

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ))



Рабочая программа дисциплины (модуля),
как компонент образовательной программы
высшего образования - программы бакалавриата
по направлению подготовки
09.03.01 Информатика и вычислительная техника,
утвержденной первым проректором РУТ (МИИТ)
Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Цифровые технологии

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Направленность (профиль): Системы автоматизированного проектирования

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная РУТ (МИИТ)
ID подписи: 2899
Подписал: И.о. заведующего кафедрой Нестеров Иван Владимирович
Дата: 21.05.2024

1. Общие сведения о дисциплине (модуле).

Целями освоения дисциплины (модуля) являются:

- изучение студентами теории и практики составления и анализа алгоритмов;

- изучение студентами теории и практики написания программного кода на языке высокого уровня.

Задачами дисциплины (модуля) являются:

- овладение методологией построения алгоритмов и написания кодов;

- формирование навыков по созданию программных интерфейсов.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ОПК-8 - Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-9 - Способен осваивать методики использования программных средств для решения практических задач.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

Знать методики использования программных средств для решения практических задач

Уметь:

Уметь использовать программные средства для решения практических задач

Владеть:

Владеть методикой использования программных средств для решения практических задач

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 4 з.е. (144 академических часа(ов)).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов	
	Всего	Семестр №4
Контактная работа при проведении учебных занятий (всего):	64	64
В том числе:		
Занятия лекционного типа	32	32
Занятия семинарского типа	32	32

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 80 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Общая структура программы под ОС Windows. Основные особенности операционной системы WINDOWS. Понятие API. Общая структура программы под ОС Windows.
2	Win32 API Базовые типы данных Win32 API. Определение класса окна. Основные макроопределения, описывающие стили класса окна.
3	Оконные функции Работа цикла обработки сообщений и функций обработки сообщений. Структура оконной функции. Описание оконной функции в классе окна.
4	Интерфейс вывода графической информации GDI Общая структура интерфейса вывода графической информации GDI. Сообщение, используемое для обновления графической информации в окне.

№ п/п	Тематика лекционных занятий / краткое содержание
5	Общая схема обработки сообщений от клавиатуры. Особенности цикла обработки системных сообщений в функции WinMain() при обработке сообщений от клавиатуры
6	Обработка сообщений от манипулятора «мышь». Основные типы сообщений. Получение координат курсора при обработке сообщений от «мыши»
7	API-функции рисования графических примитивов API-функции рисования графических примитивов в инверсном режиме. Определение ресурса WINDOWS.
8	Основные типы ресурсов. Рассмотрение основных типов ресурсов. Определение идентификатора ресурса.
9	Работа с курсорами. Функции создания пиктограмм. Работа с курсорами.
10	Ресурсы WINDOWS Работа с ресурсами падающего меню. Механизм обработки сообщений падающего меню.
11	Системное сообщение WM_COMMAND. Создание и динамическая загрузка. Системное сообщение WM_COMMAND.
12	Ресурсы падающего меню. Работа с ресурсами падающего меню. Механизм обработки сообщений падающего меню.

4.2. Занятия семинарского типа.

Практические занятия

№ п/п	Тематика практических занятий/краткое содержание
1	Структура функции WinMain(). Сообщение операционной системы. Изучение структуры функции WinMain() и сообщений операционной системы.
2	Работа цикла обработки сообщений и функций обработки сообщений. Изучение работы цикла обработки сообщений и функций обработки сообщений.
3	Структура оконной функции. Рассмотрение структуры оконной функции. Описание оконной функции в классе окна.
4	Общая структура интерфейса вывода графической информации GDI. Изучение структуры интерфейса вывода графической информации GDI. Сообщение, используемое для обновления графической информации в окне.
5	Функции получения контекста графического устройства. Изучение функций получения контекста графического устройства. Особенности их использования.
6	Функции создания основных графических инструментов. Изучение и использование функций создания (удаления) основных графических инструментов (кисть, перо).
7	Контекст графического устройства. Определение контекста графического устройства и способы работы.
8	Функции для создания шрифтов. Работа с набором функций для создания шрифтов.
9	Набор функций для создания фона. Использование функции установки режима закраски фона.
10	Активация объектов GDI (линии). Способы активации объектов GDI. Функции рисования линий.

№ п/п	Тематика практических занятий/краткое содержание
11	Активизация объектов GDI. Функции рисования простых геометрических фигур. Функции вывода текста.
12	Использование системных комбинаций на клавиатуре. Написание функций для обработки системных комбинаций на клавиатуре.
13	Обработка системных комбинаций на клавиатуре. Получение символьной информации от клавиатуры.
14	Обработка сообщений от "мыши". Написание функций обработки сообщений от манипулятора «мышь».
15	Обработка сообщений от манипулятора «мышь». Получение координат курсора при обработке сообщений от «мыши»

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение дополнительной литературы.
2	Подготовка к практическим занятиям.
3	Выполнение расчетно-графической работы.
4	Подготовка к промежуточной аттестации.
5	Подготовка к текущему контролю.

4.4. Примерный перечень тем расчетно-графических работ

Вариант 1

Написать приложение для Windows, выполняющее следующие действия:

1. При нажатии на левую клавишу мыши покрасить окно в синий цвет.
2. При нажатии на правую клавишу покрасить окно в зелёный цвет.
3. При комбинации “Shift + левая клавиша мыши” в этой точке нарисовать красный круг.
4. При нажатии на символьные клавиши выводить текст на экран из точки последнего нажатия на левую клавишу мыши.
5. По Alt+q – выход из программы.
6. По Alt+p – очистка экрана.

Вариант 2

Написать приложение для Windows, выполняющее следующие действия:

1. Разделить экран на 4 части двумя линиями.

2. В левом нижнем углу экрана постоянно выводить номер четверти, в которой находится указатель мыши.

3. При нажатии на левую клавишу мыши нарисовать линию из точки нажатия к центру экрана.

4. При нажатии на правую клавишу мыши, нарисовать круг, проходящий через точку нажатия с центром в середине экрана.

5. При нажатии на цифровые клавиши выводить текст на экран из точки последнего нажатия на правую клавишу мыши.

6. По Alt+1, Alt+2 и т.д. – очистка соответствующей четверти экрана.

7. По Alt+x – выход из программы.

Вариант 3

Написать приложение для Windows, выполняющее следующие действия:

1. Разделить экран на 2 части вертикальной линией.

2. В левом нижнем углу экрана постоянно выводить название половины в котором находится указатель мыши (левая или правая) и экранные координаты указателя мыши.

3. При нажатии на левую клавишу мыши, из точки нажатия написать произвольный текст.

4. При комбинации “Shift + левая клавиша мыши” из этой точки к центру экрана нарисовать прямую линию.

5. При нажатии на цифровые клавиши выводить текст на экран из точки последнего нажатия на правую клавишу мыши.

6. При нажатии на цифровые клавиши+Shift выводить текст на экран из точки последнего нажатия на правую клавишу мыши другим цветом.

7. “Alt+R” – очистка правой части экрана, “Alt+L” -левой.

8. “Alt+C” – выход из программы.

Вариант 4

Написать приложение для Windows, выполняющее следующие действия:

1. Разделить экран на 2 части горизонтальной линией.

2. В левом нижнем углу экрана постоянно выводить название половины в котором находится указатель мыши (верхняя или нижняя) и экранные координаты указателя мыши.

3. Как в системе AutoCAD написать процедуру рисования ломаной линии.

4. При нажатии на символьные клавиши выводить текст на экран из точки последнего нажатия на левую клавишу мыши.

5. При нажатии на символьные клавиши+Shift выводить текст из центра экрана.

6. “Alt+u” – очистка верхней части экрана, “Alt+l” – нижней.

7. “Alt+a” – выход из программы.

Вариант 5

Написать приложение для Windows, выполняющее следующие действия:

1. В середине окна нарисовать круг произвольного размера.

2. При перемещении мыши перерисовывать круг с центром в позиции указателя, удаляя предыдущий круг.

3. При нажатии левой клавиши мыши увеличивать радиус круга, правой клавиши – уменьшать в некоторых ограниченных пределах.

4. При нажатых клавишах Ctrl, Alt, Shift изменять цвет круга (произвольно).

5. По Alt+L – выход из программы.

Вариант 6

Написать приложение для Windows, выполняющее следующие действия:

1. При нажатии левой клавиши мыши отображать прямоугольник произвольного размера с серединой в позиции указателя, а при отпуске – гасить прямоугольник.

2. По нажатиям клавиш управления курсором (VK_LEFT, VK_RIGHT...) изменять ширину и высоту прямоугольника (видимого или скрытого) в некоторых ограниченных пределах.

3. По Alt+x – выход из программы.

Вариант 7

Написать приложение для Windows, выполняющее следующие действия:

Вывод прямоугольников в окно в интерактивном режиме с помощью мыши в следующем порядке:

1. Нажатие левой клавиши – первый угол прямоугольника.
2. Перемещение мыши – рисовать «резиновый» прямоугольник к текущей позиции указателя (если был начат ввод).
3. Нажатие правой клавиши – зафиксировать прямоугольник (если был начат ввод).
4. При нажатии Alt+E очищать окно.
5. По Alt+E – выход из программы.

Вариант 8

Написать приложение для Windows, выполняющее следующие действия:

1. В правой части экрана нарисовать палитру из семи цветов .
2. Нажатие левой клавиши мыши на палитре – смена текущего цвета, в остальной области рисование отрезка с “резиновой линией” .
3. Нажатие правой клавиши мыши – смена стиля пера.
4. Нажатие Shift + правая клавиша мыши – в текущей точке нарисовать круг с диаметром, равным длине “резиновой линии”.
5. По Esc – очистка окна кроме палитры.
6. По Alt+x - выход из программы.

Вариант 9

Написать приложение для Windows, выполняющее следующие действия:

1. В верхней части экрана нарисовать палитру из семи цветов .
2. Нажатие левой клавиши мыши на палитре – смена текущего цвета, в остальной области рисование круга с “резиновым контуром” (как в AutoCADe).
3. Нажатие правой клавиши – увеличение толщины пера.
4. Нажатие Shift+ правая клавиша мыши – уменьшение толщины пера.
5. По Esc – очистка окна кроме палитры.
6. По Alt+x - выход из программы.

Вариант 10

Написать приложение для Windows, выполняющее следующие действия:

Рисование вписанного правильного многоугольника с “резиновым контуром” (как в AutoCADe).

1. Разделить окно на 4 части.
2. Нажатие левой клавиши мыши – определение центра или радиуса.
3. Нажатие правой клавиши мыши – увеличение числа вершин.
4. Нажатие “Shift+ правая клавиша мыши” – уменьшение числа вершин.
5. В зависимости от четверти, где находится указатель мыши, менять текущий цвет многоугольника.
6. В левом нижнем углу экрана выводить текущее число вершин многоугольника.
7. По Alt+x - выход из программы.

Вариант 11

Написать приложение для Windows, выполняющее следующие действия:

1. В середине окна нарисовать сетку (таблицу), состоящую, например, из 7 строк и 13 столбцов, занимающую примерно 70% площади окна.
2. По нажатию левой клавиши мыши – выдать окно сообщения с номером строки и столбца элемента сетки, в котором находится указатель.
3. По нажатию правой клавиши мыши – подсветить (перекрасить) строку, в которой находится указатель, при нажатой Shift – подсветить столбец.
4. В верхней части экрана окна постоянно выводить номера строки и столбца элемента сетки, в котором находится указатель.
5. По Esc – обновить (перерисовать) сетку.
6. По Alt+x - выход из программы.

Вариант 12

Написать приложение для Windows, выполняющее следующие действия:

Организовать вывод в окно «плавающего» графика функции $y=ax^2$ следующим образом:

1. В нижней части окна отобразить палитру из 7 цветов.
2. При перемещении мыши рисовать параболу, проходящую через текущее положение курсора с экстремумом в центре экрана, удаляя предыдущий график.
3. По нажатию левой клавиши мыши в области палитры – выбрать указанный цвет текущим для отображения графика.
4. По нажатию правой клавиши мыши – увеличить или уменьшить толщину линий графика.
5. В верхней части окна постоянно выводить позицию указателя мыши и значение параметра a .
6. По Alt+x - выход из программы.

Вариант 13

Написать приложение для Windows, выполняющее следующие действия:

Организовать вывод в окно графиков аналитических функций следующим образом:

1. В левой части окна отобразить палитру из 7 цветов и в середине цветных ячеек написать уравнение аналитической функции и интервал (например: $\sin(x)$ [- π π], $\operatorname{sh}(x)$ [-1; 1], x^2 [-2;2], $\ln(x)$ [2; 4] и т.д) , из середины остальной части окна оси нарисовать оси X,Y ;
2. При перемещении мыши рисовать график функции, удаляя предыдущий график , вписывая его в прямоугольную рамку с центром на пересечении осей X,Y и углом , определяемым текущим положением указателя мыши .
3. По нажатию левой клавиши мыши в области палитры – стереть предыдущий график ,выбрать текущий цвет и уравнение аналитической функции.
4. По нажатию левой клавиши мыши вне области палитры – окончательно отрисовать текущий график аналитической функции, и сбросить режим динамической перерисовки графика;
5. По нажатию правой клавиши мыши – увеличить или уменьшить толщину линий графика.
6. По Alt+x - выход из программы.

Вариант 14

Написать приложение для Windows, выполняющее следующие действия:

Организовать вывод в окно «плавающего» графика функции $y=a*\sin(x)$ (x от -5π до $+5\pi$) следующим образом:

1. В нижней части окна отобразить палитру из 7 цветов, из середины остальной части окна оси нарисовать оси X,Y;
2. При перемещении мыши рисовать график функции, удаляя предыдущий график. При этом значение амплитуды a должно быть равным расстоянию по оси Y от середины экрана до текущего положения указателя мыши .
3. По нажатию левой клавиши мыши в области палитры – выбрать указанный цвет текущим для отображения графика.
4. По нажатию правой клавиши мыши – увеличить или уменьшить толщину линий графика.
5. В верхней части окна постоянно выводить позицию указателя мыши и значение параметра a .
6. По Alt+q - выход из программы.

Вариант 15

Написать приложение для Windows, выполняющее следующие действия:

1. Нарисовать на экране несколько вертикальных разноцветных полос фиксированной толщины.
2. При перемещении мыши в зоне какой либо полосы удлинять (укорачивать) полосу до текущего положения указателя мыши .
3. По нажатию левой клавиши мыши фиксировать полосу и больше её не перерисовывать при перемещении мыши.
4. По нажатию правой клавиши мыши в зоне зафиксированной полосы отменить режим фиксации и продолжать перерисовку при перемещении мыши.
5. По Alt+q - выход из программы.

Вариант 16

Написать приложение для Windows, выполняющее следующие действия:

1. Нарисовать на экране мишень.

2. При перемещении мыши подсвечивать сектор, в котором находится указатель.

3. По нажатию левой клавиши мыши рисовать в этом месте маленький круг и выводить на экран окно с количеством выбитых очков.

4. По нажатию правой клавиши мыши очищать экран от выстрелов.

5. По Alt+x - выход из программы.

Вариант 17

Написать приложение для Windows, выполняющее следующие действия:

1. Нарисовать на экране два прямоугольника разных цветов.

2. При попадании указателя мыши во внутреннюю область прямоугольника красить экран в цвет этого прямоугольника. Если указатель мыши находится вне зоны прямоугольников, то убрать закраску, оставив только прямоугольники.

3. По нажатию левой клавиши мыши рисовать в этом месте текст с координатами указателя мыши.

4. По нажатию правой клавиши мыши очищать экран, оставляя на нем только прямоугольники.

5. По Alt+x - выход из программы.

Вариант 18

Написать приложение для Windows, выполняющее следующие действия:

1. Разделить экран двумя диагональными линиями на четыре треугольника ;

2. При попадании указателя мыши во внутреннюю область треугольника подсвечивать только этот прямоугольник (если он ранее не закрашен).

3. При нажатии левой клавиши мыши во внутренней области треугольника закрашивать треугольник любым цветом.

4. При нажатии правой клавиши мыши во внутренней области закрашенного треугольника - убрать закраску.

5. При нажатии Esc – очищать экран, оставляя на нём только диагональные линии

6. По Alt+x - выход из программы.

Вариант 19

Написать приложение для Windows, выполняющее следующие действия:

1. Через центр экрана провести отрезок конечной длины.
2. При перемещении указателя мыши поворачивать отрезок относительно центра экрана в направлении указателя мыши (как стрелка компаса).
3. По нажатию левой клавиши мыши рисовать отрезок на экране с текущим наклоном
4. По нажатию правой клавиши мыши выводить на экран окно с текущим углом наклона отрезка к горизонтальной оси.
5. При нажатии Esc – очищать экран от нарисованных отрезков.
6. По Alt+q - выход из программы.

Вариант 20

Написать приложение для Windows, выполняющее следующие действия:

Организовать работу с набором прямоугольников.

1. По нажатию левой клавиши мыши создавать прямоугольники с центром в позиции указателя. Размеры выбирать случайным образом. Размеры нового прямоугольника не должны превышать 1/3 размеров окна. Прямоугольник изображать только заливкой (без контура). Цвет выбирать случайным образом, не светлее RGB(192, 192, 192). Допустимое количество прямоугольников – не более 30.
2. При перемещении мыши - при попадании указателя мыши на какой-либо прямоугольник – подсвечивать прямоугольник - рисовать более светлым цветом, который вычислять относительно обычного цвета прямоугольника.
3. По нажатию правой клавиши мыши, если указатель попадает в прямоугольник, удалять этот прямоугольник.
4. По нажатию цифровых и символьных клавиш – подсвечивать прямоугольник, соответствующий введённой цифре или символу, по текущему набору прямоугольников, в порядке – 0, 1, 2, ... 9, А, Б, В, ... Символ, соответствующий прямоугольнику, показывать в его центре.
5. По нажатию Esc – удалять все прямоугольники.
6. По нажатию Alt+x – выход из программы.

Вариант 21

Написать приложение для Windows, выполняющее следующие действия:

Организовать редактирование полилинии - ломаной линии, заданной набором точек с целочисленными координатами - POINT.

1. По нажатию левой клавиши мыши: в режиме изменения вершины (см. далее) – фиксировать выбранную вершину в новом положении. Если вблизи указателя нет созданной ранее вершины - добавлять в полилинию новую вершину, если есть – переходить в режим изменения (перемещения) вершины. Область указателя – круг радиусом 10 пикселей. Допустимое количество вершин – не более 30.

2. По нажатию правой клавиши – удалять вершину, находящуюся в области указателя.

3. По перемещению мыши: в режиме изменения вершины – обновлять положение выбранной вершины, иначе – искать вершину в области указателя.

4. В режиме редактирования динамически перерисовывать полилинию с учётом нового положения редактируемой вершины. Для рисования использовать функцию Polyline. Вершину в области указателя пометить (например, кружком).

5. По нажатию Esc - в режиме изменения вершины – восстановить исходное положение вершины, завершить редактирование. В режиме ожидания - удалить все вершины полилинии.

6. По нажатию Alt+x – выход из программы.

Вариант 22

Написать приложение для Windows, выполняющее следующие действия:

Создать простой однострочный редактор текста, реализующий некоторые функции стандартного элемента управления Windows (editbox). Количество символов – до 100.

1. Текстовую строку выводить из левого верхнего угла окна, курсор показывать

вертикальной линией.

2. По нажатию символьных клавиш – вставлять символы в текст в позиции курсора.

3. По нажатию Delete – удалять символ в позиции курсора.

4. По нажатию клавиш Влево/Вправо – перемещать курсор.

5. По нажатию левой клавиши мыши -- устанавливать курсор в ближайшую позицию в тексте.

6. По нажатию Alt+x – выход из программы.

Вариант 23

Написать приложение для Windows, выполняющее следующие действия:

Организовать работу с набором эллипсов.

1. По первому нажатию левой клавиши мыши задавать центр эллипса, по второму – задавать размеры полуосей (по описанному прямоугольнику) и создавать новый эллипс, цвет контура выбирать случайным образом. Допустимое количество эллипсов – не более 30.

2. При перемещении мыши перерисовывать изображение (контур) создаваемого эллипса. Если вводится первая точка (для координат центра) – размеры полуосей назначить, например: $A = 64$, $B = 32$.

3. По нажатию цифровых и символьных клавиш – удалять эллипс, соответствующий введённой цифре или символу, по текущему набору эллипсов в порядке их создания - 0, 1, 2, ... 9, А, Б, В, ... Символ, соответствующий эллипсу, показывать в его центре.

4. По нажатию правой клавиши мыши выводить информацию об эллипсе вблизи курсора (координаты центра, размеры полуосей, цвет), используя окно сообщения.

5. По нажатию Alt+x – выход из программы.

Вариант 24

Написать приложение для Windows, выполняющее следующие действия:

Организовать работу с набором отрезков прямой линии.

1. Предусмотреть 2 режима работы:

– по клавише 'N' – включить режим создания новых отрезков;

– по клавише 'E' – включить режим редактирования отрезков.

2. В режиме создания – по первому нажатию левой клавиши мыши задавать первую точку отрезка, по второму нажатию – вторую точку отрезка. Цвет выбирать случайным образом. Допустимое количество отрезков – не более 30.

3. В режиме редактирования – по первому нажатию левой клавиши – искать точку отрезка, ближайшую к указателю, и переходить к её перемещению; по второму нажатию – фиксировать конечное положение отрезка.

4. При перемещении мыши перерисовывать создаваемый или редактируемый отрезок. В режиме редактирования, если вводится первая точка, - искать точку отрезка, ближайшую к указателю и помечать её квадратом.

5. По клавише Esc в режиме редактирования – отменять редактирование указанной ранее вершины.

6. По клавише Del – в режиме редактирования - удалять отрезок, ближайший к указателю, если ожидается ввод редактируемой точки.

7. По нажатию Alt+x – выход из программы.

Вариант 25

Написать приложение для Windows, выполняющее следующие действия:

Организовать интерактивное редактирование прямоугольника.

1. В центре окна нарисовать прямоугольник. Размеры, перо и кисть выбрать произвольно.

2. По нажатию левой клавиши мыши – если найден угол или сторона прямоугольника вблизи указателя (см. далее) – переходить в режим редактирования прямоугольника. Если выполняется редактирование – фиксировать новую геометрию прямоугольника.

3. При перемещении мыши – искать ближайший угол или, если угол не найден, - сторону прямоугольника. Радиус поиска - 10 пикселей. Если найден угол – помечать его квадратиком, если найдена сторона – помечать её более толстой линией. В режиме редактирования – изменять геометрию прямоугольника в зависимости от указанного элемента – угла или стороны. Если указан угол – изменять ширину и высоту, если вертикальная сторона – ширину, горизонтальная – высоту (см.рис.). Прямоугольник динамически перерисовывать.

4. При нажатии Esc в режиме редактирования (перемещается угол или сторона прямоугольника) – отменять операцию, восстанавливать начальную геометрию прямоугольника.

5. При нажатии Shift и клавиш управления курсором – перемещать прямоугольник с шагом в 10 пикселей.

6. По нажатию Alt+x – выход из программы.

Вариант 26

Написать приложение для Windows, выполняющее следующие действия:

Организовать обработку простых геометрических фигур.

1. Создать набор из 15-20 фигур – прямоугольников и кругов. Параметры (тип, размеры и координаты) генерировать случайным образом, размеры фигур не должны превышать 12% ширины клиентской области окна.

2. Фигурам присвоить номера и буквы в порядке 0, 1, 2, ..., 9, А, Б, В, ... - в порядке создания фигур.

3. По нажатию левой клавиши мыши выводить окно сообщения с информацией о фигуре, в которую попал указатель – тип фигуры (прямоугольник/круг), номер (букву) и геометрические параметры.

4. По нажатию клавиш '0', '1', '2', ..., '9', 'А', 'Б', 'В' - выводить окно сообщения с информацией о соответствующей фигуре.

5. По нажатию клавиши <пробел> - генерировать новый набор фигур, по нажатию Alt+x – выход из программы.

Вариант 27

Написать приложение для Windows, выполняющее следующие действия:

Организовать обработку простых геометрических фигур.

1. Создать набор из 15-20 треугольников. Параметры (координаты) генерировать случайным образом, размеры фигур не должны превышать 12% ширины клиентской области окна.

2. Фигурам присвоить номера и буквы в порядке 0, 1, 2, ..., 9, А, Б, В, ... - в порядке создания фигур.

3. По нажатию левой клавиши мыши выводить окно сообщения с информацией о фигуре, в которую попал указатель – номер (букву) и координаты вершин. Для проверки попадания точки в треугольник использовать определитель (удвоенная площадь треугольника со знаком): $x_1 * (y_2 - y_3) + x_2 * (y_3 - y_1) + x_3 * (y_1 - y_2)$, где 1, 2 и 3 – номера вершин треугольника. Определитель имеет положительный знак, если точки расположены против хода часовой стрелки.

4. По нажатию клавиш '0', '1', '2', ..., '9', 'А', 'Б', 'В' - выводить окно сообщения с информацией о соответствующей фигуре.

5. По нажатию клавиши <пробел> - генерировать новый набор фигур, по нажатию Alt+x – выход из программы.

Вариант 28

Написать приложение для Windows, выполняющее следующие действия:

Организовать обработку простых геометрических фигур.

1. Создать набор из 15-20 фигур – отрезков, кругов, прямоугольников. Тип, размеры и цвета фигур генерировать случайным образом, причём размеры фигур не должны превышать 12% ширины клиентской области окна.

2. Фигурам присвоить номера и буквы в порядке 0, 1, 2, ..., 9, А, Б, В, ... - в порядке создания фигур. Номер выводить в центре фигуры.

3. По перемещению мыши искать фигуру, в которую попал указатель и перерисовывать (выделять) её специальным цветом.

4. По нажатию клавиш '0', '1', '2', ..., '9', 'А', 'Б', 'В' - выводить окно сообщения с информацией о соответствующей фигуре.

5. По нажатию клавиши <пробел> - генерировать новый набор фигур, по нажатию Alt+x – выход из программы.

Вариант 29

Написать приложение для Windows, выполняющее следующие действия:

Создать простой калькулятор.

1. Нарисовать в окне набор кнопок (в виде прямоугольников) с метками '0', '1', '2', ... '9', '+', '-', '*', '/', '=', 'C', расположив кнопки в один или несколько рядов. Предусмотреть поле – прямоугольник, отображающее исходные данные и результат вычисления, например: «10*12=120».

2. Калькулятор имеет 2 состояния: 1) ожидание ввода первого числа, операции или сброса ('C');

2) ожидание ввода второго числа, команды вычисления ('=') или сброса ('C'). Переход в состояние 2 выполняется после ввода операции, а в состояние 1 – после команды вычисления или сброса.

3. По нажатию клавиш отображать вводимое число, операцию и результат в виде текста. Например, если вводится первое число – отображается «10», после ввода операции '*' – «10*», при вводе второго числа – «10*12», а после ввода '=' - «10*12=120».

4. Обеспечить возможность ввода данных кликами левой клавиши мыши по прямоугольникам кнопок.

5. По нажатию 'С' (сброс) – переводить калькулятор в состояние 1 и очищать поле данных и результата.

6. По нажатию Alt+x – выход из программы.

Вариант 30

Написать приложение для Windows, выполняющее следующие действия:

1. В окне нарисовать прямоугольную таблицу со числовыми значениями, выбранными случайно из диапазона [0..100]. Количество строк и столбцов выбирать по случайному закону из диапазона [10..20]. Ячейки таблицы изобразить кружками. Таблицу центрировать по клиентской области окна.

2. По нажатию левой клавиши мыши выводить в окно сообщения сумму значений в строке, на которой находится указатель, а по нажатию правой клавиши – сумму значений в столбце.

3. По перемещению мыши определять элемент, над которым находится указатель и помечать (перерисовывать) элемент другим цветом.

4. По нажатию <пробел> - сгенерировать новую таблицу, по нажатию Alt+e – выход из программы.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Павловская Т. А. С/С++. Программирование на языке высокого уровня. — (Серия «Учебник для вузов»). / Т.А. Павловская. - Санкт-Петербург : Питер, 2021. - 464 с. - ISBN 978-5-4461-1350-7. - URL: https://ibooks.ru/bookshelf/376844/reading . - Текст: электронный.	https://ibooks.ru/bookshelf/376844/reading . - Текст: электронный.
2	Дадян Э.Г. Современные технологии программирования. Язык С#. / Э.Г. Дадян. - Москва : Инфра-М, 2021. - 335 с. - ISBN 978-5-16-016997-2. - URL: https://ibooks.ru/bookshelf/378034/reading . - Текст: электронный.	https://ibooks.ru/bookshelf/378034/reading . - Текст: электронный.

3	Огнева, М. В. Программирование на языке С++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/555533 .	https://urait.ru/bcode/555533
4	Лафоре Р. Объектно-ориентированное программирование в С++. Классика Computer Science. 4-е изд. — (Серия «Классика computer science»). / Р. Лафоре. - Санкт-Петербург : Питер, 2021. - 928 с. - ISBN 978-5-4461-0927-2. - URL: https://ibooks.ru/bookshelf/376836/reading . - Текст: электронный.	https://ibooks.ru/bookshelf/376836/reading . - Текст: электронный.

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

Официальный сайт РУТ (МИИТ) (<https://www.miit.ru/>).

Научно-техническая библиотека РУТ (МИИТ) (<http://library.miit.ru>).

Электронно-библиотечная система издательства «Лань» (<http://e.lanbook.com/>).

Электронно-библиотечная система ibooks.ru (<http://ibooks.ru/>).

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Microsoft Internet Explorer (или другой браузер).

Операционная система Microsoft Windows.

Microsoft Office.

MS Visual Studio C++.

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

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

9. Форма промежуточной аттестации:

Экзамен в 4 семестре.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

доцент, доцент, к.н. кафедры
«Системы автоматизированного
проектирования»

И.В. Нестеров

Согласовано:

и.о. заведующего кафедрой САП

И.В. Нестеров

Председатель учебно-методической
комиссии

М.Ф. Гуськова