

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Райхерт Татьяна Николаевна
Должность: Директор
Дата подписания: 14.02.2022 09:24:59
Уникальный программный ключ:
c914df807d771447164c08ee17f8e2f93dde816b

Министерство науки и высшего образования Российской Федерации
Нижнетагильский государственный социально-педагогический институт (филиал)
Федерального государственного автономного образовательного учреждения
высшего образования
«Российский государственный профессионально-педагогический университет»

Факультет естествознания, математики и информатики
Кафедра информационных технологий



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ Б1.О.06.07 ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

Уровень высшего образования	Бакалавриат
Направление подготовки	44.03.05 Педагогическое образование (с двумя профилями подготовки)
Профили	«Математика и информатика»
Форма обучения	Очная

Нижний Тагил
2019

Рабочая программа дисциплины «Языки и методы программирования». Нижний Тагил: Нижнетагильский государственный социально-педагогический институт (филиал) ФГАОУ ВО «Российский государственный профессионально-педагогический университет», 2019. – 19 с.

Настоящая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки).

Автор: кандидат педагогических наук,
доцент кафедры информационных технологий

М. В. Мащенко

Рецензент: кандидат физ-мат. наук,
зам. директора по УР МБОУ СОШ №61

Л. М. Ставцева

Программа рассмотрена и одобрена на заседании кафедры информационных технологий 12 сентября 2019 г., протокол № 2.

Заведующая кафедрой

М. В. Мащенко

Программа рекомендована к печати методической комиссией факультета естествознания, математики и информатики 8 октября 2019 г., протокол № 1.

Председатель МК ФЕМИ

В. А. Гордеева

Декан ФЕМИ

Т. В. Жуйкова

Главный специалист ОИР

О. В. Левинских

© Нижнетагильский государственный социально-педагогический институт (филиал) ФГАОУ ВО «Российский государственный профессионально-педагогический университет», 2019.
© Мащенко Майя Владимировна, 2019.

СОДЕРЖАНИЕ

1. Цель и задачи освоения дисциплины	4
2. Место дисциплины в структуре образовательной программы	4
3. Результаты освоения дисциплины	4
4. Структура и содержание дисциплины	5
4.1. Объем дисциплины и виды контактной и самостоятельной работы	5
4.3. Содержание тем дисциплины	7
5. Образовательные технологии	9
6. Учебно-методические материалы	10
6.1. Планирование самостоятельной работы	10
6.2. Задания и методические указания по организации самостоятельной работы	11
7. Текущий контроль качества усвоения знаний	14
8. Промежуточная аттестация	15
9. Учебно-методическое и информационное обеспечение	17

1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель дисциплины – формирование профессиональных и общеобразовательных компетенций будущих учителей информатики в области разработки программных продуктов с использованием различных структур данных, методов построения корректных и эффективных алгоритмов в современных средах программирования.

Задачи:

- формирование у студентов знаний о существующих парадигмах и современных технологиях программирования, области их применения, достоинствах и недостатках и методологии создания программных средств;
- формирование знаний о сущности и возможности применения объектно-ориентированного подхода к программированию (принципы создания классов и их применения в решении предметно-ориентированных задач);
- формирование у студентов знаний о жизненном цикле программного продукта, содержании различных этапов проектирования, правилах оформления документации на программные средства, методах их тестирования и отладки;
- практическое освоение студентами современной среды программирования и реализации базовых алгоритмических конструкций на одном из современных языков программирования;
- развитие умений проектирования, конструирования и отлаживания студентами программных продуктов в соответствии с заданными критериями;
- создание студентами различных видов пользовательского интерфейса для разработанных программных продуктов;
- систематизация полученных знаний и умений в области программирования для использования их при реализации образовательной программы по информатике.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина включена в Блок Б.1 «Дисциплины (модули)» и является составной частью раздела Б1.В «Вариативная часть» модуля «Дисциплины профильной подготовки», установлена вузом, обязательная для изучения.

3. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование и развитие следующих компетенций, согласно которым выпускник обладает:

УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.

ПК-3 Способен применять предметные знания при реализации образовательного процесса.

ПК-6 Способен формировать у обучающихся умения моделировать объекты и процессы окружающей реальности и пользоваться заданной математической или информационной моделью.

ПК-7 Способен формировать у обучающихся конкретные знания, умения и навыки в области физики и информатики.

В результате изучения данной дисциплины студент должен **знать:**

□ понятия алгоритма, исполнителя, программы, языка программирования, среды программирования, транслятора, основных алгоритмических конструкций; □ жизненный цикл программного приложения и этапы его разработки;

□ современные подходы к программированию, существующие парадигмы и технологии программирования, а также перспективы их развития;

□ объектно-ориентированный подход к программированию и возможности его реализации на одном из современных высокоуровневых языков программирования;

□ типы данных и способы задания структур данных на одном из современных языков программирования;

□ способы реализации основных алгоритмических конструкций, функций, процедур, модулей, классов на одном из современных языков программирования;
 □ понятие рекурсии и возможности ее использования при написании программ;
 □ методы сортировки и поиска данных;
 □ методы анализа и тестирования программ; □ базовые алгоритмы и стратегии решения прикладных задач в области информационно-коммуникационных технологий;

□ методы разработки пользовательского интерфейса для программных продуктов;

уметь:

□ анализировать предметную область и формализовать условия прикладных задач;

- разрабатывать эффективные алгоритмы и программы с использованием современных технологий программирования;
- использовать методологию программирования для решения практических задач обработки, представления, хранения и передачи информации;
- осуществлять тестирование, отладку программ их сопровождение, оформлять программную документацию;
- решать типовые задачи проектирования и разработки всех видов интерфейса с использованием современных технологий программирования;

□ создавать различные прикладные приложения, в том числе и для образования;

владеть:

□ технологией объектно-ориентированного программирования; □ способами тестирования и отладки программных приложений;

□ методами разработки пользовательских приложений.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Объем дисциплины и виды контактной и самостоятельной работы

Общая трудоемкость дисциплины составляет 9 зач. ед. (324 часа), их распределение по видам работ представлено в таблице.

Распределение трудоемкости дисциплины по видам работ

Вид работы	Кол-во часов
Общая трудоемкость дисциплины по учебному плану	324
Контактная работа, в том числе:	114
Лекции	52
Лабораторные занятия	110
Самостоятельная работа, в том числе:	210
Самоподготовка к текущему контролю знаний	183
Подготовка к зачету, экзамену	27

4.2. Содержание и тематическое планирование дисциплины

№	Название темы	Распределение часов	
		Трудоемкость	

			Ауд. занят	Виды занятий		Сам. работа
				Лекции	Лаборатор. работы	
1 семестр						
Всего в 1 семестре в часах		144	40	12	28	104
2 семестр						
1	Сравнительный анализ существующих парадигм и технологий программирования	6	2	2		4
2	Эволюция современных языков и сред программирования	6	2	2		4
3	Объектно-ориентированный подход в программировании и его реализация на одном из современных языков программирования.	10	6	2	4	4
4	Простейшие программы с экранной формой и элементами управления в среде программирования	10	6	2	4	4
5	Обработка событий мыши и клавиатуры в среде программирования	14	8	2	6	6
6	Класс Graphics. Работа с графическими объектами	12	6	2	4	6
7	Реализация основных алгоритмических конструкций, процедур и функций в среде программирования	23	14	4	10	9
Зачет		9	0	0	0	9
Всего во 2 семестре в часах		108	38	14	24	70
3 семестр						
8	Тестирование программных средств	18	6	2	4	12
9	Использование различных структур данных при программировании	43	20	8	12	23
10	Жизненный цикл и документирование программных средств	16	4	2	2	12
11	Построение алгоритмов с использованием рекурсии	22	10	4	6	12
Зачет с оценкой		9	0	0	0	9
Всего во 3 семестре в часах		108	40	12	28	68
4 семестр						
№	Название темы	Распределение часов				

		Трудоемкость	Ауд. занят	Виды занятий		Сам. работа
				Лекции	Лаборатор. работы	
12	Реализация поиска и сортировки данных в среде программирования	12	8	4	4	4
13	Создание прикладных Windows приложений в среде программирования	16	12	4	8	4
14	Создание различных видов интерфейса пользователя на одном из современных высокоуровневых языков программирования.	14	8	2	6	6
15	Использование объектноориентированного языка программирования для работы с базами данных.	12	8	2	6	4
Экзамен		36	0	0	0	36
Всего во 4 семестре в часах		108	44	14	30	64
Итого в часах		468	162	52	110	306

4.3. Содержание тем дисциплины

Тема 1. Сравнительный анализ существующих парадигм и технологий программирования. Понятие парадигмы программирования. Сравнительная характеристика декларативного и императивного программирования; логической, функциональной, структурной, объектно-ориентированной и параллельной парадигм программирования: достоинства и недостатки, область применения. Понятие технологии программирования. Технология программирования и информатизация общества. Современные технологии программирования (объектно-ориентированное, модульное, компонентно-ориентированное) и область их применения. Новые направления развития программирования.

Тема 2. Структурный подход к программированию: реализация основных алгоритмических конструкций.

Тема 3. Структурный подход к программированию: структуры данных (строки и массивы).

Тема 4. Структурный подход к программированию: структуры данных (структуры, коллекции и файлы).

Тема 2. Эволюция современных языков и сред программирования. Понятие языка программирования. Синтаксис, семантика и прагматика. Среда программирования и ее составляющие. Интерпретатор и компилятор языка программирования. История развития языков программирования. Классификация современных языков программирования и их сравнительная характеристика. Современные интегрированные среды разработки программ. Ввод и вывод в консольном приложении. Компиляция кода. Настройка среды. Управление исходным кодом. Средства отладки.

Тема 3. Объектно-ориентированный подход в программировании и его реализация на одном из современных языков программирования. Сущность объектно-ориентированного подхода. Объектный тип данных. Переменные объектного типа. Классы и объекты. Классификация объектов. Свойства. Статические свойства, поля и методы.

Ограничение доступа к полям класса с помощью свойств. Методы и события. События в среде Windows. Объявление класса. Создание экземпляра класса. Описание классов. Пространства имен. Конструкторы и деструкторы. Виды конструкторов. Свойства классов: наследование, инкапсуляция, полиморфизм. Перегрузка и переопределение методов на основании наследования.

Тема 4. Простейшие программы с экранной формой и элементами управления в среде программирования. Особенности использования классов объектов в среде объектно-ориентированного программирования. Архитектурные шаблоны. Иерархия пространства имен. Базовые элементы управления метка, текстовое поле, кнопка, флажок, переключатель, списки и их применение. Процедуры обработки событий. Элементы управления для ввода и вывода информации. Субклассирование. Формы и потоки.

Тема 5. Обработка событий мыши и клавиатуры в среде программирования. Инициирование события в создаваемом классе. Фокус элементов управления. Основные события, связанные с мышью. Координаты курсора мыши относительно экрана и элемента управления. Ассоциация нескольких элементов управления одним событием. Работа с клавиатурой. Контроль вводимых с клавиатуры данных.

Тема 6. Класс Graphics. Работа с графическими объектами. Видимые и невидимые графические объекты. Графический объект – точка и обращение к ней. Рисование линий, дуг и примитивов. Преобразование координат и управление ими. Собственные перья, кисти и шрифты. Работа с цветом, настройка заливки, прозрачность. Работа с готовыми изображениями. Встроенный графический редактор. Сложные фигуры и графический путь. Фигурные формы.

Тема 7. Реализация основных алгоритмических конструкций, процедур и функций в среде программирования. Алфавит и синтаксис языка программирования: структура программы, модуля, типы данных, область видимости переменных. Константы. Основные операторы. Управляющие конструкции и их реализация в системе программирования (ветвление, оператор выбора, циклы, функции и процедуры). Управление событиями с помощью процедур, функций и модулей. Встроенные функции.

Тема 8. Тестирование программных средств. Компиляция кода. Принципы и виды отладки. Средства отладки. Настройка среды. Управление исходным кодом. Структура обработки исключений. Обработка ошибок. Автономная отладка модуля. Комплексная отладка программного средства.

Тема 9. Использование различных структур данных при программировании. Понятие массива. Классификация массивов. Способы реализации массивов на языке программирования. Работа с динамическими массивами. Применение строк в системе программирования. Пользовательский тип данных. Работа с памятью компьютера. Работа с файлами в системе программирования. Файловый ввод/вывод. Чтение и запись файлов. Файловые потоки. Наблюдение за файловой системой. Организация печати. Работа с XML-документами.

Тема 10. Жизненный цикл и документирование программных средств. Специфика разработки программных средств. Жизненный цикл программного продукта. Классические технологические процессы. Основные, вспомогательные, организационные процессы и взаимосвязь между ними. Фазы. Основные подходы к описанию жизненного цикла программного продукта. Технологии коллективной разработки. Проект и программа. Документооборот в жизненном цикле проектов программных средств. Проблемы организации документирования программных средств и формирование требований к документации. Стандарты, регламентирующие документирование проектов и эксплуатацию программного средства. Назначение внешнего описания программного средства и его роль в обеспечении качества программного средства. Методы контроля внешнего описания программного средства. Документация, создаваемая в процессе разработки программных средств. Пользовательская документация по сопровождению программных средств.

Тема 11. Построение алгоритмов с использованием рекурсии. Понятие рекурсии и ее назначение. Виды рекурсии. Прямая и косвенная рекурсия. Общие принципы программной реализации рекурсии. Глубина рекурсии и общее количество рекурсивных вызовов. Использование рекурсии для расчетов и построения сложных ломаных линий.

Тема 12. Реализация поиска и сортировки данных в среде программирования. Поиск информации в файлах и массивах. Основные понятия. Методы поиска элементов в массиве (линейный поиск, линейный поиск с барьером, бинарный поиск). Слияние массивов и файлов. Методы сортировки данных (метод простого выбора, метод простого обмена, метод прямого включения, метод слияний, быстрая сортировка, поразрядная сортировка и др.).

Тема 13. Создание прикладных Windows приложений в среде программирования. Понятие прикладного приложения. Использование программирования, ориентированного на события для создания приложений. Понятие компонента. Проектирование и использование компонентов. Особенности использования среды визуального программирования. Реализация концепции визуального программирования. Макросы и шаблоны. Алгоритмы стандартной библиотеки. Программирование с использованием стандартной библиотеки. Разработка справочной системы.

Тема 14. Создание различных видов интерфейса пользователя на одном из современных высокоуровневых языков программирования. Понятие интерфейса. Особенности пользовательского интерфейса. Виды интерфейсов (командный, графический, звуковой) и способы их создания (MDI-технология, SDI-технология). Элементы интерфейса: окно, форма, меню, панель (лента). Использование библиотеки графических объектов для создания интерфейса. Создание MDI и SDI интерфейсов: разработка меню, панели инструментов, стандартных диалоговых окон, строки состояния, контекстного меню. Технология WPF и ее использование для создания пользовательских форм.

Тема 15. Использование объектно-ориентированного языка программирования для работы с базами данных. Понятие базы данных. Принципы работы поставщиков данных. Назначение технологии ADO. Использование технологии ADO для организации доступа к данным. Создание пользовательских форм и запросов. Разработка приложений с использованием технологии ADO.

Тема 17. Основы офисного программирования

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Данный курс предусматривает наличие теоретических лекционных занятий, на которых студенты знакомятся с общими современными направлениями в программировании, и практических – на которых осваивается конкретный язык программирования и соответствующие ему технологии.

Основными методами, используемыми при объяснении теоретического материала, будут: проблемное изложение; метод демонстрационных примеров, кейс-метод (с использованием элементов технологии опережающего развития); дискуссия (прессконференция; форум). Для организации практических занятий основным методом будет – лабораторная работа, подразумевающая исследовательский характер и практикумы. В конце каждой темы планируется для обобщения материала проведение деловой игры.

6. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

6.1. Планирование самостоятельной работы

Проверка качества усвоения знаний по дисциплине ведется в течение семестра в устной форме (интерактивная форма обучения (см. выше) во время лекционных и семинарских занятий) и в письменной (тестовые контрольные работы по укрупненным темам и письменные опросы на занятиях) форме. По результатам работы во время занятий студенты могут набрать определенное количество баллов и получить зачет автоматом. Формы и вопросы текущего контроля утверждаются на кафедре.

Темы занятий	Количество часов			Содержание самостоятельной работы студентов	Формы контроля СРС
	Всего	Аудиторны х	Самостоят.		
1. Сравнительный анализ существующих парадигм и технологий программирования	6	2	4	Составление сравнительной таблицы и тезисов для обсуждения на форуме.	Проверка преподавателем таблицы, обсуждение на форуме
2. Эволюция современных языков и сред программирования	6	2	4	Подготовка доклада или презентации	Проверка содержания докладов презентаций преподавателем
3. Объектноориентированный подход в программировании и его реализация на одном из современных языков программирования	10	6	4	Составление глоссария по теме, отчет по лабораторной работе №1	Проверка глоссария и отчета преподавателем
4. Простейшие программы с экранной формой и элементами управления в среде программирования	10	6	4	Краткий конспект лекции, отчет по лабораторной работе №2	Проверка конспекта и отчета преподавателем
5. Обработка событий мыши и клавиатуры в среде программирования	14	8	6	Краткий конспект лекции, подготовка индивидуального проекта, отчет по лабораторной работе №3	Проверка конспекта и отчета преподавателем, защита проекта
6. Класс Graphics. Работа с графическими объектами	12	6	6		
7. Реализация основных алгоритмических конструкций, процедур и функций в среде программирования	23	14	9	Выполнение задач для самостоятельной работы	Проверка задач преподавателем
Темы занятий	Количество часов			Содержание самостоятельной работы студентов	Формы контроля СРС
	Всего	Аудиторны х	Самостоят.		

8. Тестирование программных средств	18	6	12	Краткий конспект лекции, отчет по лабораторной работе №4	Проверка конспекта и отчета преподавателем
9. Использование различных структур данных при программировании	43	20	23	Выполнение задач для самостоятельной работы, создание теста по заданной теме	Проверка задач преподавателем
10. Жизненный цикл и документирование программных средств	16	4	12	Составление глоссария и аннотированного списка стандартов, отчет по лабораторной работе №5	Проверка преподавателем глоссария и списка, обсуждение на форуме
11. Построение алгоритмов с использованием рекурсии в среде программирования	22	10	12	Краткий конспект лекции, выполнение задач для самостоятельной работы	Проверка конспекта и задач преподавателем
12. Реализация поиска и сортировки данных в среде программирования	12	8	4		
13. Создание прикладных Windows приложений в среде программирования	16	12	4	Выполнение группового проекта	Защита проекта
14. Создание различных видов интерфейса пользователя	14	8	6		
15. Использование объектно-ориентированного языка программирования для работы с базами данных	12	8	4	Выполнение задач для самостоятельной работы	Проверка задач преподавателем
Зачеты в 2, 3 семестрах	18		18	Подготовка к зачету	Защита проекта
Экзамен в 4 семестре	36		36	Подготовка к экзамену	Собеседование
Итого	468	162	306		

6.2. Задания и методические указания по организации самостоятельной работы

Вопросы для отчета по лабораторной работе 1.

1. Перечислите основные панели инструментов, предназначенные для работы в среде Visual Studio 2010. Расскажите об их элементах.
2. Из каких частей состоит проект, созданный в среде Visual Studio 2010 на языке Visual Basic Net. В какие файлы проект компилируется при сохранении. Какие виды проектов бывают?
3. Какие ограничения доступа к полям класса с помощью свойств можно установить в Visual Studio 2010?

4. Опишите пространство имен в Visual Studio 2010. Перечислите известные вам типы данных, используемые в языке Visual Basic Net. Расскажите об известных вам функциях преобразования из одного типа данных в другой и их особенностях.

5. Каким образом на языке Visual Basic Net создаются классы? Какова структура конструктора и деструктора.

6. Каким образом в Visual Studio 2010 выполняется перегрузка и переопределение методов на основании наследования?

Вопросы для отчета по лабораторной работе 2.

1. Что такое класс Form (форма)? Перечислите основные его свойства, методы, события и возможности настройки.

2. Расскажите о панели инструментов Component Palette. Опишите базовые элементы управления и их свойства (метка, текстовое поле, кнопка, флажок, переключатель, списки).

3. Расскажите о MDI-формах (главных и дочерних), настройки принадлежности окон.

4. Что такое субклассирование?

5. Перечислите основные свойства для объекта Application.

6. Реализация форм и потоков данных в Visual Studio 2010.

Тематика индивидуальных проектов

1. Создайте элементы клавиатурного тренажера для одного ряда букв на клавиатуре нестандартной формы.

2. Создайте тренажер мыши нестандартной формы.

3. Создайте активную карту для квеста.

4. Создайте тренажер использования интерфейса Windows (разных видов меню).

5. Создайте тренажер использования интерфейса Windows (разных видов окон).

Вопросы для отчета по лабораторной работе 3.

1. Каким образом класс Graphics позволяет преобразовывать координаты точек и осуществлять управление ими?

2. Какие есть методы создания собственных перьев, кистей и шрифтов?

3. Каким образом можно осуществить настройку заливки, установить прозрачность?

4. Опишите основные функции встроенного графического редактора VB.

5. Каким образом в данном редакторе можно создавать формы нестандартной формы?

Примерные задачи для самостоятельной работы по теме 7 1.

Составить программу, которая находит сумму последних трех цифр в пятизначном числе.

2. Составить программу, которая переводит из одних единиц измерения информации в другие.

3. Составить программу, которая определяет максимальную скорость передачи сообщения при разных параметрах. Определите входные и выходные параметры для этой задачи.

4. Составьте программу, позволяющую решать квадратное уравнение. Дополните программу графической иллюстрацией.

5. Составьте программу, которая позволяет создать анимацию круга заданного радиуса с задаваемой пользователем скоростью по диагонали формы.

Вопросы для отчета по лабораторной работе 4.

1. Какие виды отладки программ существуют в Visual Studio 2010?
2. Перечислите средства отладки в Visual Studio 2010.
3. Каковы основные функции объектов Debug и Trace?
4. Каким образом в Visual Studio 2010 осуществляется управление исходным кодом?
5. Что из себя представляет структура обработки исключений в Visual Studio 2010?
6. Как работать с исключениями Try...Catch...Finally?
7. Как осуществляется обработка ошибок в Visual Studio 2010 на основе инструкции On Error?
8. Каким образом можно осуществить автономную отладку модуля?

Примерные задачи для самостоятельной работы по теме 9

1. Составьте программу, которая позволяет список учащихся сортировать по итоговым оценкам по предмету, искать ученика по классу, записывать все данные в текстовый файл.
2. Составьте программу, которая считывает из текстового файла вопросы и ответы для формирования закрытого теста.
3. Составьте программу, которая позволяет создать XML-документ, содержащий информацию о достижениях учащегося за год по всем предметам и внеучебной работе, а затем обновлять в нем данные.

Вопросы для отчета по лабораторной работе 5.

1. Перечислите основные, вспомогательные, организационные процессы жизненного цикла программного продукта.
2. Какие можно выделить фазы в жизненном цикле?
3. Каковы основные подходы к описанию жизненного цикла программного продукта?
4. Назовите основные проблемы организации документирования программных средств и формирование требований к документации?
5. Опишите стандарты, регламентирующие документирование проектов и эксплуатацию программного средства на рассмотренном вами примере.

Примерная тематика групповых проектов

1. Разработка электронного календаря-еженедельника, позволяющего планировать текущие дела и напоминать о них согласно текущей дате.
2. Разработка текстового редактора.
3. Разработка графического редактора.
4. Разработка мультимедиа плеера, поддерживающего различные форматы файлов.
5. Разработка математического редактора для работы с квадратными матрицами.
6. Разработка математического редактора для работы с комплексными числами.
7. Разработка игры «Память».
8. Разработка игры «Пятнашки».
9. Разработка игры «Пазлы»
10. Разработка игры «Крестики-нолики»
11. Разработка игры «Морской бой».

Требования к групповому проекту

Групповой проект должен представлять собой законченный программный продукт, снабженный всеми известными компонентами графического интерфейса, выполненными в едином стиле.

Примерные задачи для самостоятельной работы по теме 15

1. Создать указанную базу данных, заполнить ее не менее чем 5-ю записями. Создать указанный запрос. Сформировать формы для заполнения таблиц и просмотра запросов, представлений с русифицированными кнопками, наименованиями полей. Сформировать отчет с детальными строками, с расшифровками кодов и с итогами по двум уровням группировки (указаны в условии), по одному простому отчету для каждой таблицы, запроса. Сформировать кнопочную форму из пунктов: таблицы, формы, представления, запросы и отчеты для вызова разработанных таблиц, форм, представлений, запросов и отчетов.

Учет книг в библиотеке. Книжный фонд: код тематики, наименование, авторы, дата, число экземпляров, адрес хранения, код издательства, код отдела библиотеки.

Справочники: тематика (код, наименование), издательства (код, наименование, адрес, телефон, код города), отделы библиотеки (код и наименование), города (код и наименование).

Отчет по отделам и тематикам с итогами (число наименований и экземпляров книг) по отделам и тематикам.

7. ТЕКУЩИЙ КОНТРОЛЬ КАЧЕСТВА УСВОЕНИЯ ЗНАНИЙ

Текущий контроль сводится к систематической оценке аудиторной и самостоятельной работы студентов, в процессе которой выявляется полнота выполнения предлагаемых заданий, а также степень владения материалом.

Компоненты оценивания краткого конспекта (отчета по лабораторной работе)

Структурные компоненты краткого конспекта	Критерии оценки элементов
1. Глоссарий	☐ полнота
2. Описание области применения используемых понятий с примерами из практической деятельности	☐ логичность ☐ адекватность полнота ☐ ☐ наглядность (наличие схем, рисунков)
3. Аннотируемый список использованных источников	☐ полнота ☐ правильность оформления

Критерии оценки решения задачи для самостоятельной работы

- результативность;
- оптимальность используемых алгоритмов;
- оптимальность использования базовых элементов;
- наличие необходимого интерфейса пользователя.

Критерии оценки проекта

- ☐ соответствие предложенной тематике;
- ☐ обоснованность структуры и выбранных форм; ☐ полнота;
- ☐ результативность;
- ☐ оригинальность;
- ☐ оптимальность используемых алгоритмов; ☐ оптимальность использования базовых элементов; ☐ удобство интерфейса пользователя.
- ☐ наглядность представления проекта;
- ☐ убедительность при обосновании и защите проекта; ☐ уровень владения материалом;

□ умение вести дискуссию.

8. ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Промежуточная аттестация студентов – дифференцированный зачет в 2,3 семестрах и экзамен в 4 семестре, на которых предполагается проверка основных требований к освоению дисциплины.

Зачет выставляется при успешном освоении всех дидактических единиц, запланированных для изучения во каждом из семестров, выполнения необходимого количества самостоятельных заданий, в том числе и индивидуального проекта.

Экзаменационный билет предполагает наличие теоретического вопроса и практического задания.

Перечень теоретических вопросов для экзамена

1. Понятие языка программирования (синтаксис, семантика, прагматика) и история развития.
2. Современные системы программирования и их многообразие.
3. Парадигма программирования. Классификация. Сравнительная характеристика.
4. Технология программирования. Классификация. Сравнительная характеристика.
5. Современные технологии программирования и новые направления развития программирования.
6. Сущность объектно-ориентированного подхода. Объектный тип данных. Классификация объектов.
7. Классы и объекты. Объявление класса. Создание экземпляра класса. Свойства. Статические свойства, поля и методы. Ограничение доступа к полям класса с помощью свойств.
8. Методы и события. События в среде Windows. Процедуры обработки событий.
9. Описание классов. Конструкторы и деструкторы. Виды конструкторов.
10. Свойства классов: наследование, инкапсуляция, полиморфизм.
11. Алфавит и синтаксис языка Visual Basic Net: структура программы, модуля, типы данных область видимости переменных. Константы. Основные операторы.
12. Управляющие конструкции и их реализация в системе программирования Visual Basic Net (ветвление, оператор выбора).
13. , Управляющие конструкции и их реализация в системе программирования Visual Basic Net (циклы).
14. Управляющие конструкции и их реализация в системе программирования Visual Basic Net (функции и процедуры).
15. Управление событиями с помощью процедур, функций и модулей. Встроенные функции. Функция MsgBox и InputBox.
16. Компиляция кода в системе программирования Visual Basic Net. Средства отладки. Управление исходным кодом. Обработка ошибок и исключений.
17. Разработка форм. Класс Form.
18. Использование графики. Класс e.Graphics.
19. Базовые элементы управления: метка, текстовое поле, кнопка, флажок, переключатель, списки.
20. Классификация массивов. Способы реализации массивов на языке Visual Basic Net. Класс Array.
21. Применение строк в системе программирования Visual Basic Net.
22. Пользовательский тип данных.
23. Работа с файлами в системе программирования .
24. Работа с XML-документами.

25. Понятие компонента. Проектирование и использование компонентов.
26. Макросы и шаблоны в среде Visual Studio 2010.
27. Алгоритмы стандартной библиотеки. Программирование с использованием стандартной библиотеки.
28. Понятие рекурсии и ее реализация на языке Visual Basic Net.
29. Жизненный цикл программного продукта. Способы описания жизненного цикла программного продукта.
30. Понятие интерфейса. Виды интерфейсов (командный, графический, звуковой) и способы их создания (MDI-технология, SDI-технология).
31. Технология WPF и ее использование для создания пользовательских форм.
32. Назначение технологии ADO. Использование технологии ADO для организации доступа к данным.

Критерии оценки ответа на вопрос

- ☐ полнота;
- ☐ фактическое знание материала;
- ☐ логичность;
- ☐ наличие практических примеров; ☐ умение поддержать диалог.

Типовые экзаменационные задачи

1. Составить программу, которая в зависимости от введенного языка выдает на нем приветствие. При этом цвет текста меняется в зависимости от действий мыши (щелчка, наведения, потери фокуса и т.п.).
2. Создать программу, позволяющую вычислить значение функции и построить ее график $y = 5x, x \in [0, 2]$
 $y = 4x^2, x \in [0, 2]$
3. Создать программу-тест из 15 вопросов на основе мультимедийной формы, с возможностью выбора варианта и вычисления результата тестирования. Время тестирования ограничено таймером (5 мин).
4. Создать программу-тест на сопоставление из 5 вопросов, позволяющую переносить варианты ответов мышью.
5. Создать программу, рисующую кривые Гильберта
6. Создать средствами языка программирования однотабличную базу данных с возможностью поиска данных.
7. Создать программу, позволяющую из текстового файла импортировать данные, содержащие в каждой строке данные о фамилии, имени и возрасте человека, разделенные пробелом. Замените все имена инициалами и создайте возможность поиска самого старшего человека.
8. Создать программу, позволяющую в зависимости от выбора пользователя рисовать необходимую фигуру, заданных размеров и нужного цвета.
9. Создать программу, имитирующую будильник.
10. Составить программу, которая строку с целым числом в двоичной системе счисления, переводит в десятичный код.
11. Составить программу, которая в данной матрице 5×5 , заданной случайным образом в промежутке от -2 до 2, вычисляет сумму в каждой строке и считает количество положительных чисел.

Критерии оценивания знаний и умений на зачете

Оценка "**Отлично**" выставляется студентам, показавшим глубокое знание теоретических вопросов курса, умение проиллюстрировать изложение практическими

примерами, полно и подробно ответившим на вопрос билета и дополнительные вопросы преподавателя, а также выполнившим практическое задание.

Оценка **"Хорошо"** выставляется студентам, сдавшим экзамен с незначительными замечаниями, показавшим глубокое знание теоретических вопросов, умение проиллюстрировать изложение практическими примерами, полностью ответившим на вопрос билета и дополнительные вопросы преподавателя и выполнившим практическое задание, но допустившим при ответах незначительные ошибки, указывающие на наличие некоторых (несущественных) пробелов в знаниях.

Оценка **"Удовлетворительно"** выставляется студентам, сдавшим экзамен со значительными замечаниями, показавшим знание основных положений теории при наличии существенных пробелов в деталях, испытывающим затруднения в практическом применении теории, допустившим существенные ошибки при ответе на вопрос билета и дополнительные вопросы преподавателя.

Оценка **"Неудовлетворительно"** выставляется, если студент показал существенные пробелы в знаниях основных положений теории, не умеет применять теоретические знания на практике, не ответил на вопрос билета, не выполнил практическое задание.

Критерии оценивания ответов на экзамене

Оценка **"Отлично"** выставляется студентам, показавшим глубокое знание теоретических вопросов курса, умение проиллюстрировать изложение практическими примерами, полно и подробно ответившим на вопрос билета и дополнительные вопросы преподавателя, а также выполнившим практическое задание.

Оценка **"Хорошо"** выставляется студентам, сдавшим экзамен с незначительными замечаниями, показавшим глубокое знание теоретических вопросов, умение проиллюстрировать изложение практическими примерами, полностью ответившим на вопрос билета и дополнительные вопросы преподавателя и выполнившим практическое задание, но допустившим при ответах незначительные ошибки, указывающие на наличие некоторых (несущественных) пробелов в знаниях.

Оценка **"Удовлетворительно"** выставляется студентам, сдавшим экзамен со значительными замечаниями, показавшим знание основных положений теории при наличии существенных пробелов в деталях, испытывающим затруднения в практическом применении теории, допустившим существенные ошибки при ответе на вопрос билета и дополнительные вопросы преподавателя.

Оценка **"Неудовлетворительно"** выставляется, если студент показал существенные пробелы в знаниях основных положений теории, не умеет применять теоретические знания на практике, не ответил на вопрос билета, не выполнил практическое задание.

9. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Основная литература

1. Давыдова, Н.А. Программирование : учебное пособие. [Электронный ресурс]: Учебные пособия / Н.А. Давыдова, Е.В. Боровская. М.: Издательство «Лаборатория знаний», 2012. — 238 с. Режим доступа: <http://e.lanbook.com/book/8764> — Загл. с экрана.
2. Культин Н. Б. Visual C++ в задачах и примерах [Текст] / Н. Культин. – СПб.: БХВ-Петербург, 2011. - 264 с. (10 экз).
3. Орлов С. Теория и практика языков программирования. Учебник для вузов. Стандарт 3-го поколения. СПб.: Питер, 2014. [Электронный ресурс]. Режим доступа:
4. Тюгашев А. Языки программирования: Учебное пособие. Стандарт третьего поколения. СПб.: Питер, 2014. [Электронный ресурс]. Режим доступа:

Дополнительная литература

1. Бабушкина, И.А. Практикум по объектно-ориентированному программированию [Электронный ресурс]: учебное пособие / И.А. Бабушкина, С.М. Окулов. М.: Издательство

"Лаборатория знаний", 2015. — 369 с. Режим доступа: <https://e.lanbook.com/book/66121>

2. Давыдова, Н.А. Программирование [Электронный ресурс] : учебное пособие /

Н.А. Давыдова, Е.В. Боровская. – М. : Издательство "Лаборатория знаний", 2015. — 241 с. — Режим доступа: <https://e.lanbook.com/book/66124>. — Загл. с экрана. Режим доступа: <https://e.lanbook.com/book/66124>

3. Златопольский, Д.М. Основы программирования на языке Python [Электронный ресурс]: учебник / Д.М. Златопольский. – М.: ДМК Пресс, 2017. — 284 с. — Режим доступа: <https://e.lanbook.com/book/97359>. — Загл. с экрана. Режим доступа: <https://e.lanbook.com/book/97359>

4. Москвитина, О.А. Сборник примеров и задач по программированию [Электронный ресурс] : учебное пособие / О.А. Москвитина, В.С. Новичков, А.Н. Пылькин. – М.: Горячая линия-Телеком, 2014. — 245 с. Режим доступа: <https://e.lanbook.com/book/64090>

Интернет-ресурсы

1. INTUIT.ru: Учебный курс - Работа в Microsoft Visual Studio [Электронный ресурс]. URL: <http://www.intuit.ru/department/itmngt/workinmsvistudio/> (дата обращения: 2013 г.).

2. INTUIT.ru: Учебный курс – Visual Basic. Net [Электронный ресурс]. URL: <http://www.intuit.ru/department/pl/vbnet/> (дата обращения: 2013 г.).

3. Microsoft Visual Studio 2010 — официальный сайт Visual Studio [Электронный ресурс]. URL: <http://www.microsoft.com/visualstudio/ru-ru/> (дата обращения: 2013 г.).

4. Библиотека полнотекстовых учебников и учебных пособий по гуманитарноэкономическим и техническим дисциплинам [Электронный ресурс]. URL: <http://window.edu.ru/window/library> (дата обращения: 2013 г.).

5. Вирт Н. Алгоритмы и структуры данных [Электронный ресурс]. М.: ДМК Пресс, 2010. 272 с. URL: <http://www.biblioclub.ru/book/86483/> (дата обращения: 2013 г.).

6. Кауфман В. Ш. Языки программирования. Концепции и принципы [Электронный ресурс]. М.: ДМК Пресс, 2011. 464 с. URL: <http://www.biblioclub.ru/book/86526/> (дата обращения: 2013 г.).

7. Ковалевская Е. В., Комлева Н. В. Методы программирования. Учебнометодический комплекс [Электронный ресурс]. М.: Евразийский открытый институт, 2011. 319 с. URL: <http://www.biblioclub.ru/book/90390/> (дата обращения: 2013 г.).

8. Курс технологии программирования [Электронный ресурс]. URL: <http://sp.cmc.msu.ru/info/3/techprog.htm> (дата обращения: 2013 г.).

9. Поддержка Visual Studio и устранение неполадок [Электронный ресурс]. URL: <http://msdn.microsoft.com/ru-ru/vstudio/cc136615> (дата обращения: 2013 г.).

10. Самоучитель по VB.NET [Электронный ресурс]. URL: <http://starcats.ru/vbnet/> (дата обращения: 2013 г.).