

УТВЕРЖДАЮ
Заведующий кафедрой информатики и
программного обеспечения
Т.М. Хвостенко

Рабочая программа учебной дисциплины

ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль) подготовки:
Прикладная информатика

Квалификация (степень) выпускника:
Бакалавр

Форма обучения:
Очная, очно-заочная, заочная

Составитель программы:
Гришанова Т.В.

Брянск 2026

СОДЕРЖАНИЕ

1. Аннотация к дисциплине.....	3
2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы.....	3
3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся.....	6
3.1 Объем дисциплины по видам учебных занятий (в часах).....	6
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий.....	6
4.1 Тематическая структура дисциплины.....	6
4.2 Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах).....	7
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	10
6. Оценочные материалы для проведения текущей и промежуточной аттестации обучающихся по дисциплине «Высокоуровневые методы информатики и программирования».....	11
6.1. Описание показателей и критериев оценивания компетенций, описание шкал оценивания.....	11
6.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения основной профессиональной образовательной программы.....	13
6.3. Типовые контрольные задания или иные материалы, необходимые для процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения основной профессиональной образовательной программы.....	14
6.3.1. Типовые задания для проведения текущего контроля обучающихся.....	14
6.3.1.1. Примерные тестовые задания для текущего контроля.....	14
6.3.2. Типовые задания для проведения промежуточной аттестации обучающихся.....	18
6.3.2.1. Типовые вопросы к экзамену.....	21
6.3.2.2. Рекомендуемые задачи.....	23
6.3.2.3. Итоговое тестирование.....	31
6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.....	37
7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.....	38
8. Методические указания для обучающихся по освоению дисциплины.....	39
9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.....	43
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, в том числе комплект лицензионного программного обеспечения, электронно-библиотечные системы, современные профессиональные базы данных и информационные справочные систем.....	44
10.1 Лицензионное программное обеспечение.....	44
10.2. Электронно-библиотечная система.....	44
10.3. Современные профессиональные баз данных.....	44
10.4. Информационные справочные системы.....	45

1. Аннотация к дисциплине

Рабочая программа дисциплины «Высокоуровневые методы информатики и программирования» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденный приказом Министерства образования и науки Российской Федерации от «19» сентября 2017 г. №922 (с изменениями и дополнениями от: 26 ноября 2020 г., 8 февраля 2021 г.).

Место дисциплины в структуре основной профессиональной образовательной программы

Настоящая дисциплина включена в обязательную часть Блока1 учебных планов по направлению подготовки 09.03.03 Прикладная информатика уровень бакалавриата.

Дисциплина изучается на 1 курсе во 2 семестре, экзамен при очной форме обучения, на 3 курсе в 5 семестре, экзамен при очно-заочной и заочной формах обучения.

Цель изучения дисциплины:

формирование у обучающихся системы знаний в области теории и практики высокоуровневых методов информатики и программирования.

Задачи:

- научиться программировать приложения и создавать программные прототипы решения прикладных задач,
- принимать участие в управлении проектами создания информационных систем на стадиях жизненного цикла;
- освоить методы решения практических задач на основе готовых пакетов прикладных программ (Microsoft Office 2007),
- изучить основы объектно-ориентированного программирования применительно к интегрированной среде Delphi и уметь на высоком профессиональном уровне создавать проекты будущих программных приложений с использованием стандартных компонент самой среды и возможностей встроенного языка программирования высокого уровня Delphi (Object Pascal), создавая в рамках проектируемых программ современный пользовательский графический интерфейс.

Компетенции обучающегося, формируемые в результате освоения дисциплины

ПК-2. Способность осуществлять описание бизнес-процессов на основе исходных данных

ПК-2.3 Применять программные средства для описания бизнес-процессов на основе исходных данных с использованием средств программирования

2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы

Процесс изучения дисциплины направлен на формирование компетенций, предусмотренных ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата) и на основе профессионального стандарта «Специалист по информационным системам», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2014 г. № 896н (зарегистрирован Министерством юстиции Российской Федерации 24 декабря 2014 г., регистрационный № 35361), с изменением, внесенным приказом Министерства труда и социальной защиты Российской Федерации от 12 декабря 2016 г. № 727н (зарегистрирован Министерством юстиции Российской Федерации 13 января 2017 г., регистрационный № 45230)

Код компетенции	Результаты освоения ОПОП (содержание компетенций)	Индикаторы достижения компетенций	Формы образовательной деятельности, способствующие формированию и развитию компетенции
ПК-2	Способность осуществлять описание бизнес-процессов на основе исходных данных	ПК-2.3 Применять программные средства для описания бизнес-процессов на основе исходных данных с использованием средств программирования Знать: возможности типовой информационной системы; основы программирования Уметь: применять программные средства для описания бизнес-процесса; применять языки программирования высокого уровня в практической деятельности. Владеть: навыками формирования нормативно-технической документации; современными средствами программирования	<u>Контактная работа:</u> Лекции Практические занятия <u>Самостоятельная работа</u>

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 6 зачетных единиц.

3.1 Объём дисциплины по видам учебных занятий (в часах)

Объём дисциплины	Всего часов		
	очная форма обучения	очно-заочная форма обучения	заочная форма обучения
Общая трудоемкость дисциплины	216		
Контактная работа обучающихся с преподавателем (всего)	95	40	16
Аудиторная работа (всего):	95	40	16
в том числе:			
Лекции	38	20	6
семинары, практические занятия	57	20	10
лабораторные работы			
Внеаудиторная работа (всего):			
в том числе:			
Самостоятельная работа обучающихся (всего)	85	140	191
Вид промежуточной аттестации обучающегося – экзамен	36	36	9

4.Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Тематическая структура дисциплины

№ п/п	Наименование модуля	№ п.п.	Тема	Перечень планируемых результатов обучения (ПРО)
1	Основы программирования	1	Сущность технологий программирования	ПК-2.3
		2	Объектно-ориентированный подход к программированию.	
		3	Принципы объектно-ориентированного программирования	
		4	Функционально-ориентированный подход к программированию.	
		5	Модульное программирование	
2	Алгоритмизация и программирование	6	Основы алгоритмизации и программирования.	ПК-2.3
		7	Основные инструментальные системы создания программ обработки информации.	
		8	Организация ввода и вывода данных в языке Python. Работа со строковыми данными в Python	
		9	Логические и условные операторы в Python	
		10	Циклы в Python	
		11	Работа со списками в Python	
		12	Словари в Python	
3	Технологии программирования. Языки программирования высокого уровня	13	Системы и технологии программирования. Языки программирования высокого уровня	ПК-2.3
		14	Работа с модулями в Python	
		15	Создание графических изображений с использованием модуля Turtle в Python	
		16	Создание игр с использованием модуля Pygame	
4	Программные средства реализации информационных процессов	17	Среда разработки Visual Basic	ПК-2.3
		18	Основные элементы среды разработки Visual Basic	
		19	Основные возможности Visual Basic	
		20	Основы создания графического интерфейса пользователя.	
		21	Расширенные средства создания приложений	
		22	Построение меню в Visual Basic	
		23	Создание панелей инструментов в Visual Basic	
5	Основы визуального программирования	24	Создание приложения Excel как макроса Visual Basic.	ПК-2.3
		25	Создание приложения Word как макроса Visual Basic.	

		26	Использование графики для создания приложений	
		27	Использование средств мультимедиа для создания приложений	
		28	Создание новых классов объектов и их использование в приложениях.	
		29	Работа приложений с базами данных.	
		30	Разработка интерфейса и использование мастера форм.	
		31	Создание справочной системы приложения.	
6	Модели решения функциональных и вычислительных задач	32	Разработка приложений в Delphi.	ПК-2.3
		33	Последовательность разработки справочной системы приложения.	
		34	Создание приложений для работы с БД в Delphi.	
		35	Выполнение операций по обработке данных в БД.	
		36	Создание запросов к БД.	

4.2 Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

для очной формы обучения

№ п/п	Разделы дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)							Вид оценочного средства текущего контроля успеваемости, промежуточной аттестации (по семестрам)	
			Всего	Из них аудиторные занятия				Самостоятел. работа	Контрольна		Курсовая
				Лекции	Практикум. Лаборатор	Практическ.занятия /семинары					
1	Основы программирования	5	30	8		10		12			Опрос, тестирование
2	Алгоритмизация и программирование	5	30	6		10		14			Опрос, тестирование
3	Технологии программирования. Языки программирования высокого уровня	5	30	6		10		14			Опрос, тестирование
4	Программные средства реализации информационных про-	5	30	6		10		14			Опрос, тестирование

	цессов									
5	Основы визуального программирования	5	30	6		10		14		Опрос, тестирование
6	Модели решения функциональных и вычислительных задач	5	30	6		7		17		Опрос, тестирование
	Контроль	5	36							
			236	38		57		85		36 (экзамен)

для очно-заочной формы обучения

№ п/п	Разделы дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)							Вид оценочного средства текущего контроля успеваемости, промежуточной аттестации (по семестрам)	
			Всего	Из них аудиторные занятия				Самостоятел	Контрольн я работа		Курсовая
				Лекции	.Практикум. Лаборатор	Практическ.занятия /семинары					
1	Основы программирования	5	30	4		4		22			Опрос, тестирование
2	Алгоритмизация и программирование	5	30	4		4		22			Опрос, тестирование
3	Технологии программирования. Языки программирования высокого уровня	5	30	2		4		24			Опрос, тестирование
4	Программные средства реализации информационных процессов	5	30	4		2		24			Опрос, тестирование
5	Основы визуального программирования	5	30	4		2		24			Опрос, тестирование
6	Модели решения функциональных и вычислительных задач	5	30	2		4		24			Опрос, тестирование
	Контроль	5	36								
			236	20		20		140			36 (экзамен)

для заочной формы обучения

№ п/п	Разделы дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Вид оценочного средства текущего контроля успеваемости, промежуточной аттестации (по семестрам)		
			Всего	Из них аудиторные занятия				Самостоятел		Контрольна я работа	Курсовая
				Лекции	Практикум. Лаборатор	Практическ. занятия /семинары					
1	Основы программирования	5	38	1		2		35		Опрос, тестирование	
2	Алгоритмизация и программирование	5	38	1		2		35		Опрос, тестирование	
3	Технологии программирования. Языки программирования высокого уровня	5	38	1		1		26		Опрос, тестирование	
4	Программные средства реализации информационных процессов	5	38	1		2		35		Опрос, тестирование	
5	Основы визуального программирования	5	38	1		2		35		Опрос, тестирование	
6	Модели решения функциональных и вычислительных задач	5	37	1		1		25		Опрос, тестирование	
	Контроль	5	9								
			236	6		10		191		9 (экзамен)	

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа обучающихся при изучении курса «Высокоуровневые методы информатики и программирования» предполагает, в первую очередь, работу с основной и дополнительной литературой. Результатами этой работы становятся выступления на практических занятиях, участие в обсуждении.

Методика самостоятельной работы предварительно разъясняется преподавателем и в последующем может уточняться с учетом индивидуальных особенностей обучающихся. Время и место самостоятельной работы выбираются обучающимися по своему усмотрению с учетом рекомендаций преподавателя.

Самостоятельную работу над дисциплиной следует начинать с изучения рабочей программы дисциплины «Высокоуровневые методы информатики и программирования», которая содержит

основные требования к знаниям, умениям и навыкам обучаемых. Обязательно следует вспомнить рекомендации преподавателя, данные в ходе установочных занятий. Затем – приступить к изучению отдельных разделов и тем в порядке, предусмотренном программой.

Получив представление об основном содержании раздела, темы, необходимо изучить материал с помощью учебников, указанных в разделе 7 указанной программы. Целесообразно составить краткий конспект или схему, отображающую смысл и связи основных понятий данного раздела и включенных в него тем. Затем, как показывает опыт, полезно изучить выдержки из первоисточников. При желании можно составить их краткий конспект. Обязательно следует записывать возникшие вопросы, на которые не удалось ответить самостоятельно.

Наименование раздела	Вопросы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Основы программирования	Модульное программирование	Работа в библиотеке, включая ЭБС. Подготовка доклада-презентации.	Литература к теме, работа с интернет источниками	Опрос, доклад-презентация
Алгоритмизация и программирование	Основы алгоритмизации и программирования.	Работа в библиотеке, включая ЭБС. Подготовка доклада-презентации	Литература к теме, работа с интернет источниками	Опрос, доклад-презентация
Технологии программирования. Языки программирования высокого уровня	Работа с модулями в Python	Работа в библиотеке, включая ЭБС. Подготовка доклада-презентации.	Литература к теме, работа с интернет источниками	Опрос, доклад-презентация
Программные средства реализации информационных процессов	Создание панелей инструментов в Visual Basic	Работа в библиотеке, включая ЭБС. Подготовка доклада-презентации.	Литература к теме, работа с интернет источниками	Опрос, доклад-презентация
Основы визуального программирования	Создание приложения Excel как макроса Visual Basic.	Работа в библиотеке, включая ЭБС. Подготовка доклада-презентации.	Литература к теме, работа с интернет источниками	Опрос, доклад-презентация
Модели решения функциональных и вычислительных задач	Разработка приложений в Delphi.	Работа в библиотеке, включая ЭБС. Подготовка доклада-презентации.	Литература к теме, работа с интернет источниками	Опрос, доклад-презентация

6. Оценочные материалы для проведения текущей и промежуточной аттестации

обучающихся по дисциплине «Высокоуровневые методы информатики и программирования»

6.1. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Шкала и критерии оценки, балл	Критерии оценивания компетенции
1.	Опрос	Сбор первичной информации по выяснению уровня усвоения пройденного материала	«Зачтено» - если обучающийся демонстрирует знание материала по разделу, основанные на знакомстве с обязательной литературой и современными публикациями; дает логичные, аргументированные ответы на поставленные вопросы. Также оценка «зачтено» ставится, если обучающимся допущены незначительные неточности в ответах, которые он исправляет путем наводящих вопросов со стороны преподавателя. «Не зачтено» - имеются существенные пробелы в знании основного материала по разделу, а также допущены принципиальные ошибки при изложении материала.	ПК-2.3
2	Доклад-презентация	Публичное выступление по представлению полученных результатов в программе Microsoft PowerPoint	«отлично» – доклад выполнен в соответствии с заявленной темой, презентация легко читаема и ясна для понимания, грамотное использование терминологии, свободное изложение рассматриваемых проблем, докладчик правильно ответил на все вопросы в ходе дискуссии; «хорошо» – некорректное оформление презентации, грамотное использование терминологии, в основном свободное изложение рассматриваемых проблем, докладчик частично правильно ответил на все вопросы в ходе дискуссии; «удовлетворительно» – отсутствие презентации, докладчик испытывал затруднения при выступлении и ответил на вопросы в ходе дискуссии; «неудовлетворительно» - докладчик не раскрыл тему	ПК-2.3
3	Тестирование	Тестирование можно проводить в форме: • компьютерного тестирования, т.е. компьютер произвольно выбирает вопросы из базы данных по степени сложности; • письменных ответов, т.е. преподаватель задает вопрос и дает несколько вариантов ответа, а студент на отдельном листе записывает номера вопросов и номера соответствующих ответов	«отлично» - процент правильных ответов 80-100%; «хорошо» - процент правильных ответов 65-79,9%; «удовлетворительно» - процент правильных ответов 50-64,9%; «неудовлетворительно» - процент правильных ответов менее 50%.	ПК-2.3

6.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения основной профессиональной образовательной программы

№	Форма контроля/ коды оцениваемых компетенций	Процедура оценивания	Шкала и критерии оценки, балл
3.	Экзамен – ПК-2.3	Правильность ответов на все вопросы (верное, четкое и достаточно глубокое изложение идей, понятий, фактов и т.д.); Сочетание полноты и лаконичности ответа; Наличие практических навыков по дисциплине (решение задач или заданий); Ориентирование в учебной, научной и специальной литературе; Логика и аргументированность изложения; Грамотное комментирование, приведение примеров, аналогий; Культура ответа.	1. оценка «отлично» - обучающийся должен дать полные, исчерпывающие ответы на вопросы экзаменационного билета, в частности, ответ должен предполагать знание основных понятий и их особенностей, умение правильно определять специфику соответствующих отношений, правильное решение практического задания. Оценка «отлично» предполагает наличие системы знаний по предмету, умение излагать материал в логической последовательности, систематично, грамотным языком; 2. оценка «хорошо» - обучающийся должен дать полные ответы на вопросы, указанные в экзаменационном билете. Допускаются неточности при ответе, которые все же не влияют на правильность ответа. Ответ должен предполагать знание основных понятий и их особенностей, умение правильно определять специфику соответствующих отношений, правильное решение практического задания. Оценка «хорошо» предполагает наличие системы знаний по предмету, умение излагать материал в логической последовательности, систематично, грамотным языком, однако, допускаются незначительные ошибки, неточности по названным критериям, которые все же не искажают сути соответствующего ответа; 3. оценка «удовлетворительно» - обучающийся должен в целом дать ответы на вопросы, предложенные в экзаменационном билете, ориентироваться в системе дисциплины «Высокоуровневые методы информатики и программирования», продемонстрировать правильный ход решения практического задания, знать основные категории предмета. Оценка «удовлетворительно» предполагает, что материал в основном изложен грамотным языком; 4. оценка «неудовлетворительно» предполагает, что обучающимся либо не дан ответ на вопрос билета, либо обучающийся не знает основных категорий, не может определить предмет дисциплины.
4.	Тестирование (на экзамене) – ПК-2.3	Полнота знаний теоретического контролируемого материала. Количество правильных ответов	«отлично» - процент правильных ответов 80-100%; «хорошо» - процент правильных ответов 65-79,9%; «удовлетворительно» - процент правильных ответов 50-64,9%; «неудовлетворительно» - процент правильных ответов менее 50%.

6.3. Типовые контрольные задания или иные материалы, необходимые для процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения основной профессиональной образовательной программы

6.3.1. Типовые задания для проведения текущего контроля обучающихся

6.3.1.1. Примерные тестовые задания для текущего контроля

**Тема 6. Модели решения функциональных и вычислительных задач
Тема 1. Основы программирования**

Задание 1

Объектно-ориентированный подход использует:

объектную декомпозицию

структурный алгоритм

методологию программирования на основе системного подхода к анализу, проектированию и реализации программного обеспечения

последовательное разложение функций обработки данных на простые функциональные элементы

Задание 2

Методология программирования на основе системного подхода к анализу, проектированию и реализации программного обеспечения – это _____ программирование

Ответ: структурное

Задание 3

Главной целью информатизации является

наиболее полное удовлетворение потребностей общества в информации во всех сферах деятельности

создание новой инфраструктуры и сектора услуг, способных поддержать национальную экономику

внедрение компьютерных и телекоммуникационных технологий

замена экономической структуры, основанной на тяжелой промышленности, структурой, базирующейся на наукоемких областях

Задание 4

Установите соответствие между основными понятиями технологии программирования и их определением

Понятие	Определение
1. программирование	А. программные продукты и техническая документация к ним
2. программа	Б. сфера деятельности, направленная на создание программ
3. приложение	В. последовательность команд компьютера, приводящая к решению задачи
4. программное обеспечение	Г. комплекс взаимосвязанных программ, предназначенный для реализации определенной задачи массового спроса
5. программный продукт	Д. программная реализация на компьютере решения задачи

Запишите выбранные цифры под соответствующими буквами:

А	Б	В	Г	Д

Ответ: 4, 1, 2, 5, 3

Задание 5

Установить правильную последовательность действий при разработке функциональной структуры алгоритма приложения

№	Этапы разработки функциональной структуры алгоритма приложения
1	Устанавливается состав приложений, обеспечивающих реализацию поставленных целей
2	определяются необходимые для решения задач функции обработки данных
3	определяются цели автоматизации предметной области и их иерархия
4	Выполняется декомпозиция функций обработки данных до необходимой структурной сложности
5	уточняется характер взаимосвязи приложений и их основные характеристики

Запишите соответствующую последовательность цифр слева направо:

--	--	--	--	--

Ответ: 3, 1, 5, 2, 4

Тема 2.Алгоритмизация и программирование

Задание 6

Язык манипулирования данными используется для ...

- создания и изменения структуры базы данных и ее составных частей
- управления правами доступа к данным и выполнением процедур в многопользовательской среде
- манипулирования данными в таблицах баз данных**
- создания запросов к БД

Задание 7

Язык определения данных используется для ...

- создания и изменения структуры базы данных и ее составных частей**
- управления правами доступа к данным и выполнением процедур в многопользовательской среде
- манипулирования данными в таблицах баз данных
- создания запросов к БД

Задание 8

Разработка программы начинается с этапа:

- разработки алгоритма
- перевода алгоритма на язык программирования
- анализа требований**
- тестирования

Задание 9

Алгоритм называется циклическим:

если его выполнение предполагает многократное повторение одних и тех же действий
если ход его выполнения зависит от истинности тех или иных условий
если его команды выполняются в порядке их естественного следования независимо от каких-либо условий
если он включает в себя вспомогательный алгоритм

Тема 3. Технологии программирования. Языки программирования высокого уровня

Задание 10

Представлен фрагмент программы:

```
while логическое выражение:  
    операторы циклической части программы  
print()
```

это:

- форма записи оператора цикла с предусловием**
- форма записи оператора цикла с постусловием
- форма записи оператора цикла с параметром
- форма записи массива данных

Задание 11

Задан фрагмент программы

```
for i in range(1, m):  
    s=0  
    for j in range(1, n):  
        if a[i,j]>0:  
            s=s+a[i,j]
```

Эта программа позволяет:

- для матрицы А вычислить и запомнить сумму положительных элементов каждого столбца матрицы**
- для матрицы А вычислить и запомнить сумму отрицательных элементов каждого столбца матрицы.
- для матрицы А вычислить и запомнить сумму положительных элементов каждой строки матрицы
- для матрицы А вычислить и запомнить сумму отрицательных элементов каждой строки матрицы.

Задание 12

Ошибки, связанные с неправильной записью конструкций языка программирования, называют:

- логическими
- орфографическими
- синтаксическими**
- методическими.

Задание 13

Из каких двух основных частей состоит пользовательский интерфейс:

- из простого интерфейса и руководства по использованию этого интерфейса
- из сложного меню и кнопки "Выход из программы"
- из монитора и клавиатуры
- из форм и объектов**

Тема 4. Программные средства реализации информационных процессов

Задание 14

В записи функции «InputBox (подсказка,[заголовок], [значение], [x], [y])», значение – это ...

строка для отображения в окне подсказки относительно вводимого значения
принимаемое по умолчанию значение, которое выводится автоматически в текстовом поле

строковое значение, задающее название окна
координаты левого верхнего угла окна диалога

Задание 15

Функция Sheets(“лист3”).Select в Visual Basic:

позволяет выбрать рабочий лист книги

выводит на экран окно, которое используется для отображения какой-либо информации или сообщения

позволяет очищать область вывода данных

отображает окно диалога для ввода данных в текстовое поле, описание которых может быть задано как значения некоторых параметров

Задание 16

Объект в Visual Basic, позволяющий манипулировать графикой в различных форматах

Picture

PictureBox

Image

Image Line

Задание 17

Достоинством языка программирования Visual Basic является:

оптимальное сочетание простоты использования, доступности и большого набора разнообразных возможностей, позволяющие охватить все основные области программной деятельности

возможность работы с таймером, доступ к базам данных

реализация разработчиком гибкого и удобного интерфейса для своего приложения

поддержка сети Интернет, позволяющая организовать связь с другими компьютерами, подключенными к сети

Задание 18

Функция в Visual Basic, которая выводит на экран окно, используемое для отображения какой-либо информации или сообщения: : _____

Ответ: MsgBox

Задание 19

Оператор Sub в программе на языке программирования Visual Basic задает:

изменяет значение аргумента

ввод значений переменных

название программы

описание переменных

Задание 20

Верхний уровень меню это ...

строка меню, в которой находятся элементы главного меню

список элементов управления

раскрывающийся список элементов

панель задач

Тема 5. Основы визуального программирования

Задание 21

Для построения диаграммы в Visual Basic используется объект:

Chart

ChartWizard

Gallery

PlotBy

Задание 22

Элементы синтаксиса метода ChartWizard при построении графика, который задает место расположения рядов данных:

PlotBy

Source

Gallery

Format

Задание 23

Команда в Visual Basic, предназначенная для проверки орфографии:

CategoryTitle

CheckSpelling

PlotBy

CategoryLabels

Задание 24

Для построения диаграммы в Visual Basic используется объект:

Chart

ChartWizard

Gallery

PlotBy

Задание 25

Функция Sheets(“лист3”).Select в Visual Basic:

1. позволяет выбрать рабочий лист книги

2. выводит на экран окно, которое используется для отображения какой-либо информации или сообщения

3. позволяет очищать область вывода данных

4. отображает окно диалога для ввода данных в текстовое поле, описание которых может быть задано как значения некоторых параметров

Ответ:

Обоснование:

Ответ: 1

Обоснование: команда Sheets позволяет указать лист рабочей книги, а Select определяет переход на данный лист

Задание 26

Для создания справочной системы в Visual Basic предназначена утилита:

Single record

Grid (Datasheet)

Тема 6. Модели решения функциональных и вычислительных задач

Задание 27

Логический оператор, описывающих условие в SQL запросах, который сравнивает поле с заданным значением, имеет вид:

=
Like
In
Between

Задание 28

Тип поля в Delphi - число в денежном формате:

Alpha
Number
Money
Short

Задание 29

Выражения для получения значений вычисляемых полей разработчик должен разместить в методе-обработчике набора данных ...

DBDEMOS
TTable
TQuery
OnCalcFields

Задание 30

Фильтры, которые можно создавать и редактировать во время выполнения приложения, для них используется только свойство Filter, называются _____

Ответ: динамические

Задание 31

Свойство DataBaseName компонента Query служит для ...

задания базы данных, с которой осуществляется связь путем выбора из выпадающего списка псевдонимов или указанием полного пути к файлу
управления отображением данных в компоненте Query
использования динамических запросов, содержащих параметры.
получения результата запроса после задания параметра, надо повторно выполнить запрос

Задание 32

Запись оператора SELECT * означает, что

в результирующий набор данных необходимо включить все поля
в результирующий набор данных не включается ни одного поля
в результирующий набор данных необходимо включить только первое поле
в результирующий набор данных необходимо включить первые 3 поля

Задание 33

Стандартные компоненты Delphi, который позволяет поместить главное меню в программу:

TMainMenu
TPopupMenu
TLabel

TEdit

Задание 34

Вкладка Delphi, которая содержит основные свойства окна, такие как тип окна: произвольная строка, задаваемая пользователем:

General

Buttons

Position

Styles

Задание 25

Набор DLL-библиотек, обеспечивающих низкоуровневый доступ к локальным и клиент-серверным БД:

Borland Database Engine

SQL Links

BDE Administrator

Database Desktop

Задание 36

Свойство, которое соответствует заголовку столбца данного поля:

Alignment

Display Label

Display Width

Readonly

Задание 37

SQL, который используется в специальных утилитах, позволяющих в интерактивном режиме вводить запросы с использованием команд SQL, посылать их для выполнения на сервер и получать результаты в предназначенном для этого окне – это _____ SQL

Ответ: интерактивный

Задание 38

SQL, который используется в прикладных программах, позволяя им посылать запросы к серверу и обрабатывать полученные результаты, в том числе комбинируя set-ориентированный и record-ориентированный подходы.

интерактивный SQL

встроенный SQL

оперативный SQL

многомерный SQL

6.3.2. Типовые задания для проведения промежуточной аттестации обучающихся

Промежуточная аттестация по дисциплине «Высокоуровневые методы информатики и программирования» проводится в форме экзамена.

6.3.2.1. Типовые вопросы к экзамену

1. Сущность технологий программирования
2. Объектно-ориентированный подход к программированию
3. Основные понятия алгоритмизации и программирования
4. Простейшие конструкции языка Python – числа, константы, переменные, функции, выражения.
5. Организация ввода и вывода данных в языке Python.
6. Программирование условий в Python.

7. Программирование алгоритмов циклической структуры. Операторы циклов.
8. Программирование задач обработки одномерных массивов данных.
9. Основные функции для работы со строками в Python
10. Использование словарей в Python
11. Средство создания приложений Visual Basic и его основные возможности.
12. Основы создания графического интерфейса пользователя
13. Основы программирования на Visual Basic.
14. Порядок создания приложений Windows с помощью Visual Basic.
15. Как подготовить приложение Visual Basic к распространению?
16. Что такое интерфейс пользователя в Windows-приложениях? Как он создается в Visual Basic?
Перечислите основные стандартные элементы управления, из которых строится пользовательский интерфейс. Какие свойства являются общими для этих элементов?
17. Какие базовые типы данных поддерживает Visual Basic? Какой оператор должен присутствовать в разделе глобальных объявлений, чтобы исключить ошибки в случае появления неявно объявленных переменных?
18. Работа с диалоговыми окнами Visual Basic и программирование вычислений как макросов для Excel.
19. Как добавить новый элемент управления в панель инструментов? Какие средства создания меню приложений используются в Visual Basic?
20. Что можно создать в приложении с помощью объектов общих диалогов? Как создать диалоговое окно с предупреждающим сообщением? Что такое окно ввода и как его использовать в разрабатываемом приложении?
21. Что такое динамические массивы и как они декларируются?
22. Каково различие между переменной и константой? Какие типы констант поддерживает Visual Basic?
23. Каково различие между линейным и блочным оператором If ? Какой оператор Visual Basic позволяет сделать выбор из нескольких альтернативных вариантов?
24. Как определяется цикл с известным числом повторений? Какие циклы с условием поддерживает Visual Basic?
25. В чем различие циклов с предусловием и постусловием?
26. Что такое функция и как она определяется?
27. Как производится обращение к функции?
28. Какие типы встроенных функций есть в Visual Basic?
29. Назовите основные события формы. В каких случаях они происходят? Назовите основные события мыши. Назовите основные события клавиатуры.
30. В чем заключается сущность класса? Что определяется в объявлении класса? Какие процедуры используются для включения свойств класса? Как определяются методы класса?
31. Укажите последовательность действий при создании хорошо структурированного приложения, облегчающего его сопровождение в Delphi?
32. Почему начинать проектирование целесообразно с формирования списка изображений ImageList и зачем надо связывать с этим списком диспетчер ActionList.
33. Последовательность создания БД в Delphi.
34. Подключение формы приложения к БД в Delphi.
35. Принципы объектно-ориентированного программирования
36. Функционально-ориентированный подход к программированию
37. Модульное программирование
38. Основные инструментальные средства создания программ обработки данных
39. Понятие о двумерных массивах данных. Порядок составления программ с использованием массивов.
40. Модуль time в Python
41. Модуль random в Python
42. Создание графических изображений с помощью модуля Turtle в Python

43. Основы создания игр с использованием модуля Pygame. Понятие игрового цикла
44. Основные элементы разработки среды Visual Basic.
45. Основные возможности Visual Basic.
46. В чем заключаются особенности среды программирования Visual Basic? Какие окна входят в интегрированную среду разработки Visual Basic?
47. Что такое программирование, управляемое событиями? В чем заключаются его особенности?
48. Что такое проект Visual Basic? Какое средство среды Visual Basic можно использовать для просмотра структуры проекта? Какие возможности обеспечивает это средство?
49. В чем сущность модульного программирования? Чем отличается процедура-функция от подпрограммы?
50. Назовите типы файлов, используемых в Visual Basic. Какие операции предусмотрены в Visual Basic для работы с файлами?
51. В чем отличия файлов прямого и последовательного доступа?
52. Какой объект нужно использовать в приложении, чтобы можно было работать с базой данных, созданной в Access?
53. Какие формы взаимодействия с приложениями Windows предусматривает Visual Basic?
54. Как осуществляется связь программы Visual Basic с файлами приложений?
55. Как можно установить связь программы Visual Basic с данными активного приложения?
56. Как включается приложение MS Office в форму проекта?
57. Каким образом можно получить сведения о функциях приложений MS Office и включить их в программу Visual Basic?
58. С какой целью в проектах Visual Basic используются графические изображения?
59. Как можно добавить растровое изображение в форму? Как можно заменить во время выполнения программы один растровый рисунок на другой?
60. Какие средства графического управления есть в Visual Basic? Какие графические методы Visual Basic вы знаете?
61. Зачем при работе с базами данных используются индексы?
62. Как упорядочиваются записи, если индекс содержит несколько полей?
63. Как создать приложение, использующее две таблицы — главную и вспомогательную?
64. Как можно создать вычисляемое поле набора данных?
65. Как организуется фильтрация данных?
66. Компонент Query и его использование для создания приложений в Delphi.

6.3.2.2. Рекомендуемые задачи

Задание 1

Пользователь вводит целое число. Выведите его строку-описание вида "отрицательное четное число", "нулевое число", "положительное нечетное число", например, численным описанием числа 190 является строка "положительное четное число". Если число не является четным - выведите сообщение «число не является четным»

Решение:

```
some_number = int(input())
if some_number < 0:
    if some_number % 2 > 0:
        print("Отрицательное нечетное число")
        print("Число не является четным")
    else:
        print("Отрицательное четное число")
elif some_number == 0:
    print("Нулевое число")
else:
    if some_number % 2 > 0:
```

```

    print("Положительное нечетное число")
    print("Число не является четным")
else:
    print("Положительное четное число")

```

Задание 2

На вход подается 1 строка без пробелов. По данной строке определите, является ли она палиндромом (то есть, можно ли прочесть ее наоборот, как, например, слово "шалаш"). Необходимо вывести "yes", если строка является палиндромом, и "no" в противном случае.

Решение:

```

some_word = input("Введите слово, которое необходимо проверить: ")
reversed_word = ""

```

```

for i in range(len(some_word), 0, -1):
    reversed_word += some_word[i - 1]

```

```

if (reversed_word == some_word):
    print("Yes")
else:
    print("No")

```

Задание 3

В первую строчку вводится число N ($1 \leq N \leq 100\,000$). В следующую строку через пробел вводятся N чисел ($1 \leq A_i \leq 10^9$). Вам требуется написать метод, который получает на вход массив и изменяет его таким образом, чтобы на первом месте стоял последний элемент, на втором - первый, на третьем - второй и т. д. Выведите N чисел - измененный массив.

Решение:

```

N = int(input("Введите N (1..100000): ").strip())
if not (1 <= N <= 100000):
    print("Ошибка: N должно быть в диапазоне 1..100000.")
    raise SystemExit

```

```

parts = input("Введите N чисел через пробел (каждый 1..10^9): ").split()
if len(parts) != N:
    print("Ошибка: количество чисел не равно N.")
    raise SystemExit

```

```

arr = []
for i, token in enumerate(parts, 1):
    try:
        x = int(token)
    except ValueError:
        print(f"Ошибка: элемент #{i} не является целым числом.")
        raise SystemExit
    if not (1 <= x <= 10**9):
        print(f"Ошибка: элемент #{i} выходит за пределы 1..10^9.")
        raise SystemExit
    arr.append(x)

```

```

rotated = [arr[-1]] + arr[:-1] if N > 1 else arr

```

```

print("Измененный массив:")

```

```
print(*rotated)
```

Задание 4

На берегу реки стояли n рыбаков, все они хотели перебраться на другой берег. Одна лодка может выдержать не более m килограмм, при этом в лодку помещается не более 2 человек. Определите, какое минимальное число лодок нужно, чтобы перевезти на другой берег всех рыбаков. В первую строку вводится число m ($1 \leq m \leq 10^6$) - максимальная масса, которую может выдержать одна лодка. Во вторую строку вводится число n ($1 \leq n \leq 100$) - количество рыбаков. В следующие N строк вводится по одному числу A_i ($1 \leq A_i \leq m$) - вес каждого путешественника. Программа должна вывести одно число - минимальное количество лодок, необходимое для переправки всех рыбаков на противоположный берег.

Решение:

```
m = int(input("Введите грузоподъёмность лодки m (1..1_000_000): ").strip())
```

```
if not (1 <= m <= 10**6):
```

```
    print("Ошибка: m должно быть в диапазоне 1..1_000_000.")
```

```
    raise SystemExit
```

```
n = int(input("Введите количество рыбаков n (1..100): ").strip())
```

```
if not (1 <= n <= 100):
```

```
    print("Ошибка: n должно быть в диапазоне 1..100.")
```

```
    raise SystemExit
```

```
weights = []
```

```
for i in range(1, n + 1):
```

```
    w = int(input(f"Введите вес рыбака #{i} (1..{m}): ").strip())
```

```
    if not (1 <= w <= m):
```

```
        print(f"Ошибка: вес рыбака #{i} должен быть в диапазоне 1..{m}.")
```

```
        raise SystemExit
```

```
    weights.append(w)
```

```
weights.sort()
```

```
i, j = 0, n - 1
```

```
boats = 0
```

```
while i <= j:
```

```
    boats += 1
```

```
    if weights[i] + weights[j] <= m:
```

```
        i += 1
```

```
    j -= 1
```

```
print("Минимальное количество лодок:", boats)
```

Задание 5

С помощью цикла создайте словарь, в котором ключи будут, например от числа 10, до -5 (включительно). А значениями этих ключей будут сами эти числа возведённые в степени равных этим числам

Например:

```
my_dict = {
```

```
    10: 100000000000,
```

```
    9: 387420489,
```

```
    # и так далее ...
```

```
    -5: -0.00032
```

```

}
Решение:
start = int(input("Введите начальное целое число (начало диапазона): ").strip())
end = int(input("Введите конечное целое число (конец диапазона): ").strip())

my_dict = {}
step = 1 if start <= end else -1
for k in range(start, end + step, step):
    my_dict[k] = k ** k

print("Сформированный словарь:")
print(my_dict)

```

Задание 6

Построение меню приложения. Объекты и коллекции, применяемые для работы с меню.

Содержание разработать приложение «Цвет и размер», в окне которого имеются меню следующего вида:

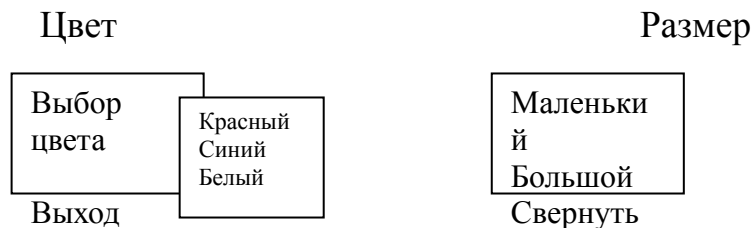


Рис. 1. Исходное задание

При выборе цвета в меню поле формы должно окрашиваться в соответствующий цвет. При задании соответствующего размера в меню окно формы изменяется от стандартного до полноэкранного, либо свернутого.

Указания по порядку и методике выполнения задания:

1. Запустить Visual Basic. В меню File щелкнуть мышью на команде New Project.
2. В окне проекта выделить щелчком мыши форму Form1 и задать ее свойства:
 - a) Name – frmColors; б) Caption – Wdtn b hfpvth; d) StartUpPostion – 2-Center Screen.
3. В меню Tools выбрать команду Menu Editor. На экране появится окно редактора меню (рис.2).

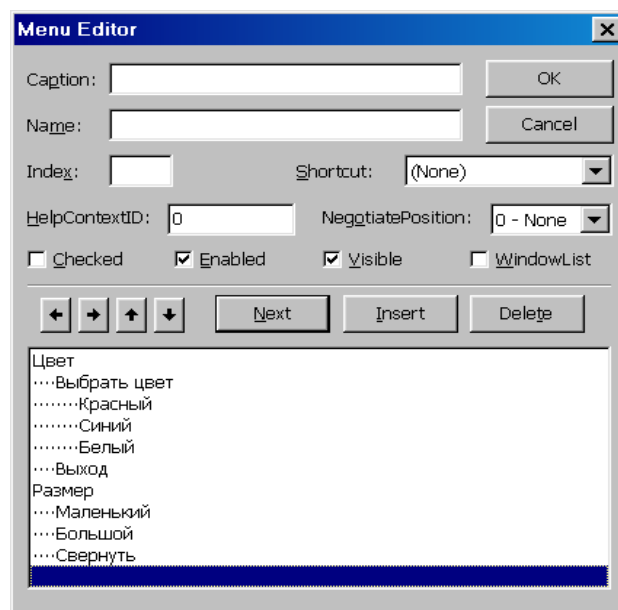


Рис.2. Окно редактора меню

4. Набрать «Цвет» в поле Caption и нажать клавишу Tab для перехода в следующее поле Name. При этом в нижней половине окна редактора появится надпись «Цвет».

5. Ввести имя «mnuColors» в поле Name.

6. Щелкнуть по кнопке Next для добавления заголовка меню **Цвет** в программу. Меню **Цвет** добавится к строке меню, а редактор меню очистит окно для ввода следующего пункта.

7. Набрать «Выбрать цвет» в поле Caption, нажать Tab и в поле Name набрать «mnuSetColor».

8. Задание подчиненного положения введенного пункта меню по отношению к предыдущему задается щелчком по кнопке со значком ➔.

9. Далее, с помощью редактора меню, ввести все остальные пункты меню, в соответствии с таблицей 1.

Таблица 1 - Пункты меню

Caption	Name	Управляющая кнопка
Цвет	mnuColors	
Выбрать цвет	mnuSetColor	➔
Красный	mnuRed	➔
Синий	mnuBlue	
Белый	mnuWhite	
Выход	mnuExit	⬅
Размер	mnuSize	⬅
Маленький	mnu Small	➔
Большой	mnuLarge	
Свернуть	mnuMin	

10. Щелкнуть по кнопке **ОК**, чтобы закрыть окно редактора. На форме «Цвет и размер» появится строка меню (рис. 3).

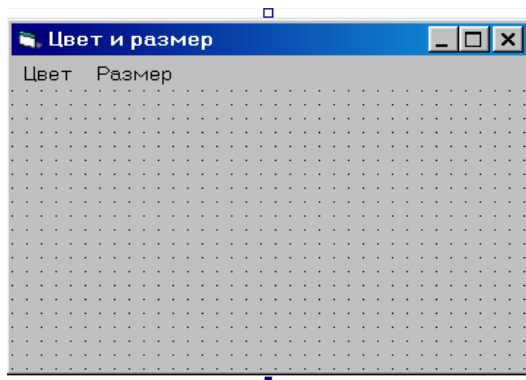


Рис.3. Форма со строкой меню

Далее следует задать программный код для всех пунктов меню. Начинать надо с программного кода, соответствующего открытию формы при запуске приложения. Необходимо дважды щелкнуть по полю формы, после чего открывается окно Code, где набирается код:

```
Private Sub Form_Load()
    frmColors.BackColor = vbWhite
    mnuWhite.Enabled = False
    mnuSmall.Enabled = False
End Sub
```

При открытии формы цвет фона задается белым, что выполняется с помощью задания свойства `frmColors.BackColor = vbWhite` (`vbWhite` – встроенная системная константа, список которых имеется в окне Object Browser меню View). Размер окна остается таким, каким он был установлен при проектировании формы. При открытии приложения пункты меню «Белый» и «Маленький» будут заблокированы, т.к. они уже задействованы. Блокирование выполняется командами:

```
mnuWhite.Enabled = False
mnuSmall.Enabled = False
```

Далее задается программный код для всех команд меню. Выполнить щелчок по соответствующей команде меню, после чего открывается окно Code, в котором набрать текст, приведенный ниже:

```
Private Sub Form_Load()
    frmColors.BackColor = vbWhite
    mnuWhite.Enabled = False
    mnuSmall.Enabled = False
End Sub
```

```
Private Sub mnuBlue_Click()
    frmColors.BackColor = vbBlue
    mnuRed.Enabled = True
    mnuBlue.Enabled = False
    mnuWhite.Enabled = True
End Sub
```

```
Private Sub mnuLarge_Click()
    frmColors.WindowState = 2
    mnuSmall.Enabled = True
    mnuLarge.Enabled = False
    mnuMin.Enabled = True
End Sub
```

```
Private Sub mnuMin_Click()
```

```
frmColors.WindowState = 1
mnuSmall.Enabled = True
mnuLarge.Enabled = True
mnuMin.Enabled = False
End Sub
```

```
Private Sub mnuRed_Click()
frmColors.BackColor = vbRed
mnuRed.Enabled = False
mnuBlue.Enabled = True
mnuWhite.Enabled = True
End Sub
```

```
Private Sub mnuSmall_Click()
frmColors.WindowState = 0
mnuSmall.Enabled = False
mnuLarge.Enabled = True
mnuMin.Enabled = True
End Sub
```

```
Private Sub mnuWhite_Click()
frmColors.BackColor = vbWhite
mnuRed.Enabled = True
mnuBlue.Enabled = True
mnuWhite.Enabled = False
End Sub
```

```
Private Sub mnuExit_Click()
End
End Sub
```

Задание 7

Создание БД в среде Delphi.

Цель работы: закрепить теоретические знания по созданию базы данных в Delphi на практике.

Содержание задания: создать БД в среде Delphi.

Указания по порядку и методике выполнения задания:

Для создания используются соответствующие СУБД. Но вместе с BDE и Delphi поставляется программа Database Desktop. Вызов Database Desktop включен в главное меню Delphi в раздел Toolse. Создадим с помощью Database Desktop таблицы данных Pers и Gro.

Таблица 1 - Пример табличных данных Pro

Номер	Группа (класс, от- дел)	ФИО	Оценка по информатике	Дата рождения	Пол	Характери- стика	Фото- графия
Num	Gr	Fam	Mark	Born	Sex	Charact	Photo
1	A1-01	Иванов И.И.	5	17.10.1985	М	...	...
2	A1-02	Петров П.П.	3	01.09.1986	М	...	...
3	A3-01	Сидоров С.С.	4	01.02.1984	М	...	...
4	Абитуриент	Иванова И. И.	1	17.01.1987	Ж	...	...
...	...	...	...	...	...	...	...

Таблица 2 - Пример таблицы данных о группах Gro

Группа (класс, отдел)	Преподаватель информатики (руководитель)
Gr	Teach
A1-01	Иванов И.И.
A1-02	Иванов И.И.
A3-01	Сидоров С.С.
Абитуриент	

Выполните в окне Database Desktop команду File/New. В открывшемся меню выберите раздел Table – создание новой таблицы. Откроется небольшое диалоговое окно. В нем из выпадающего списка вы можете выбрать СУБД, для которой хотите создать таблицу. Можете посмотреть в нем, таблицы, каких СУБД могут создаваться с помощью Database Desktop. Выберите Paradox 7. Вы увидите окно, представленное на рис. 1. В этом окне вы можете задать структуру таблицы (поля и их типы), создать вторичные индексы, ввести диапазоны допустимых значений полей, значения по умолчанию и ввести много иной полезной информации о создаваемой таблице.

Для каждого поля создаваемой, таблицы, прежде всего, указывает имя (Filed Name) – идентификатор поля. Он может включать до 25 символов. Затем надо выбрать тип (Type) данных этого поля. Для этого перейдите в раздел Type поля и щелкните правой кнопкой мыши.

Появится список доступных типов, из которого вы можете выбрать необходимый вам. В нашем примере таблице Pets (см. табл. 1) для поля Num целесообразно выбрать тип Autoincrement. Это тип целого числа, автоматически увеличивающегося на 1 для каждой новой записи. Следовательно, поле этого типа обеспечит уникальность каждой записи. Поэтому поле Num следует сделать ключевым. Ключевое поле должно быть, отмечено символом “*” в последней колонке Key. Для того чтобы поставить или удалить этот символ, надо или сделать двойной щелчок в соответствующей графе информации о поле, или выделить эту графу и нажать клавишу пробела.

Когда заполнена информация об одном поле, можно начинать вводить описание следующего поля. Для этого достаточно нажать клавишу со стрелкой вниз или клавишу табуляции. В полях Gr, Fam и Sex надо задать тип Alpha (строка). Для полей этого типа надо в столбце Size указать размер поля – число символов. Для поля Mark выберите тип Short (короткое целое). Для поля Born задайте тип Date (дата), для поля Charact – Formatted Memo (форматированный текст), для поля Photo – Graphic (графика). Значения полей типов Formatted Memo и Graphic будут храниться в отдельных файлах и могут иметь неограниченный размер. Но для этих полей так же, как для строк, нужно задавать размер в столбце Size. Это число байтов, хранящихся непосредственно в таблице. Практически все равно, какое число задать (например, 10). Но что – то задать надо, иначе Database Desktop сообщит об ошибке.

Выделив в окне рис. 1 то или иное поле, вы можете видеть в правой части окна элементы, позволяющие накладывать на значения поля некоторые ограничения. Индикатор Required Field надо включать для тех полей, значения которых обязательно должны содержаться в каждой записи. Для нашего примера таким полем, очевидно, должно быть поле Fam. Окно Default позволяет задать значение поля по умолчанию. Это свойство полезно задавать для полей чисел, дат и некоторых символьных. В нашем примере полезно задать значения по умолчанию для поля Born и поля Sex («пример» "М" - иначе у пользователя могут возникнуть проблемы при вводе информации). Окна Minimum и Maximum позволяют задать минимальное и максимальное значения. В нашем примере полезно указать эти значения для поля Born.

В последующих примерах нам потребуется индексировать таблицу различным образом. Так, что вам надо создать в таблице соответствующие вторичные индексы. В дальнейшем вам будут полезны следующие индексы:

Имя индекса	Поля	Пояснения
fio	Fam	Упорядочивание таблицы по алфавиту
grfio	Gr, Fam	Упорядочивание таблицы по группам, а внутри каждой группы упорядочивание по алфавиту
B	Born	Упорядочивание таблицы по году рождения
MarkGr	Mark, Gr	Упорядочивание по оценкам и группам

Чтобы создать новый вторичный индекс, выберите в выпадающем списке в верхнем правом

углу окна рис. 1 раздел Secondary Indexes. Нажмите кнопку Define, которая станет доступной. Откроется диалоговое окно. В его левой панели Fields содержится список доступных полей, а правой панели Indexed fields вы можете подобрать и упорядочить список полей включаемых в индекс. Для переноса поля из левого окна в правое надо выделить интересующее вас поле или группу полей и нажать кнопку со стрелкой вправо. Стрелками Change order (изменить последовательность) можно изменить порядок следования полей в индексе.

Панель радиокнопок Index Option(опции индекса) позволяет установить следующие характеристики:

Unique	Установка этой опции не позволяет индексировать таблицу, если в ней находятся дубликаты совокупности включенных в индекс полей. Например, установка этой опции для fio не допустила бы наличие в таблице однофамильцев.
Descending	При установке этой опции таблица будет упорядочиваться по степени убывания значений (по умолчанию упорядочивание производится по степени нарастания значений). Например, этот индикатор можно заключить для индекса MarkGr, чтобы в таблице сначала были заданы отличники
Case Sensitive	При установке этой опции будет приниматься во внимание регистр, в котором введены символы.
Maintained	Если эта опция установлена, то индекс обновляется при каждом изменении в таблице. В противном случае индекс обновляется только в момент связывания с таблицей или передачи в нее запроса. Это несколько замедляет обработку запросов. Поэтому полезно включить эту опцию для обновляемых таблиц. Если таблица используется только для чтения, эту опцию лучше не включать.

После того как индекс сформирован, щелкаете по кнопке ОК. Открывается окно, в котором вы задаете имя индекса.

После того как все необходимые данные о структуре таблицы введены, щелкнете по кнопке Save as (сохранить, как) и перед вами откроется окно, напоминающее обычный диалог сохранения в файле. От обычного это окно отличается выпадающим списком Alias. Этот список содержит псевдонимы различных баз данных (о них пойдет речь позднее), из которого вы можете выбрать базу данных, в которую будете сохранять свою таблицу. Если вам не надо сохранять таблицу в одной из существующих баз данных, то вы можете воспользоваться обычным списком Сохранить в верхней части окна. При этом вы с помощью обычной быстрой кнопки можете создать новую папку (каталог). Мы создаем базу данных типа Paradox. Для Paradox база данных – это каталог, в котором сохраняется таблица. Так что создайте с помощью быстрой кнопки, имеющейся в окне диалога, каталог, который будет вашей базой данных, и сохраните в нем сформированную таблицу, задав ей имя Pers. Если вы работаете в сети, то можете сохранить таблицу не на своем, а на другом, серверном компьютере.

Выше было рассмотрено задание структуры таблицы Pers в табл. 1. Вам также надо создать таблицу Gro, описанную в табл. 2. Она проще: в ней всего два поля Grl и Teach . Типы обоих полей – строки. Поле Grl является ключевым. Вторичные индексы создавать не надо. Сохраните эту таблицу в том же каталоге, в котором вы сохранили предыдущую таблицу.

После того как вы создали таблицы, вы можете их открыть командой File/Open. После выполнения этой команды и выбора файла таблицы вы увидите окно вида, представленного ранее на рис. 3. Программа Database Desktop автоматически не настраивается на русский язык. Но при использовании таблицы в приложении все надписи будут выглядеть нормально.

Команда Table/Restructure позволяет изменить структуру таблицы или какие-то ее характеристики. При выполнении этой команды вы попадаете в окно, аналогичное рассмотренному ранее при разработке структуры, и сможете внести в структуру таблицы необходимые изменения, задать дополнительные индексы и т. п.

Вы создали базу данных, содержащую две таблицы. Теперь вам надо создать для нее псевдоним, по которому приложения будут ссылаться на эту БД. Псевдонимы можно просматривать и создавать в Database Desktop, выполнив команду Tools/Alias Manager. Вы увидите

диалоговое окно Alias Manager (диспетчер псевдонимов), вид которого представлен на рис. 4. Впрочем, вид этого окна существенно зависит от того, псевдоним какой базы данных просматривается. Индикатор Public alias (открытый псевдоним) в верхней части окна показывает, будет ли создаваться открытый, доступный всем псевдоним, или псевдоним проекта. Ниже расположен выпадающий список Database Alias, в котором вы можете выбрать интересующий вас псевдоним из числа уже созданных.

При выборе псевдонима в списке Database Alias автоматически изменяется тип драйвера в выпадающем списке Driver type и расположенная ниже информация о драйвере. Для драйвера типа STANDARD, который используется для баз данных PARADOX и многих других, эта информация сводится к указанию пути к базе данных в окне Path. Если псевдоним базы данных уже существует, но путь к ней изменился, достаточно изменить файл ID API32.CFG, в котором сохраняется конфигурация BDE. Все приложения, использующие данный псевдоним, даже не почувствуют, что местоположение базы данных изменилось.

Но в нашем случае псевдонима еще нет, и его надо создать. Щелкните для этого на кнопке New (новый). В результате окно несколько преобразуется: вместо кнопки New появится кнопка Keep New – сохранить новый псевдоним, и появится возможность в списке Database Alias не только выбирать существующий псевдоним, но и писать новый.

В нашем примере вам надо в открывшемся окне:

- Выбрать драйвер STANDARD в списке Driver type.
- Указать каталог хранения таблиц. В этом может помочь кнопка просмотра Browse.
- В окне Database Alias написать новый псевдоним. Давайте зададим псевдоним Learn.
- Щелкнуть на кнопке Keep New, чтобы сохранить введенную информацию, сохранить всю информацию кнопкой Save as и выйти из диалога.

6.3.2.3. Итоговое тестирование

Задание 1

Главной целью информатизации является

наиболее полное удовлетворение потребностей общества в информации во всех сферах деятельности

создание новой инфраструктуры и сектора услуг, способных поддержать национальную экономику

внедрение компьютерных и телекоммуникационных технологий

замена экономической структуры, основанной на тяжелой промышленности, структурой, базирующейся на наукоемких областях

Задание 2

Установите соответствие между основными понятиями технологии программирования и их определением

Понятие	Определение
1. программирование	А. программные продукты и техническая документация к ним
2. программа	Б. сфера деятельности, направленная на создание программ
3. приложение	В. последовательность команд компьютера, приводящая к решению задачи
4. программное обеспечение	Г. комплекс взаимосвязанных программ, предназначенный для реализации определенной задачи массового спроса
5. программный продукт	Д. программная реализация на компьютере решения задачи

Запишите выбранные цифры под соответствующими буквами:

А	Б	В	Г	Д

Задание 3

Установить правильную последовательность действий при разработке функциональной структуры алгоритма приложения

№	Этапы разработки функциональной структуры алгоритма приложения
1	Устанавливается состав приложений, обеспечивающих реализацию поставленных целей
2	определяются необходимые для решения задач функции обработки данных
3	определяются цели автоматизации предметной области и их иерархия
4	Выполняется декомпозиция функций обработки данных до необходимой структурной сложности
5	уточняется характер взаимосвязи приложений и их основные характеристики

Запишите соответствующую последовательность цифр слева направо:

--	--	--	--	--

Задание 4

Разработка программы начинается с этапа:

- разработки алгоритма
- перевода алгоритма на язык программирования
- анализа требований
- тестирования

Задание 5

Алгоритм называется циклическим:

- если его выполнение предполагает многократное повторение одних и тех же действий
- если ход его выполнения зависит от истинности тех или иных условий
- если его команды выполняются в порядке их естественного следования независимо от каких-либо условий
- если он включает в себя вспомогательный алгоритм

Задание 6

Ошибки, связанные с неправильной записью конструкций языка программирования, называют:

- логическими
- орфографическими
- синтаксическими
- методическими.

Задание 7

Из каких двух основных частей состоит пользовательский интерфейс:

- из простого интерфейса и руководства по использованию этого интерфейса
- из сложного меню и кнопки "Выход из программы"
- из монитора и клавиатуры
- из форм и объектов

Задание 8

Достоинством языка программирования Visual Basic является:

- оптимальное сочетание простоты использования, доступности и большого набора разнообразных возможностей, позволяющие охватить все основные области программной деятельности
- возможность работы с таймером, доступ к базам данных
- реализация разработчиком гибкого и удобного интерфейса для своего приложения
- поддержка сети Интернет, позволяющая организовать связь с другими компьютерами, подключенными к сети

Задание 9

Функция в Visual Basic, которая выводит на экран окно, используемое для отображения какой-либо информации или сообщения: : _____

Задание 10

Оператор Sub в программе на языке программирования Visual Basic задает:

- изменяет значение аргумента
- ввод значений переменных
- название программы
- описание переменных

Задание 11

Верхний уровень меню это ...

- строка меню, в которой находятся элементы главного меню
- список элементов управления
- раскрывающийся список элементов
- панель задач

Задание 12

Для построения диаграммы в Visual Basic используется объект:

- Chart
- ChartWizard
- Gallery
- PlotBy

Задание 13

Функция Sheets(“лист3”).Select в Visual Basic:

1. позволяет выбрать рабочий лист книги
2. выводит на экран окно, которое используется для отображения какой-либо информации или сообщения
3. позволяет очищать область вывода данных
4. отображает окно диалога для ввода данных в текстовое поле, описание которых может быть задано как значения некоторых параметров

Ответ:

Обоснование:

Задание 14

Для создания справочной системы в Visual Basic предназначена утилита:

- Single record
- Grid (Datasheet)
- Master/Detail
- HTML Help WorkShop

Задание 15

Стандартные компоненты Delphi, который позволяет поместить главное меню в программу:

- TMainMenu
- TPopupMenu
- TLabel
- TEdit

Задание 16

Вкладка Delphi, которая содержит основные свойства окно, такие как тип окна: произвольная строка, задаваемая пользователем:

- General
- Buttons
- Position
- Styles

Задание 17

Набор DLL-библиотек, обеспечивающих низкоуровневый доступ к локальным и клиент-серверным БД:

- Borland Database Engine
- SQL Links
- BDE Administrator
- Database Desktop

Задание 18

Свойство, которое соответствует заголовку столбца данного поля:

- Alignment
- Display Label
- Display Width
- Readonly

Задание 19

SQL, который используется в специальных утилитах, позволяющих в интерактивном режиме вводить запросы с использованием команд SQL, посылая их для выполнения на сервер и получать результаты в предназначенном для этого окне – это _____ SQL

Ответ:

Задание 20

SQL, который используется в прикладных программах, позволяя им посылать запросы к серверу и обрабатывать полученные результаты, в том числе комбинируя set-ориентированный и record-ориентированный подходы.

- интерактивный SQL
- встроенный SQL
- оперативный SQL
- многомерный SQL

Задание 21

Объектно-ориентированный подход использует:

- объектную декомпозицию
- структурный алгоритм
- методологию программирования на основе системного подхода к анализу, проектированию и реализации программного обеспечения

последовательное разложение функций обработки данных на простые функциональные элементы

Задание 22

Методология программирования на основе системного подхода к анализу, проектированию и реализации программного обеспечения – это _____ программирование

Ответ:

Задание 23

Язык манипулирования данными используется для ...

- создания и изменения структуры базы данных и ее составных частей
- управления правами доступа к данным и выполнением процедур в многопользовательской среде
- манипулирования данными в таблицах баз данных
- создания запросов к БД

Задание 24

Язык определения данных используется для ...

- создания и изменения структуры базы данных и ее составных частей
- управления правами доступа к данным и выполнением процедур в многопользовательской среде
- манипулирования данными в таблицах баз данных
- создания запросов к БД

Задание 25

Представлен фрагмент программы:

while логическое выражение:

операторы циклической части программы

print()

это:

- форма записи оператора цикла с предусловием
- форма записи оператора цикла с постусловием
- форма записи оператора цикла с параметром
- форма записи массива данных

Задание 26

Задан фрагмент программы

```
for i in range(1, m):
```

```
    s=0
```

```
    for j in range(1, n):
```

```
        if a[i,j]>0:
```

```
            s=s+a[i,j]
```

Эта программа позволяет:

- для матрицы A вычислить и запомнить сумму положительных элементов каждого столбца матрицы
- для матрицы A вычислить и запомнить сумму отрицательных элементов каждого столбца матрицы.
- для матрицы A вычислить и запомнить сумму положительных элементов каждой строки матрицы
- для матрицы A вычислить и запомнить сумму отрицательных элементов каждой строки матрицы.

Задание 27

В записи функции «InputBox (подсказка,[заголовок], [значение], [x], [y])», значение – это ...
строка для отображения в окне подсказки относительно вводимого значения
принимаемое по умолчанию значение, которое выводится автоматически в текстовом поле
строковое значение, задающее название окна
координаты левого верхнего угла окна диалога

Задание 28

Функция Sheets(“лист3”).Select в Visual Basic:

- позволяет выбрать рабочий лист книги
- выводит на экран окно, которое используется для отображения какой-либо информации или сообщения
- позволяет очищать область вывода данных
- отображает окно диалога для ввода данных в текстовое поле, описание которых может быть задано как значения некоторых параметров

Задание 29

Объект в Visual Basic, позволяющий манипулировать графикой в различных форматах

- Picture
- PictureBox
- Image
- Image Line

Задание 30

Для построения диаграммы в Visual Basic используется объект:

- Chart
- ChartWizard
- Gallery
- PlotBy

Задание 31

Элементы синтаксиса метода ChartWizard при построении графика, который задает место расположения рядов данных:

- PlotBy
- Source
- Gallery
- Format

Задание 32

Команда в Visual Basic, предназначенная для проверки орфографии:

- CategoryTitle
- CheckSpelling
- PlotBy
- CategoryLabels

Задание 33

Логический оператор, описывающих условие в SQL запросах, который сравнивает поле с заданным значением, имеет вид:

- =
- Like
- In
- Between

Задание 34

Тип поля в Delphi - число в денежном формате:

Alpha
Number
Money
Short

Задание 35

Выражения для получения значений вычисляемых полей разработчик должен разместить в методе-обработчике набора данных ...

DBDEMOS
TTable
TQuery
OnCalcFields

Задание 36

Фильтры, которые можно создавать и редактировать во время выполнения приложения, для них используется только свойство Filter, называются _____

Ответ:

Задание 37

Свойство DataBaseName компонента Query служит для ...

задания базы данных, с которой осуществляется связь путем выбора из выпадающего списка псевдонимов или указанием полного пути к файлу
управления отображением данных в компоненте Query
использования динамических запросов, содержащих параметры.
получения результата запроса после задания параметра, надо повторно выполнить запрос

Задание 38

Запись оператора SELECT * означает, что

в результирующий набор данных необходимо включить все поля
в результирующий набор данных не включается ни одного поля
в результирующий набор данных необходимо включить только первое поле
в результирующий набор данных необходимо включить первые 3 поля

6.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

С целью определения уровня овладения компетенциями, закрепленными за дисциплиной, в заданные преподавателем сроки проводится текущий и промежуточный контроль знаний, умений и навыков каждого обучающегося. Все виды текущего контроля осуществляются на практических занятиях. Исключение составляет устный опрос, который может проводиться в начале или конце лекции в течение 10-15 мин. с целью закрепления знаний терминологии по дисциплине. При оценке компетенций принимается во внимание формирование профессионального мировоззрения, определенного уровня включённости в занятия, рефлексивные навыки, владение изучаемым материалом.

Процедура оценивания компетенций обучающихся основана на следующих стандартах:

1. Периодичность проведения оценки.
2. Многоступенчатость: оценка (как преподавателем, так и обучающимися группы) и самооценка обучающегося, обсуждение результатов и комплекс мер по устранению недостатков.

3. Единство используемой технологии для всех обучающихся, выполнение условий сопоставимости результатов оценивания.

4. Соблюдение последовательности проведения оценки.

Текущая аттестация обучающихся.

Текущая аттестация по дисциплине «Высокоуровневые методы информатики и программирования» проводится в форме опроса и контрольных мероприятий по оцениванию фактических результатов обучения обучающихся и осуществляется преподавателем дисциплины.

Объектами оценивания выступают:

1. учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине);
2. степень усвоения теоретических знаний в качестве «ключей анализа»;
3. уровень овладения практическими умениями и навыками по всем видам учебной работы;
4. результаты самостоятельной работы (изучение книг из списка основной и дополнительной литературы).

Активность обучающегося на занятиях оценивается на основе выполненных обучающимся работ и заданий, предусмотренных данной рабочей программой дисциплины.

Кроме того, оценивание обучающегося проводится на текущем контроле по дисциплине. Оценивание обучающегося на контрольной неделе проводится преподавателем независимо от наличия или отсутствия обучающегося (по уважительной или неуважительной причине) на занятии. Оценка носит комплексный характер и учитывает достижения обучающегося по основным компонентам учебного процесса за текущий период.

Оценивание обучающегося носит комплексный характер и учитывает достижения обучающегося по основным компонентам учебного процесса за текущий период с выставлением оценок в ведомости.

Промежуточная аттестация обучающихся. Промежуточная аттестация по дисциплине «Высокоуровневые методы информатики и программирования» проводится в соответствии с учебным планом для очной, очно-заочной и заочной форм обучения в виде экзамена в период экзаменационной сессии в соответствии с графиком проведения.

Обучающиеся допускаются к экзамену по дисциплине в случае выполнения им учебного плана по дисциплине: выполнения всех заданий и мероприятий, предусмотренных программой дисциплины.

Оценка знаний обучающегося на экзамене определяется его учебными достижениями в семестровый период и результатами текущего контроля знаний и выполнением им заданий.

Знания умения, навыки обучающегося на зачете оцениваются как: «зачтено», «не зачтено»; на экзамене – как: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Основой для определения оценки служит уровень усвоения обучающимися материала, предусмотренного данной рабочей программой.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебник для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 241 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583784>

2. Жилко Е.П. Информатика и программирование. Часть 1 [Электронный ресурс]: учебное пособие/ Жилко Е.П., Титова Л.Н., Дямина Э.И.— Электрон. текстовые данные.— Москва: Ай Пи Ар Медиа, 2022.— 195 с.— Режим доступа: <http://www.iprbookshop.ru/95153.html>.— ЭБС «IPRbooks»

3. Сузи, Р. А. Язык программирования Python: учебное пособие / Р. А. Сузи. — 4-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024.

— 350 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/142310.html>

б) дополнительная литература

1. Гаско, Р. Простой Python просто с нуля / Р. Гаско. — Москва : СОЛОН-ПРЕСС, 2023. — 256 с. — ISBN 978-5-91359-334-4. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/142050.html>

2. Давыдова Н.А. Программирование [Электронный ресурс]: учебное пособие/ Давыдова Н.А., Боровская Е.В.— Электрон. текстовые данные.— Москва: Лаборатория знаний, 2020.— 239 с.— Режим доступа: <http://www.iprbookshop.ru/6485.html>.— ЭБС «IPRbooks»

3. Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс]/ Златопольский Д.М.— Электрон. текстовые данные.— Москва: Лаборатория знаний, 2020.— 224 с.— Режим доступа: <http://www.iprbookshop.ru/12264.html>.— ЭБС «IPRbooks»

4. Терехов А.Н. Технология программирования [Электронный ресурс]: учебное пособие/ Терехов А.Н.— Электрон. текстовые данные.— Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.— 148 с.— Режим доступа: <http://www.iprbookshop.ru/97587.html>.— ЭБС «IPRbooks»

5. Файфель Б.Л. Основы языка Python : учебное пособие / Файфель Б.Л., Колобова А.Е.. — Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2024. — 284 с. — ISBN 978-5-7433-3614-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/147706.html> (дата обращения: 09.03.2026). — Режим доступа: для авторизир. пользователей

8. Методические указания для обучающихся по освоению дисциплины

Вид деятельности	Методические указания по организации деятельности студента
Лекция	Лекция – форма обучения студентов, при которой преподаватель последовательно излагает основной материал темы учебной дисциплины. Лекция – это важный источник информации по каждой учебной дисциплине. Она ориентирует студента в основных проблемах изучаемого курса, направляет самостоятельную работу над ним. Для лекций по каждому предмету должна быть отдельная тетрадь для лекций. Прежде всего, запишите имя, отчество и фамилию лектора, оставьте место для списка рекомендованной литературы, пособий, справочников. Будьте внимательны, когда лектор объявляет тему лекции, объясняет Вам место, которое занимает новый предмет в Вашей подготовке и чему новому Вы сможете научиться. Опытный студент знает, что, как правило, на первой лекции преподаватель обосновывает свои требования, раскрывает особенности чтения курса и способы сдачи зачета или экзамена. Отступите поля, которые понадобятся для различных пометок, замечаний и вопросов. Запись содержания лекций очень индивидуальна, именно поэтому трудно пользоваться чужими конспектами. Не стесняйтесь задавать вопросы преподавателю! Чем больше у Вас будет информации, тем свободнее и увереннее Вы будете себя чувствовать! Базовые рекомендации: - не старайтесь дословно конспектировать лекции, выделяйте основные положения, старайтесь понять логику лектора; - точно записывайте определения, законы, понятия, формулы и т.д.; - передавайте излагаемый материал лектором своими словами; - наиболее важные положения лекции выделяйте подчеркиванием; - создайте свою систему сокращения слов; - привыкайте просматривать, перечитывать перед новой лекцией предыдущую информацию; - дополняйте материал лекции информацией; - задавайте вопросы лектору;

	- обязательно вовремя восполняйте возникшие пробелы. Правила тактичного поведения и эффективного слушания на лекциях: - Слушать (и слышать) другого человека - это настоящее искусство, которое очень пригодится в будущей профессиональной деятельности экономиста. - Если преподаватель «скучный», но Вы чувствуете, что он действительно владеет материалом, то скука - это уже Ваша личная проблема (стоит вообще спросить себя, а настоящий ли Вы студент, если Вам не интересна лекция специалиста?). Существует очень полезный прием, позволяющий студенту-экономисту оставаться в творческом напряжении даже на лекциях заведомо «неинтересных» преподавателей. Прием прост – постарайтесь всем своим видом показать, что Вам «все-таки интересно» и Вы «все-таки верите», что преподаватель вот-вот скажет что-то очень важное. И если в аудитории найдутся хотя бы несколько таких студентов, внимательно и уважительно слушающих преподавателя, то может произойти «маленькое чудо», когда преподаватель «вдруг» заговорит с увлечением, начнет рассуждать смело и с озорством (иногда преподаватели сами ищут в аудитории внимательные и заинтересованные лица и начинают читать свои лекции, частенько поглядывая на таких студентов, как бы «вдохновляясь» их доброжелательным вниманием). Если это кажется невероятным (типа того, что «чудес не бывает»), просто вспомните себя в подобных ситуациях, когда с приятным собеседником-слушателем Вы вдруг обнаруживаете, что говорите намного увереннее и даже интереснее для самого себя. - Если Вы в чем-то не согласны с преподавателем, то совсем не обязательно тут же перебивать его и, тем более, высказывать свои представления, даже если они и кажутся Вам верными. Перебивание преподавателя на полуслове - это верный признак невоспитанности. Вопросы следует задавать либо после занятий (для этого их надо кратко записать, чтобы не забыть), либо выбрав момент, когда преподаватель сделал хотя бы небольшую паузу, и обязательно извинившись. Правила конспектирования на лекциях: - Не следует пытаться записывать подряд все то, о чем говорит преподаватель. Даже если студент владеет стенографией, записывать все высказывания просто не имеет смысла: важно уловить главную мысль и основные факты. - Желательно оставлять на страницах поля для своих заметок (и делать эти заметки либо во время самой лекции, либо при подготовке к семинарам и экзаменам). - Естественно, желательно использовать при конспектировании сокращения, которые каждый может «разработать» для себя самостоятельно (лишь бы самому легко было потом разобраться с этими сокращениями). - Стараться поменьше использовать на лекциях диктофоны, поскольку потом трудно будет «декодировать» неразборчивый голос преподавателя, все равно потом придется переписывать лекцию (а с голоса очень трудно готовиться к ответственным экзаменам), наконец, диктофоны часто отвлекают преподавателя тем, что студент ничего не делает на лекции (за него, якобы «работает» техника) и обычно просто сидит, глядя на преподавателя немигающими глазами (взглядом немного скучающего «удава»), а преподаватель чувствует себя неудобно и вместо того, чтобы свободно размышлять над проблемой, читает лекцию намного хуже, чем он мог бы это сделать (и это не только наши личные впечатления: очень многие преподаватели рассказывают о подобных случаях).
Практические занятия	Практическое занятие – это одна из форм учебной работы, которая ориентирована на закрепление изученного теоретического материала, его более глубокое усвоение и формирование умения применять теоретические знания в практических, прикладных целях.

	Особое внимание на практических занятиях уделяется выработке учебных или профессиональных навыков. Такие навыки формируются в процессе выполнения конкретных заданий – упражнений, задач и т.п. – под руководством и контролем преподавателя. Готовясь к практическому занятию, тема которого всегда заранее известна, студент должен освежить в памяти теоретические сведения, полученные на лекциях и в процессе самостоятельной работы, подобрать необходимую учебную и справочную литературу. Только это обеспечит высокую эффективность учебных занятий. Отличительной особенностью практических занятий является активное участие самих студентов в объяснении вынесенных на рассмотрение проблем, вопросов; преподаватель, давая студентам возможность свободно высказаться по обсуждаемому вопросу, только помогает им правильно построить обсуждение. Такая учебная цель занятия требует, чтобы учащиеся были хорошо подготовлены к нему. В противном случае занятие не будет действенным и может превратиться в скучный обмен вопросами и ответами между преподавателем и студентами. При подготовке к практическому занятию: - проанализируйте тему занятия, подумайте о цели и основных проблемах, вынесенных на обсуждение; - внимательно прочитайте материал, данный преподавателем по этой теме на лекции; - изучите рекомендованную литературу, делая при этом конспекты прочитанного или выписки, которые понадобятся при обсуждении на занятии; - постарайтесь сформулировать свое мнение по каждому вопросу и аргументировать его обосновать; - запишите возникшие во время самостоятельной работы с учебниками и научной литературой вопросы, чтобы затем на практическом занятии получить на них ответы. В процессе работы на практическом занятии: - внимательно слушайте выступления других участников занятия, старайтесь соотнести, сопоставить их высказывания со своим мнением; - активно участвуйте в обсуждении рассматриваемых вопросов, не бойтесь высказывать свое мнение, но старайтесь, чтобы оно было подкреплено убедительными доводами; - если вы не согласны с чьим-то мнением, смело критикуйте его, но помните, что критика должна быть обоснованной и конструктивной, т.е. нести в себе какое-то конкретное предложение в качестве альтернативы; - после практического занятия кратко сформулируйте окончательный правильный ответ на вопросы, которые были рассмотрены. Практическое занятие помогает студентам глубоко овладеть предметом, способствует развитию у них умения самостоятельно работать с учебной литературой и первоисточниками, освоению ими методов научной работы и приобретению навыков научной аргументации, научного мышления. Преподавателю же работа студента на практическом занятии позволяет судить о том, насколько успешно и с каким желанием он осваивает материал курса.
Самостоятельная работа	Самостоятельная работа проводится с целью: систематизации и закрепления полученных теоретических знаний и практических умений обучающихся; углубления и расширения теоретических знаний студентов; формирования умений использовать нормативную, правовую, справочную документацию, учебную и специальную литературу; развития познавательных способностей и активности обучающихся: творческой инициативы, самостоятельности, ответственности, организованности; формирование самостоятельности мышления, способностей к саморазвитию, совершенствованию и самоорганизации; формирования профессиональных компетенций; развитию исследовательских умений обучающихся. Формы и

	виды самостоятельной работы: чтение основной и дополнительной литературы – самостоятельное изучение материала по рекомендуемым литературным источникам; работа с библиотечным каталогом, самостоятельный подбор необходимой литературы; работа со словарем, справочником; поиск необходимой информации в сети Интернет; конспектирование источников; реферирование источников; составление аннотаций к прочитанным литературным источникам; составление рецензий и отзывов на прочитанный материал; составление обзора публикаций по теме; составление и разработка терминологического словаря; составление хронологической таблицы; составление библиографии (библиографической картотеки); подготовка к различным формам текущей и промежуточной аттестации (к тестированию, зачету, экзамену); выполнение домашних контрольных работ; самостоятельное выполнение практических заданий репродуктивного типа (ответы на вопросы, тесты; выполнение творческих заданий). Технология организации самостоятельной работы обучающихся включает использование информационных и материально-технических ресурсов образовательного учреждения: библиотеку с читальным залом, укомплектованную в соответствии с существующими нормами; учебно-методическую базу учебных кабинетов, лабораторий и зала кодификации; компьютерные классы с возможностью работы в сети Интернет; аудитории (классы) для консультационной деятельности; учебную и учебно-методическую литературу, разработанную с учетом увеличения доли самостоятельной работы студентов, и иные методические материалы. Перед выполнением обучающимися внеаудиторной самостоятельной работы преподаватель проводит консультирование по выполнению задания, который включает цель задания, его содержания, сроки выполнения, ориентировочный объем работы, основные требования к результатам работы, критерии оценки. Во время выполнения обучающимися внеаудиторной самостоятельной работы и при необходимости преподаватель может проводить индивидуальные и групповые консультации. Самостоятельная работа может осуществляться индивидуально или группами обучающихся в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности, уровня умений обучающихся. Контроль самостоятельной работы предусматривает: • соотнесение содержания контроля с целями обучения; объективность контроля; • валидность контроля (соответствие предъявляемых заданий тому, что предполагается проверить); • дифференциацию контрольно-измерительных материалов. Формы контроля самостоятельной работы: • просмотр и проверка выполнения самостоятельной работы преподавателем; • организация самопроверки, • взаимопроверки выполненного задания в группе; обсуждение результатов выполненной работы на занятии; • проведение письменного опроса; • проведение устного опроса; • организация и проведение индивидуального собеседования; организация и проведение собеседования с группой; • защита отчетов о проделанной работе.
Опрос	Опрос - это средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выявление объема знаний по определенному разделу, теме, проблеме и т.п. Проблематика, выносимая на опрос определена в заданиях для самостоятельной работы обучающегося, а также может определяться преподавателем, ведущим дисциплину. Во время проведения устного опроса обучающийся должен уметь обсудить с

Тестирование	преподавателем соответствующую проблематику на уровне диалога. Контроль в виде тестов может использоваться после изучения каждой темы курса. Итоговое тестирование можно проводить в форме: • компьютерного тестирования, т.е. компьютер произвольно выбирает вопросы из базы данных по степени сложности; • письменных ответов, т.е. преподаватель задает вопрос и дает несколько вариантов ответа, а обучающийся на отдельном листе записывает номера вопросов и номера соответствующих ответов. Для достижения большей достоверности результатов тестирования следует строить текст так, чтобы у обучающихся было не более 40 – 50 секунд для ответа на один вопрос. Итоговый тест должен включать не менее 40 вопросов по всему курсу. Значит, итоговое тестирование займет целое занятие. Оценка результатов тестирования может проводиться двумя способами: 1) по 5-балльной системе, когда ответы студентов оцениваются следующим образом: - «отлично» – более 80% ответов правильные; - «хорошо» – более 65% ответов правильные; - «удовлетворительно» – более 50% ответов правильные. Обучающиеся, которые правильно ответили менее чем на 70% вопросов, должны в последующем пересдать тест. При этом необходимо проконтролировать, чтобы вариант теста был другой; 2) по системе зачет-незачет, когда для зачета по данной дисциплине достаточно правильно ответить более чем на 70% вопросов.
Подготовка к экзамену	При подготовке к экзамену необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и др. Основное в подготовке к сдаче экзамена по дисциплине «Высокоуровневые методы информатики и программирования» - это повторение всего материала дисциплины, по которому необходимо сдавать экзамен. При подготовке к сдаче экзамена обучающийся весь объем работы должен распределять равномерно по дням, отведенным для подготовки к экзамену, контролировать каждый день выполнение намеченной работы. Подготовка к экзамену включает в себя три этапа: • самостоятельная работа в течение семестра; • непосредственная подготовка в дни, предшествующие экзамену по темам курса; • подготовка к ответу на задания, содержащиеся в билетах (тестах) экзамена. Для успешной сдачи экзамена по дисциплине «Высокоуровневые методы информатики и программирования» обучающиеся должны принимать во внимание, что: • все основные вопросы, указанные в рабочей программе, нужно знать, понимать их смысл и уметь его разъяснить; • указанные в рабочей программе формируемые компетенции в результате освоения дисциплины должны быть продемонстрированы студентом; • практические занятия способствуют получению более высокого уровня знаний и, как следствие, более высокой оценке на экзамене; • готовиться к экзамену необходимо начинать с первой лекции и первого семинара.

9. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Реализация образовательного процесса по дисциплине «Высокоуровневые методы информатики и программирования» осуществляется в следующих аудиториях:

1. Занятия **лекционного типа** - аудитория №503: 40 мест (20 столов, 40 стульев), 1 доска, 5 стендов, 1 стол преподавателя, 1 кафедра, вешалка напольная – 2 шт.

2. Для проведения **практических занятий** используется лаборатория для проведения практических занятий №404: 44 места (22 стола, 44 стула), 1 доска, 5 стендов, 1 кафедра, вешалка напольная – 1 шт, 12 ПЭВМ с выходом в Интернет, принтер – 1

3. Для **самостоятельной работы** студентов используется аудитория №506: 22 места (11 столов, 22 стула), 1 доска, 4 стенда, 1 кафедра, вешалка напольная – 1 шт, 10 ПЭВМ с выходом в Интернет, принтер - 1

4. Для **проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации** используется аудитория для текущего контроля и промежуточной аттестации №503: 40 мест (20 столов, 40 стульев), 1 доска, 5 стендов, 1 стол преподавателя, 1 кафедра, вешалка напольная – 2 шт.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, в том числе комплект лицензионного программного обеспечения, электронно-библиотечные системы, современные профессиональные базы данных и информационные справочные систем

10.1 Лицензионное программное обеспечение:

1. Операционная система Microsoft Windows Professional XP
2. Операционная система Microsoft Windows Professional 7
3. Программные средства Microsoft Office 2007, 2010, 2013 Russian
4. Программные средства Microsoft Office Professional Plus 2007, 2013 Russian
5. Программные средства Microsoft Windows Server Standard 2008 Russian
6. Программные средства Total Commander 7.x User license
7. Программные средства WinRAR 3.x Standard license
8. Программное средство Dreamweaver CS5 11.0 Win AOO License RU,
9. Программное средство ABBYY FineReader 10 Professional Edition,
10. Программное средство Total Commander 7.x User license,
11. Программное средство Visual Basic 6 по программе MSDN Academic Alliance
12. Программное средство Visual Studio Code
13. Пакет Embarcadero RAD Studio, среда программирования Delphi

10.2. Электронно-библиотечная система:

Электронная библиотечная система (ЭБС): <http://www.iprbookshop.ru/>

10.3. Современные профессиональные баз данных:

1. Официальный интернет-портал базы данных правовой информации <http://pravo.gov.ru>
2. Портал "Информационно-коммуникационные технологии в образовании" <http://www.ict.edu.ru>
3. Научная электронная библиотека <http://www.elibrary.ru/>
4. Национальная электронная библиотека <http://www.nns.ru/>
5. Электронные ресурсы Российской государственной библиотеки <http://www.rsl.ru/ru/root3489/all>
6. Web of Science Core Collection — политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных — <http://webofscience.com>
7. Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН) <http://neicon.ru>

8. www.minfin.ru Сайт Министерства финансов РФ
9. <http://gks.ru> Сайт Федеральной службы государственной статистики
10. www.skrin.ru База данных СКРИН (крупнейшая база данных по российским компаниям, отраслям, регионам РФ)
11. www.expert.ru Электронная версия журнала «Эксперт»
12. <http://ecsn.ru/> «Экономические науки»

10.4. Информационные справочные системы:

1. Справочно-правовая система «КонсультантПлюс»
2. Справочная правовая система «Гарант»

Рабочую программу дисциплины составила:

Гришанова Татьяна Валерьевна, преподаватель кафедры информатики и программного обеспечения Частного образовательного учреждения высшего образования «Брянский институт управления и бизнеса».