



Проректор по учебной работе

А.Л. Толстик

Регистрационный № УД- 1562 /уч.

ПРОГРАММИРОВАНИЕ В C++ BUILDER

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности

1-31 03 01	Математика (по направлениям)
(1-31 03 01-05	Математика (информационные технологии))

2015 г.

Учебная программа составлена на основе ОСВО 1-31 03 01-2008 (30.08.2008) и учебного плана рег. № G31з-037/уч. от 01.09.2008г. для специальности 1-31 03 01 Математика (по направлениям) (направление 1-31 03 01-05 информационные технологии).

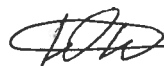
СОСТАВИТЕЛИ:

Юрий Евгеньевич Нагорный, старший преподаватель кафедры Веб-технологий и компьютерного моделирования механико-математического факультета Белорусского государственного университета.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой веб-технологий и компьютерного моделирования
(протокол № 10 от 14.05.2015г.);

Учебно-методической комиссией механико-математического факультета
Белорусского государственного университета
(протокол № 6 от 26.05.2015г.).



(Ю.Е.Нагорный)



(В.С. Романчик)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время от программных продуктов требуются не только хорошие функциональные возможности, но и удобство в их использовании. Это связано как с широким распространением вычислительной техники и её применением практически во всех сферах жизнедеятельности, так и появлением целого класса программно-аппаратных устройств как новых, так и основанных на ранее существующих аппаратах, приобретших возможности управления и контроля. Использование всего этого класса техники осуществляется не профессиональными программистами, а в лучшем случае, просто квалифицированными работниками других специальностей или, даже, обычными гражданами. Поэтому в последнее время встал вопрос о создании программ с простым и интуитивно понятным интерфейсом.

Всё это определяет важность курса «*Программирование в C++Builder*» в учебном процессе, а также обуславливает необходимость внесения своевременных изменений и дополнений в его содержание.

Учебная программа по дисциплине «*Программирование в C++Builder*» разработана для студентов V курса заочной формы обучения по специальности 1-31 03 01 Математика (по направлениям) (направление 1-31 03 01-05 «информационные технологии») механико-математического факультета Белорусского государственного университета.

Центральной идеей образования по дисциплине “Программирование в C++Builder” является необходимость обучения студентов современным подходам в визуальном программировании.

Дисциплина «*Программирование в C++Builder*» имеет прикладную направленность. Ее основная цель развить и закрепить логико-алгоритмическое мышление, изучить современные методы визуального программирования и информационные технологии и научиться с их помощью решать разнообразные задачи на компьютерах.

Задачи дисциплины состоят в изучении парадигмы объектно-ориентированного программирования, освоении среды визуальной разработки C++Builder, формировании знаний об основных методах проектирования интерфейсов программ; приобретение навыков тестирования программного обеспечения в рамках среды C++Builder.

Программа курса «*Программирование в C++Builder*» составлена с учетом межпредметных связей и программ по смежным дисциплинам. Его изучение базируется на знаниях университетского курса по программированию на языке C++ «*Методы программирования и информатика*».

В результате изучения дисциплины студент должен

знать:

- теоретические основы объектно-ориентированного подхода к программированию;
- типовые алгоритмы и границы их применимости;
- типы и структуры данных;
- встроенные (библиотечные) возможности языка программирования C++.
- основные элементы VCL;

уметь:

- составлять алгоритм на основе содержательной постановки задачи;
- описывать алгоритмы в программном коде на языках C++;
- создавать интерфейс программы;
- осуществлять обмен данными между интерфейсом и исполнительной частью программы;
- применять встроенные (библиотечные) возможности языка;
- структурировать и документировать программный код, отлаживать программы: локализовать и устранять ошибки.

владеть:

- навыками работы в визуальной среде программирования;
- навыками формирования интерфейса;
- методами и средствами решения практических задач из осваиваемой предметной области с использованием среды C++Builder.

В соответствии с учебными планами специальности на изучение данной дисциплины для формы обучения отводится:

	Экзамен, семестр	Зачет, семестр	Всего часов	В том числе ауди- торных	Из них	
					Лекций	Лабораторных занятий
V курс, 10 семестр			184	24	10	14
VI курс, 11 семестр	11					

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Создание Windows-приложений средствами C++ Builder.

Основные файлы проекта. Главное меню C++ Builder. Object Inspector. Object tree View. Формы. Инспектор объектов. Хранилище объектов. Вывод сообщений. Ввод/вывод информации с использованием компонентов и классов. Ввод/вывод строк и чисел и с использованием текстовых компонентов.

Тема 2. Работа с файлами. Использование меню.

Открытие текстовых и двоичных файлов. Компонента OpenFileDialog. Чтение данных из файлов. Запись данных в файл. Компонента SaveDialog. Заккрытие файлов. Использование графических компонент MainMenu и PopupMenu.

Тема 3. Работа с графикой.

Графические компоненты. Использование канвы для рисования. Построение графиков функций на плоскости.

Тема 4. Анимация. Игры. Обработка событий мыши, клавиатуры, таймера

Методика анимации изображения. Какие свойства имеет компонент Timer. Основные события мыши. Координаты курсора мыши. Обработка нажатия кнопки мыши. Когда наступают события мыши OnStartDrag, OnDragOver, OnDragDrop, OnEndDrag. Что можно распознать при обработке этих событий. Чем отличаются события клавиатуры OnKeyDown и OnKeyPress. Как распознать нажатые функциональные клавиши. Куда поступают события клавиатуры.

Тема 5. Построение распределенных приложений типа Client/Server.

Организация сетевых соединений (сокетов). Использование компонентов TclientSocket TserverSocket. Пересылка сообщений между клиентом и сервером в обоих направлениях с постоянным отслеживанием информации.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы,	Название раздела, темы	Количество аудиторных часов				Количество часов УСР	Литература	Формы контроля знаний
		лекции	практические (семинарские) занятия	лабораторные занятия	Иное			
1	Создание Windows-приложений средствами C++ Builder.	2		2			[1] - [6]	Отчеты по лабораторной работе
2	Работа с файлами. Использование меню.	2		3			[1] - [6]	Отчеты по лабораторной работе
3	Работа с графикой.	2		2			[1] - [6]	Отчеты по лабораторной работе
4	Анимация. Игры. Обработка событий мыши, клавиатуры, таймера	2		3			[1] - [6]	Отчеты по лабораторной работе
5	Построение распределенных приложений типа Client/Server.	2		4			[1] - [6]	Отчеты по лабораторной работе
	Всего	10		14				

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

Основная литература

1. *Архангельский А. Я.* Программирование в C++Builder 6 --- М: ЗАО “Издательство БИНОМ” 2002г.
2. *В.В. Подбельский.* Язык C++. Учебное пособие. – М.: Финансы и статистика, 2000. –560с.
3. *Г. Шилдт.* Самоучитель C++, 3-е издание: — СПб.: BHV — Санкт-Петербург, 1998. — 688с.

Дополнительная литература

4. *Архангельский А. Я.* Справочное пособие по C++Builder 6. Книга 1. Язык C++ --- М: ЗАО “Издательство БИНОМ ”. 2002г.
5. *Архангельский А. Я.* C++Builder 6. Справочное пособие. Книга 2. Классы и компоненты. --- М: ЗАО “Издательство БИНОМ ”. 2002г.
6. *Архангельский А. Я.* Решение типовых задач в C++Builder 6.--- М: ЗАО “Издательство БИНОМ ”. 2003 г.

ПЕРЕЧЕНЬ ЗАДАНИЙ И КОНТРОЛЬНЫХ МЕРОПРИЯТИЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Контрольные мероприятия СР по дисциплине «*Программирование в C++Builder*» проводятся преподавателем, как правило, во время аудиторных занятий.

Полученные студентом количественные результаты СР учитываются как составная часть итоговой оценки по дисциплине в рамках рейтинговой системы.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ЗАДАНИЙ

Индивидуальные задания для самостоятельной работы включают решения не менее трёх практических задач различного уровня сложности, которые сдаются на проверку в электронном виде, а также вопросы практической работы в среде C++Builder.

Тема 1. Создание Windows-приложений средствами C++ Builder

При выполнении заданий необходимо использовать как главную форму, так и новую форму, которая может быть создана самостоятельно или взята из Хранилища объектов (например, форма About Box). Для вывода сообщений использовать диалоговые окна (функции ShowMessage (...), MessageDlg (...), MessageBox (...)) и другие. Для ввода информации использовать компоненты Edit, UpDown (кнопка-счётчик), SpinEdit (кнопка-счётчик с окном редактирования) и другие, а также диалоговое окно для ввода текста (функции InputBox (...)). Для указания на окончание ввода исходных данных использовать как кнопки, так и нажатие на клавишу Enter. При вводе числовых данных в текстовые компоненты предусмотреть контроль вводимых данных.

1. Разработать калькулятор, реализующий арифметические операции.
2. Реализовать перевод чисел в двоичную, восьмеричную и шестнадцатеричную системы счисления.
3. Реализовать поразрядные логические операции AND, OR, NOT, XOR для двоичных векторов.
4. Реализовать вычисление скалярного произведения двух векторов, проверку, являются ли векторы коллинеарными или ортогональными.
5. Выполнить сортировку одномерного массива различными способами.
6. Найти произведение элементов одномерного массива, расположенных между максимальным и минимальным элементами.
7. Найти произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.
8. Преобразовать одномерный массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные.

9. Элементы одномерного массива циклически сдвинуть на k мест влево (вправо).
10. Сложить и умножить два полинома заданных степеней (коэффициенты полиномов представляют собой одномерные массивы).
11. Ввести одномерный массив. Найти сумму элементов массива, расположенных между первым и последним положительными элементами. Преобразовать массив таким образом, чтобы сначала располагались все элементы равные нулю, а потом – все остальные.
12. Ввести одномерный массив. Найти количество элементов массива, равных нулю. Упорядочить элементы массива по возрастанию модулей элементов.
13. В одномерном массиве выбрать без повторений все элементы, встречающиеся более одного раза.
14. Ввести одномерный массив и числа a и b . Найти номер максимального по модулю элемента массива. Преобразовать массив таким образом, чтобы сначала располагались все элементы, которые лежат в промежутке $[a, b]$, а потом – все остальные.
15. Ввести одномерный массив. Найти сумму элементов массива, расположенных между первым и вторым положительными элементами. Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

Тема 2. Работа с файлами. Использование меню.

При выполнении заданий необходимость использовать меню. Предусмотреть ввод исходных данных как с помощью текстовых компонентов, так и из файлов (с отображением введенных данных в текстовых компонентах Memo, RichEdit, Edit и использованием компонента OpenFileDialog). Полученные результаты также отображаются с использованием текстовых компонентов и выводятся в файлы (при выборе соответствующего пункта меню; использовать компонент SaveDialog). Для преобразования строк, содержащих несколько числовых данных (например, строку матрицы), в последовательность чисел, и наоборот, использовать потоки ввода/вывода из/в строк/строки.

1. Транспонировать квадратную матрицу.
2. Ввести матрицу $n \times m$. Характеристикой строки матрицы назовем сумму ее положительных элементов, находящихся на четных позициях. Переставляя строки матрицы, расположить их в соответствии с ростом характеристик.
3. Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями.
4. Осуществить циклический сдвиг заданной матрицы на k позиций вправо (влево).
5. Осуществить циклический сдвиг заданной матрицы на k позиций вниз (вверх).

6. Упорядочить строки матрицы по возрастанию максимального количества одинаковых элементов в строке.
7. Путем перестановки элементов квадратной целочисленной матрицы добиться того, чтобы ее максимальный элемент находился в позиции (0,0), следующий по величине элемент – в позиции (1,1) и т.д., заполнив, таким образом, всю главную диагональ.
8. Удалить из заданной матрицы строку и столбец, на пересечении которых находится минимальный элемент матрицы.
9. В матрице найти минимальный элемент и переместить его на место заданного элемента путем перестановки строк и столбцов.
10. Ввести множество векторов. Найти среди них все коллинеарные.
11. Разработать текстовый редактор.
12. Создать записную книжку.
13. В заданном тексте выполнить замену заданного слова (если слово встречается несколько раз, то замену выполнить каждый раз) другим словом, также заданным.
14. Удалить в тексте строки, содержащие заданное слово.
15. Текст представляет собой сведения об успеваемости студентов. Строка содержит фамилию студента и результаты сдачи экзаменов. Упорядочить строки в соответствии с убыванием среднего балла.

Тема 3. Работа с графикой.

Графики вычерчивать в графических компонентах Image и Paint Box, используя методы и свойства класса Canvas. Перед вычерчиванием графика установить пределы изменения x и y . Выполнить масштабирование для вывода графика в заданную прямоугольную область графического компонента. Предусмотреть возможность очистки области компонента. Выводить также оси координат. Реализовать возможность выбора цвета и толщины линий.

1. $y = \operatorname{tg} x; y = \operatorname{ctg} x$.

2. $y = -2x^2 + 3x; y = 3x^2 + 2$.

3. $y = \frac{1}{x}, x > 0$.

4. Эллипс с полуосями r_1 и r_2 , расположенными параллельно осям координат:

$$x = r_1 \cos \varphi; y = r_2 \sin \varphi; \varphi \in [0, 2\pi]$$

5. $y = \frac{x+2}{x-3}$

Спираль Архимеда (задана в полярных координатах) :
 $\rho = a\varphi$

6. $y = 3 + \frac{2}{x} + \frac{3}{x^2}$

Кардиоида:

$$\rho = a(1 + \cos \varphi); a > 0, \varphi \in [0, 2\pi]$$

$$7. \quad y = \frac{1}{3x^2 + 2x + 1}$$

Астроида:

$$x = b \cos^3 \varphi;$$

$$y = b \sin^3 \varphi; \varphi \in [0, 2\pi]$$

$$8. \quad y = \sin x + \cos x$$

Улитка Паскаля:

$$9. \quad \rho = b + a \cos \varphi; \varphi \in [0, 2\pi]$$

$$10. \quad y = x + \cos|x|$$

Логарифмическая спираль: $\rho = a \cdot e^{b\varphi}$

$$11. \quad y = -0,5 \cdot e^x$$

Лемниската Бернулли: $\rho^2 = 2a^2 \cos 2\varphi$

$$12. \quad y = x^2 - \sin(\pi x^2)$$

$$13. \quad y = \frac{1}{x^2 + 4x + 5}$$

Циссоида:

$$x = \frac{at^2}{1+t^2},$$

$$y = \frac{at^3}{1+t^2}, t \in (-\infty, \infty), a > 0$$

$$14. \quad y = \frac{\sin x}{x}$$

Спираль с n витками:

$$\rho = \frac{\varphi}{2}$$

$$x = \rho \cos \varphi$$

$$15. \quad y = \rho \sin \varphi; 0 \leq \varphi \leq 2n\pi$$

$$16. \quad y = e^{-x^2}$$

Окружность:

$$x = r \cos \varphi$$

$$y = r \sin \varphi; \varphi \in [0, 2\pi]$$

$$17. \quad y = x^5 - 3x^2 + 2$$

Фигура Лиссажу:

$$x = a \sin(k_1 \varphi + b_1)$$

$$18. \quad y = b \sin(k_2 \varphi + b_2)$$

Параметры a, b, k_1, k_2, b_1, b_2 задать.

$$19. \quad y = \frac{x}{x^2 + 2x + 1}$$

Строфоида:

$$x = \frac{a(t^2 - 1)}{t^2 + 1},$$

$$y = \frac{at(t^2 - 1)}{t^2 + 1},$$

$$t \in (-\infty, \infty), a > 0$$

Тема 4. Анимация. Игры. Обработка событий мыши, клавиатуры, таймера

При выполнении заданий использовать графические компоненты Image и PaintBox. Для отсчета нужных интервалов времени использовать компонент Timer. Для динамических рисунков предусмотреть кнопку для запуска и остановки.

1. Вывести аналоговые часы (со стрелками). Предусмотреть кнопку для запуска и остановки часов. Использовать функцию Time() и класс TDateTime для получения текущего времени.
2. “Летающий” шарик. По достижении границы компонента он отражается от границы по правилам отражения. Предусмотреть кнопку остановки и запуска шарика.
3. Реализовать игру “крестики-нолики”.
4. Точка равномерно движется по окружности. Предусмотреть возможность увеличения скорости движения.
5. Построить простейший графический редактор с возможностью выбора цвета рисования.
6. Равносторонний треугольник вращается вокруг своего центра. Предусмотреть возможность увеличения скорости вращения.
7. Изобразить движущегося человечка. По достижении левой и правой границы компонента человечек поворачивает в обратную сторону.
8. Изобразить прямоугольник (квадрат), вращающийся вокруг своего центра. Предусмотреть возможность увеличения скорости вращения.
9. Вращаются два отрезка, каждый вокруг своей концевой точки. Предусмотреть возможность изменения скорости вращения каждого отрезка отдельно.
10. “Расстановка трех чисел”. В каждой из 9 клеток квадрата размером 3×3 клетки поставить одно из чисел 1,2,3 так, чтобы сумма чисел, стоящих в каждом вертикальном ряду, в каждом горизонтальном ряду, а также на любой диагонали, равнялась 6.
11. “Расстановка девяти чисел”. Квадрат размером 3×3. числа выбираются из диапазона 1,2,3,...,9. Суммы чисел, стоящих в каждом вертикальном ряду, каждом горизонтальном ряду, а также по диагоналям должны быть равны.
12. “Ипподром”. Играющий выбирает одну из трех лошадей, соревнующихся на бегах, и выигрывает, если его лошадь приходит первой.

Скорость передвижения лошадей на разных этапах выбирается программой с помощью датчика случайных чисел.

13. Игра в слова. Программа выбирает слово и рисует на экране столько прочерков, сколько букв в слове. Отгадать, какое слово загадано программой. За один ход играющий указывает одну букву. Если буква не угадана, то играющий теряет одно очко. В начальный момент у играющего 15 очков.

14. “Коровы и быки”. Программа выбирает случайное четырехзначное число с разными цифрами. Угадать это число. На каждом шаге играющий называет четырехзначное число, а программа сообщает, сколько цифр числа угадано (быки) и сколько цифр угадано и стоит на нужном месте (коровы). Например, если задано число 1294, а названо 1423, то программа ответит “1 корова, 3 быка”.

15. Требуется ввести курсор в область экрана (небольшой круг), расположение которого неизвестно играющему. Если курсор приближается к области, то его цвет становится ярче, если удаляется, то тускнеет.

Тема 5. Построение распределенных приложений типа Client/Server.

Для выполнения заданий необходимо создать два приложения: сервер и клиент.

1. Организовать пересылку текстовых сообщений между клиентом и сервером в обоих направлениях с постоянным отслеживанием информации.
2. Реализовать игру «крестики/нолики» в сети.
3. Реализовать игру в слова (следующее слово начинается с буквы, на которую заканчивается предыдущее слово; слова не должны повторяться) в сети.
4. Реализовать игру в «догонялки» в сети.
5. Реализовать сетевой вариант игры «ипподром» (лаб. работа №4, задание 12).
6. Реализовать игру в «морской бой» в сети.
7. Реализовать сетевой вариант игры в отгадывание слов (см. лаб. работу №4, задание 13). Слово задаётся пользователем программы-сервера.
8. Реализовать сетевой вариант игры «коровы и быки» (лаб. работа №4, задание 14).
9. Требуется ввести курсор в область экрана (небольшой круг), расположение которого неизвестно играющему (клиент), но известно приложению-серверу (второй игрок). Если курсор приближается к области, то клиенту передается соответствующее сообщение («теплее», «горячо» и т.д.). Если удаляется, то – («холоднее» и т.д.).

СРЕДСТВА ДИАГНОСТИКИ РЕЗУЛЬТАТОВ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ

Рекомендуются следующие формы диагностики компетенций.

Устно-письменная форма

1. Отчеты по лабораторным работам.
2. Курсовые работы с их устной защитой.
3. Экзамен.

ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

Курсовой работа, как форма текущей аттестации студентов, является видом самостоятельной работы студентов, носит учебно-исследовательский характер и представляет собой решение учебной задачи по изучаемой учебной дисциплине в соответствии с установленными требованиями.

Порядок организации курсового проектирования и защиты курсовых работ определяется учреждением высшего образования.

Студент вправе выбрать тему курсовой работы из числа утвержденных на кафедре или самостоятельно предложить тему курсовой работы с обоснованием ее целесообразности. Для формирования умений и навыков работы в команде возможно выполнение группового задания, предусматривающего работу нескольких обучающихся над одной курсовой работой. В этом случае каждому из них устанавливается индивидуальный объем задач в соответствии с объемом и уровнем общих требований.

Цель курсовых работ

по дисциплине «Программирование в C++Builder»:

а) закрепить, углубить и расширить теоретические знания по дисциплине с учетом современных тенденций в развитии методов программирования в визуальной среде;

б) овладеть навыками самостоятельной работы с научной литературой; навыками работы с применением современных программных средств;

в) закрепить и углубить практические методов программирования в визуальной среде C++Builder;

г) выработать умение публичной защиты.

На выполнение курсовой работы по дисциплине «Программирование в C++Builder» отводится 40 часов внеаудиторной управляемой самостоятельной работы студента на V курсе в 10 семестрах.

Требования к структуре курсовой работы. Структура курсовой работы должна способствовать раскрытию избранной темы и быть аналогична структуре дипломной работы: иметь титульный лист, реферат, содержание,

введение, основную часть, заключение, список использованных источников и приложения.

Требования к содержанию (основной части). Требования к реферату и содержанию (основной части) курсовых работ аналогичны правилам оформления реферата и содержания дипломных работ. Во введении обосновывается актуальность выбранной темы, определяется общая цель курсовой работы, конкретные ее задачи и методы исследования. Основная часть работы включает две – четыре главы, которые разбивают на разделы и подразделы. Каждая глава посвящается решению задач, сформулированных во введении, и заканчивается констатацией итогов. Серьезные теоретические положения необходимо давать со ссылкой на источник. Написание курсовой работы предполагает более глубокое изучение избранной темы, нежели она раскрывается в учебной литературе. В работах, носящих в основном теоретический характер, анализируя литературу по теме исследования, изучая и описывая опыт наблюдаемых событий (явлений), студент обязательно высказывает свое мнение и отношение к затрагиваемым сторонам проблемы.

Требования к оформлению. Объем курсовой работы – до 25–30 страниц печатного текста размера 14pt, выполненного через 1,5 межстрочных интервала. Оформление заключения, списка использованных источников и приложения осуществляется в соответствии с требованиями ГОСТа. Работу сшивают в папку-скоросшиватель или переплетают.

Выполненная студентом курсовая работа проверяется руководителем работы в срок до 10 дней до защиты. Защита курсовых работ производится до начала экзаменационной сессии перед комиссией, которая формируется заведующим кафедрой в составе не менее двух человек с участием руководителя курсовой работы.

На защите студент обязан кратко изложить содержание работы, дать исчерпывающие ответы на вопросы членов комиссии. Оценка курсовой работы выставляется комиссией по итогам защиты и качеству выполненной работы.

Примерный перечень тем курсовых работ

1. Написать простейший чат.
2. Реализовать одну из игр в теме 4 с возможностью фиксации количества шагов, затраченного времени. Сохранения результатов в файл. Загрузка результатов прошлых игр. Возможность составления таблицы результатов с сортировкой по всем параметрам.

**ПРОТОКОЛ
СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ ПО ИЗУЧАЕМОЙ
УЧЕБНОЙ ДИСЦИПЛИНЕ С ДРУГИМИ ДИСЦИПЛИНАМИ
СПЕЦИАЛЬНОСТИ**

[illegible]

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
на ____/____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры Веб-технологий и компьютерного моделирования (протокол № ____ от ____ 201_ г.)

Заведующий кафедрой

канд. физ.-мат. наук, доцент _____ В.С. Романчик

УТВЕРЖДАЮ

Декан факультета

канд. физ.-мат. наук, доцент _____ Д.Г. Медведев