обработчиков событий). - разрабатывать программы для решения учебных задач из различных предметных областей; Владеть: - навыками выбора типов данных и проектирования структур данных; - навыками отладки и проверки работоспособности консольных и визуальных приложений.	языка C# Тема 6. Разработка и перегрузка функций Семестр 2 Тема 7. Обработка массивов и строк Тема 8. Разработка структур данных Тема 9. Работа с файлами Тема 10. Введение в визуальную технологию программирования Тема 11. Разработка интерфейса приложения Тема 12. Построение событийной модели приложения Тема 13. Разработка многодокументных приложений Тема 14. Основы разработки СУБД в визуальной технологии	- практические работы по темам 7 - диктант по теме 8 - практические работы по темам 8 - результаты командной разработки по теме 9 - устный опрос по теме 10 - практические работы по темам 11, 12; - индивидуальное проектное задание по теме 13; - доклад по теме 14 - устный ответ на экзамене; - выполнение практического задания на экзамене;
ОПК-8 Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла	ОПК-8.1. Планирует работы проекта в соответствии с полученным заданием на разных стадиях жизненного цикла информационных систем, в том числе используя облачные сервисы и приложения для управления проектами в сфере ИТ	Знать: - состав команды разработчиков программного проекта, принципы и способы формирования проектной группы; Уметь: - распределять работу в группе с учетом профессиональных компетенций отдельных участников; Владеть: - навыками работы в проектных группах по разработке программного обеспечения, в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.);	Семестр 1 Тема 1. Жизненный цикл программного продукта. Технология командной разработки программного продукта Тема 6. Разработка и перегрузка функций Семестр 2 Тема 9. Работа с файлами	- устный опрос по теме 1 - результаты командной разработки по теме 6; - защита командной разработки по теме 6; - результаты командной разработки по теме 9 - защита командной разработки по теме 9 - представление эссе по теме 9; - устный ответ на экзамене;
	ОПК-8.2. Осуществляет мониторинг и оценку выполнения работ по	Знать: - технологические средства управления работой команды; Уметь: - подводить итоги	Семестр 1 Тема 1. Жизненный цикл программного продукта. Технология командной	- устный опрос по теме 1 - результаты командной разработки по теме 6;

	проекту, в том числе используя облачные сервисы и приложения для управления командами разработчиков в	командной работы и оценивать собственную деятельность и деятельность отдельных участников; Владеть: - навыками оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной деятельности (сервисы Yandex, Google и др.)..	разработки программного продукта Тема 6. Разработка и перегрузка функций Семестр 2 Тема 9. Работа с файлами	- защита командной разработки по теме 6; - результаты командной разработки по теме 9 - защита командной разработки по теме 9 - представление эссе по теме 9; - устный ответ на экзамене;
ОПК-9 Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп	ОПК-9.1. Осуществляет профессиональные коммуникации с участниками проектных групп, в том числе с применением средств видеоконференцсвязи, облачных сервисов и приложений для управления проектными командами	Знать: - особенности и средства взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения. Уметь: - взаимодействовать внутри проектной группы и с ее окружением во время разработки программного проекта, в том числе с применением средств удаленного взаимодействия; Владеть: - навыками проведения обсуждений проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	Семестр 1 Тема 1. Жизненный цикл программного продукта. Технология командной разработки программного продукта Тема 6. Разработка и перегрузка функций Семестр 2 Тема 9. Работа с файлами	- устный опрос по теме 1 - защита командной разработки по теме 6; - защита командной разработки по теме 9 - устный ответ на экзамене;
	ОПК-9.2. Осуществляет профессиональные коммуникации с заказчиком проектной деятельности, в том числе с применением средств удаленного взаимодействия и видеоконференцсвязи	Знать: - способы демонстрации результатов программной разработки; Уметь: - представлять программную разработку потенциальному Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.); Владеть: - навыками ответов на вопросы и обоснования принятых программных решений.	Семестр 1 Тема 1. Жизненный цикл программного продукта. Технология командной разработки программного продукта Тема 6. Разработка и перегрузка функций Семестр 2 Тема 9. Работа с файлами	- устный опрос по теме 1 - защита командной разработки по теме 6; - защита командной разработки по теме 9 - устный ответ на экзамене;

2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ОБЕСПЕЧИВАЮЩИХ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Устные опросы

*Примерные вопросы для проведения устного опроса
по теме «Жизненный цикл программного продукта. Технология командной разработки
программного продукта»*

1. Понятие программного продукта.
2. Этапы жизненного цикла программного продукта.
3. Характеристика отдельных этапов жизненного цикла программного продукта (исследование, проектирование, кодирование, тестирование, внедрение).
4. Командный характер современных программных разработок.
5. Состав команды разработчиков программного проекта.
6. Принципы и способы формирования проектной группы.
7. Особенности жизненного цикла ИТ-проекта по разработке программного продукта, его структуризация.
8. Особенности и средства управления командной разработкой.
9. Особенности и средства коммуникационного взаимодействия внутри команды.
10. Понятие программирования.
11. Виды программирования (машинные коды, мнемокоды, структурное программирование, процедурное программирование и т.д.)
12. Технологии программирования (модульная, объектная, визуальная).
13. Основные понятия платформы .NET Framework, ее состав и особенности работы, требования к архитектуре ПК, установка платформы .NET Framework и Visual Studio на компьютер.

*Примерные вопросы для проведения устного опроса
по теме «Программная модель. Синтаксис языка C#»*

1. Алфавит языка C#.
2. Синтаксис языка: константы и переменные
3. Идентификаторы и общие правила их написания, основные конструкции (директивы, лексемы, комментарии)
4. Виды лексем, разделители кода.
5. Понятие операции. Виды операций.
6. Унарные операции.
7. Арифметические операции.
8. Операции сравнения и логические операции.
9. Запись математических функций и выражений. Приоритет выполнения операций.
10. Понятие программной модели, ее виды (блочная, модульная, объектная) и их особенности.
11. Структура программы на языке C#. Порядок выполнения программы.
12. Назначение и особенности функции Main.
13. Глобальные и локальные данные программы, их область видимости и время жизни.

*Примерные вопросы для проведения устного опроса
по теме «Введение в визуальную технологию программирования»*

1. Отличия работы консольных приложений от визуальных.
2. Принципы разработки визуальных приложений.
3. Технологии, применяемые для разработки визуальных приложений.
4. Понятие проекта и его состав.
5. Назначение объекта Application.
6. Событийная модель работы приложения.
7. Цикл работы приложения.
8. Механизм передачи события из ядра Windows в приложение.
9. Перечислите события, связанные с консолью.
10. Формат заголовка обработчика события.
11. Перечислить события, связанные с RAM.

Критерии оценивания устного опроса:

- оценка «*отлично*» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов, активно дополнял ответы других обучающихся или задавал им дополнительные вопросы;
- оценка «*хорошо*» выставляется обучающемуся, если он ответил развернуто на один из заданных вопросов или ответил кратко на ряд вопросов;
- оценка «*удовлетворительно*» выставляется обучающемуся, если он кратко ответил на один из задаваемых вопросов или просто дополнял ответы других обучающихся, задавал им дополнительные вопросы;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если он не смог ответить правильно на заданный вопрос, либо не отвечал на вопросы и не участвовал в их обсуждении.

Тестирование

Примерные вопросы для тестирования

по темам «Программная модель. Синтаксис языка C#», «Типовая структура языка C#»

1. Что входит в алфавит языка C#:
 - ☐ латинские буквы
 - ☐ арабские цифры
 - ☐ литералы
 - ☐ спецсимволы
 - ☐ знаки операций
 - ☐ идентификаторы
 - ☐ служебные слова
 - ☐ операторы
2. Выделите основные синтаксические конструкции языка:
 - ☐ инструкция
 - ☐ идентификатор
 - ☐ комментарий
 - ☐ слово
 - ☐ лексема
 - ☐ строка
 - ☐ разделитель
3. Без каких спецсимволов невозможно написать простейшую программу на языке C#:
 - a. ()
 - b. {}
 - c. :

d. ;

e. //

f. =

g. ==

4. Какие логические операции применяются в языке?

☐ not

☐ &&

☐ or

☐ and

☐ !

☐ !=

5. Выделите унарные операции:

☐ +

☐ -

☐ &

☐ %

☐ ++

☐ !

☐ ==

6. Какие конструктивы позволяют строить эффективные формы присваивания?

☐ составные операторы

☐ инкремент

☐ декремент

☐ обычные операторы

☐ присваивание

7. О чем говорит «пустой» тип?

☐ У данного нет значения

☐ Данное отсутствует

☐ У функции нет результата

8. Перечислить все числовые типы данных, занимающие в памяти 2 байта и менее:

☐ byte

☐ sbyte

☐ short

☐ ushort

☐ int

☐ uint

☐ long

☐ ulong

☐ float

☐ double

☐ extended

9. Выделите все целые типы со знаком:

☐ byte

☐ sbyte

☐ short

☐ ushort

- ☐ int
- ☐ uint
- ☐ long
- ☐ ulong
- ☐ float
- ☐ double
- ☐ extended

10. Выделите типы, позволяющие хранить целое число из диапазона 0...65535:

- ☐ byte
- ☐ sbyte
- ☐ short
- ☐ ushort
- ☐ int
- ☐ uint
- ☐ long
- ☐ ulong
- ☐ float
- ☐ double
- ☐ extended

11. Выделите все типы, позволяющие хранить число 0,34567891291:

- ☐ byte
- ☐ sbyte
- ☐ short
- ☐ ushort
- ☐ int
- ☐ uint
- ☐ long
- ☐ ulong
- ☐ float
- ☐ double
- ☐ extended

12. Введите название типа, используемого для описания функции, у которой нет результата (открытый вопрос)

13. Выделите простые типы данных

- ☐ символ
- ☐ короткое целое
- ☐ длинное целое
- ☐ строка
- ☐ вещественное с двойной точностью
- ☐ структура
- ☐ одномерный массив
- ☐ логическое данное

14. Выделите структурированные типы данных:

- ☐ символ
- ☐ короткое целое
- ☐ длинное целое

- ☐ строка
- ☐ вещественное с двойной точностью:
- ☐ структура
- ☐ одномерный массив
- ☐ логическое данное

15. Выделите типы-значения:

- ☐ символ
- ☐ короткое целое
- ☐ длинное целое
- ☐ строка
- ☐ вещественное с двойной точностью
- ☐ структура
- ☐ одномерный массив
- ☐ логическое данное

16. Что не входит в характеристики типа?

- ☐ объем выделяемой памяти
- ☐ идентификатор данного
- ☐ диапазон возможных значений
- ☐ перечень допустимых операций
- ☐ модификатор доступа
- ☐ внутренний формат представления

17. В каком сегменте программы хранятся локальные данные (или ссылки на них)?

- ☐ сегмент данных
- ☐ сегмент стека
- ☐ сегмент кода

18. Выберите безошибочные объявления переменных:

- ☐ float b, float c;
- ☐ int a, b, c;
- ☐ int a=0, b;
- ☐ char d=!;

19. Какой модификатор позволяет описать константу?

- ☐ public
- ☐ var
- ☐ const
- ☐ static

20. К какому типу программ относится программа на языке C#:

- a. блочный
- b. модульный

21. Какие структурные элементы присутствуют в программе, написанной на языке C#?

- ☐ разделы
- ☐ подпрограммы
- ☐ процедуры
- ☐ функции
- ☐ модули
- ☐ классы

22. Выделите верные утверждения о функции Main:

- ☐ Должна обязательно присутствовать в тексте программы
 - ☐ в простейшем случае может отсутствовать
 - ☐ С нее начинается работа программы и ею же заканчивается
 - ☐ Получает исходные данные от операционной системы
 - ☐ Всегда должна возвращать результат работы программы
23. От чего зависит время жизни переменной?
- от используемого модификатора во время объявления
 - от местоположения объявления и используемого модификатора
 - от местоположения объявления

Критерии оценивания тестирования:

- оценка «отлично» выставляется обучающемуся, если он правильно ответил на 85% вопросов теста;
- оценка «хорошо» выставляется обучающемуся, если он правильно ответил на 75% вопросов теста;
- оценка «удовлетворительно» выставляется обучающемуся, если он правильно ответил на 50% вопросов теста;
- оценка «неудовлетворительно» выставляется обучающемуся, если он ответил менее, чем на 50% вопросов теста.

Диктант

*Примерные вопросы для проверочного диктанта
по теме «Операторы языка C#»,*

1. Формат оператора присваивания и принцип его работы.
2. Эффективные формы оператора присваивания
3. Особенность префиксной формы записи оператора инкремента.
4. Особенность постфиксной формы записи оператора инкремента.
5. Формат объявления переменной
6. Формат объявления константы
7. Формат оператора ввода данного с клавиатуры.
8. Формат оператора ввода данного с клавиатуры с преобразованием типа.
9. Формат операторы вывода одного данного на экран.
10. Формат операторы вывода нескольких данных на экран
11. Формат условного оператора и порядок его работы.
12. Формат цикла с предусловием и порядок его работы.
13. Формат оператора вариантного выбора (переключатель). Оператор break.
14. Формат цикла с постусловием и порядок его работы.
15. Формат цикла с известным количеством повторений

*Примерные вопросы для проверочного диктанта
по темам «Обработка массивов и строк», «Разработка структур данных»*

1. Отличие структурированного типа от простого.
2. Отличие типа-значения от типа-ссылки
3. Виды структурированных типов.
4. Понятие массива.
5. Понятие структуры

6. Понятие строки
7. Что такое размерность массива?
8. Как определяется адрес элемента одномерного массива?
9. Как определить длину строки?
10. Формат объявления одномерного массива.
11. Формат объявления двумерного массива.
12. Формат описания типа структуры
13. Формат объявления переменной структуры
14. Обращение к элементу массива.
15. Обращение к полю структуры
16. Обращение к отдельному символу строки
17. Особенности передачи массивов между функциями?
18. Ограничения в применении цикла `foreach`
19. Условия применения цикла `for` для обработки массивов

Критерии оценивания результатов диктанта:

- оценка «*отлично*» выставляется обучающемуся, если он правильно ответил на 85% вопросов диктанта;
- оценка «*хорошо*» выставляется обучающемуся, если он правильно ответил на 75% вопросов диктанта;
- оценка «*удовлетворительно*» выставляется обучающемуся, если он правильно ответил на 50% вопросов диктанта;
- оценка «*неудовлетворительно*» выставляется обучающемуся, если он ответил менее, чем на 50% вопросов диктанта.

Подготовка и выступление с докладом

Темы для подготовки докладов

по теме «Основы разработки СУБД в визуальной технологии»

1. Реляционные базы данных: суть, назначение, понятийный аппарат.
2. Системы управления базами данных: назначение и решаемые задачи, основной и вспомогательный функционал.
3. Основные компоненты для разработки СУБД и связи между ними.
4. Визуальный способ разработки СУБД
5. Программный способ разработки СУБД
6. Построение отчетов по БД в визуальной технологии

Критерии оценивания доклада и выступления:

- оценка «*отлично*» выставляется обучающемуся, если доклад полностью раскрывает тему, структура доклада логично выстроена, обучающийся хорошо ориентируется в материале и текст доклада, в основном, рассказывает (а не читает), отвечает на дополнительные вопросы; сопровождающая доклад презентация выполнена в соответствии с требованиями и дополняет доклад, докладчик уложился в регламент, текст доклада оформлен в соответствии с требованиями;
- оценка «*хорошо*» выставляется обучающемуся, если доклад в целом раскрывает тему, но остаются не освещенные стороны вопросы, не на все дополнительные вопросы обучающийся

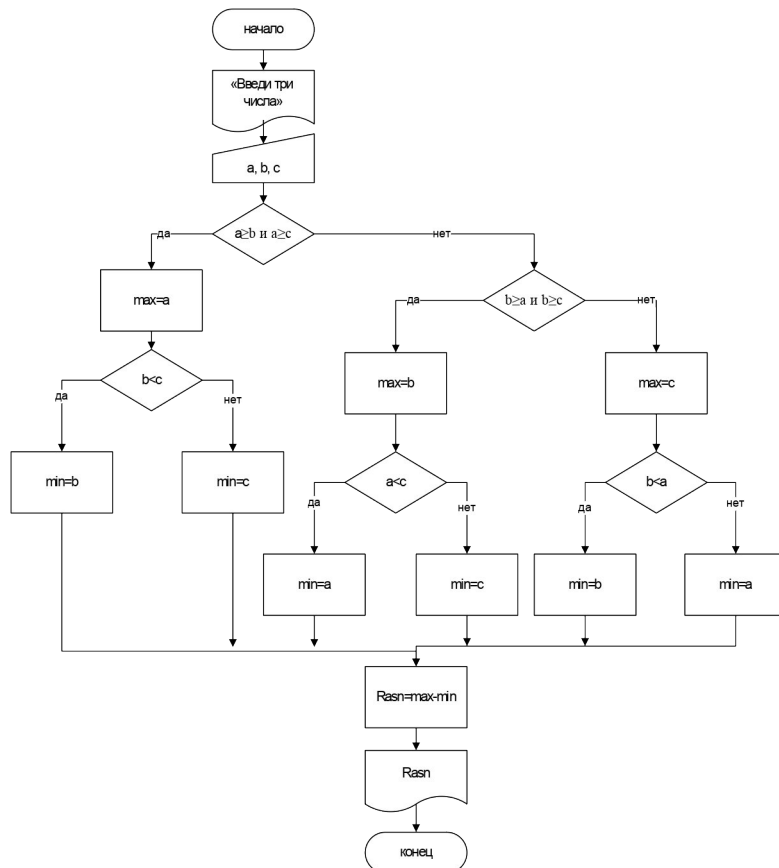
может дать ответ; структура доклада логично выстроена, текст доклада обучающийся, в основном, рассказывает (а не читает); презентация выполнена в соответствии с требованиями, но есть ряд замечаний по ее оформлению, доклад и презентация дополняют друг друга, докладчик уложился в регламент;

- оценка «удовлетворительно» выставляется обучающемуся, если его доклад не раскрывает тему в достаточной мере, не отвечает на дополнительные вопросы; к оформлению доклада много замечаний; презентация выполнена не по требованиям, доклад дублирует презентацию, а не дополняет ее, обучающийся не укладывается в регламент или наоборот;
- оценка «неудовлетворительно» выставляется обучающемуся, если доклад отсутствует или доклад совсем не раскрывает тему, не оформлен по требованиям; отсутствует презентация или презентация не соответствует более половины требованиям.

Проверочная работа по теме «Операторы языка C#»

Задание: Реализовать алгоритмы на языке C#

1. Найти разность между минимальным и максимальным числом из трех введенных с клавиатуры.

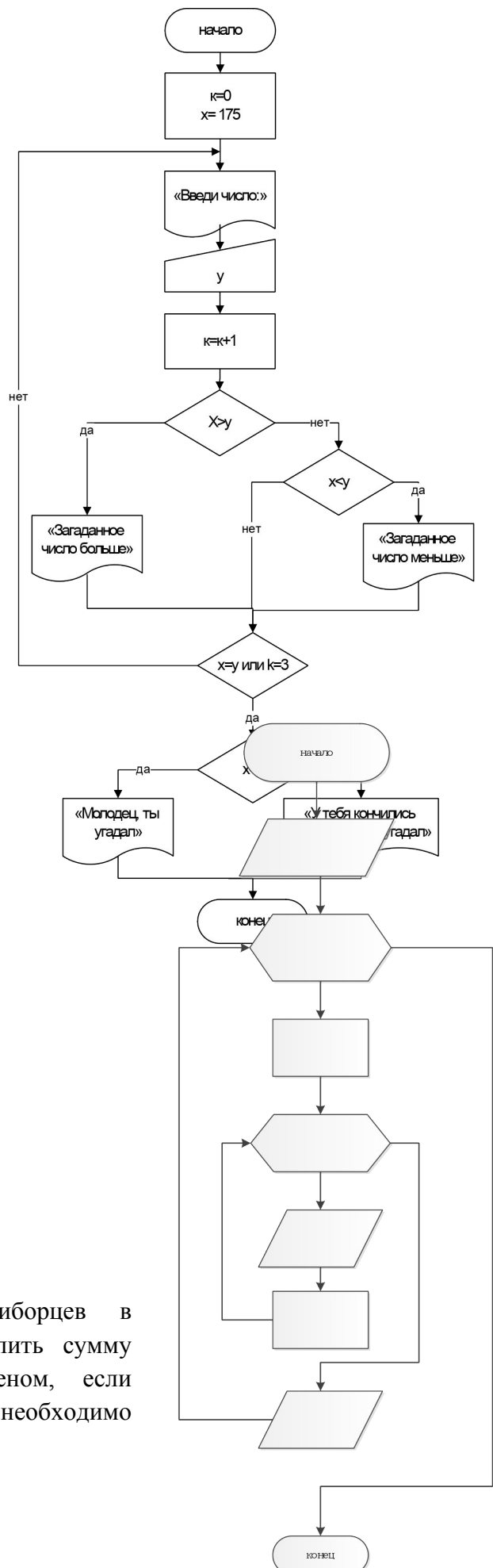
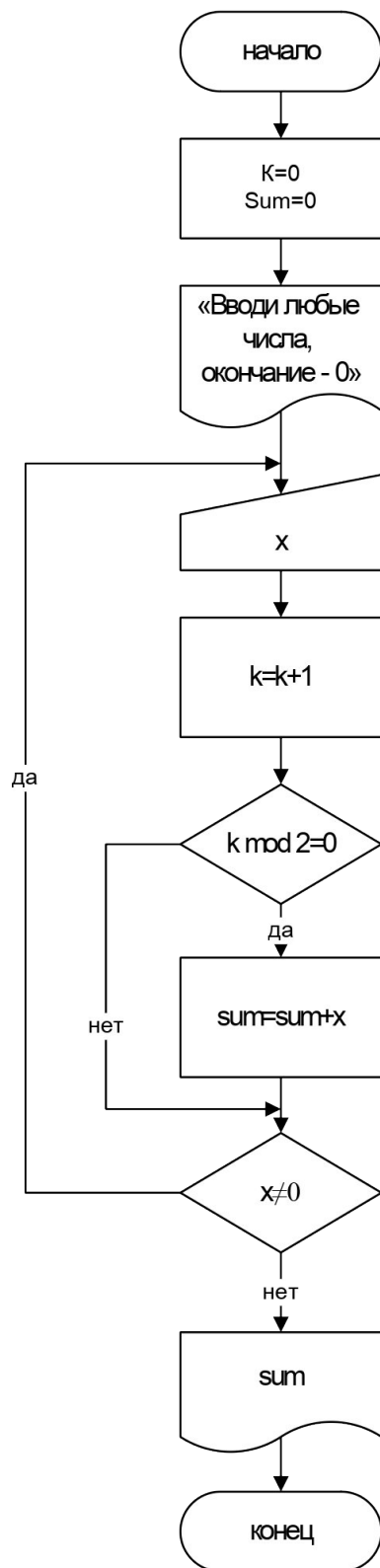


Для вводимой последовательности чисел (окончанием ввода считать число – 0) определить сумму чисел, стоящих на четных местах (не путать с

Разработать алгоритм «угадывания числа», если загаданное число присвоено переменной X, а вариант пользователя – вводится с клавиатуры в переменную Y. Возможны только три попытки. Угадывание

четностью самого числа).

сопровождается подсказками: «больше», «меньше».



4. Известны результаты спортсменов-пятиборцев в каждом виде спорта в баллах. Определить сумму баллов, полученных каждым спортсменом, если количество спортсменов – K, а баллы необходимо ввести с клавиатуры.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если он выполнил задание в полном объеме: 4 задачи реализованы, программы работают без ошибок, все вводы и выводы сопровождаются корректными комментариями, типы данных выбраны адекватно;
- оценка «хорошо» выставляется обучающемуся, если он выполнил в полном объеме 3 задачи из 4: программы работают без ошибок, все вводы и выводы сопровождаются корректными комментариями, типы данных выбраны адекватно; допустимы незначительные отклонения от требований;
- оценка «удовлетворительно» выставляется обучающемуся, если он выполнил в полном объеме 2 задачи из 4: программы работают без ошибок, все вводы и выводы сопровождаются корректными комментариями, типы данных выбраны адекватно; либо 3 задачи из 4, но со значительными отклонениями от требований;
- оценка «неудовлетворительно» выставляется обучающемуся, если он выполнил только 1 задачу, или 2 задачи, но со значительными отклонениями от требований.

Практические работы

Практическое задание по теме «Программная модель. Синтаксис языка C#. Алгоритмизация»

Задача: разработать алгоритм программы для расчета значения функции $f(x) = a \cdot x^n + b$, если аргумент x изменяется на интервале $[\min; \max]$ с шагом step , а все необходимые данные вводятся пользователем с клавиатуры. По результатам расчета определить минимальное и максимальное значение функции на заданном интервале. Предусмотреть возможность повторения расчетов для других исходных данных.

Технология работы:

1. Определить по условию задачи входные данные и результаты работы, формализованно описать модель работы программы.
2. Используя структурно-функциональный подход, предложить деление программы на подпрограммы (функции) и описать работу каждой подпрограммы (название, исходные данные, результат работы).
3. Используя правила построения блок-схем (ГОСТ), детально разработать алгоритм каждой функции. Для каждого алгоритма определить используемые базовые алгоритмические структуры.
4. С помощью языка блок-схем объединить все полученные алгоритмы в один.

Самостоятельно: используя данные Вашего варианта, формализованно описать и детально разработать алгоритм для каждой задачи (5 задач).

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме – безошибочно решены 6 задач, оформление алгоритмов соответствует требованиям (есть описание, перечислены алгоритмические структуры, соблюден ГОСТ);

- оценка «хорошо» выставляется обучающему, если он выполнил задание почти в полном объеме (5-6 задач) с незначительными ошибками в логике или оформлении алгоритма;
- оценка «удовлетворительно» выставляется обучающему, если он выполнил задание частично (не менее 4-х задач), есть незначительные ошибки в логике или оформлении алгоритмов;
- оценка «неудовлетворительно» выставляется обучающему, если он выполнил менее 4 задач, с грубыми ошибками в логике и оформлении алгоритмов.

Практическое задание по теме «Разработка и перегрузка функций»

Задача: смоделировать работу теста по проверке знаний элементарных математических операций (сложение, вычитание, умножение). Задать в тесте решение 10 различных примеров. Оценку выводить в текстовом формате: «отлично» (10-9), «хорошо» (8-7), «удовлетворительно» (6-5), «неудовлетворительно». При ошибочном вводе сразу выводить сообщение об ошибке и правильный ответ. Предусмотреть возможность выбора двух вариантов теста: с двумя операндами и с тремя операндами, а также возможность повтора прохождения теста.

Технология работы:

1. Разработать функции для расчета бинарных операций сложения, вычитания, умножения. Функции принимают два целочисленных аргумента и возвращают результат расчета.
2. Разработать функцию генерации случайной операции – генерация числа из диапазона [1; 3], которое определяет вид операции (1 – сложение, 2 – вычитание, 3 – умножение). Функция возвращает символ операции.
3. Разработать функцию генерации двух случайных операндов – целых чисел из диапазона [1; 20]. Полученные данные являются выходными и оформляются модификатором out.
4. Разработать функцию расчета правильного значения. Функция принимает знак операции и два операнда, а возвращает результат расчета.
5. Разработать функцию формирования на экране задания теста в виде примера («5*7=»). Функция принимает знак операции и два операнда.
6. Разработать функцию анализа ответа. Функция принимает правильное решение и ответ учащегося, сравнивает их и возвращает true или false.
7. Разработать функцию анализа счетчика правильных ответов. Функция принимает счетчик в качестве аргумента, анализирует его и возвращает строковое данное (отлично, хорошо и т.д.).
8. Собрать функции в единую программу, проверить работу программы и отладить при необходимости.
9. Спроектировать механизм управления количеством аргументов в выражениях. Разработать функцию выбора количества аргументов. Перегрузить все функции, зависящие от количества аргументов. Функцию суммирования не перегружать, а отредактировать так, чтобы она имела переменное количество аргументов (параметров). Откорректировать те фрагменты программы, которые зависят от количества аргументов.
10. *Продумать, какие изменения в программе необходимо сделать, если влиять на интервал генерируемых операндов (в частности, для сложения и вычитания диапазон может быть шире – до 100, а для умножения – до 10). Внести изменения в программу и проверить ее работу.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме, обучающийся хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;

- оценка «хорошо» выставляется обучающемуся, если он выполнил задание почти в полном объеме (может отсутствовать пункт со звездочкой, часть перегрузок), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «удовлетворительно» выставляется обучающемуся, если он выполнил первые 8 пунктов задания;
- оценка «неудовлетворительно» выставляется обучающемуся, если он выполнил менее 8 пунктов задания.

Практическое задание по теме «Обработка массивов и строк»

Задание 1:



Разработать функцию заполнения одномерного массива фамилиями (для реального массива количество фамилий – 10).

Разработать функцию вывода содержимого массива на экран. Использовать оператор foreach.

Разработать функцию вывода одной фамилии из массива по ее индексу. Использовать *перегрузку* функций. Вывод на экран осуществлять в формате: **5-я фамилия списка – Иванов.**

Разработать функцию поиска заданной фамилии в массиве, возвращающей ее индекс или значение «-1», если такой фамилии в массиве нет.

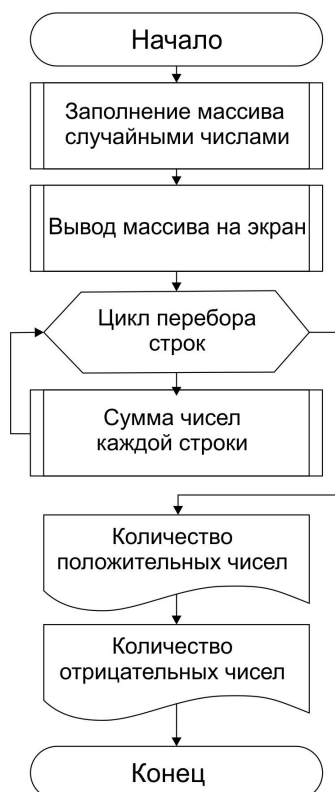
Разработать функцию поиска заданной фамилии в массиве, возвращающей количество таких фамилий в массиве или значение «0», если такой фамилии в массиве нет.

Разработать функцию вывода на экран всех фамилий из массива, начинающихся на заданный символ. Использовать *перегрузку* функций.

.. Разработать функцию, возвращающую весь список в одну строку с одним пробелом между фамилиями.

Разработать функцию сортировки списка фамилий по алфавиту – по возрастанию (см. алгоритм). Функция работает в фоновом режиме (без вывода результата).

Составить программу демонстрации разработанного функционала (функция main) с помощью «меню» (*выбор пункта из меню – отдельная функция, возвращающая номер пункта).



Задание 2:

1. Разработать функцию заполнения двумерного массива случайными числами (реальный массив должен состоять из 5 строк и 6 столбцов).

2. Разработать функцию вывода «прямоугольного массива» на экран.

3. Разработать функцию, возвращающую сумму чисел указанной строки.

13. Разработать функцию, возвращающую количество положительных нулевых или отрицательных чисел в массиве.
14. Составить программу анализа числового массива по заданному алгоритму (функция `main`)
15. *Разработать функцию, сортирующую одномерный массив по возрастанию или по убыванию (в зависимости от заданного направления сортировки), используя алгоритм последовательной сортировки (или любой другой).
16. *Используя функцию сортировки, отсортировать сроки двумерного массива.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме, обучающийся хорошо ориентируется в технологии разработки функций пользователя, программа логично и корректно демонстрирует работу всех функций;
- оценка «хорошо» выставляется обучающемуся, если он выполнил задание почти в полном объеме (может отсутствовать пункт со звездочкой, часть перегрузок), допускает небольшие «проектно-технологические» ошибки в разработке функций;
- оценка «удовлетворительно» выставляется обучающемуся, если он выполнил первые 8 пунктов задания;
- оценка «неудовлетворительно» выставляется обучающемуся, если он выполнил менее 8 пунктов задания.

Практическое задание по теме «Разработка структур данных»

Задание:

1. Разработать тип структуры FIO, для хранения фамилии, имени и отчества абитуриентов;
2. Разработать тип структуры EGE для хранения баллов ЕГЭ (математика, русский, профиль) абитуриента;
3. Используя разработанные ранее структуры, разработать тип структуры Abiturient для сбора информации об абитуриенте: присвоенный код; фамилия, имя, отчество; дата рождения; направление подготовки (основное); направление подготовки (резервное); баллы ЕГЭ (математика, русский, профиль).
4. Разработать функцию заполнения структур вводимыми с клавиатуры значениями.
5. Разработать функцию вывода полей структур на экран.
6. Разработать функцию сохранения (добавления) данных по абитуриенту в текстовом файле Abitur_2020.txt. Одна запись должна занимать одну строку, разделитель полей задать самостоятельно.
7. Разработать функцию чтения одной записи из файла и размещение ее в структуре.
8. Разработать функцию чтения всех записей из файла и сохранение их в массиве записей (массив структур).
9. *Разработать функцию, возвращающую количество абитуриентов, подавших заявление на заданное направление подготовки.
10. Разработать функцию, выводящую всю информацию по абитуриентам, подавшим заявление на заданное направление подготовки с одновременным расчетом среднего балла ЕГЭ.
11. Составить программу демонстрации разработанного функционала (функция `main`) с помощью «меню»* (выбор пункта из меню – отдельная функция, возвращающая номер пункта).

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание выполнены в полном объеме, включая задания со «*», программа работает без ошибок, обучающийся хорошо ориентируется в технологии разработки структуры и работы с файлами;
- оценка «хорошо» выставляется обучающемуся, если он выполнил задание почти в полном объеме (может отсутствовать пункты со звездочкой), программы работают без ошибок;
- оценка «удовлетворительно» выставляется обучающемуся, если он выполнил первые 7-8 заданий, соблюдая технологию работы;
- оценка «неудовлетворительно» выставляется обучающемуся, если он выполнил менее 7 заданий.

Практическое задание по теме «Разработка интерфейса приложения»

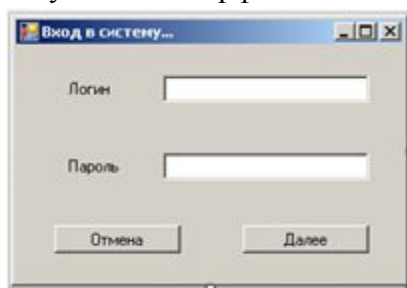
Задание: Необходимо разработать ИС для транспортной компании, осуществляющей пассажирские перевозки. Компания обладает следующими транспортными средствами: легковые автомобили, пассажирские газели, миниавтобусы. Компания работает с наличной и безналичной оплатой, с юридическими и физическими лицами.

Функционал ИС:

1. Регистрация новых заявок на перевозку с сохранением заявок в файл.
2. Редактирование и отслеживание заявок.
3. Построение аналитических отчетов.

Технология работы:

1. В Блокноте создать файл «input_system.txt» и разместить в нем три учетные записи по формату «логин+пароль+ФИО». Сохранить в Вашей папке.
2. Открыть визуальное приложение.
3. На первой форме сформировать следующий интерфейс окна:



4. Настроить tab-очередь: TextBox1 – TextBox2 – Button2 – Button1
5. Добавить в проект новую форму для регистрации заявки на перевозку. Продумать интерфейс окна для ввода следующей информации:
 - данные по заказчику (ФИО/Название, адрес, контактный телефон, паспортные данные/Реквизиты);
 - данные по заказу (вид транспорта, количество человек (и количество детей среди них), наличие груза и его масса и размер, место прибытия, дата и время отправления, место отправления);
 - ориентировочное расстояние и сумма, ответственное лицо за перевозку (водитель).
6. Разместить необходимые компоненты на форме, всем компонентам присвоить идентификаторы, соответствующие вводимой информации, и настроить логические переходы по компонентам (tab-очередь).
7. Перейти на первую форму, создать обработчик события «Нажатие кнопки «Далее», написать код по переходу ко второй форме. Проверить работу события и tab-очереди на второй форме.

Разработка интерфейса форм для решения задач «Редактирование и отслеживание заявок», «Построение аналитических отчетов» реализуется в часы, выделенные на самостоятельную работу обучающихся.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме, программа работает без ошибок, обучающийся хорошо ориентируется в технологии построения визуального интерфейса, а разработанные формы отвечают современным требованиям эргономики;
- оценка «хорошо» выставляется обучающемуся, если он выполнил задание в полном объеме, компоненты не достаточно полно настроены и/или не настроены tab-очереди; созданный интерфейс не в полной мере отвечает современным требованиям эргономики;
- оценка «удовлетворительно» выставляется обучающемуся, если он выполнил работу не в полном объеме, но не менее 60%; компоненты настроены частично; интерфейс не в полной мере отвечает современным требованиям эргономики;
- оценка «неудовлетворительно» выставляется обучающемуся, если он выполнил менее 60% заданий.

Практическое задание по теме «Построение событийной модели приложения»

Задание: для предыдущего задания разработать все необходимые обработчики событий для реализации прикладной задачи и корректного функционирования интерфейса: нажатие клавиш [Enter] (проверка заполненного текстового поля и переход к следующему компоненту) и [Esc] (очистка текстового поля) в текстовых окнах, щелчок по кнопкам «Отмена» (очистка текстовых полей, переход к первому заполняемому полю) и «Далее» (проверка правильности введенного логина и пароля, переход к следующей форме с переносом данных пользователя). При отработке событий необходимо максимально учитывать ошибки ввода, отсутствие файлов и т.д.

Технология работы:

1. Для первой формы входа в ИС событийная модель может быть представлена следующими событиями:

№	Компонент	Возможный идентификатор	Событие	Действие (реакция)
	TextBox1	txtLogin	KeyPress	[Enter] – переход к следующему компоненту ввода (по очереди) [Esc] – очистка текущего компонента
	TextBox2	txtPass	KeyPress	[Enter] – переход к следующему компоненту (по очереди) [Esc] – очистка текущего компонента
	Button1	btnEsc	Click	Выход из приложения с подтверждением
	Button2	btnNext	Click	1. Проверка корректности ввода данных пользователя 2. В случае корректности данных вывод сообщения «Добро пожаловать, user!» и загрузка второй пустой формы 3. В случае некорректности данных вывод сообщения об этом, очистка компонентов ввода, передача управления (курсора) первому компоненту ввода

2. Разработать вспомогательную функцию приведения логина и пароля к корректному виду, которая убирает незначимые пробелы до и после введенных данных. Использовать данную функцию в любом «месте» событийной модели (можно при обработке события Click, можно при обработке события KeyPress). Проверить работу формы.
 3. Сформировать счетчик попыток ввода данных. Количество реализованных (или оставшихся) попыток выводить на форму. Обработать событие Change, связанное с количеством попыток, для выгрузки приложения при их завершении с информированием пользователя об этом.
 4. *Предложите и обработайте хотя бы еще одно событие (можно добавить компонент) для более комфортной работы интерфейса.
 5. Для второй формы самостоятельно разработать все необходимые обработчики событий для реализации прикладной задачи и корректного функционирования интерфейса.
- Работа может быть доделана в часы, выделенные на самостоятельную работу обучающихся.

Критерии оценивания:

- оценка «отлично» выставляется обучающемуся, если задание выполнено в полном объеме, включая задания со звездочкой, программа работает без ошибок, обучающийся хорошо ориентируется в технологии построения событийной модели, разработанная модель реализует прикладную задачу и эргономику интерфейса;
- оценка «хорошо» выставляется обучающемуся, если задание выполнено в полном объеме, но не выполнены задания со звездочкой, программа работает без ошибок, обучающийся хорошо ориентируется в технологии построения событийной модели, разработанная модель реализует прикладную задачу, но не достаточно «сервисности» в ее работе, эргономика интерфейса реализована частично;
- оценка «удовлетворительно» выставляется обучающемуся, если он выполнил работу не в полном объеме, но не менее 60%; событийная модель реализована частично; интерфейс не в полной мере отвечает современным требованиям эргономики;
- оценка «неудовлетворительно» выставляется обучающемуся, если он выполнил менее 60% заданий.

Командная разработка «Сбор данных»

к теме «Разработка и перегрузка функций»

Задача: разработать и защитить программу для ввода и подтверждения данных клиента компании.

Формат работы - малая проектная группа (4 человека)

ИТ-менеджер – постановка задачи, распределение работ, контроль выполненных работ, итоговая сборка программы.

ИТ-аналитик (проектировщик) – определение набора функций, входных и выходных данных, разработка алгоритма контроля и тестирования, отдельных фрагментов программы.

Кодировщик – разработка и отладка функций;

Тестер – проверка работоспособности всех функций по заданному алгоритму.

Данные клиента: Фамилия, имя, отчество, дата рождения, пол, семейное положение (женат/холост), фамилия до брака, количество детей, образование (высшее/среднее профессиональное/среднее), место проживания, место работы (название компании, адрес), занимаемая должность, средний уровень дохода за последние 3 месяца, данные паспорта (серия, номер, дата), прописка, СНИЛС (номер), ИНН (номер), наличие иждивенцев и их количество.

Рассчитываемые данные: возраст на текущий момент (полных лет), количество несовершеннолетних детей, «подушевой» доход семьи (включая клиента).

Метаданные: дата и время ввода данных, Фамилия И.О. оператора, факт подтверждения клиентом.

Технология разработки:

- данные клиента вводятся с клавиатуры и сопровождаются понятными подсказками;
- вводимые с клавиатуры данные подлежат контролю на корректность (средства контроля продумать самостоятельно);
- при отсутствии каких-либо данных предусмотреть «замещающее» значение;
- после ввода и расчетов все данные выводятся на экран для контроля клиентом (продумать шаблон вывода данных в виде отдельной универсальной функции);
- предусмотреть необходимость повторного ввода (при выявлении ошибок клиентом), но не всех данных, а только тех, которые введены неправильно;

Требования к разработке:

- весь код представлен функциями, в Main – вызов укрупненных функций (сборщиков мелких задач);
- максимальное разделение визуальных и не визуальных действий (в расчетных и контролирующих функциях не должно быть никаких вводов/выводов);
- объем функций – оптимально-минимальный для сохранения универсальности;
- комментарии к каждой функции (назначение, описание входных/выходных данных, ФИО разработчика).

Отчетность группы:

- код программы (program.cs);
- отчет каждого разработчика – таблица с указанием названий функций, количества операторов в функции, возможности повторного применения в другом проекте (универсальность), самостоятельности разработки по 10-балльной шкале;
- отчет по группе ИТ-менеджера – таблица с указанием ФИО разработчика, количество разработанных функций, оценка качества разработки по 10-балльной шкале.

Защита разработки:

- ИТ-менеджер описывает общую концепцию и представляет работу программы;
- все остальные участники перечисляют выполненные задачи и отвечают на вопросы преподавателя и других присутствующих.

Командная разработка «Тестовая система»

к темам «Разработка структур данных», «Работа с файлами»

Задача: разработать консольную программу для прохождения тестирования по выбранному предмету.

Формат работы - малая проектная группа (4 человека)

ИТ-менеджер – постановка задачи, распределение работ, контроль выполненных работ, итоговая сборка программы.

ИТ-аналитик (проектировщик) – определение набора функций, входных и выходных данных, разработка алгоритма контроля и тестирования, отдельных фрагментов программы.

Кодировщик – разработка и отладка функций;

Тестер – проверка работоспособности всех функций по заданному алгоритму.

Требования к тесту:

- один тест проверяет знания по одному предмету (теме предмета);
- вопросы в тесте могут быть следующих видов «один из многих», «несколько из многих», «открытый ответ»;

- по тесту подсчитывается количество правильных ответов и итоговая оценка в процентном виде (все правильные ответы – 100%);
- тесты хранятся в файлах, название файла отражает предмет (тему);
- один вопрос и ответы на него занимают в файле одну строку;
- результаты тестирования (ФИО, дата и время прохождения теста, название теста, полученный результат) записываются в файл результатов, название которого фиксировано;

Технология разработки:

- данные тестируемого вводятся с клавиатуры и сопровождаются понятными подсказками;
- для доступа к тесту тестируемому выдается временный пароль, который можно не хранить;
- вводимые с клавиатуры данные подлежат контролю на корректность (средства контроля продумать самостоятельно);
- название теста, текущая дата и время, номер вопроса, ФИО сдающего всегда находятся на экране;
- вопросы теста выводятся в том порядке, как они записаны в файле;
- предусмотреть возможность досрочного завершения теста с соответствующей записью в файле результатов;

Требования к разработке:

- весь код представлен функциями, в Main – вызов укрупненных функций (сборщиков мелких задач);
- максимальное использование структурированных типов (строк, массивов, структур);
- максимальное разделение визуальных и не визуальных действий (в расчетных и контролирующих функциях не должно быть никаких вводов/выводов);
- объем функций – оптимально-минимальный для сохранения универсальности;
- комментарии к каждой функции (назначение, описание входных/выходных данных, ФИО разработчика).

Отчетность группы:

- код программы (program.cs);
- отчет каждого разработчика – таблица с указанием названий функций, количества операторов в функции, возможности повторного применения в другом проекте (универсальность), самостоятельности разработки по 10-балльной шкале;
- отчет по группе ИТ-менеджера – таблица с указанием ФИО разработчика, количество разработанных функций, оценка качества разработки по 10-балльной шкале.

Защита разработки:

- ИТ-менеджер описывает общую концепцию и представляет работу программы;
- все остальные участники перечисляют выполненные задачи и отвечают на вопросы преподавателя и других присутствующих.

Критерии оценивания командной разработки:

Первоначальная оценка работы группы определяется качеством разработки и работы всей программы:

- оценка «отлично» выставляется группе, если программа выполнена в полном объеме, работает без ошибок и удовлетворяет всем требованиям, обучающиеся демонстрируют способность разрабатывать универсальные и прикладные функции, использовать разные типы данных, полно отвечают на все вопросы во время защиты;
- оценка «хорошо» выставляется группе, если программа выполнена почти в полном объеме (не менее 80%) и удовлетворяет всем требованиям, но есть незначительные ошибки и/или

отклонения от требований, обучающиеся демонстрируют способность разрабатывать универсальные и прикладные функции, использовать разные типы данных, отвечают на вопросы во время защиты;

- оценка «удовлетворительно» выставляется группе, если в программе не все выполнено (не менее 60%), есть ошибки и/или значительные отклонения от требований, обучающиеся слабо демонстрируют способность разрабатывать универсальные и прикладные функции, использовать разные типы данных, не могут ответить на все вопросы во время защиты;
- оценка «неудовлетворительно» выставляется группе, если программа выполнена менее 60%, во время защиты обучающиеся не могут пояснить функционал программы и ее код, не отвечают на вопросы аудитории.

Оценка может понижаться:

ИТ-менеджеру – за некорректное распределение работ, за недостоверные данные в отчетах.

ИТ-аналитику (проектировщику) – за недостаточную универсальность функций, за отсутствие достаточных комментариев.

Кодировщику – за низкую эффективность кода;

Тестеру – за ошибки в работе функций.

Подготовка и выступление с эссе по командной разработке

Задача: представить кратко теорию командной разработки программных продуктов, оценить корректность организации и эффективность выполнения работ в конкретной группе, выделить положительные и негативные моменты.

Требования к эссе:

- 1) Устное выступление не более 5 минут;
- 2) Теоретическая часть может содержать углубленное представление одного из следующих вопросов:
 - a) Состав современных проектных команд по разработке программных продуктов
 - b) Принципы и способы формирования проектных групп
 - c) Современные технологии командной разработки программных продуктов
 - d) Методологии управления командными работами
 - e) Организация профессиональных коммуникаций в команде
 - f) Факторы эффективности и качество командных разработок
 - g) Влияние корпоративной культуры на эффективность и качество работы в команде
 - h) Средства эффективного взаимодействия в проектных командах
- 3) Печатный объем представляемого материала 5-8 страниц, оформленных по требованиям Академии.

Критерии оценивания эссе:

- оценка «отлично» выставляется обучающемуся, если эссе полностью раскрывает теоретическую часть задания, обучающийся хорошо ориентируется в материале, приведены оценочные суждения о работе проектной команды и обоснования сделанным выводам, обучающийся отвечает на дополнительные вопросы, работа оформлена в соответствии с требованиями;
- оценка «хорошо» выставляется обучающемуся, если эссе в целом раскрывает теоретический вопрос, но остаются не освещенные некоторые моменты, не на все дополнительные вопросы обучающийся может дать ответ; приведены оценочные суждения о работе проектной команды, но не всегда обоснованы сделанные выводы, работа оформлена в соответствии с требованиями;

- оценка «удовлетворительно» выставляется обучающемуся, если его эссе не раскрывает тему в достаточной мере, обучающийся с трудом отвечает на дополнительные вопросы; оценочные суждения о работе проектной команды отсутствуют или сделаны очень скупо, без достаточных обоснований, работа оформлена с нарушением требований;
- оценка «неудовлетворительно» выставляется обучающемуся, если эссе отсутствует или эссе совсем не раскрывает теоретический вопрос, обучающийся не отвечает на дополнительные вопросы; оценочные суждения о работе проектной команды отсутствуют, работа оформлена с нарушением требований.

Индивидуальная программная разработка

к теме «Разработка многодокументных приложений»

Задача: разработать приложение для реализации задачи выданного варианта (возможен собственный вариант).

Требования к программной разработке:

- приложение состоит из нескольких визуальных форм: Заставка, О программе, не менее двух форм для решения прикладной задачи;
- интерфейс окон разрабатывается с учетом требований эргономики;
- событийная модель разрабатывается с учетом «дублирования» действия пользователя;
- на главной форме реализовать основные интерфейсные элементы: главной и контекстное меню, панель инструментов, информационную (статусную) строку;
- между окнами должна быть реализована передача данных;
- данные и/или результаты работы программы «отчуждать» от программы любым известным способом (файлы, база данных);

Сдача работы:

- производится на последнем занятии по дисциплине или в часы КСР;
- работы приложения демонстрируется преподавателю;
- при необходимости преподаватель может задать уточняющие вопросы по разработке.

Варианты примерных заданий для индивидуальной разработки

1. Демонстрация возможностей шрифтов (на основе фрагмента текста)
2. Обработка пароля (ввод, изменение, контроль)
3. Тестирование знаний по курсу ... (ООП, С#, мультипрограммирование и т.д.)
4. Проверка знаний таблицы умножения
5. Расчет стоимости услуг ... (прокат дисков, химчистка и т.д.)
6. Формирование списка группы ... (учащихся, туристов, пациентов и т.д.)
7. Анкетирование соискателя вакантного рабочего места
8. Психологическое тестирование
9. Калькулятор элементарных операций
10. Обработчик строк (не менее 6 различных действий над строкой)
11. Модуль ввода данных о ... (поступающем товаре, грузе и т.д.)
12. Расчет функции $y=a*xn+b$ на заданном интервале
13. Будильник текущего дня (не менее 10 событий)
14. Формирование справки об ... (обучении, лечении, отпуске и т.д.)
15. Формирование приглашения на ... (мероприятие, обучение и т.д.)
16. Модуль ввода данных об элементе домашней библиотеки (книга, журнал, диски и т.д.)

17. Преобразование данных печатного издания в элемент библиографии (по заданному приближенному шаблону)
18. Расчет площадей и периметров выбранной геометрической фигуры

Критерии оценивания индивидуальной разработки:

Первоначальная оценка работы группы определяется качеством разработки и работы всей программы:

- оценка *«отлично»* выставляется обучающемуся, если программа выполнена в полном объеме, работает без ошибок и удовлетворяет всем требованиям; обучающийся демонстрирует способность корректно разрабатывать интерфейс экранных форм и событийную модель работы приложения, умеет передавать данные между окнами и хранить данные и/или результаты работы за пределами приложения, полно отвечает на уточняющие вопросы во время демонстрации работы программы и хорошо ориентируется как в самом приложении, так и в теоретическом материале, что свидетельствует о самостоятельности разработки;
- оценка *«хорошо»* выставляется обучающемуся, если программа выполнена почти в полном объеме (не менее 80%) и удовлетворяет всем требованиям, но есть незначительные ошибки и/или отклонения от требований; обучающийся умеет передавать данные между окнами и хранить данные и/или результаты работы за пределами приложения, отвечает на уточняющие вопросы во время демонстрации работы программы, но не всегда дает развернутые ответы, хорошо ориентируется в программном коде, но не достаточно хорошо в теоретическом материале, что свидетельствует о самостоятельности практической разработки и о наличии небольших пробелов в знаниях;
- оценка *«удовлетворительно»* выставляется обучающемуся, если в программе не все выполнено (не менее 60%), есть ошибки и/или значительные отклонения от требований, обучающийся не достаточно хорошо ориентируется в программном коде и теоретическом материале, слабо отвечает на уточняющие вопросы, что свидетельствует о значительных пробелах в знаниях и о ставит под сомнение самостоятельность программной разработки;
- оценка *«неудовлетворительно»* выставляется обучающемуся, если разработка не выполнена или на выполнена менее, чем на 60%, не соответствует требованиям, во время демонстрации обучающийся не ориентируется в коде и не может доказать самостоятельность разработки, не отвечает на уточняющие вопросы преподавателя.

3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

ОЦЕНКА ЗНАНИЙ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ – 1 СЕМЕСТР

Список вопросов для подготовки к экзамену

Компетенция ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

Индикатор ОПК-7.1. Составляет алгоритмы для решения стандартных профессиональных задач
Обучающийся знает: основные понятия теории алгоритмов (алгоритм, исполнитель, виды и типы алгоритмов), правила описания алгоритмов.

1. Алгоритмизация: особенности и виды алгоритмических языков программирования. Понятие и свойства алгоритмов.

2. Алгоритмизация: ГОСТы на разработку алгоритмов. Способы и правила описания алгоритмов.

3. Алгоритмизация: базовые алгоритмические структуры.

Компетенция ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

Индикатор ОПК-7.2. Разрабатывает программы, производит их отладку и проверку работоспособности

Обучающийся знает: этапы проектирования программ в соответствии с жизненным циклом информационных систем; существующие технологии и средства программирования, синтаксис высокоуровневого языка C#, структуру программы, операторы и управляющие конструкции, типовую структуру данных.

1. Жизненный цикл разработки программного продукта: понятие, этапы, характеристика.
2. Основные этапы разработки программ.
3. Понятие программной модели, ее виды (блочная, модульная) и их особенности.
4. Модель программы на языке C#, программные единицы, функция main и ее особенности, порядок выполнения C# программы.
5. Глобальные и локальные данные программы, их область видимости и время жизни. Особенности глобальных и локальных данных в C# программах.
6. Сегментированная модель программы. Типы-значения и типы-ссылки.
7. Технологии программирования (модульная, объектная, визуальная).
8. Платформа .NET Framework, ее состав, требования к архитектуре ПК, установка платформы .NET Framework и Visual Studio на компьютер.
9. Платформа .NET Framework, используемые языки программирования и общезыковая исполняющая среда, общий промежуточный язык, новый тип программ – сборки.
10. Основные понятия языка: алфавит языка, имена и идентификаторы (служебные и пользовательские), описания (объявления), операторы.
11. Концепция данных: определение данных, константы и переменные, концепция типов, объединение данных в структуры.
12. Структура программы. Правила записи текста программы. Комментарии и их использование. Ввод и вывод данных.
13. Операторы (операции) языка, приоритет их выполнения, правила построения выражений.
14. Простые стандартные типы данных: целочисленный и вещественный, символьный (литерный) и логические типы. Основные операции над ними.
15. Операторы языка: оператор присваивания, эффективные формы оператора присваивания, явное и неявное преобразование типов.
16. Операторы языка: условный оператор, оператор-переключатель.
17. Операторы цикла.
18. Разработка функций пользователя. Перегрузка функций.

Компетенция ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.

Индикатор ОПК-8.1. Планирует работы проекта в соответствии с полученным заданием на разных стадиях жизненного цикла информационных систем, в том числе используя облачные сервисы и приложения для управления проектами в сфере ИТ

Обучающийся знает: состав команды разработчиков программного проекта, принципы и способы формирования проектной группы.

1. Командный характер современных программных разработок и состав команды.
2. Принципы и способы формирования проектной группы.

3. Особенности жизненного цикла ИТ-проекта по разработке программного продукта, его структуризация.
4. Функциональные обязанности ИТ-менеджера и системного аналитика.
5. Функциональные обязанности кодировщика и тестера.

Компетенция ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.

Индикатор ОПК-8.2. Осуществляет мониторинг и оценку выполнения работ по проекту, в том числе используя облачные сервисы и приложения для управления командами разработчиков

Обучающийся знает: технологические средства управления работой команды.

1. Способы и средства управления командной разработчиков.
2. Специфика управления виртуальной командой разработчиков.

Компетенция ОПК-9. Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп

Индикатор ОПК-9.1. Осуществляет профессиональные коммуникации с участниками проектных групп, в том числе с применением средств видеоконференцсвязи, облачных сервисов и приложений для управления проектными командами

Обучающийся знает: особенности и средства взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения.

1. Способы и средства взаимодействия с командной разработчиков.
2. Специфика коммуникационного (удаленного) взаимодействия с командой разработчиков.

Компетенция ОПК-9. Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп

Индикатор ОПК-9.2. Осуществляет профессиональные коммуникации с заказчиком проектной деятельности, в том числе с применением средств удаленного взаимодействия и видеоконференцсвязи

Обучающийся знает: способы демонстрации результатов программной разработки.

1. Заинтересованные стороны в разработке программного продукта.
2. Способы и правила взаимодействия с заказчиком программного продукта.
3. Способы демонстрации результатов программной разработки, в том числе коллективного характера и с применением средств удаленного взаимодействия.

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ – 1 СЕМЕСТР

Типовые задания для подготовки обучающихся к экзамену

Компетенция ОПК-7. - Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Индикатор ОПК-7.1. - Составляет алгоритмы для решения стандартных профессиональных задач.

Обучающийся умеет: формализовано описывать поставленные задачи; разрабатывать оптимальные алгоритмы для решения поставленных задач.

Обучающийся владеет: навыками оформления алгоритмов в соответствии с требованиями ГОСТ.

Индикатор ОПК-7.2. - Разрабатывает программы, производит их отладку и проверку работоспособности.

Обучающийся умеет: определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм (обработчиков событий); разрабатывать программы для решения учебных задач из различных предметных областей.

Обучающийся владеет: навыками выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам разработки алгоритма для решения поставленной задачи и консольной программы по нему, в которой обучающемуся необходимо адекватно выбрать типы данных и определить количество разрабатываемых функций.

Задание 1. Разработать алгоритм и консольную программу для тестирования знаний обучающихся по дисциплине «Информатика». В тест включить 5 вопросов. Реализовать проверки корректности ввода. После теста выводить: ФИО тестируемого и количество набранных баллов. Предусмотреть возможность повторного прохождения теста. В программе обязательно наличие нескольких функций с передачей данных между ними и обоснованный выбор типов данных.

Задание 2. Разработать алгоритм и консольную программу для анкетирования покупателя о качестве обслуживания. Результаты сохранять в массив. По результатам определять общую оценку покупателя: негативная - если «отрицательные» ответы превышают «положительные», позитивная - в обратном случае. Предусмотреть возможность повторного анкетирования. В программе обязательно наличие нескольких функций с передачей данных между ними и обоснованный выбор типов данных.

Задание 3. Разработать алгоритм и консольную программу для формирования маршрута экскурсии пользователя: для женщин предусмотреть возможность побывать на выставке цветов или балльных платьев, для мужчин – на экспозиции автомобилей или спортивных достижений; для несовершеннолетних – посещение кинозала, для совершеннолетних – ресторана. Выбор осуществлять случайным образом. В программе обязательно наличие нескольких функций с передачей данных между ними и обоснованный выбор типов данных.

Компетенция ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.

Индикатор ОПК-8.1. Планирует работы проекта в соответствии с полученным заданием на разных стадиях жизненного цикла информационных систем, в том числе используя облачные сервисы и приложения для управления проектами в сфере ИТ.

Обучающийся умеет: распределять работу в группе с учетом профессиональных компетенций отдельных участников.

Обучающийся владеет: навыками работы в проектных группах по разработке программного обеспечения, в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.).

Индикатор ОПК-8.2. Осуществляет мониторинг и оценку выполнения работ по проекту, в том числе используя облачные сервисы и приложения для управления командами разработчиков.

Обучающийся умеет: подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников.

Обучающийся владеет: навыками оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной деятельности (сервисы Yandex, Google и др.).

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в командной разработке «Сбор данных», по результатам защиты выполненной разработки.

Компетенция ОПК-9. Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп;

Индикатор ОПК-9.1. Осуществляет профессиональные коммуникации с участниками проектных групп, в том числе с применением средств видеоконференцсвязи, облачных сервисов и приложений для управления проектными командами.

Обучающийся умеет: взаимодействовать внутри проектной группы и с ее окружением во время разработки программного проекта, в том числе с применением средств удаленного взаимодействия.

Обучающийся владеет: навыками проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.).

Индикатор ОПК-9.2. Осуществляет профессиональные коммуникации с заказчиком проектной деятельности, в том числе с применением средств удаленного взаимодействия и видеоконференцсвязи.

Обучающийся умеет: представлять программную разработку потенциальному Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.).

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в командной разработке «Сбор данных», по результатам защиты выполненной разработки.

ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА – 1 СЕМЕСТР

Экзамнационный билет для проверки знаний в первом семестре состоит из теоретического вопроса и практического задания.

Тольяттинская академия управления	Дисциплина Алгоритмизация и программирование
Билет	
1.	Поточный ввод/вывод. Виды файлов. Чтение и запись данных в файл.
2.	Практическое задание Разработать алгоритм и консольную программу для формирования маршрута экскурсии пользователя: для женщин предусмотреть возможность побывать на выставке цветов или бальных платьев, для мужчин – на экспозиции автомобилей или спортивных достижений; для несовершеннолетних – посещение кинозала, для совершеннолетних – ресторана. Выбор осуществлять случайным образом. В программе обязательно наличие нескольких функций с передачей данных между ними и обоснованный выбор типов данных.

ОЦЕНКА ЗНАНИЙ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ – 2 СЕМЕСТР

Список вопросов для подготовки к экзамену

Компетенция ОПК-7. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

Индикатор ОПК-7.1. Составляет алгоритмы для решения стандартных профессиональных задач
Обучающийся знает: основные понятия теории алгоритмов (алгоритм, исполнитель, виды и типы алгоритмов), правила описания алгоритмов.

Индикатор ОПК-7.2. Разрабатывает программы, производит их отладку и проверку работоспособности

Обучающийся знает: этапы проектирования программ в соответствии с жизненным циклом информационных систем; существующие технологии и средства программирования, синтаксис высокоуровневого языка C#, структуру программы, операторы и управляющие конструкции, типовую структуру данных.

1. Сложные типы данных: строка (понятие, объявление, обработка, передача между функциями, библиотечные функции обработки строк).
2. Сложные типы данных: одномерные массивы (понятие, характеристики, типы индексов, функциональное назначение, представление в памяти). Основные операции над массивами. Передача массивов между функциями.
3. Двумерные массивы (типы индексов, функциональное назначение, представление в памяти, передача массивов между функциями).
4. Пользовательский тип – структура: понятие, описание типа, объявление переменной, передача между функциями, обработка.
5. Поточный ввод/вывод. Виды файлов. Чтение и запись данных в файл.
6. Работа с файлами и каталогами.
7. Визуальное программирование: отличие работы Windows-приложений от работы DOS-программ,
8. Назначение визуальной технологии, сопутствующие технологии (объектная, двунаправленной разработки), принципы разработки визуальных приложений
9. Объект Application и его роль в Windows-приложении.
10. Понятие проекта, файловый и компонентный состав проекта, событийная модель работы приложения.
11. Организация проекта в Visual C#, процесс создания проекта, порядок разработки визуального приложения.
12. Понятие события и событийной модели работы приложения, виды событий (пользовательские, системные, программные и т.д.), схема обработки события,
13. Понятие многодокументного приложения, SDI и MDI-приложения.
14. Способы организации заставки к приложению.
15. Принципы разработки пользовательского интерфейса
16. Библиотека компонентов: понятие компонента, классификация компонентов (визуальные и не визуальные; стандартные, дополнительные, системные и т.д.);
17. Свойства компонентов: виды свойств (этапа проектирования, динамические, только для чтения); общие свойства и методы визуальных компонентов
18. Программное обращение к свойствам и методам компонентов (привести примеры).
19. Компоненты ввода и вывода данных в визуальное приложение. Проверка корректности ввода данных.
20. Характеристика объекта TForm и его основных свойств.
21. Описание компонентов-контейнеров.

22. Разработка отклика на событие, вызов обработчиков событий, построение ссылок на существующие обработчики событий.
23. Организация взаимодействия форм и передачи данных между ними.
24. Многодокументные приложения: виды окон приложения и их общие свойства.
25. Понятие диалогового окна, его специфические особенности, создание диалоговых окон приложения.
26. Системные диалоговые окна и порядок работы с ними.
27. Работа с каталогами и файлами с помощью системных диалоговых окон.
28. Организация печати с помощью системного диалогового окна.
29. Работа с буфером обмена в визуальном приложении.
30. Использование в приложениях панелей сообщений.
31. Разработка основного и контекстного меню.
32. Разработка инструментальных и информационных панелей.
33. Понятие исключения и обработка исключительных ситуаций.

Компетенция ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.

Индикатор ОПК-8.1. Планирует работы проекта в соответствии с полученным заданием на разных стадиях жизненного цикла информационных систем, в том числе используя облачные сервисы и приложения для управления проектами в сфере ИТ

Обучающийся знает: состав команды разработчиков программного проекта, принципы и способы формирования проектной группы.

Индикатор ОПК-8.2. Осуществляет мониторинг и оценку выполнения работ по проекту, в том числе используя облачные сервисы и приложения для управления командами разработчиков.

Обучающийся знает: технологические средства управления работой команды.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам представления эссе на защите командной разработки по теме «Тестовая система»

Компетенция ОПК-9. Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп

Индикатор ОПК-9.1. Осуществляет профессиональные коммуникации с участниками проектных групп, в том числе с применением средств видеоконференцсвязи, облачных сервисов и приложений для управления проектными командами

Обучающийся знает: особенности и средства взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения.

Индикатор ОПК-9.2. Осуществляет профессиональные коммуникации с заказчиком проектной деятельности, в том числе с применением средств удаленного взаимодействия и видеоконференцсвязи

Обучающийся знает: способы демонстрации результатов программной разработки.

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам представления эссе на защите командной разработки по теме «Тестовая система»

ОЦЕНКА УМЕНИЙ И НАВЫКОВ ОБУЧАЮЩИХСЯ НА ЭКЗАМЕНЕ – 2 СЕМЕСТР

Типовые задания для подготовки обучающихся к экзамену

Компетенция ОПК-7. - Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Индикатор ОПК-7.1. - Составляет алгоритмы для решения стандартных профессиональных задач.

Обучающийся умеет: формализовано описывать поставленные задачи; разрабатывать оптимальные алгоритмы для решения поставленных задач.

Обучающийся владеет: навыками оформления алгоритмов в соответствии с требованиями ГОСТ.

Индикатор ОПК-7.2. - Разрабатывает программы, производит их отладку и проверку работоспособности.

Обучающийся умеет: определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм (обработчиков событий); разрабатывать программы для решения учебных задач из различных предметных областей.

Обучающийся владеет: навыками выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений.

Задание 1. Спроектировать интерфейс и событийную модель приложения для вывода должников библиотеки, если их фамилии и срок сдачи книг записаны в отдельный файл. Задолженность определять сравнением текущей даты с датой сдачи книг.

Задание 2. Спроектировать интерфейс и событийную модель приложения для вывода фотографии сотрудника компании на отдельной форме, если фотографии хранятся в отдельном каталоге и названия файлов соответствуют фамилии сотрудника.

Задание 3. Спроектировать интерфейс и событийную модель приложения для тестирования знаний обучающихся по дисциплине «Информатика». В тесте должно быть не менее 3-х вопросов. Вопросы и результаты тестирования должны отражаться на разных формах. ФИО тестируемого и результат тестирования должен храниться в текстовом файле.

Задание 4. Спроектировать интерфейс и событийную модель приложения для анкетирования покупателя. Анкета ориентирована на вопросы о качестве обслуживания. Результаты вопросов должны храниться в БД.

Компетенция ОПК-8. Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла.

Индикатор ОПК-8.1. Планирует работы проекта в соответствии с полученным заданием на разных стадиях жизненного цикла информационных систем, в том числе используя облачные сервисы и приложения для управления проектами в сфере ИТ.

Обучающийся умеет: распределять работу в группе с учетом профессиональных компетенций отдельных участников.

Обучающийся владеет: навыками работы в проектных группах по разработке программного обеспечения, в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.).

Индикатор ОПК-8.2. Осуществляет мониторинг и оценку выполнения работ по проекту, в том числе используя облачные сервисы и приложения для управления командами разработчиков.

Обучающийся умеет: подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников.

Обучающийся владеет: навыками оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной деятельности (сервисы Yandex, Google и др.).

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в командной разработке «Тестовая система».

Компетенция ОПК-9. Способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп;

Индикатор ОПК-9.1. Осуществляет профессиональные коммуникации с участниками проектных групп, в том числе с применением средств видеоконференцсвязи, облачных сервисов и приложений для управления проектными командами.

Обучающийся умеет: взаимодействовать внутри проектной группы и ее окружением во время разработки программного проекта, в том числе с применением средств удаленного взаимодействия.

Обучающийся владеет: навыками проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.).

Индикатор ОПК-9.2. Осуществляет профессиональные коммуникации с заказчиком проектной деятельности, в том числе с применением средств удаленного взаимодействия и видеоконференцсвязи.

Обучающийся умеет: представлять программную разработку потенциальному Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.).

Оценка достижения обучающимся запланированного результата обучения осуществляется по результатам его участия в командной разработке «Тестовая система».

ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА – 2 СЕМЕСТР

Экзаменационный билет состоит из теоретического вопроса и практических заданий: первое на проектирование приложения (программы), второе – на программирование спроектированного приложения.

Тольяттинская академия управления	Дисциплина Алгоритмизация и программирование
Билет	
1. Назначение визуальной технологии, сопутствующие технологии (объектная, двунаправленной разработки), принципы разработки визуальных приложений.	
2. Спроектировать интерфейс и событийную модель приложения для тестирования знаний студентов по дисциплине «Информатика». В тесте должно быть не менее 3-х вопросов. Вопросы и результаты тестирования должны отражаться на разных формах. ФИО тестируемого и результат тестирования должны храниться в отдельном текстовом файле.	
3. По спроектированной модели приложения написать программный код для тестирования знаний студентов по дисциплине «Информатика».	

4. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

ПРОЦЕДУРА ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Для контроля усвоения обучающимися данной дисциплины и достижения запланированных результатов обучения учебным планом предусмотрены экзамен по окончании первого семестра обучения и экзамен по окончании второго семестра обучения в форме устного ответа на теоретический вопрос и выполнения практического задания (заданий). Для допуска к сдаче экзамена обучающиеся должны участвовать в мероприятиях текущего контроля: отвечать на вопросы устных опросов, выполнять практические и индивидуальные задания, выполнять проверочные работы, тесты и диктанты, участвовать в индивидуальных и командных разработках и выступать с докладами. Для допуска к экзамену в первом семестре обучающемуся необходимо, пройти тест и написать хотя бы один диктант с результатом не менее 60%, выполнить проверочную работу (тема 5), участвовать в командной разработке и ее защите, выполнить в полном объеме проверяемые практические задания. Для допуска к экзамену во втором семестре обучающемуся необходимо участвовать в командной разработке и ее защите, выступить с эссе и докладом, выполнить в полном объеме проверяемые практические задания (по темам 7, 8, 11, 12) и индивидуальную разработку (по теме 13).

Экзамен проходит в форме устного ответа на теоретический вопрос и выполнения практического задания (заданий) на компьютере с помощью программных средств. Ключевым критерием сдачи экзамена выступает работоспособность написанной программы. Неработающая программа приводит к получению оценки «неудовлетворительно» за экзамен.

КРИТЕРИИ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
Компетенция ОПК-7 - способен разрабатывать алгоритмы и программы, пригодные для практического применения					
Индикатор ОПК-7.1. Составляет алгоритмы для решения стандартных профессиональных задач					
Знать: - основные понятия теории алгоритмов (алгоритм, исполнитель, виды и типы	Не знает	Фрагментарные знания основных понятий теории алгоритмов (алгоритм, исполнитель, виды и типы алгоритмов),	Общие, но не структурированные знания основных понятий теории алгоритмов (алгоритм, исполнитель,	Сформированные, но содержащие отдельные пробелы знания основных понятий	Сформированные систематизированные прочные знания основных понятий

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
алгоритмов), правила описания алгоритмов		правил описания алгоритмов	виды и типы алгоритмов), правил описания алгоритмов	теории алгоритмов (алгоритм, исполнитель, виды и типы алгоритмов), правил описания алгоритмов	теории алгоритмов (алгоритм, исполнитель, виды и типы алгоритмов), правил описания алгоритмов
Уметь: - формализовано описывать поставленные задачи; - разрабатывать оптимальные алгоритмы для решения поставленных задач;	Не умеет	Частично освоенные умения формализовано описывать поставленные задачи; разрабатывать оптимальные алгоритмы для решения поставленных задач;	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполнению умения формализовано описывать поставленные задачи; разрабатывать оптимальные алгоритмы для решения поставленных задач;	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения формализовано описывать поставленные задачи; разрабатывать оптимальные алгоритмы для решения поставленных задач;	Успешно сформированные, самостоятельные и осознанно выполняемые умения формализовано описывать поставленные задачи; разрабатывать оптимальные алгоритмы для решения поставленных задач;
Владеть: - навыками оформления алгоритмов в соответствии с требованиями ГОСТ;	Не владеет	Фрагментарно сформированные навыки оформления алгоритмов в соответствии с требованиями ГОСТ;	В целом сформированные, но выполняемые на элементарном уровне навыки оформления алгоритмов в соответствии с требованиями ГОСТ;	Сформированные, но не устойчивые в сложных ситуациях навыки оформления алгоритмов в соответствии с требованиями ГОСТ;	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки оформления алгоритмов в соответствии с требованиями ГОСТ;
Компетенция ОПК-7 - способен разрабатывать алгоритмы и программы, пригодные для практического применения Индикатор ОПК-7.2. Разрабатывает программы, производит их отладку и проверку работоспособности					
Знать: этапы	Не знает	Фрагментарные знания этапов	Общие, но не структурирова	Сформированные, но	Сформированные

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
проектирование программ в соответствии с жизненным циклом; существующие технологии и средства программирования; синтаксис высокоуровневого языка C#, структуру программы, операторы и управляющие конструкции, типовую структуру данных		проектирование программ в соответствии с жизненным циклом; существующих технологий и средств программирования; синтаксиса высокоуровневого языка C#, структур программы, операторов и управляющих конструкций, типовой структуры данных	нные знания этапов проектирования программ в соответствии с жизненным циклом; существующих технологий и средств программирования; синтаксиса высокоуровневого языка C#, структур программы, операторов и управляющих конструкций, типовой структуры данных	содержащие отдельные пробелы знания этапов проектирования программ в соответствии с жизненным циклом; существующих технологий и средств программирования; синтаксиса высокоуровневого языка C#, структур программы, операторов и управляющих конструкций, типовой структуры данных	систематизированные прочные знания этапов проектирования программ в соответствии с жизненным циклом; существующих технологий и средств программирования; синтаксиса высокоуровневого языка C#, структур программы, операторов и управляющих конструкций, типовой структуры данных
Уметь: определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм (обработчиков событий); разрабатывать программы для решения учебных задач из различных предметных областей	Не умеет	Частично освоенные умения определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм (обработчиков событий); разрабатывать программы для решения учебных задач из различных предметных областей	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм (обработчиков событий);	В целом сформированные и самостоятельно выполняемые, но не всегда интеллектуально обоснованные умения определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм	Успешно сформированные, самостоятельное и осознанно выполняемые умения определять концептуальную (функциональную, событийную) модель программы и перечень разрабатываемых подпрограмм (обработчиков событий); разрабатывать программы для

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
			разрабатывать программы для решения учебных задач из различных предметных областей	(обработчик в событий); разрабатывать программы для решения учебных задач из различных предметных областей	решения учебных задач из различных предметных областей
Владеть: - навыками выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений	Не владеет	Фрагментарно сформированные навыки выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений	В целом сформированные, но выполняемые на элементарном уровне навыки выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений	Сформированные, но не устойчивые в сложных ситуациях навыки выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений	Успешно сформированные, устойчивые, технологически и грамотно выполняемые навыки выбора типов данных и проектирования структур данных; навыками отладки и проверки работоспособности консольных и визуальных приложений
Компетенция ОПК-8 - Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла Индикатор ОПК-8.1. Планирует работы проекта в соответствии с полученным заданием на разных стадиях жизненного цикла информационных систем, в том числе используя облачные сервисы и приложения для управления проектами в сфере ИТ					
Знать: состав команды разработчиков программного проекта, принципы и способы формирования проектной группы	Не знает	Фрагментарные знания состава команды разработчиков программного проекта, принципов и способов формирования проектной группы	Общие, но не структурированные знания состава команды разработчиков программного проекта, принципов и способов формирования проектной группы	Сформированные, но содержащие отдельные пробелы знания состава команды разработчиков программного проекта, принципов и способов формирования	Сформированные систематизированные прочные знания состава команды разработчиков программного проекта, принципов и способов формирования

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
				проектной группы	проектной группы
Уметь: распределять работу в группе с учетом профессиональных компетенций отдельных участников	Не умеет	Частично освоенные умения распределять работу в группе с учетом профессиональных компетенций отдельных участников	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполнением умения распределять работу в группе с учетом профессиональных компетенций отдельных участников	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения распределять работу в группе с учетом профессиональных компетенций отдельных участников	Успешно сформированные, самостоятельно и осознанно выполняемые умения распределять работу в группе с учетом профессиональных компетенций отдельных участников
Владеть: навыками работы в проектных группах по разработке программного обеспечения, в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.)	Не владеет	Фрагментарно сформированные навыки работы в проектных группах по разработке программного обеспечения в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.)	В целом сформированные, но выполняемые на элементарном уровне навыки работы в проектных группах по разработке программного обеспечения в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.)	Сформированные, но не устойчивые в сложных ситуациях навыки работы в проектных группах по разработке программного обеспечения в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.)	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки работы в проектных группах по разработке программного обеспечения в том числе с помощью средств удаленного взаимодействия (сервисы Yandex, Google и др.)
Компетенция ОПК-8 - Способен принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла Индикатор ОПК-8.2. - Осуществляет мониторинг и оценку выполнения работ по проекту, в том числе используя облачные сервисы и приложения для управления командами разработчиков					
Знать:	Не знает	Фрагментарные	Общие, но не	Сформирован	Сформирован

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
технологическое средство управления работой команды		знания технологических средств управления работой команды	структурированные знания технологических средств управления работой команды	ые, но содержащие отдельные пробелы знания технологических средств управления работой команды	ные систематизированные прочные знания технологических средств управления работой команды
Уметь: подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников	Не умеет	Частично освоенные умения подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников	В целом освоенные, но не подкрепляемые знаниями и самостоятельно выполняемые умения подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников	Успешно сформированные, самостоятельно и осознанно выполняемые умения подводить итоги командной работы и оценивать собственную деятельность и деятельность отдельных участников
Владеть: навыками оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной деятельности (сервисы Yandex, Google и др.).	Не владеет	Фрагментарно сформированные навыки оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной деятельности (сервисы Yandex, Google и др.).	В целом сформированные, но выполняемые на элементарном уровне навыки оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной	Сформированные, но не устойчивые в сложных ситуациях навыки оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для совместной	Успешно сформированные, устойчивые, технологически грамотно выполняемые навыки оформления документов по деятельности проектной команды, в том числе с помощью облачных средства для

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
			деятельности (сервисы Yandex, Google и др.).	деятельности (сервисы Yandex, Google и др.).	совместной деятельности (сервисы Yandex, Google и др.).
Компетенция ОПК-9 - способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп Индикатор ОПК-9.1. Осуществляет профессиональные коммуникации с участниками проектных групп, в том числе с применением средств видеоконференцсвязи, облачных сервисов и приложений для управления проектными командами					
Знать: особенности и средства взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения	Не знает	Фрагментарные знания особенностей и средств взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения	Общие, но не структурированные знания особенностей и средств взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения	Сформированные, но содержащие отдельные пробелы знания особенностей и средств взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения	Сформированные систематизированные прочные знания особенностей и средств взаимодействия (в том числе коммуникационного) внутри команды разработчиков программного обеспечения
Уметь: - взаимодействовать внутри проектной группы и ее окружением во время разработки программного проекта, в том числе с применением средств удаленного взаимодействия	Не умеет	Частично освоенные умения по взаимодействию внутри проектной группы и ее окружением во время разработки программного проекта, в том числе с применением средств удаленного взаимодействия	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения по взаимодействию внутри проектной группы и ее окружением во время разработки программного проекта, в том числе с применением средств удаленного взаимодействия	В целом сформированные и самостоятельные выполняемые, но не всегда интеллектуально обоснованные умения по взаимодействию внутри проектной группы и ее окружением во время разработки программного проекта, в том	Успешно сформированные, самостоятельные и осознанно выполняемые умения по взаимодействию внутри проектной группы и ее окружением во время разработки программного проекта, в том числе с применением средств удаленного

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
				числе с применением средств удаленного взаимодействия	взаимодействи я
Владеть: - -навыками проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	Не владеет	Фрагментарно сформированные навыки проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	В целом сформированные, но выполняемые на элементарном уровне навыки проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	Сформированные, но не устойчивые в сложных ситуациях навыки проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	Успешно сформированные, устойчивые, технологические и грамотно выполняемые навыки проведения обсуждений внутри проектной группы, в том числе средствами удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)
Компетенция ОПК-9 - способен принимать участие в реализации профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп Индикатор ОПК-9.2. Осуществляет профессиональные коммуникации с заказчиком проектной деятельности, в том числе с применением средств удаленного взаимодействия и видеоконференцсвязи					
Знать: способы демонстрации результатов программной разработки	Не знает	Фрагментарные знания способов демонстрации результатов программной разработки	Общие, но не структурированные знания способов демонстрации результатов программной разработки	Сформированные, но содержащие отдельные пробелы знания способов демонстрации результатов программной разработки	Сформированные систематизированные прочные знания способов демонстрации результатов программной разработки
Уметь: представлять программную разработку потенциальному Заказчику и демонстрирова	Не умеет	Частично освоенные умения представлять программную разработку потенциальному	В целом освоенные, но не подкрепляемые знаниями и самостоятельностью выполнения умения	В целом сформированные и самостоятельно выполняемые, но не всегда	Успешно сформированные, самостоятельные и осознанно выполняемые умения

Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения				
	1	2	3	4	5
ть ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)		Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	представлять программную разработку потенциальному Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	интеллектуально обоснованные умения представлять программную разработку потенциальному Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)	представлять программную разработку потенциальному Заказчику и демонстрировать ее функционал, в том числе с применением средств удаленного взаимодействия (Zoom, Yandex.Телемост, Telegram и др.)
Владеть: навыками ответов на вопросы и обоснования принятых программных решений	Не владеет	Фрагментарно сформированные навыки ответов на вопросы и обоснования принятых программных решений	В целом сформированные, но выполняемые на элементарном уровне навыки ответов на вопросы и обоснования принятых программных решений	Сформированные, но не устойчивые в сложных ситуациях навыки ответов на вопросы и обоснования принятых программных решений	Успешно сформированные, устойчивые, технологически и грамотно выполняемые навыки ответов на вопросы и обоснования принятых программных решений

КРИТЕРИИ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

При выставлении итоговой оценки за дисциплину необходимо учитывать результаты текущего контроля и критерии оценивания сформированности компетенций.

Оценка «**отлично**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос были продемонстрированы глубокие, систематизированные и прочные знания (в соответствии с критериями сформированности компетенций), умения логично и обоснованно излагать материал, быстро ориентироваться в дополнительных вопросах и давать полные ответы;

- практическое задание (заданий) выполнено безошибочно и в полном соответствии с требованиями, обучающийся демонстрирует устойчивые навыки работы и осознанно исполняемые действия для разработки программных продуктов, способность вносить коррективы и обосновывать свои действия;
- обучающийся участвовал во всех видах текущего контроля и получал по ним максимальные баллы (допустимо отдельные отметки «хорошо»).

Все вышеперечисленное говорит о сформированности запланированных компетенций на высоком уровне.

Оценка «**хорошо**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано хорошее, но недостаточно полное изложение материала, с отдельными пробелами в знаниях (возможно, незначительными неточностями в употреблении терминов), логичное изложение материала, но недостаточно быстрая ориентация в нем при ответе на дополнительные вопросы;
- практическое задание (задания) выполнено с незначительными замечаниями и ошибками, в соответствии с требованиями, обучающийся демонстрирует навыки работы и осознанно исполняемые действия для разработки программных продуктов, но затрудняется внести коррективы;
- обучающийся участвовал во всех видах текущего контроля и получал по ним в основном отметки «хорошо».

Все вышеперечисленное говорит о сформированности запланированных компетенций на хорошем уровне.

Оценка «**удовлетворительно**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано поверхностное владение материалом и фрагментарные знания, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, плохая ориентация в материале при ответе на дополнительные вопросы;
- практическое задание (задания) выполнено с грубыми ошибками и множеством замечаний, с отклонениями от требований, обучающийся демонстрирует непрочные навыки работы, затрудняется в объяснении применяемых средств и конструктивов для разработки программных продуктов, но способен внести отдельные коррективы в работу;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о сформированности запланированных компетенций на низком уровне.

Оценка «**неудовлетворительно**» выставляется обучающемуся, если:

- при ответе на теоретический вопрос было продемонстрировано незнание значительного объема материала, «фрагментарность имеющихся знаний, отсутствие стройного и логично выстроенного ответа, затруднение в употреблении терминов и приведении примеров, неумение выделить главное, сделать выводы и обобщения, неспособность отвечать на дополнительные вопросы;
- практическое задание (задания) не выполнено (программа не работает) или выполнено со значительными отклонениями от требований, с грубыми ошибками и множеством замечаний, обучающийся демонстрирует частично сформированные навыки работы, затрудняется в объяснении применяемых средств и конструктивов для разработки программных продуктов, не способен внести коррективы в работу;
- обучающийся участвовал не во всех мероприятиях текущего контроля и получал по ним в основном удовлетворительные отметки.

Все вышеперечисленное говорит о частичной сформированности запланированных компетенций.

5. ЛИСТ СОГЛАСОВАНИЯ

Составил:

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)

Заведующий выпускающей кафедрой

Н.Б. Стрекалова, д.п.н., доцент



(подпись)