

Министерство науки и высшего образования
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Кафедра «Математическое обеспечение и применение ЭВМ»

УТВЕРЖДАЮ
Первый проректор
И.В. Макурин
28 2017 г.



РАБОЧАЯ ПРОГРАММА

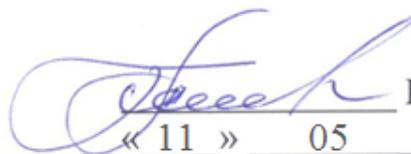
дисциплины «Разработка интерфейса пользователя»

основной профессиональной образовательной программы
подготовки бакалавров
по направлению 09.03.01 «Информатика и вычислительная техника»
профиль «Программное обеспечение средств вычислительной техники и
автоматизированных систем»

Форма обучения	заочная
Технология обучения	традиционная

Комсомольск-на-Амуре 2017

Автор рабочей программы
профессор, к.т.н

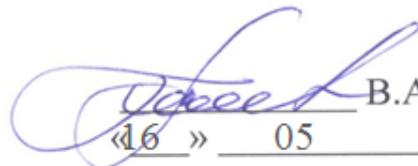
 В.А.Тихомиров
« 11 » 05 2017 г.

СОГЛАСОВАНО

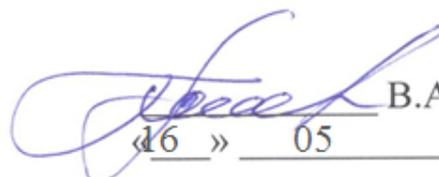
Директор библиотеки

 И.А. Романовская
« 18 » 05 2017 г.

Заведующий кафедрой «Математическое обеспечение и применение ЭВМ», кандидат технических наук, профессор

 В.А. Тихомиров
« 16 » 05 2017 г.

Заведующий выпускающей кафедрой «Математическое обеспечение и применение ЭВМ», кандидат технических наук, профессор

 В.А. Тихомиров
« 16 » 05 2017 г.

Декан ФЗДО , кандидат технических наук, доцент

 М.В.Семибратова
« 20 » 05 2017 г.

Начальник учебно-методического управления

 Е.Е. Поздеева
« 20 » 05 2017 г.

Введение

Рабочая программа дисциплины «Разработка интерфейса пользователя» составлена в соответствии с требованиями федерального государственного образовательного стандарта, утвержденного приказом Министерства образования и науки Российской Федерации от 12.01.2016 № 5, и образовательной программы подготовки бакалавров по направлению 09.03.01 «Информатика и вычислительная техника». Набор заочного отделения 2017, 2018 года.

1 Аннотация дисциплины

Наименование дисциплины	Разработка интерфейса пользователя							
Цель дисциплины	формирование у студентов практических навыков организации интерфейса с помощью средств языков высокого уровня							
Задачи дисциплины	<ul style="list-style-type: none">- обучить тенденциям развития пользовательских интерфейсов;- обучить новым компьютерным технологиям и методам повышения полезности разрабатываемых и используемых программных систем;- обучить особенностям восприятия информации человеком;- обучить устройству и режимам человеко-машинного диалога;- обучить компьютерному представлению и визуализации информации;- обучить критериям оценки полезности диалоговых систем;- обучить описанию взаимодействия пользователя с компьютерной средой в заданной проблемной области;- обучить использованию программ поддержки разработки пользовательских интерфейсов;- обучить создавать среду, описывать события и реализовывать интерактивную систему.							
Основные разделы дисциплины	Пользовательский интерфейс Психология человека и компьютера. Проектирование пользовательского интерфейса. Правила проектирования пользовательского интерфейса. Этапы проектирование пользовательского интерфейса. Инструментарий разработчика пользовательского интерфейса. Тестирование пользовательского интерфейса.							
Общая трудоемкость дисциплины	4 з.е. / 144 академических часов							
	Семестр	Аудиторная нагрузка, ч				СРС, ч	Промежуточная аттестация, ч	Всего за семестр, ч
		Число недель	Лекции	Лаб. работы	Курсовое проектирование			
8 семестр	18	4	8		123	9	144	
ИТОГО:		18	4	8		123	9	144

Трудоемкость дисциплины в таблице представлена для двух случаев учебного года, когда учебный семестр составляет 18 недель.

2 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы

Дисциплина «Разработка интерфейса пользователя» нацелена на формирование компетенций, знаний, умений и навыков, указанных в таблице 1.

Таблица 1 – Компетенции, знания, умения, навыки

Наименование и шифр компетенции, в формировании которой принимает участие дисциплина	Перечень формируемых знаний, умений, навыков, предусмотренных образовательной программой		
	Перечень знаний (с указанием шифра)	Перечень умений (с указанием шифра)	Перечень навыков (с указанием шифра)
способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования (ПК-2)	Методы и средства проектирования программных интерфейсов 33(ПК-2-7)	Использовать существующие типовые решения и шаблоны проектирования программного обеспечения УЗ(ПК-2-7)	Навыками проектирования программных интерфейсов НЗ(ПК-2-7)

3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина(модуль) «Разработка интерфейса пользователя» изучается на 4 курсе в 8 семестре.

Дисциплина является дисциплиной по выбору, входит в состав блока Б1 «Дисциплины (модули)» и относится к вариативной части.

Для освоения дисциплины необходимы общие знания, умения, навыки работы с ПЭВМ, навыки программирования на языках высокого уровня.

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

Общая трудоемкость (объем) дисциплины составляет 4 зачетных единиц, 144 академических часов.

Распределение объема дисциплины (модуля) по видам учебных занятий представлено в таблице 2.

Таблица 2 – Объем дисциплины (модуля) по видам учебных занятий

Объем дисциплины	Заочная форма обучения (Час)
Число недель в семестре	18
Общая трудоемкость дисциплины	144
Контактная аудиторная работа обучающихся с преподавателем (по видам учебных занятий), всего	12
В том числе:	
занятия лекционного типа (лекции и иные учебные занятия, предусматривающие преимущественную передачу учебной информации педагогическими работниками)	4
занятия семинарского типа (семинары, практические занятия, практикумы, лабораторные работы, коллоквиумы и иные аналогичные занятия)	8
Самостоятельная работа обучающихся и контактная работа, включающая групповые консультации, индивидуальную работу обучающихся с преподавателями (в том числе индивидуальные консультации); взаимодействие в электронной информационно-образовательной среде вуза	123
Промежуточная аттестация обучающихся	9

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

Таблица 3 – Структура и содержание дисциплины (модуля)

Наименование разделов, тем и содержание материала	Компонент учебного плана	Трудоемкость (в часах)			Форма проведения	Планируемые (контролируемые) результаты освоения	
		Для графика 18 недель в семестрах				Компетенции	Знания, умения, навыки
Раздел 1 Проектирование человеко-машинного интерфейса							
Тема 1 Особенности графического интерфейса. Объектный подход к проектированию интерфейса. Компоненты графического интерфейса. Взаимодействие пользователя с приложением. Общие правила взаимодействия с объектами.	Лекция	1			Традиционная	ПК-2	ЗЗ(ПК-2-7)
Тема 2 Понятие удобства применения программного продукта. Важность тестирования на удобство применения программного обеспечения. Цели и задачи тестирования. Условие успеха программных продуктов. Отчетные результаты теста.	Лекция	1			Традиционная	ПК-2	ЗЗ(ПК-2-7)
Тема 3 Пользовательский интерфейс WEB- приложений. WEB – страницы и сайты. Пользовательский интерфейс системы реального времени. Средства разработки WEB- документов.	Лекция	2			Традиционная	ПК-2	ЗЗ(ПК-2-7)
Стандартные элементы пользовательского интерфейса	Лабораторная работа	2			Традиционная	ПК-2	УЗ(ПК-2-7) НЗ(ПК-3-1)

Наименование разделов, тем и содержание материала	Компонент учебного плана	Трудоемкость (в часах)			Форма проведения	Планируемые (контролируемые) результаты освоения	
		Для графика 18 недель в семестрах				Компетенции	Знания, умения, навыки
Прототипирование и визуализация пользовательского интерфейса	Лабораторная работа	2			Традиционная	ПК-2	У3(ПК-2-7) НЗ(ПК-2-7)
Диалоговый интерфейс: разработка сценариев и структуры диалога	Лабораторная работа	4			Традиционная	ПК-2	У3(ПК-2-7) НЗ(ПК-2-7)
Тенденции и перспективы развития человеко-машинных интерфейсов	Самостоятельная работа обучающихся	120			Чтение основной и дополнительной литературы, конспектирование	ПК-2	У3(ПК-2-7) НЗ(ПК-2-7)
		3			Индив. консультации с препод.	ПК-2	У3(ПК-2-7) НЗ(ПК-2-7)
Текущий контроль по разделу 1		Экзамен					
ИТОГО по дисциплине	Лекции	4			-	-	-
	Лабораторные работы	8			-	-	-
	Самостоятельная работа обучающихся	123			-	-	-
	Промежуточная аттестация	9 (Экзамен)					
ИТОГО: общая трудоемкость дисциплины (часов)		144					
в том числе с использованием активных методов обучения		20 часов					

6 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся, осваивающих дисциплину «Разработка интерфейса пользователя», состоит из следующих компонентов:

- изучение теоретических разделов дисциплины;
- подготовка к лабораторным занятиям;
- подготовка и оформление РГР.

Для успешного выполнения всех разделов самостоятельной работы учащимся рекомендуется использовать следующее учебно-методическое обеспечение:

Комплект электронных УММ для выполнения лабораторных работ и РГР по дисциплине «Разработка интерфейса пользователя» в локальной сети ФКТ по адресу \\3k316m04\Share\МОП_ЭВМ\1. Заочное\Бакалавры\ЧМВ.

Рекомендуемый график выполнения самостоятельной работы студента, для варианта семестра в 18 недель, представлен в таблице 4.

Самостоятельная работа студентов, реализуемая вне рамок аудиторных занятий, имеет следующую структуру:

- подготовка к лекциям;
- теоретическая подготовка к лабораторным занятиям;
- самостоятельное изучение отдельных теоретических разделов дисциплины «Разработка интерфейса пользователя»;
- выполнение РГР, подготовка к сдаче;
- подготовка к экзамену (отдельные 9 часов).

При подготовке к лекциям студент должен восстановить в памяти материал, разобранный в предыдущих лекциях, и освежить навыки практического использования этого материала на лабораторных работах.

Теоретическая подготовка к лабораторным занятиям требует знания пройденного лекционного материала, предварительного изучения методического материала по предстоящей к выполнению (или защите) лабораторной работы. Необходимо дополнительно ознакомиться с аналогичными темами, проиллюстрированными в интернет сообществе и на порталах вузов - аналогов.

РГР выполняется, начиная с середины семестра, когда у студента накапливается необходимый набор знаний и умений для её выполнения.

Таблица 4 – Рекомендуемый график выполнения самостоятельной работы студентов при 18-недельном семестре

Вид самостоятельной работы	Часов в неделю в семестре 8																		Итого по видам работ	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18		
Подготовка к лабораторным занятиям	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	54
Изучение теоретических разделов дисциплины	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3		41
Подготовка, оформление и защита РГР					2	2	2	3	4	4	4	4								25
Индивидуальные консультации с преподавателем				0,5			0,5			0,5			0,5			0,5		0,5		3
ИТОГО в 8 семестре	5	5	5	5,5	7	7	7,5	8	9	9,5	9	9	5,5	6	6	6,5	6	6,5		123

7 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)

Таблица 5 – Паспорт фонда оценочных средств

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	Показатели оценки
Стандартные элементы пользовательского интерфейса	ПК-2	Отчет по лабораторной работе № 1	Умеет использовать стандартные элементы интерфейса
Прототипирование и визуализация пользовательского интерфейса	ПК-2	Отчет по лабораторной работе № 2	Умеет выполнять визуализацию интерфейса в среде программирования
Диалоговый интерфейс: разработка сценариев и структуры диалога	ПК-2	Отчет по лабораторной работе № 3	Умеет использовать стандартные диалоговые окна
Тенденции и перспективы развития человеко-машинных интерфейсов	ПК-2	Отчет по выполнению РГР, экзамен	Умеет строить и тестировать программный интерфейс

Промежуточная аттестация в семестре проводится в форме экзамена.

Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, представлены в виде технологической карты дисциплины (таблица 6).

Таблица 6 – Технологическая карта

	Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
4 семестр <i>Промежуточная аттестация в форме экзамена</i>				
1	Отчеты по лабораторным работам (3 работы)	На сессии	10 баллов за одну работу	10 баллов – студент правильно выполнил практическое задание. Показал отличные знания в рамках освоенного учебного материала. 5 баллов – студент выполнил практическое задание с небольшими неточностями. Показал хорошие знания в рамках освоенного учебного материала.

	Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
	ты)			3 баллов – студент выполнил практическое задание с существенными неточностями. Показал удовлетворительные знания в рамках усвоенного учебного материала. 1 балл – при выполнении практического задания студент продемонстрировал недостаточный уровень знаний. 0 баллов – задание не выполнено.
2	Отчет по выполнению РГР	В конце семестра	20 баллов	20 баллов – студент правильно выполнил задание. Показал отличное владение навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. Ответил на все дополнительные вопросы на защите. 15 баллов – студент выполнил задание с небольшими неточностями. Показал хорошие владения навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. Ответил на большинство дополнительных вопросов на защите. 5 баллов – студент выполнил задание с существенными неточностями. Показал удовлетворительное владение навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. При ответах на дополнительные вопросы на защите было допущено много неточностей. 0 баллов – при выполнении задания студент продемонстрировал недостаточный уровень владения навыками применения полученных знаний и умений при решении профессиональных задач в рамках усвоенного учебного материала. При ответах на дополнительные вопросы на защите было допущено множество неточностей.
Текущий контроль:		-	50 баллов	
ИТОГО:		-	50 баллов	
3	Экзамен	Два вопроса оценивание уровня усвоенных знаний	25 баллов	25 баллов – студент дал полный и точный ответ на теоретический вопрос, ответил на дополнительные вопросы, 16 - студент ответил на теоретический вопрос билета с небольшими неточностями, показал хорошие знания в рамках усвоенного учебного материала, ответил на большинство дополнительных вопросов, 12-студент ответил на теоретический вопрос билета с существенными неточностями, показал удовлетворительные знания в рамках усвоенного учебного материала, при ответах на дополнительные вопросы было допущено много неточностей, 0 баллов - в ответе более трех грубых ошибок или задание не выполнено.

	Наименование оценочного средства	Сроки выполнения	Шкала оценивания	Критерии оценивания
		Практическое задание оценивание уровня усвоенных умений и навыков	25 баллов	25 баллов - студент правильно выполнил практическое задание билета, показал отличные умения в рамках освоенного учебного материала, продемонстрировал владение навыками разработки ПО, ответил на все дополнительные вопросы, 16- студент выполнил практическое задание билета с небольшими неточностями, показал хорошие умения и владение навыками разработки ПО в рамках освоенного учебного материала, ответил на большинство дополнительных вопросов, 10 - студент выполнил практическое задание билета с существенными неточностями, показал удовлетворительные умения и владение навыками разработки ПО в рамках освоенного учебного материала, при ответах на дополнительные вопросы было допущено много неточностей, 0 баллов - при выполнении практического задания билета студент продемонстрировал недостаточный уровень умений и навыков, при ответах на дополнительные вопросы были допущены грубые ошибки.
Текущий контроль:	-	50	-	-
Экзамен:	-	50	-	-
ИТОГО		100		
Критерии оценки результатов обучения по дисциплине: 0 – 64 % от максимально возможной суммы баллов – «неудовлетворительно» (недостаточный уровень для промежуточной аттестации по дисциплине); 65 – 74 % от максимально возможной суммы баллов – «удовлетворительно» (пороговый (минимальный) уровень); 75 – 84 % от максимально возможной суммы баллов – «хорошо» (средний уровень); 85 – 100 % от максимально возможной суммы баллов – «отлично» (высокий (максимальный) уровень)				

Задания для текущего контроля

Примеры заданий на лабораторные работы ЛАБОРАТОРНАЯ РАБОТА

Прикладной интерфейс. Меню

Составить иерархию предложенного понятия. Написать подпрограмму вызова иерархического меню. Для нижних узлов иерархии должны выводиться свойства соответствующих объектов. В вариантах заданий в скобках дан верхний уровень иерархии.

Варианты заданий

1. Блюда на завтрак (бутерброды, каши, холодные закуски).
2. Кухня народов СНГ (русская, узбекская, украинская).
3. Мясные блюда (говядина, баранина, птица).
4. Виды спорта (командные, некомандные).
5. Виды спорта (летние, зимние).
6. Виды спорта (на суше, в воздухе, на воде).
7. Птицы (оседлые, кочующие, перелетные).
8. Посуда (фарфоровая, стеклянная, деревянная, металлическая).
9. Обувь (зимняя, осенняя, летняя).
10. Одежда (зимняя, осенняя, летняя).
11. Грибы (съедобные, несъедобные).
12. Животные Африки (птицы, млекопитающие).
13. Ядовитые животные (насекомые, змеи).
14. Животные тропических лесов (насекомые, пресмыкающиеся, птицы, млекопитающие).
15. Горные животные (млекопитающие, птицы, насекомые).
16. Природные ресурсы (богатства недр, растительный мир, водные ресурсы).
17. Деревья (лиственные, хвойные).
18. Птицы (степные, лесные, горные).
19. Цветы (домашние, полевые, культурные).
20. Рыбы (морские, речные, озерные).
21. Холодные блюда и закуски (из овощей, из рыбы, мясные).
22. Морские животные (птицы, рыбы, млекопитающие).
23. Лесные животные (птицы, звери, насекомые).
24. Животные за полярным кругом (птицы, звери, ластоногие).
25. Города СНГ (Россия, Украина, Белоруссия).
26. Автомобили (легковые, грузовые, автобусы).
27. Суда (гражданские, военные).
28. Магазины (продовольственные, промышленных товаров, хозяйственные)
29. Самолеты (военные, гражданские, специализированные).
30. Легковые автомобили (отечественные, импортные).

ЛАБОРАТОРНАЯ РАБОТА

Модальные и немодальные диалоги

Написать программу, в которой используется диалог для работы со структурами данных, определяющими списки. Ввод элементов данных осуществляется с помощью окна редактирования **Edit Box**. Состояние списка выводится в окно списка **List Box**. Элемент списка удаляется, если по нему дважды щелкнуть курсором мыши.

Варианты заданий

Виды структур данных: очередь (S1), стек (S2), односвязный циклический список (S3). Тип данных элементов: char (T1), int (T2), unsigned (T3), float (T4), char* (T5). Вид диалога: модальный (D1), немодальный (D2). Предлагаются следующие варианты:

- | | | | | |
|-------------|--------------|--------------|--------------|--------------|
| 1) S1-T1-D1 | 7) S1-T2-D1 | 13) S1-T3-D1 | 19) S1-T4-D1 | 25) S1-T5-D1 |
| 2) S2-T1-D2 | 8) S2-T2-D2 | 14) S2-T3-D2 | 20) S2-T4-D2 | 26) S2-T5-D2 |
| 3) S3-T1-D1 | 9) S3-T2-D1 | 15) S3-T3-D1 | 21) S3-T4-D1 | 27) S3-T5-D1 |
| 4) S1-T1-D2 | 10) S1-T2-D2 | 16) S1-T3-D2 | 22) S1-T4-D2 | 28) S1-T5-D2 |
| 5) S2-T1-D1 | 11) S2-T2-D1 | 17) S2-T3-D1 | 23) S2-T4-D1 | 29) S2-T5-D1 |
| 6) S3-T1-D2 | 12) S3-T2-D2 | 18) S3-T3-D2 | 24) S3-T4-D2 | 30) S3-T5-D2 |

ЛАБОРАТОРНАЯ РАБОТА

Стандартные компоненты ввода и отображения текстовой информации

Написать программу в Borland C++ Builder, в которой вводятся два числа, затем к ним применяется заданная операция и результат выводится с помощью компонента заданного типа. Над окнами сделать надписи, указывающие о их назначении.

Варианты заданий

- E1 – Числа вводятся в два окна редактирования Edit
E2 – Числа вводятся в многострочное окно редактирования Memo
O1 – Вывод осуществляется на компонент Panel
O2 – Вывод осуществляется в окно списка ListBox
D1 – тип данных чисел – float
D2 – тип данных чисел – integer
+, -, *, / – выполняемые арифметические операции

Во всех вариантах проверить, введены ли числа правильно. Предлагаются следующие варианты:

- | | | | | |
|---------------|----------------|----------------|----------------|----------------|
| 1) E1-O1-D1 + | 7) E1-O2-D2 + | 13) E1-O1-D2 – | 19) E1-O2-D1 * | 25) E1-O1-D1 / |
| 2) E2-O1-D1 + | 8) E2-O2-D2 + | 14) E2-O1-D2 – | 20) E2-O2-D1 * | 26) E2-O1-D1 / |
| 3) E1-O2-D1 + | 9) E1-O1-D1 – | 15) E1-O2-D2 – | 21) E1-O1-D2 * | 27) E1-O2-D1 / |
| 4) E2-O2-D1 + | 10) E2-O1-D1 – | 16) E2-O2-D2 – | 22) E2-O1-D2 * | 28) E2-O2-D1 / |
| 5) E1-O1-D2 + | 11) E1-O2-D1 – | 17) E1-O1-D1 * | 23) E1-O2-D2 * | 29) E1-O1-D2 / |
| 6) E2-O1-D2 + | 12) E2-O2-D1 – | 18) E2-O1-D1 * | 24) E2-O2-D2 * | 30) E2-O1-D2 / |

Примеры заданий для РГР семестр 8

«Организация меню»

1. Написать и протестировать программу, имитирующую работу пользователя в режиме редактирования. При нажатии обычных клавиш на экран выводятся соответствующие им буквы. Если нажата одна из клавиш: «стрелка влево», «стрелка вверх», «стрелка вправо», «стрелка вниз», то курсор продвигается на одну позицию в указанном направлении. Исчезающее меню вызывается с помощью клавиши <F2>. В зависимости от варианта исчезающее меню выводится горизонтально, либо вертикально, и состоит из четырех опций.

2. Составить иерархию предложенного понятия. Написать подпрограмму вызова иерархического меню. Для нижних узлов иерархии должны выводиться свойства соответствующих объектов. В вариантах заданий в скобках дан верхний уровень иерархии.

«Построение диалоговых приложений»

1. Написать и протестировать программу, в которой используется диалог для работы со структурами данных, определяющими списки. Ввод элементов данных осуществляется с помощью окна редактирования Edit Box. Состояние списка выводится в окно списка List Box. Элемент списка удаляется, если по нему дважды щелкнуть курсором мыши.

«Интерактивная система для работы с числовыми данными»

1. Написать и протестировать программу, в которой вводятся два числа, затем к ним применяется заданная операция и результат выводится с помощью компонента заданного типа. Над окнами сделать надписи, указывающие о их назначении.

«Интерактивная система для исследования кривых на плоскости»

1. Написать и протестировать программу в среде с визуальным редактором интерфейса, в которой в качестве входных данных вводятся параметры (границы области, число точек, параметры функции, цвет графика), а выводится график функции одной переменной. Над окнами сделать надписи, указывающие о их назначении.

3. Написать функцию, получающую байт атрибутов и возвращающую 0, если цвета фона и тона в этом байте совпадают, и ненулевое значение, если они различны. При этом, интерфейс программы должен быть реализован с использованием одного из стандартных визуальных конструкторов.

4. Разработать описание пользовательского интерфейса для программы «Будильник».

Исходными данными для программы являются:

- время, на которое установлен будильник;
- флажок, показывающий, включен будильник или выключен;
- период повторения будильника: однократное срабатывание, включение каждый день, включение раз в неделю.

Для программы требуется разработать консольный интерфейс, интерфейс-меню и интерфейс со свободной навигацией. Результатом разработки является текстовое описание запросов, пунктов меню или компонентов интерфейса соответственно.

5. Написать программу квадратного уравнения, имеющую консольный интерфейс или интерфейс, основанный на простом меню. Все коэффициенты уравнения вводятся пользователем. При вводе коэффициентов должна предоставляться возможность редактирования вводимого значения.

Если уравнение имеет один корень, то нужно выводить только одно значение. Если уравнение не имеет корней, выводится соответствующее сообщение.

Замечание: при написании программы необходимо обращать внимание на отделение пользовательского интерфейса от вычислительной части.

6. Написать программу, строящую график функции, выбранной пользователем. Выбор функции осуществляется из некоторого фиксированного списка жестко заданных функций. Кроме того, программа должна предоставлять возможность ввода функции с клавиатуры.

Помимо выбора функции, требуется организовать ввод координат области, для которой будет строиться график.

Замечание: парсинг введенного текста можно реализовать рекурсией.

7. Реализовать программно элемент управления «Список с прокруткой» для текстового режима. Работа со списком должна сводиться к вызову одной функции, в которую передаются координаты области на экране, которую занимает список, и массив строк, составляющих список. Функция самостоятельно организует интерфейс для выбора элемента.

Каждая строка представляет один элемент списка. В строках не должно быть символов возврата каретки и перевода строки. Элементы должны выводиться в заданном окне. Количество строк в списке может как превышать высоту окна, так и быть меньше ее. В первом случае необходимо предусмотреть прокрутку содержимого в окне, во втором - обеспечить корректный выбор только тех позиций в окне, которые содержат строки списка (другими словами, курсор не должен выходить за пределы и окна, и списка).

Текущий элемент подсвечивается каким-либо цветом, заданным разработчиком. Выбор элемента осуществляется нажатием клавиш «Up» (перемещение курсора вверх) и «Down» (перемещение курсора вниз).

Подтверждение выбора (и выход из функции) осуществляется при нажатии клавиш Enter, Esc, Left, Right, Tab, Shift+Tab. Каждой из этих клавиш должен соответствовать определенный код, который также возвращается функцией выбора из списка. Этот код может использоваться вызывающей функцией для того, чтобы определить дальнейшее действие: переход к предыдущему (Shift+Tab, Left) или следующему (Tab, Right) элементу управления, выполнение основного действия программы (Enter), выход из программы (Esc).

Помимо указанного кода функция должна возвращать индекс выбранного элемента списка.

8. Написать функцию ввода текстовой строки с редактированием. При вызове функции ей передается длина поля ввода (максимальная длина отображаемой на экране части строки), а также координаты первого символа поля ввода на экране. Поле ввода однострочное, т.е. имеет высоту, равную 1 знакоместу.

Длина вводимой строки может превышать размеры поля ввода. Если в процессе ввода или перемещения курсора по строке текущая позиция ввода выходит за пределы поля, необходимо производить прокрутку строки в окне.

По усмотрению разработчика максимальная длина строки может либо передаваться в функцию, либо определяться объемом доступной динамической памяти. В первом случае буфер под строку резервирует вызывающая функция, во втором - функция ввода.

Функция ввода должна предоставлять следующие возможности по редактированию текста: перемещение курсора, ввод символа, удаление символа над курсором (клавиша Delete), удаление символа перед курсором (клавиша Backspace).

Если часть строки находится за пределами поля, желательно показывать это каким-либо образом (например, ставить выделенные цветом символы «<» и «>» в начале и конце поля ввода, если есть текст за пределами поля с соответствующей стороны).

Технологии и методическое обеспечение промежуточной аттестации

Согласно рабочему учебному плану, по дисциплине «Разработка интерфейса пользователя» предусмотрено проведение промежуточной аттестации в форме экзамена. Студент допускается до экзамена при условии сдачи и защиты всех лабораторных работ и расчетно-графических заданий.

Форма проведения промежуточной аттестации в виде экзамена – письменная индивидуальная. Продолжительность проведения экзамена – два академических часа. Вопрос разрешения или запрещения использования литературы и конспекта лекций остается на усмотрение экзаменатора.

Экзаменационный билет включает два теоретических вопроса и одну практическую задачу, заключающуюся, как правило, в разработке программы на языке Си++.

Теоретические вопросы экзамена

1. Эргономические характеристики интерфейса.
2. Составные части интерфейса.
3. Задачи процесса диалога.
4. Сообщения.
5. Виды диалога.
6. Диалог типа «вопрос-ответ».
7. Процессы ввода-вывода.
8. Методы разработки подпрограмм нижнего уровня на языке Си.
9. Строение видеопамати в текстовом режиме.
10. Исчезающее меню и алгоритм его построения.
11. Сохранение и восстановление экрана при создании исчезающего меню.
12. Вывод меню на экран.
13. Подпрограмма определения выбора пользователя.
14. Структура подпрограммы исчезающего меню.
15. Иерархическое меню и его фрейм.
16. Подпрограмма создания иерархического меню.
17. Активизация меню.
18. Организация иерархического меню, управляющая подпрограмма.
19. Всплывающие окна, оконные функции ввода-вывода.
20. Приложения системы Windows, типы данных, главная подпрограмма и оконные функции.
21. Создание окна в системе Windows, стили окон.
22. Цикл обработки сообщений.
23. Обработка сообщений в системе Windows.
24. Обработка нажатия клавиши.
25. Контекст устройства.
26. Перо, кисть и графические функции.
27. Обработка сообщения о нажатии клавиши.
28. Обработка запроса на перерисовку окна.
29. Обработка сообщений мыши.
30. Генерация запроса на перерисовку окна.
31. Генерация сообщения таймера.
32. Окна сообщений.
33. Меню в системе Windows, файл ресурсов.
34. Обработка команд меню.
35. Горячие клавиши в системе Windows.
36. Элементы управления диалогом.
37. Обработка сообщений в диалоге и активизация диалога.
38. Определение ресурсов диалога.
39. Создание немодального диалога.
40. Элементы управления listbox и editbox и их обработка.

41. Основные инструменты Borland C++ Builder для организации интерфейса.
42. Палитра компонентов и их описание.
43. Стандартные компоненты ввода и отображения текстовой информации.
44. Компонент для ввода и отображения матриц.
45. Компонент Image.
46. Сохранение и восстановление графических данных.
47. Компонент PaintBox.
48. Программирование графики в Borland C++ Builder с помощью функций API.
49. Создание модального диалога на основе новой формы.

Экзаменационные задачи

1. Написать подпрограмму, определяющую номер текущего видеорежима.
2. Написать подпрограмму установки размеров курсора.
3. Написать подпрограмму вывода символа и его атрибута в текущую позицию курсора.
4. Написать подпрограмму изменения палитры.
5. Написать подпрограмму установки текущей позиции курсора.
6. Написать подпрограмму вывода строки символов с одинаковыми атрибутами на экран.
7. Написать подпрограмму вывода символа и атрибута в заданную позицию экрана.
8. Написать подпрограмму сохранения области вывода меню.
9. Написать подпрограмму восстановления области вывода меню.
10. Написать подпрограмму вывода меню и рамки на экран.
11. Написать подпрограмму определения выбора пользователя для меню.
12. Написать подпрограмму создания фрейма иерархического меню.
13. Написать управляющую программу для иерархического меню, главный фрейм которого состоит из опций «Файл», «Параметры», «Работа» и «Выход», из которых опция «Файл» имеет подопции «Чтение», «Запись» и «Печать», опция «Параметры» имеет подопции «Матрица А» и «Вектор В», а опция «Работа» имеет подопции «По шагам» и «Непрерывно».
14. Написать приложение системы Windows, оконная функция которого обрабатывает нажатие клавиши и выводит ее код на экран.
15. Написать приложение системы Windows, оконная функция которого выводит сообщения в позиции, где была нажата левая или правая кнопка мыши.
16. Составить файл ресурсов для иерархического меню.

17. Написать приложение системы Windows, обрабатывающего команды меню.
18. Написать диалоговую программу, содержащую listbox и edit box.
19. С помощью Borland C++ Builder написать диалоговую программу, в которой два числа вводятся из окон редактирования и их сумма выводится в компонент Label.
20. С помощью Borland C++ Builder написать диалоговую программу, в которой два числа вводятся из окна Memo и их сумма выводится в компонент Panel.
21. С помощью Borland C++ Builder написать диалоговую программу, в которой несколько чисел вводятся из таблицы GridString и их сумма выводится в компонент TextEdit. Коэффициенты матрицы вводятся пользователем с клавиатуры.
22. Написать программу, выводящую правильный n-угольник на компонент Image. Число n вводится из окна редактирования.
23. Написать программу, выводящую правильный n-угольник на компонент PaintBox.
24. Написать программу, рисующую случайный треугольник и сохраняющую это изображение в файл с именем t.bmp.
25. Написать программу, отображающую файл bmp на компонент Image.

Примерный вариант экзаменационного билета

1. Исчезающее меню и методы его построения.
2. Каркас приложения Windows.
3. С помощью Borland C++ Builder написать диалоговую программу, в которой два числа вводятся из окон редактирования и их сумма выводится в компонент Label.

8 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

8.1 Основная литература

- 1 Корнеев, В. И. Программирование графики на C++. Теория и примеры [Электронный ресурс] : учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. — М.: ИД «ФОРУМ» : ИНФРА-М, 2017. — 517 с. // ZNANIUM.COM: электронно-библиотечная система. – Режим доступа: <http://znanium.com/catalog.php>, ограниченный. – Загл. с экрана.

8.2 Дополнительная литература

- 1 Ткаченко, О. Н. Взаимодействие пользователей с интерфейсами информационных систем для мобильных устройств: исследование опыта [Электронный ресурс] : учебное пособие / О. Н. Ткаченко — М. : Магистр :

ИНФРА-М, 2018.— 152 с. // ZNANIUM.COM : электронно-библиотечная система. – Режим доступа: <http://znanium.com/catalog.php>, ограниченный. – Загл. с экрана.

- 2 Купер, А. Интерфейс. Основы проектирования взаимодействия / А. Купер, Р. Рейман, Д. Кронин, К. Носсел; Пер. с англ. Е.Матвеев. - 4-е изд. - СПб. : Питер, 2017. – 719 с.

9 Методические указания для обучающихся по освоению дисциплины (модуля)

Для повышения качества выживаемости знаний, задачи лабораторных работ и РГР должны подбираться с учетом необходимости применения знаний в последующих дисциплинах.

Начиная со второго семестра необходимо на первом лабораторном занятии проводить тестирование остаточных знаний студентов. Для тестирования можно использовать задачи для лабораторных работ, включающие в качестве вспомогательных элементов, действия, связанные с тематикой работ первого семестра. Варианты тестовых задач приводятся в приложении.

Результаты тестирования могут быть учтены при промежуточной аттестации студентов в текущем семестре.

Проведение контроля текущей успеваемости, с одной стороны, позволяет получать адекватную информацию о степени усвоения учебного материала, с другой стороны, стимулирует ритмичность учебной деятельности.

Информация о выполненных лабораторных работах отражается в рейтинговых таблицах, выкладываемых на соответствующем сайте ФКТ. Уровень рейтинга студента рассчитывается и отображается на сайте автоматически.

РГР способствует лучшему освоению практических навыков по данному предмету. Студент получает задания в начале изучаемого раздела, а сдает выполненное задание после прохождения основных лабораторных занятий по данному разделу.

Качество освоения учебного материала и выполнения РГР контролируется преподавателем в виде защиты лабораторных работ. На защите РГР преподаватель в устной (или письменной) форме проверяет знание основных определений и положений дидактической единицы, являющейся темой РГР, а также проверяет навыки практического использования современного программного обеспечения.

Студент, успешно выполнивший и защитивший плановые лабораторные работы допускается к экзамену.

10 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Освоение дисциплины «Разработка интерфейса пользователя» основывается на активном использовании систем визуального программирования в процессе изучения теоретических разделов дисциплины и подготовки к практическим занятиям.

С целью повышения качества ведения образовательной деятельности в университете создана электронная информационно-образовательная среда. Она подразумевает организацию взаимодействия между обучающимися и преподавателями через систему личных кабинетов студентов, расположенных на официальном сайте университета в информационно-телекоммуникационной сети «Интернет» по адресу <https://student.knastu.ru>. Созданная информационно-образовательная среда позволяет осуществлять взаимодействие между участниками образовательного процесса посредством организации дистанционного консультирования по вопросам выполнения практических заданий. В учебном процессе по дисциплине активно используется контрольно-рейтинговая система факультета компьютерных технологий по контролю уровня выполнения лабораторных работ «ЛабДиспетчер», расположенная по адресу <http://biblserver/LD> в локальной сети ФКТ.

При изучении дисциплины, для выполнения лабораторных работ используется лицензионное программное обеспечение – Builder C++ или демонстрационно-учебная версия системы RAD Studio.

11 Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Для реализации программы дисциплины «Разработка интерфейса пользователя» используется материально-техническое обеспечение, перечисленное в таблице 7.

Таблица 7 – Материально-техническое обеспечение дисциплины

Аудитория	Наименование аудитории (лаборатории)	Используемое оборудование	Назначение оборудования
Аудитория с компьютерами и проектором	Компьютерные классы ФЗДО	Компьютеры IBM PC Corel-3, 4Мб ОЗУ, 25 шт. в классе, проектор	Выполнение лабораторных работ, проведение лекций

Лист регистрации изменений к РПД

№ п/п	Содержание изменения/основание	Кол-во стр. РПД	Подпись автора РПД
1	Изменение листа подписей в связи со сменой декана ФКТ /пр.№ 271-ЛС «к» от 29.12.2016	1	
2	Изменение КУГ/пр. № 326-О «а» от 04.09.2017	9	
3	Изменение титульного листа в связи с переименованием вуза/пр. №997-О от 03.11.2017	1	
4	Актуализация литературы/ 28.11.2017	2	