

Частное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ
УПРАВЛЕНИЯ И ЭКОНОМИКИ»

РАССМОТРЕНО И ОДОБРЕНО

УТВЕРЖДАЮ

На заседании кафедры
информационных технологий и
математики
Протокол № 9 от 25.05.2023 г.

Первый проректор
С.В. Авдашкевич
28.06.2023

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина:	Б1.О.28 Алгоритмизация и программирование
Направление подготовки:	38.03.05 Бизнес-информатика
Направленность (профиль):	Цифровые решения для бизнеса
Уровень высшего образования:	Бакалавриат
Форма обучения:	очная, заочная, очно-заочная
Разработчики:	Старший преподаватель, Смирнова С. Л.

Санкт-Петербург
2023

1. Цели и задачи дисциплины:*Цель освоения дисциплины:*

формирование у студентов достаточно полного и строгого представления об алгоритмах программирования и о современных языках программирования (на примере языка С#).

Задачи дисциплины:

- способствовать освоению студентами теоретических основ алгоритмизации задач, практических приемов структурирования данных и программирования на алгоритмических языках высокого уровня (на примере языка С#);

- развить у студентов умения и навыки проектирования и разработки программных систем с использованием структурной и объектно-ориентированной технологий программирования.

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы высшего образования

Планируемые результаты освоения ОП ВО (код и содержание компетенций)	Планируемые результаты обучения по ОП ВО (индикаторы достижения компетенций)	Примечание
ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации;	ОПК-3.1 Знает основные процессы создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий.	-
	ОПК-3.2 Умеет планировать и организовывать процессы создания и использования информационных продуктов и услуг.	
	ОПК-3.3 Владеет навыками разработки алгоритмов и программ для их практической реализации в сфере информационно-коммуникационных технологий.	

Планируемые результаты обучения по ОП ВО (индикаторы достижения компетенций)	Планируемые результаты обучения по дисциплине
ОПК-3.1. Знает основные процессы создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий.	- общие принципы построения алгоритмов, основные алгоритмические конструкции; - современные программные средства для практической реализации программного обеспечения на персональных компьютерах; - основы программирования на языке высокого уровня объектно-ориентированной методологии С#; - основы функционирования программ, управляемых событиями; - основные принципы разработки пользовательского интерфейса.
ОПК-3.2. Умеет планировать и организовывать процессы создания и использования информационных продуктов и услуг.	- реализовывать алгоритмы решения задач на языке высокого уровня (С#); - разрабатывать программы, управляемые событиями; - разрабатывать программные приложения в среде визуального программирования с использованием технологий визуальной разработки интерфейсов (на примере среды Microsoft Visual Studio).
ОПК-3.3. Владеет навыками разработки алгоритмов и программ для их практической реализации в сфере информационно-коммуникационных технологий.	- владеет практическими навыками составления алгоритмов для решения задач; реализации алгоритмов на языке программирования С#.

3. Содержание, объем дисциплины и формы проведения занятий

№ п/п	Наименование темы дисциплины	Компетенции	Оценочные средства текущего контроля		
			ЗНАТЬ	УМЕТЬ	ВЛАДЕТЬ
			ОПК-3.1	ОПК-3.2	ОПК-3.3
1	Основы алгоритмизации	ОПК-3	Собеседование, опрос/ Контрольная работа №1 (10)	Задача №1 (10)	Расчетно-графическая работа №1 (20)
2	Введение в программирование на C#	ОПК-3	Собеседование, опрос/ Контрольная работа №2 (10)	Задача №2 (10)	Расчетно-графическая работа №1 (20)
3	Объектно-ориентированное программирование на C#	ОПК-3	Собеседование, опрос/ Контрольная работа №3 (10)	Задача №3 (10)	Расчетно-графическая работа №1 (20)
4	Разработка программ с графическим интерфейсом	ОПК-3	Собеседование, опрос/ Контрольная работа №4 (10)	Задача №4 (10)	Расчетно-графическая работа №1 (20)
Количество баллов (100 баллов):			100		

Содержание учебного материала, лабораторные работы и практические занятия, курсовая работа
<p>Тема 1: Основы алгоритмизации Основные понятия алгоритмизации. Понятие алгоритма. Свойства алгоритма. Формы записи алгоритмов. Основные алгоритмические конструкции, используемые при проектировании алгоритмов линейных, ветвящихся и циклических вычислительных процессов. Правила построения блок-схем алгоритмов. Принципы построения алгоритмов: использование базовых структур, метод последовательной детализации, сборочный метод. Языки программирования, классификация языков программирования.</p> <p>Практические занятия/самостоятельная работа: Разработка линейных алгоритмов и алгоритмов ветвления. Разработка циклических алгоритмов. Анализ готовых алгоритмов.</p> <p>Лабораторная работа: -</p>
<p>Тема 2: Введение в программирование на C# Платформа .NET Framework. Обзор сборок .Net. Общая система типов (CTS). Основные конструкции C#. Базовые типы данных. Тип string. Типы DateTime и TimeSpan. Тип Var.</p> <p>Практические занятия/самостоятельная работа: Выполнение расчетов. Консольный ввод-вывод. Ввод-вывод с использованием файлов. Форматирование значений данных. Управляющие операторы для организации циклов. Массивы. Строки.</p> <p>Лабораторная работа: -</p>
<p>Тема 3: Объектно-ориентированное программирование на C# Основные принципы ООП. Инкапсуляция. Классы. Создание объектов. Члены классов. Модификаторы доступа. Ключевое слово this. Конструкторы. Свойства, автоматические свойства. Константные поля. Ключевое слово readonly. Статические члены класса. Атрибут partial. Наследование и полиморфизм. Защищенный доступ. Запечатанные классы, ключевое слово sealed. Соккрытие имен, ключевое слово base. Виртуальные члены класса. Ключевые слова virtual и override. Абстрактные классы. Создание иерархии наследования. Соккрытие членов класса. Правила приведения к базовому и производному классу. Ключевое слово as. Ключевое слово is. Главный родительский класс System.Object. Перегрузка функций-членов класса. Интерфейсы. Понятие интерфейсных типов. Реализация интерфейса. Проектирование иерархий интерфейсов. Коллекции и Обобщения. Пространство имен System.Collections. Инициализация коллекций. Делегаты и события.</p> <p>Практические занятия/самостоятельная работа: Классы и методы. Параметризованные коллекции. Иерархия классов.</p> <p>Лабораторная работа: -</p>
<p>Тема 4: Разработка программ с графическим интерфейсом</p>

Содержание учебного материала, лабораторные работы и практические занятия, курсовая работа
<p>Событийно-управляемое программирование. Создание оконных приложений Windows Forms. Формы в приложениях C#. Настройка внешнего вида и поведения формы. Обработка событий. Автоматически генерируемый код графического интерфейса. Делегаты и механизм обработки событий. Обзор основных элементов управления. Класс Control. Свойства элементов управления. Редактирование макета графического интерфейса. Текстовые элементы управления. Надписи, текстовые поля. Классы, способные порождать текстовые элементы управления. Кнопки, переключатели. Кнопки. Абстрактный класс ButtonBase. Класс Button (Кнопка). Флажок CheckBox. Переключатель RadioButton и контейнеры. Контейнеры GroupBox и Panel. Списки и полосы прокрутки. Списки CheckedListBox, ListBox, ComboBox. Графическое окно PictureBox. Полосы прокрутки и абстрактный класс ScrollBar. Работа с таймером. Ползунок TrackBar и индикатор хода процесса ProgressBar. Элементы для работы с датой и временем, древовидного и спискового представления информации. Элемент выбора даты и времени DateTimePicker. Элемент управления MonthCalendar. Элемент древовидного представления TreeView. Элемент спискового представления ListView. Совместная работа элементов TreeView и ListView в Проводнике файловой системы. Табличное представление информации. Элемент табличного представления DataGridView. Программное управление элементом. Контейнеры, меню и панели инструментов. Контейнер изображений ImageList. Строка состояния. Стандартные диалоги. Диалоги открытия и сохранения файлов, изменения шрифта и цвета, выбора принтера и печати. Свойства диалогов. Методы диалогов.</p> <p>Практические занятия/самостоятельная работа: Разработка программ с оконным интерфейсом. Использование элементов управления, стандартных диалогов. Графика.</p> <p>Лабораторная работа: -</p> <p>Курсовая работа: не предусмотрено учебным планом</p>

Очная форма обучения

Вид учебной работы	Всего часов	Семестр 5
Аудиторные занятия (АЗ):	72	72
Лекционные занятия (Лек)	36	36
Лабораторные занятия (Лаб)	0	0
Практические занятия (Пр)	36	36
Самостоятельная работа студента (СР)	74	74
Курсовая работа	0	0
Другие виды самостоятельной работы*	74	74
Контроль самостоятельной работы (КСР)	7	7
Контактная работа (КоР)	79	79
Форма промежуточной аттестации	0	Экзамен
Подготовка к экзамену и сдача экзамена (СР, КоР)	27	27
Общая трудоемкость дисциплины, часы/ЗЕТ	180/5	180/5

* Подготовка к аудиторным занятиям, подготовка к зачету (при наличии)

№	Наименование темы дисциплины	Семестр/ Курс	Количество учебных часов				СР	Практическая подготовка
			В том числе по видам аудиторных занятий					
			Лек	Пр	Лаб			
1	Основы алгоритмизации	5	6	6	0	12	6	
2	Введение в программирование на C#	5	10	10	0	14	10	
3	Объектно-ориентированное программирование на C#	5	10	8	0	16	8	
4	Разработка программ с графическим интерфейсом	5	10	12	0	32	12	
Итого:			36	36	0	74	36	

* Практическая подготовка при реализации дисциплин организована путем проведения практических занятий и (или) выполнения лабораторных и (или) курсовых работ и предусматривает выполнение работ, связанных с будущей профессиональной деятельностью.

Заочная форма обучения

Вид учебной работы	Всего часов	Семестр 5
Аудиторные занятия (АЗ):	12	12
Лекционные занятия (Лек)	6	6

Вид учебной работы	Всего часов	Семестр 5
Лабораторные занятия (Лаб)	0	0
Практические занятия (Пр)	6	6
Самостоятельная работа студента (СР)	151	151
Курсовая работа	0	0
Другие виды самостоятельной работы*	151	151
Контроль самостоятельной работы (КСР)	8	8
Контактная работа (КоР)	20	20
Форма промежуточной аттестации	0	Экзамен
Подготовка к экзамену/зачету и сдача экзамена/зачета (СР, КоР)	9	9
Общая трудоемкость дисциплины, часы/ЗЕТ	180/5	180/5

* Подготовка к аудиторным занятиям

№	Наименование темы дисциплины	Семестр/ Курс	Количество учебных часов				Практическая подготовка
			В том числе по видам аудиторных занятий			СР	
			Лек	Пр	Лаб		
1	Основы алгоритмизации	5	2	0	0	28	6
2	Введение в программирование на С#	5	2	2	0	36	10
3	Объектно-ориентированное программирование на С#	5	0	2	0	38	8
4	Разработка программ с графическим интерфейсом	5	2	2	0	49	12
Итого:			6	6	0	151	36

* Практическая подготовка при реализации дисциплин организована путем проведения практических занятий и (или) выполнения лабораторных и (или) курсовых работ и (или) путем выделения часов из часов, отведенных на самостоятельную работу, и предусматривает выполнение работ, связанных с будущей профессиональной деятельностью.

Очно-заочная форма обучения

Вид учебной работы	Всего часов	Семестр 5
Аудиторные занятия (АЗ):	54	54
Лекционные занятия (Лек)	18	18
Лабораторные занятия (Лаб)	0	0
Практические занятия (Пр)	36	36
Самостоятельная работа студента (СР)	94	94
Курсовая работа	0	0
Другие виды самостоятельной работы*	94	94
Контроль самостоятельной работы (КСР)	5	5
Контактная работа (КоР)	59	59
Форма промежуточной аттестации	0	Экзамен
Подготовка к экзамену и сдача экзамена (СР, КоР)	27	27
Общая трудоемкость дисциплины, часы/ЗЕТ	180/5	180/5

* Подготовка к аудиторным занятиям, подготовка к зачету (при наличии)

№	Наименование темы дисциплины	Семестр/ Курс	Количество учебных часов				Практическая подготовка
			В том числе по видам аудиторных занятий			СР	
			Лек	Пр	Лаб		
1	Основы алгоритмизации	5	4	6	0	14	6
2	Введение в программирование на С#	5	6	10	0	20	10
3	Объектно-ориентированное программирование на С#	5	4	8	0	22	8
4	Разработка программ с графическим интерфейсом	5	4	12	0	38	12
Итого:			18	36	0	94	36

* Практическая подготовка при реализации дисциплин организована путем проведения практических занятий и (или) выполнения лабораторных и (или) курсовых работ и (или) путем выделения часов из часов, отведенных на самостоятельную работу, и предусматривает выполнение работ, связанных с будущей профессиональной деятельностью.

4. Способ реализации дисциплины

Без использования онлайн-курса.

5. Учебно-методическое обеспечение дисциплины:

Основная литература:

1. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++: ПРАКТИЧЕСКИЙ КУРС. Учебное пособие для вузов / Огнева М. В., Кудрина Е. В. - Национальный исследовательский Саратовский государственный университет имени Н.Г. Чернышевского (г. Саратов)., 2022 г. - 335 с. - ISBN 978-5-534-05123-0 – Режим доступа: <https://urait.ru/book/programmirovanie-na-yazyke-s-prakticheskiy-kurs-492984>

2. МАТЕМАТИЧЕСКАЯ ЛОГИКА И ТЕОРИЯ АЛГОРИТМОВ 5-е изд. Учебник и практикум для вузов / Судоплатов С. В., Овчинникова Е. В. - Новосибирский государственный технический университет (г. Новосибирск)., 2022 г. - 207 с. - ISBN 978-5-534-12274-9 – Режим доступа: <https://urait.ru/book/matematiceskaya-logika-i-teoriya-algoritmov-447321>

3. АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ. Учебник для вузов / Трофимов В. В., Павловская Т. А. ; Под ред. Трофимова В.В. - Санкт-Петербургский государственный экономический университет (г. Санкт-Петербург).; Национальный исследовательский университет ИТМО (г. Санкт-Петербург)., 2022 г. - 137 с. - ISBN 978-5-534-07834-3 – Режим доступа: <https://urait.ru/book/algoritmizaciya-i-programmirovanie-491215>

Дополнительная литература:

1. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ C#. Учебное пособие для вузов / Кудрина Е. В., Огнева М. В. - Национальный исследовательский Саратовский государственный университет имени Н.Г. Чернышевского (г. Саратов)., 2020 г. - 322 с. - ISBN 978-5-534-09796-2 – Режим доступа: <https://urait.ru/book/osnovy-algoritmizacii-i-programmirovaniya-na-yazyke-c-456182>

2. ПРОГРАММИРОВАНИЕ. ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ ПОДХОД. Учебник и практикум для вузов / Зыков С. В. - Национальный исследовательский университет «Высшая школа экономики» (г. Москва)., 2022 г. - 155 с. - ISBN 978-5-534-00850-0 – Режим доступа: <https://urait.ru/book/programmirovanie-obektno-orientirovannyy-podhod-490423>

3. ТЕОРИЯ АЛГОРИТМОВ. ВВЕДЕНИЕ В СЛОЖНОСТЬ ВЫЧИСЛЕНИЙ 2-е изд., испр. и доп. Учебное пособие для вузов / Крупский В. Н. - Московский государственный университет имени М.В. Ломоносова (г. Москва)., 2022 г. - 117 с. - ISBN 978-5-534-04817-9 – Режим доступа: <https://urait.ru/book/teoriya-algoritmov-vvedenie-v-slozhnost-vychisleniy-492937>

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения

1. Операционная система
2. Пакет прикладных офисных программ
3. Антивирусное программное обеспечение
4. LMS Moodle
5. Вебинарная платформа
6. MS Visual Studio

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», информационных справочных систем и профессиональных баз данных, необходимых для освоения дисциплины

1. ibooks.ru : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <https://ibooks.ru>. - Текст: электронный

2. Электронно-библиотечная система СПбУТУиЭ : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <http://libume.ru>. - Текст: электронный

3. Юрайт : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <https://urait.ru>. - Текст: электронный

4. eLibrary.ru : научная электронная библиотека [Электронный ресурс] : профессиональная база данных. - Режим доступа: <http://elibrary.ru>. - Текст: электронный

5. Архив научных журналов НЭИКОН [Электронный ресурс] : профессиональная база данных. - Режим доступа: arhiv.neicon.ru. - Текст: электронный

6. КиберЛенинка : научная электронная библиотека [Электронный ресурс] : информационная справочная система. - Режим доступа: <http://cyberleninka.ru>. - Текст: электронный

7. Лань : электронно-библиотечная система [Электронный ресурс] : профессиональная база данных. - Режим доступа: <https://e.lanbook.com>. - Текст: электронный

8. it-world.ru [Электронный ресурс] : информационная справочная система . - Режим доступа: <https://www.it-world.ru>. - Текст: электронный

9. Connect: IT-технологии : информационная справочная система. - Режим доступа: <https://www.connect-wit.ru/>. - Текст: электронный

10. Квант [Электронный ресурс] : информационная справочная система . - Режим доступа: <http://kvant.mcsme.ru>. - Текст: электронный

11. Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации: профессиональная база данных. - Режим доступа: <https://digital.gov.ru>. - Текст: электронный

12. Бизнес-информатика: профессиональная база данных . - Режим доступа: <https://bijournal.hse.ru/>. - Текст: электронный

13. Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций: профессиональная база данных . - Режим доступа: <https://rkn.gov.ru>. - Текст: электронный

8. Материально-техническое обеспечение дисциплины

1. Учебная аудитория для проведения занятий лекционного типа, семинарского типа - практических занятий, для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оборудованная: рабочими местами для обучающихся, оснащёнными специальной мебелью; рабочим местом преподавателя, оснащённым специальной мебелью, персональным компьютером с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета; техническими средствами обучения - мультимедийным оборудованием (проектор, экран, колонки) и маркерной доской; лицензионным программным обеспечением

2. Учебная аудитория для проведения занятий семинарского типа - практических занятий – компьютерный класс, оборудованный рабочими местами для обучающихся, оснащёнными специальной мебелью, персональными компьютерами с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета; рабочим местом преподавателя, оснащённым специальной мебелью, персональным компьютером с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета; техническими средствами обучения - мультимедийным оборудованием (проектор, экран, колонки) и маркерной доской; лицензионным программным обеспечением

3. При применении электронного обучения, дистанционных образовательных технологий используются: виртуальные аналоги учебных аудиторий - вебинарные комнаты на вебинарных платформах, рабочее место преподавателя, оснащённое персональным компьютером (планшет, мобильное устройство) с возможностью подключения к сети «Интернет», доступом к

электронной информационно-образовательной среде Университета и к информационно-образовательному portalу Университета imeos.ru, веб-камерой, микрофоном и гарнитурой (в т.ч. интегрированными в устройствами), программным обеспечением; рабочее место обучающегося оснащено персональным компьютером (планшет, мобильное устройство) с возможностью подключения к сети «Интернет», доступом к электронной информационно-образовательной среде Университета и к информационно-образовательному portalу Университета imeos.ru, веб-камерой, микрофоном и гарнитурой (в т.ч. интегрированными в устройства). Авторизация на информационно-образовательном portalе Университета imeos.ru и начало работы осуществляются с использованием персональной учетной записи (логина и пароля). Лицензионное программное обеспечение

4. Помещение для самостоятельной работы, оборудованное специальной мебелью, персональными компьютерами с возможностью подключения к сети «Интернет» и доступом к электронной информационно-образовательной среде Университета, лицензионным программным обеспечением

9. Оценочные материалы по дисциплине

Описание оценочных средств (показатели и критерии оценивания, шкалы оценивания) представлено в приложении к основной профессиональной образовательной программе «Каталог оценочных средств текущего контроля и промежуточной аттестации».

Процедуры оценивания знаний, умений, навыков и (или) опыта деятельности приводятся в соответствующих методических материалах и локальных нормативных актах Университета.

Для оценивания учебных достижений студентов в Университете действует балльно-рейтинговая система.

Если оценка, соответствующая набранной в семестре сумме рейтинговых баллов, удовлетворяет студента, то она является итоговой оценкой по дисциплине при проведении промежуточной аттестации в форме экзамена/зачета с оценкой/зачета.

Условием сдачи экзамена/зачета с оценкой/зачета с целью повышения итоговой оценки по дисциплине является сдача студентом экзамена, за который он получает экзаменационные баллы без учета баллов, полученных за текущий контроль:

Шкала оценивания учебных достижений по дисциплине, завершающейся зачетом без оценки

Баллы по дисциплине	60 и менее		61-73		74-90		91-100
Итоговая оценка по дисциплине	Незачет		Зачет				
Баллы в международной шкале ECTS с буквенным обозначением уровня	50 и менее	51-60	61-67	68-73	74-83	84-90	91-100
	F	Fx	E	D	C	B	A
Уровень сформированности компетенций	Не сформированы		Пороговый		Высокий		Повышенный

Шкала оценивания учебных достижений по дисциплине, завершающейся экзаменом/зачетом с оценкой

Баллы по дисциплине	60 и менее		61-73		74-90		91-100
Итоговая оценка по дисциплине	Неудовлетворительно		Удовлетворительно		Хорошо		Отлично
Баллы в международной шкале ECTS с буквенным обозначением уровня	<50	51-60	61-67	68-73	74-83	84-90	91-100
	F	Fx	E	D	C	B	A
Уровень сформированности компетенций	Не сформированы		Пороговый		Высокий		Повышенный

9.1. Типовые контрольные задания для текущего контроля

Задача №1

1. Напишите название свойства, соответствующее каждому определению.

_____ - описание процесса решения на языке исполнителя.

_____ - алгоритм приводит к решению задачи.

_____ - алгоритм дает правильное решение при различных наборах начальных данных.

_____ - процесс решения разбит на отдельные шаги.

_____ - запись алгоритма четкая, полная, ясная.

_____ - задача решается за известное количество шагов.

2. Запишите исполнителей для приведённых ниже видов работ:

Уборка мусора во дворе –

Перевозка пассажиров в поезде –

Выдача заработной платы –

Приём экзаменов в школе –

Сдача экзамена в университете –

Набор текста на компьютере –

Приготовление еды в ресторане –

3. Определите полный набор данных для решения следующих задач обработки информации:

Вычисление стоимости покупок в магазине –

Вычисление суммы сдачи от данных вами продавцу денег –

Определение времени показа по телевизору интересующего вас фильма –

Вычисление площади треугольника –

Определение времени падения кирпича с крыши дома –

Определение месячной платы за расход электроэнергии –

Перевод русского текста на транслит –

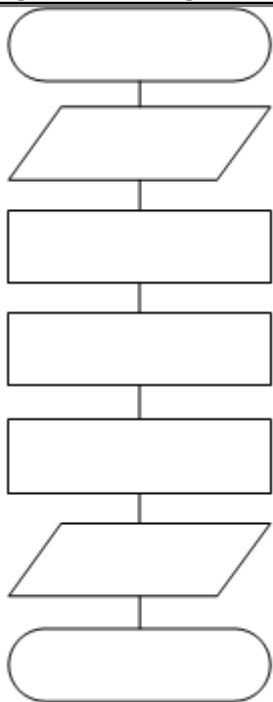
4. Есть исполнитель «Перевозчик», который перевозит через реку волка, козу и капусту.

Напишите алгоритм перевоза через реку волка, козы и капусты, если СКИ «Перевозчика» содержит 5 команд: ВЗЯТЬ КОЗУ, ВЗЯТЬ ВОЛКА, ВЗЯТЬ КАПУСТУ, ВЫСАДИТЬ, ПЕРЕПЛЫТЬ. В лодку может поместиться только один предмет или животное. Нельзя оставлять на берегу одних волка с козой и козу с капустой.

5. Составить алгоритм в *словесной форме* для решения задачи нахождения минимального из трех значений.

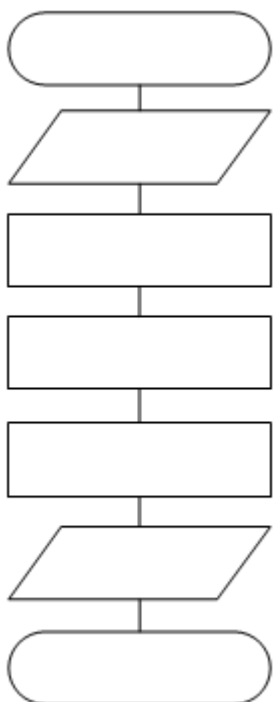
6. Составить алгоритм в *словесной форме* для решения задачи деления отрезка пополам с помощью циркуля и линейки.

7. Дана длина ребра куба. Составьте блок-схему алгоритма нахождения площади грани, полной поверхности и объема этого куба.



8. Дано a . Не используя никаких функций и операций, кроме умножения, получить a^8 за три операции. Заполните шаблон, используя таблицу значений при $a=2$.

шаг алгоритма	аргумент	промежуточные величины		результат	пояснения
	a	b	c	y	
1	2				ввод a
2		4			
3			16		
4				256	
5					вывод $y=256$
6					конец



9. Дано действительное число a . Не пользуясь никакими арифметическими операциями, кроме умножения, получить:

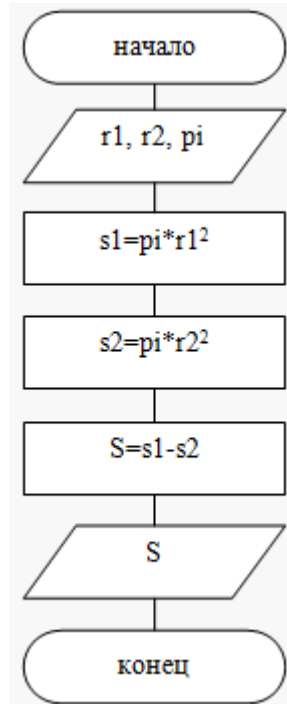
а) a^4 за две операции;

б) a^6 за три операции;

в) a^7 за четыре операции;

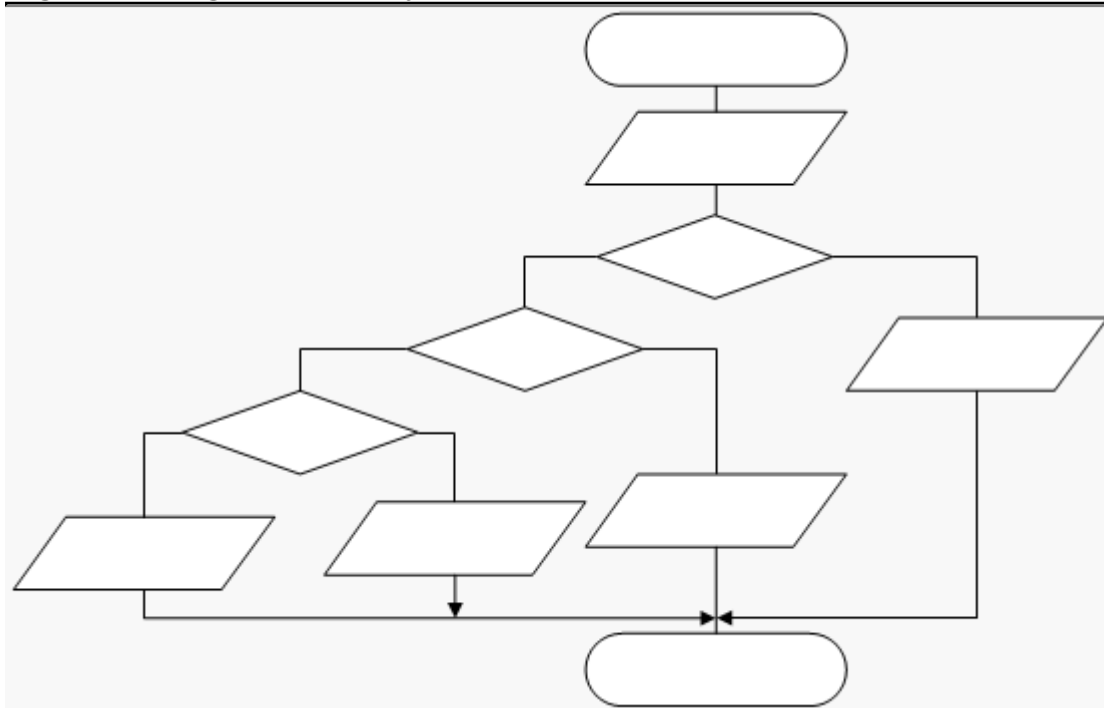
г) a^{12} за четыре операции.

10. По заданной блок-схеме алгоритма восстановите условие задачи.

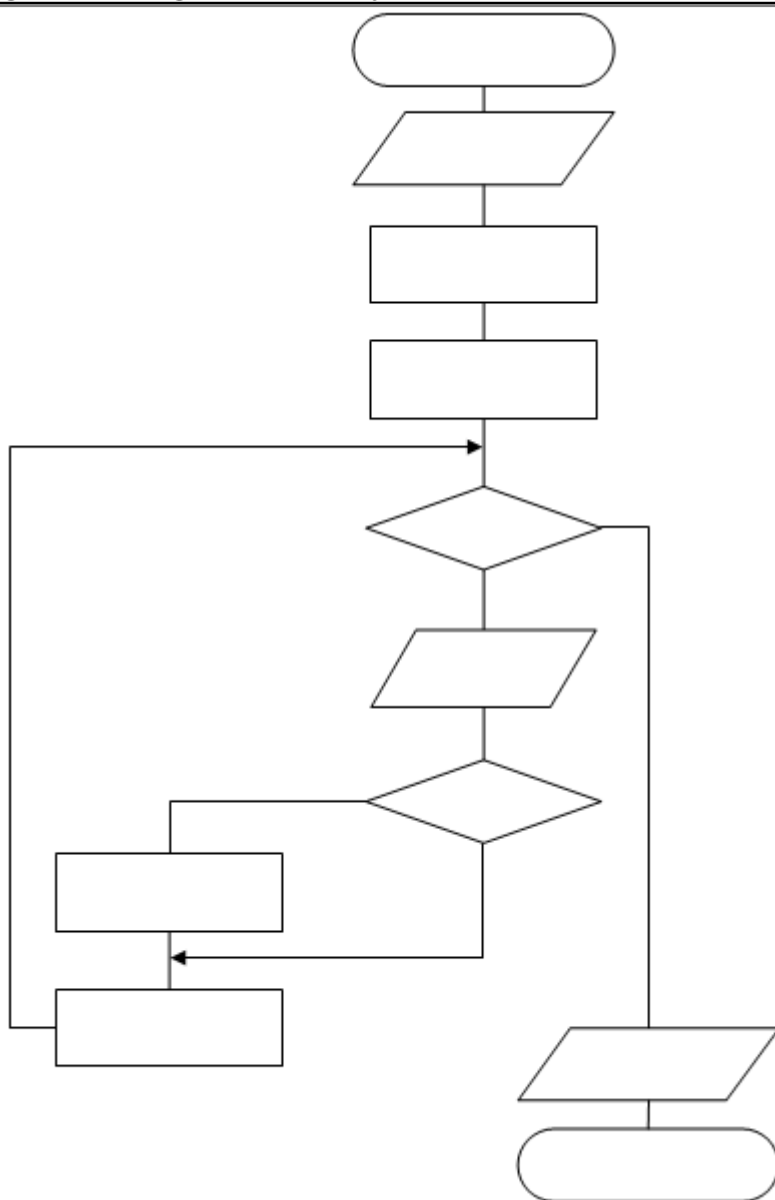


шаг алгоритма	аргумент			промежуточные величины		результат	пояснения
	r1	r2	pi	s1	s2	S	
1							
2							
3							
4							
5							
6							

11. Артериальное кровяное давление у студентов колледжа (16-20 лет) считается нормальным, если верхнее артериальное давление $H < 120$, а нижнее $h > 70$ мм рт.ст. Причем $H - h > 30$. Составьте алгоритм, который в зависимости от измеренных значений H и h выдает одно из следующих значений: «нормальное», «повышенное», «пониженное».



12. Составьте блок-схему алгоритма нахождения максимального числа среди заданных n чисел ($n > 2$).



Собеседование, опрос/Контрольная работа №1

- Определение алгоритма.
- Свойства алгоритма.
- Данные в алгоритмах.
- Формы представления алгоритмов.
- Правила построения блок схем.
- Определение программы.
- Этапы решения задач.
- Критерии оценки алгоритма
- Базовые алгоритмические структуры.
- Дополнительные структуры. Реализация с помощью базовых структур.
- Разветвленная структура. Ветвление, полная и неполная формы.
- Определение цикла. Виды циклов.
- Определение языка программирования.

Задача №2

1. Составить программу для выполнения расчетов функции:

$$A = \left(x + \frac{7}{6}\right)^{\frac{4}{3}} + \sin e^x + \arcsin(\cos 2x)$$

Значение аргумента должно задаваться в диалоге с пользователем (с использованием метода Console.ReadLine()).

2. Осуществить ввод необходимых данных, выполнить реализацию алгоритма с использованием операторов цикла for, обеспечить вывод полученных результатов. Не разрешается использовать другие операторы цикла. Не допускается использование массивов и операторов, прерывающих ход программы (break, goto).

Ввести целое число $N > 1$ и две вещественные точки на числовой оси: A, B ($A < B$). Отрезок $[A, B]$ разбит на N равных отрезков. Вывести N — длину каждого отрезка, а также значения функции $f(x) = 1 - \sin(x)$ в точках, разбивающих отрезок $[A, B]$: $f(A), f(A + H), f(A + 2H), \dots, f(B)$.

3. Ввод значений массивов осуществить из файла или с клавиатуры. Выполнить реализацию алгоритма с использованием операторов цикла foreach, обеспечить вывод полученных результатов.

Задан одномерный массив размером N . Определить количество отрицательных чисел, количество положительных чисел и среднее арифметическое всех чисел.

4. Решить задачу с использованием типа string. Ввод данных должен осуществляться в диалоге с пользователем.

Дана строка, состоящая из групп нулей и единиц. Посчитать количество нулей и единиц.

Собеседование, опрос/Контрольная работа №2

1. Общая характеристика платформы MSDN. Структура программы на языке программирования C#.
2. Язык C#, CLI, основы синтаксиса языка, ссылочные типы и типы-значения, преобразования типов, представление объектов в памяти.
3. Методы: способы передачи параметров, абстрактные, виртуальные и статические методы, модификаторы видимости.
4. Платформа .NET: общее описание, CLR, IL, CTS.
5. Сборки: понятие сборки, сильные и слабые имена, загрузка сборки. Понятие целевой платформы.
6. Классификация типов данных. Встроенные типы данных. Преобразование типов.
7. Основные операторы. Чтение данных с клавиатуры. Вывод данных на экран.
8. Понятие исключительной ситуации.
9. Массивы. Размерность массива. Методы и свойства класса Array.
10. Регулярные и буквальные строковые литералы. Ссылки типа string. Операции над строками.
11. Методы и свойства класса String. Форматирование строк. Неизменяемость объектов класса String. Тип System.Text.StringBuilder.
12. Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками. Режимы доступа к файлу, режимы открытия файла, режимы совместного использования файла.
13. Потоки байтов FileStream.
14. Потоки символов StreamWriter и StreamReader.

Задача №3

1. Написать программу, осуществляющую заданные вычисления. Вид используемых классов и методов определить самостоятельно. В конструкторе без параметров для инициализации полей использовать случайные числа.

Заданы две окружности $(x - a)^2 + (y - b)^2 = r^2$ и $(x - c)^2 + (y - d)^2 = R^2$ и координаты четырех точек. Определить количество точек, лежащих внутри каждой окружности.

2. Описать структуру с именем ZNAK, содержащую следующие поля: фамилия, имя; знак Зодиака; дата рождения (массив из трех чисел). Создать параметризованную коллекцию для хранения описанной структуры. Вид коллекции выбрать самостоятельно. Написать приложение для работы с этой коллекцией, позволяющее выполнять:

- добавление элемента в коллекцию с клавиатуры;
- считывание данных из файла;
- запись данных в указанный файл;
- сортировку данных по различным критериям;
- поиск элемента по заданному полю;
- вывод всех элементов, удовлетворяющих заданному условию;
- удаление элемента из коллекции;
- ввод с клавиатуры данных (записи должны быть упорядочены по дате рождения);
- вывод на экран информации о человеке, чья фамилия введена с клавиатуры (если такого нет, вывести соответствующее сообщение).

Приложение должно предусматривать обработку ошибок.

3. Построить иерархию классов: журнал, книга, печатное издание, учебник. Разработать методы и свойства для каждого из определяемых классов. Реализовать программу на C#. Расширить иерархию классов с использованием виртуального класса в качестве основы иерархии.

Показать пример использования полиморфизма методов. Реализовать для иерархии механизм интерфейсов, при этом один из классов должен реализовывать как минимум два интерфейса. Использовать для проверки всех методов данного класса многоадресный делегат.

Собеседование, опрос/Контрольная работа №3

1. Определение класса. Поля. Методы. Объекты. Конструкторы. Деструктор. Сбор мусора.
2. Ключевое слово this. Ключевое слово static. Статические конструкторы. Статические классы.
3. Перегрузка операций.
4. Основные свойства ООП. Инкапсуляция. Спецификаторы доступа. Свойства. Автоматические свойства. Индексаторы.
5. Основные свойства ООП. Иерархии классов. Наследование. Виртуальные методы. Механизм позднего связывания.
6. Абстрактные классы. Класс object. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET
7. Понятие делегата. Описание и использование.
8. События.
9. Механизм обработки исключений. Системные исключения и их обработка. Свойства и методы класса Exception. Исключения, создаваемые программистом.
10. Основные абстрактные структуры данных. Физическое представление данных.
11. Пространство имен System.Collections. Интерфейсы пространства имен System.Collections.
12. Коллекции общего назначения. Основные элементы класса ArrayList.
13. Коллекции общего назначения. Основные элементы класса Hashtable.
14. Коллекции общего назначения. Основные элементы класса SortedList.
15. Коллекции общего назначения. Основные элементы класса Stack.
16. Коллекции общего назначения. Основные элементы класса Queue.

17. Коллекции общего назначения. Доступ к коллекциям с помощью нумератора.
18. Классы-прототипы. Основные преимущества использования обобщений. Ограничения.
19. Параметризованные коллекции библиотеки .NET.
20. Исключения: бросание, перебрасывание и обработка, библиотечные исключения, свойства исключений, хорошие практики.
21. LINQ: основные методы, синтаксис, основные реализации.

Задача №4

1. Создайте программу с кнопкой, меткой и полем ввода. При щелчке на соответствующий элемент на форме динамически должен создаваться подобный ему элемент. Предусмотреть возможность вывода количества кнопок, меток и полей ввода.

2. Напишите Windows-приложение, предназначенное для создания простого калькулятора.

Требования:

- Выполнение основных арифметических операций с целыми (и вещественными) числами.
- Текущее число должно отображаться в «строке ввода»; для очистки строки ввода должна быть предусмотрена кнопка «С» – очистить.
- Ввод данных должен осуществляться с помощью кнопок (цифры 0-9; знака «,» – для разделителя целой и дробной части; знак «=»).
- В приложении должно быть предусмотрено наличие меню (по крайней мере, с командами «файл/выход» и «?/о программе»).
- В приложении должно быть предусмотрено переключение режима: простой/расширенный. В расширенном режиме должны добавляться дополнительные функции (в зависимости от варианта).
- Сообщение от цифровых кнопок (0-9) должно обрабатываться единым обработчиком.
- (*) Ввод данных должен также осуществляться с помощью клавиатуры.
- (**) Калькулятор должен позволять производить операции сложения и вычитания целых чисел в N-ичной системе счисления (1

3. Напишите Windows-приложение, предназначенное для имитации движения «шарика».

Под «шариком» понимается некий графический объект (графический примитив – квадрат, прямоугольник, круг – или произвольная картинка).

Окно приложения должно быть разделено на 2 части:

- Левая – «поле» для движения «шарика».
- Правая – настройка параметров.

Пуск и остановка движения должна производиться с помощью соответствующей кнопки.

В программе должно быть предусмотрено изменение следующих параметров:

- Цвет поля/«шарика».
- Скорость движения «шарика».
- Угол движения «шарика» (или шаг по горизонтали/вертикали).

Для анимирования движения «шарика» следует использовать компонент Timer. Для изменения параметров должны быть использованы компоненты: группа радиокнопок, поле со списком, трекбар (синхронизировать с элементом NumericUpDown).

4. Реализуйте проект с двумя формами. Обеспечьте сохранение выбранных параметров при закрытии формы с настройками. В программе должна быть предусмотрена обработка ошибочного ввода пользователя. Настройте переходы между формами согласно следующему условию:

Создать меню, состоящее из пунктов Input color, Change, Clear.

При выборе команды Input color открывается диалоговое окно, содержащее:

- группу из двух флажков (Up, Down) типа CheckBox;
- группу из трех переключателей (Red, Green, Blue) типа RadioButton;
- кнопку типа Button.

Обеспечить возможность:

- выбора цвета с помощью переключателей;
- ввода режима, определяющего, какая область закрашивается: все окно, его верхняя или нижняя половина.

При выборе команды Change цвет главного окна изменяется на заданный (верхняя, нижняя или обе половины в зависимости от введенного режима). При выборе команды Clear восстанавливается первоначальный цвет окна.

5. Разработать оконное приложение для обработки двумерного массива размера $M \times N$ целых чисел.

Для визуального отображения процесса работы с матрицей, а также ввода и редактирования ее элементов, предполагается использование элемента dataGridView.

Требования к программе:

- Предусмотреть возможность изменения размеров матрицы.
- Предусмотреть возможность ввода только числовых значений.

Выполнить обработку матрицы согласно заданию:

- Умножение матрицы на число.
- Поиск наименьшего значения среди максимальных элементов строк.

Собеседование, опрос/Контрольная работа №4

1. Создание оконных приложений Windows Forms.
2. Классы Application и Form.
3. События в Windows-приложениях.
4. Основные свойства элементов управления.
5. Текстовые элементы управления.
6. Текстовые элементы управления списками.
7. Элементы управления для создания кнопок.
8. Табличное представление информации. Элемент DataGridView.
9. Общие элементы управления.
10. Задание порядка перехода по Tab.
11. Формирование меню приложения.
12. Контейнеры, меню и панели инструментов.
13. Модальные и немодальные формы.
14. Организация взаимодействия между формами.
15. Общие диалоговые окна.
16. Рисование графических примитивов.

Расчетно-графическая работа №1

Вариант 1. "Графики функций"

Напишите приложение, обеспечивающее построение графика заданной функции одной переменной $y=f(x)$ в заданных границах изменения аргумента $(x \in [m, n])$ при заданных значениях параметров a и b .

Список функций для построения:

- $y = \frac{a}{bx}$
- $y = a \cdot \sin\{bx\}$
- $y = a \cdot \cos\{bx\}$
- $y = \sqrt[a]{x^b}$

где a и b – числовые [вещественные] константы, задаваемые пользователем.

Основные требования к программе:

- Выбор функции для построения осуществляется пользователем.

- Значения параметров a и b задаются пользователем.

- Границы изменения аргумента m и n задаются пользователем.

Вариант 2. "Диаграммы"

Напишите приложение, обеспечивающее построение диаграммы одного из двух типов: линейной или столбчатой. Исходными данными является последовательность положительных действительных чисел a_1, a_2, \dots, a_n , обозначающая результаты каких-либо измерений. Требуется построить визуализированное представление этой последовательности с целью сравнения полученных результатов. Исходные данные задаются с помощью текстового файла `in.txt`.

Выбор типа диаграммы должен осуществляться пользователем с помощью окна, вызываемого командой «Настройки».

Основные команды приложения продублировать на панели инструментов.

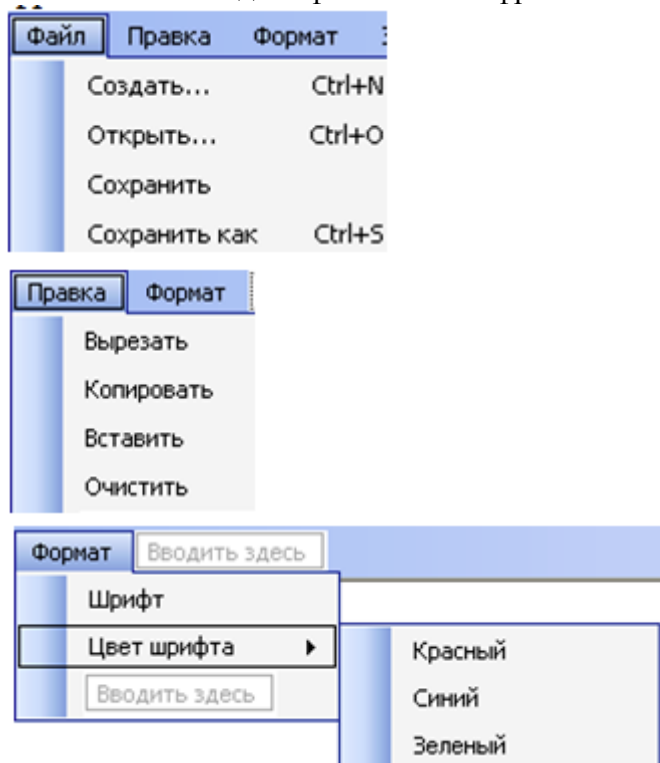
Вариант 3. "Текстовый редактор"

Разработайте проект – оконное приложение «Текстовый редактор», который должен иметь возможность работать с файлами формата `.rtf`.

Интерфейс текстового редактора должен содержать следующие элементы:

- Окно редактирования, заполняющего всю рабочую область формы для ввода, редактирования текста и форматирования символов шрифта.

- Оконное меню с основными пунктами «Файл», «Правка», «Формат» – опциями, позволяющими выполнять команды с файлом и его фрагментами.



- Команды «Создать...», «Открыть...», «Сохранить как...» открывают стандартный диалог для открытия/сохранения файла.

- Пункт меню «Правка» содержит команды: Вырезать, Копировать, Вставить, Выделить все, Очистить.

- Пункт меню «Формат» содержит команды: Шрифт, Цвет шрифта (Синий, Оранжевый, Серый).

- Команда «Шрифт» открывает стандартный диалог для выбора шрифта.

- Добавить в оконное меню пункт «Задача», с опцией «Решить» при активизации которой параметры шрифта текстового редактора менялись бы следующим образом: `Name=Arial Narrow, Size=16, Style=Bold`.

Задача	Вводить здесь
Решить	

Вариант 4. "Графический редактор"

Разработайте проект – оконное приложение «Графический редактор» для работы с растровыми изображениями.

В программе должны быть реализованы следующие функции:

- Открыть графический файл.
- Сохранить графический файл.
- Вывод на печать.
- Редактирование растрового изображения с помощью инструментов: Изменение цвета, Толщина линии, Линия, Круг.

Вариант 5. «Файловый менеджер»

Создать программу-оболочку для работы с файловой системой. Реализовать следующие функции: выбор диска, перемещение по иерархии каталогов диска, отображение списка файлов каталога, копирование, перемещение, удаление файлов, запуск программ.

9.2. Примерный перечень тем курсовой работы

Не предусмотрено учебным планом

9.3. Типовые контрольные задания для промежуточной аттестации: экзамен

Перечень теоретических вопросов к экзамену (Вопрос №1)

1. Определение алгоритма. Свойства алгоритма. Данные в алгоритмах. Формы представления алгоритмов. Критерии оценки алгоритма.
2. Базовые алгоритмические структуры. Правила построения блок схем.
3. Определение программы. Этапы решения задач. Определение языка программирования.
4. Разветвленная структура. Ветвление, полная и неполная формы.
5. Определение цикла. Виды циклов.
6. Общая характеристика платформы MSDN. Структура программы на языке программирования C#.
7. Язык C#, CLI, основы синтаксиса языка, ссылочные типы и типы-значения, преобразования типов, представление объектов в памяти.
8. Методы: способы передачи параметров, абстрактные, виртуальные и статические методы, модификаторы видимости.
9. Платформа .NET: общее описание, CLR, IL, CTS.
10. Сборки: понятие сборки, сильные и слабые имена, загрузка сборки. Понятие целевой платформы.
11. Классификация типов данных. Встроенные типы данных. Преобразование типов.
12. Основные операторы. Чтение данных с клавиатуры. Вывод данных на экран.
13. Массивы. Размерность массива. Методы и свойства класса Array.
14. Регулярные и буквальное строковые литералы. Ссылки типа string. Операции над строками.
15. Методы и свойства класса String. Форматирование строк. Неизменяемость объектов класса String. Тип System.Text.StringBuilder.
16. Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками. Режимы доступа к файлу, режимы открытия файла, режимы совместного использования файла.
17. Потоки байтов FileStream. Потоки символов StreamWriter и StreamReader.

18. Определение класса. Поля. Методы. Объекты. Конструкторы. Деструктор. Сбор мусора.
19. Ключевое слово this. Ключевое слово static. Статические конструкторы. Статические классы.
20. Перегрузка операций.
21. Основные свойства ООП. Инкапсуляция. Спецификаторы доступа. Свойства. Автоматические свойства. Индексаторы.
22. Основные свойства ООП. Иерархии классов. Наследование. Виртуальные методы. Механизм позднего связывания.
23. Абстрактные классы. Класс object. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET
24. Понятие делегата. Описание и использование.
25. События.
26. Механизм обработки исключений. Системные исключения и их обработка. Свойства и методы класса Exception. Исключения, создаваемые программистом.
27. Основные абстрактные структуры данных. Физическое представление данных.
28. Пространство имен System.Collections. Интерфейсы пространства имен System.Collections.
29. Коллекции общего назначения. Основные элементы класса ArrayList.
30. Коллекции общего назначения. Основные элементы класса Hashtable.
31. Коллекции общего назначения. Основные элементы класса SortedList.
32. Коллекции общего назначения. Основные элементы класса Stack.
33. Коллекции общего назначения. Основные элементы класса Queue.
34. Коллекции общего назначения. Доступ к коллекциям с помощью нумератора.
35. Классы-прототипы. Основные преимущества использования обобщений. Ограничения.
36. Параметризованные коллекции библиотеки .NET.
37. Исключения: бросание, перебрасывание и обработка, библиотечные исключения, свойства исключений, хорошие практики.
38. LINQ: основные методы, синтаксис, основные реализации.
39. Создание оконных приложений Windows Forms. Классы Application и Form. События в Windows-приложениях.
40. Основные свойства элементов управления.
41. Текстовые элементы управления.
42. Элементы управления списками.
43. Элементы управления для создания кнопок.
44. Табличное представление информации. Элемент DataGridView.
45. Общие элементы управления.
46. Формирование меню приложения.
47. Контейнеры, меню и панели инструментов.
48. Модальные и немодальные формы. Организация взаимодействия между формами.
49. Общие диалоговые окна.
50. Рисование графических примитивов.

Перечень практических заданий к экзамену (Вопрос №2)

1. Ввести целое число $N > 0$ и вещественное $a > 0$. Последовательность вещественных чисел определяется следующим образом $x_{n+1} = (x_n + a/x_n)/2$. Считая $x_0 = a$ вывести первые N членов последовательности. Такой способ применяли еще в древнем Вавилоне для вычисления квадратного корня числа a . После выдачи последовательности распечатать значение квадратного корня из a , вычисленное стандартной функцией. См. также: Квадратный корень, Итерационная формула Герона.
2. Ввести вещественное число X и целое число $N > 0$. Найти значение выражения $X - X^3/(3!) + X^5/(5!) - \dots + (-1)^N X^{2N+1}/((2N+1)!)$, которое является приближенным значением функции

\sin в точке X . Отобразить сумму ряда и рассчитанное с помощью функции \sin значения. См. также: Факториал, Ряд Тейлора.

3. Ввести целое число $N > 0$. Среди цифр этого числа выделить только чётные, из которых составить другое число и вывести. Например, при $N=3854972$ ответом будет число 842.

4. Ввести положительные числа A, B, C . На прямоугольнике размера $A \times B$ размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике. Операции умножения и деления не использовать.

5. Ввести натуральное (целое неотрицательное) число a и целое положительное число d . Вычислить частное q и остаток r при делении a на d , не используя операций целочисленного деления и нахождения остатка. Разрешается использовать только целые переменные и целочисленные операции.

6. В заданном одномерном массиве размером N поменять местами первый и последний положительные элементы.

7. Написать программу для определения индексов положительных элементов, заданного массива A , состоящего из 20-ти элементов.

8. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Удалить первую букву в каждом слове.

9. Дана строка символов, содержащая некоторый текст. Разработать программу, которая определяет, является ли данный текст палиндромом, т.е. читается ли он слева направо так же, как и справа налево (например, «А роза упала на лапу Азора»)

10. Написать программу, осуществляющую заданные вычисления. Вид используемых классов и методов определить самостоятельно. В конструкторе без параметров для инициализации полей использовать случайные числа. Заданы две матрицы $B(4,4)$ и $D(3,3)$. Написать программу транспонирования каждой из заданных матриц с последующим перемножением транспонированной матрицы на соответствующую ей исходную.

11. Написать программу, осуществляющую заданные вычисления. Вид используемых классов и методов определить самостоятельно. Заданы координаты четырёх точек A, B, C, D на плоскости. Определить наибольший из периметров треугольников ABC, ABD, ACD .

12. Построить иерархию классов: тест, экзамен, выпускной экзамен, испытание. Разработать методы и свойства для каждого из определяемых классов.

13. Разработать программу, динамически порождающую при щелчке на окне различные элементы (поля ввода, кнопки, метки). Тип элементов определяется с помощью радиокнопок. Все элементы располагаются горизонтально в ряд. При достижении правой границы окна начинается новый ряд элементов.

14. Написать программу для обработки строки: исходная строка делится на две отдельных строки (первая содержит исходную строку до первой точки, а вторая – исходную строку после первой точки). Разработать графический интерфейс для решения данной задачи.

15. Составить программу, генерирующую цвет по трем его составляющим (красный, зеленый, синий). Разработать графический интерфейс для решения данной задачи.

16. Создать проект, позволяющий сохранять произвольный текст в текстовом файле с указанным пользователем именем. Загружать текст из файла.

17. Проект содержит три текстовых окна. Используя буфер обмена, организовать обмен информацией между приложениями.

18. Вывести алфавит введенного пользователем текста.

19. Создать проект, позволяющий построить треугольник по координатам вершин. Разработать графический интерфейс для решения данной задачи.

20. Написать программу поиска номера первого из двух последовательных элементов в целочисленном массиве из 30 элементов, сумма которых максимальна. Если таких элементов несколько, то вывести номер первой пары.

38.03.05 Бизнес-информатика, направленность (профиль) "Цифровые решения для бизнеса"

Рабочая программа дисциплины

Дисциплина: Б1.О.28 Алгоритмизация и программирование

Форма обучения: очная, заочная, очно-заочная

Разработана для приема 2023/2024 учебного года

Раздел билета	Компетенции	Планируемые результаты обучения по дисциплине	Количество баллов
Вопрос №1 Теоретический вопрос (проверяет знания («знать»), сформированные дисциплиной)	ОПК-3	- общие принципы построения алгоритмов, основные алгоритмические конструкции; - современные программные средства для практической реализации программного обеспечения на персональных компьютерах; - основы программирования на языке высокого уровня объектно-ориентированной методологии С#; - основы функционирования программ, управляемых событиями; - основные принципы разработки пользовательского интерфейса.	60
Вопрос №2 Практическое задание (проверяет умения («уметь»), проверяет практические навыки («владеть»), сфор- мированные дисциплиной)	ОПК-3	- реализовывать алгоритмы решения задач на языке высокого уровня (С#); - разрабатывать программы, управляемые событиями; - разрабатывать программные приложения в среде визуального программирования с использованием технологий визуальной разработки интерфейсов (на примере среды Microsoft Visual Studio). - владеет практическими навыками составления алгоритмов для решения задач; реализации алгоритмов на языке программирования С#.	40