

Утверждено

на заседании каф. ПОКС

Протокол № _____, от _____

Зав. каф. ПОКС _____ Тен И.Г.

**Темы курсовых проектов по дисциплине
«Объектно-ориентированное программирование»**

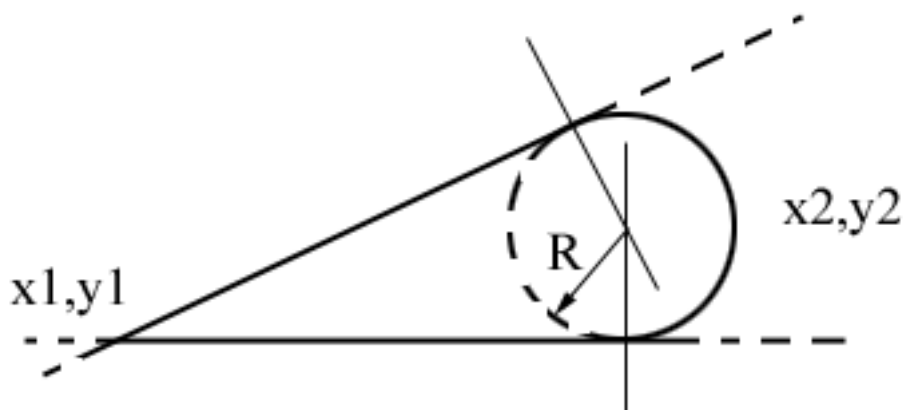
1. *Логическая игра “Крестики-нолики” на неограниченном поле* Приложение является реализацией известной логической игры “Крестики-нолики” В данной реализации предусматривается игра двух игроков на неограниченном поле. Цель игры . построить непрерывную линию из пяти или более фишек (крестиков или ноликов) по горизонтали, вертикали или диагонали. Возможны два режима игры, когда выигрывает тот, кто первым построит линию, или кто больше наберет очков за определенное количество времени. Во втором случае количество построенных линий не ограничено, каждая фишка в линии приносит игроку одно очко. Приложение должно обеспечивать начало новой игры на чистом поле, а также проверку соответствия действий игроков правилам игры и условия окончания игры. Роль одного из игроков (по выбору пользователя) может выполнять компьютер. Для исключения возможности образования изолированных игр на одном поле и неоправданного увеличения размера поля следует установить максимально допустимое расстояние (не более пяти) от уже существующих фишек до новой, размещаемой игроком.
2. *Логическая игра “Точки”* Данное приложение является реализацией известной логической игры. В данной реализации предусматривается игра двух игроков на квадратном поле фиксированного размера. Цель игры . заполнить максимальное количество клеток игрового поля своими фишками. Игроки выполняют ходы по очереди. Ход игрока заключается в произвольной установке линии на границе двух ячеек, причем, если какая-либо ячейка оказывается со всех четырех сторон обрамлена линиями, то она отмечается фишкой данного игрока, а самому игроку предоставляется дополнительный ход, и т.д. Игра заканчивается, когда все игровое поле оказывается заполнено фишками. Выигрывает тот игрок, чьих фишек на поле больше на момент окончания игры. Приложение должно обеспечивать начало новой игры на чистом поле, а также проверку соответствия действий игроков правилам игры и условия окончания игры. Роль одного из игроков (по выбору пользователя) может выполнять компьютер.
3. *Логическая игра “Шарики”* Разрабатываемое приложение представляет собой программную реализацию известной логической игры. Цель игры состоит в том чтобы набрать максимальное количество очков. Суть игры состоит в следующем: на игровом поле отображаются круги разного цвета, игроку предлагается создать линию из кругов одинакового цвета, расположенных по горизонтали или по вертикали, созданная линия .сгорает.. Причем линия может .сгореть., только в том случае если количество элементов в ней равно 3. После этой процедуры пустые места на игровом поле заполняются новыми элементами. Линию можно создать путем перестановки по горизонтали или по вертикали соседних элементов. Игра заканчивается тогда когда не возможно создать ни одной линии. Приложение должно обеспечивать возможность задания количества цветов элементов (кругов). Следует учесть что цвета кругов выбираются произвольным образом исходя из заданного количества. Кроме того, линии (3 и более элементов) получаемые при произвольной расстановки элементов “сгорают”.
4. *Логическая игра “Морской бой”* Разрабатываемое приложение представляет собой программную реализацию известной логической игры. Приложение должно обеспечивать:
 - a. Расстановку .кораблей. на игровом поле 10x10
 - b. Выбор противника (человек, компьютер).
 - c. Изменение интерфейса в зависимости от выбора противника.
 - d. Фиксацию имен противников и число побед.

5. *Логическая игра “Реверси”* Разрабатываемое приложение представляет собой программную реализацию логической игры “Реверси”. Игра идет на поле произвольного размера. Два игрока по очереди устанавливают фишки своего цвета на поле. Фишку можно ставить только на те клетки, рядом с которыми уже стоят фишки. Если между установленной фишкой и какой либо другой фишкой того же цвета находятся фишки другого цвета, все они меняют свой цвет.
6. *Логическая игра “Водопроводчик”* Суть игры заключается в постройке трубопровода от одного края игрового поля к другому за определенное время. Играет один игрок. Компьютер предлагает очередной узел трубопровода, игрок может присоединить его к концу трубопровода, предварительно повернув его нужной стороной. В определенный момент времени с начала игры, в трубопровод начинает подаваться вода, которая медленно заполняет трубопровод. Если трубопровод будет заполнен до того, как игрок закончит строительство - игрок проиграл.
7. *Логическая игра “Тетрис”* Программа представляет собой Тетрис на поле произвольного размера. Программа должна предоставлять возможность выбирать размер фигур (4, 5, 6, 7 клеток). Скорость падения управляется автоматически в зависимости от времени игры.
8. *Логическая игра “Авеле”* Разрабатываемое приложение представляет собой программную реализацию африканской игры “Авеле” (другое название “Манкала”). Игровое поле состоит из двух рядов, по 6 лунок в каждом. Каждая лунка в начале игры содержит 4 камня. Нижний ряд принадлежит игроку, верхний - его противнику. Игроки делают ходы по очереди. Для хода игрок берет камни из любой своей непустой лунки, и раскладывает их по одному в каждую следующую лунку, двигаясь против часовой стрелки. Начальная лунка всегда пропускается. Если игрок кладет последний камень в лунку противника И лунка содержит 2 или 3 камня (включая только что положенный) то игрок собирает камни из лунок, двигаясь по часовой стрелке, до тех пор, пока два вышеприведенных условия соблюдаются. Если игрок не может сделать очередной ход, он собирает все оставшиеся камни, игра заканчивается и определяется победитель.
9. *Игра “Шадокки”* Великий колдун сказал, что шадокские ракеты терпят неудачу потому, что не хватает транзисторов в системах безопасности. Транзисторы нужного типа собирают с растений, произрастающих на огородах Гибби. Вы должны отправиться на планету Гибби и привести нужные транзисторы. Будьте осторожны, Гибби очень умны. Они позволяют Шадоккам забираться в свои огороды, но окружают огород со всех сторон. Всякий раз, когда растение зацветает и дает транзистор, они мчатся, чтобы собрать урожай раньше шадокка. Обратите внимание, что растения обладают разной плодородностью.
10. *Логическая игра “Шашки”* Разрабатываемое приложение представляет собой программную реализацию логической игры “Шашки”. Необходимо реализовать подмножество правил классических шашек на доске 8x8.
11. *Логическая игра «Обратный тетрис» (тетрис наоборот).* Игрок выбирает фигуру и бросает, компьютер пытается установить ее в стакан. Цель игры – завалить компьютер.
12. *Логическая игра «Шахматы».* Необходимо реализовать разумное подмножество международных шахматных правил, как они определены FIDE. Учить игре в шахматы компьютер не нужно, достаточно реализовать игру двух человек за одним компьютером.
13. Игра-аркада «Bounce»
14. Игра-аркада «Snake».
15. Игра-аркада «Космическая дуэль»
16. Разработать программу для управления базой данных «Абитуриенты». Запись в базе данных содержит следующие поля: ФИО, год рождения, школа, средний балл. Реализовать функции создания в ней записи, ее редактирования, удаления, поиска записи по значению поля, поиска номера школы с максимальным количеством абитуриентов со средним баллом большим 4.
17. Разработать программу для расчета площади фигуры, ограниченной осью OX, функцией $y=ax^2+b$ и прямыми $x=k$, $x=t$, методом Монте-Карло (имитационное моделирование). Построить график функции с осями координат, графики прямых и закрасить вычисляемую

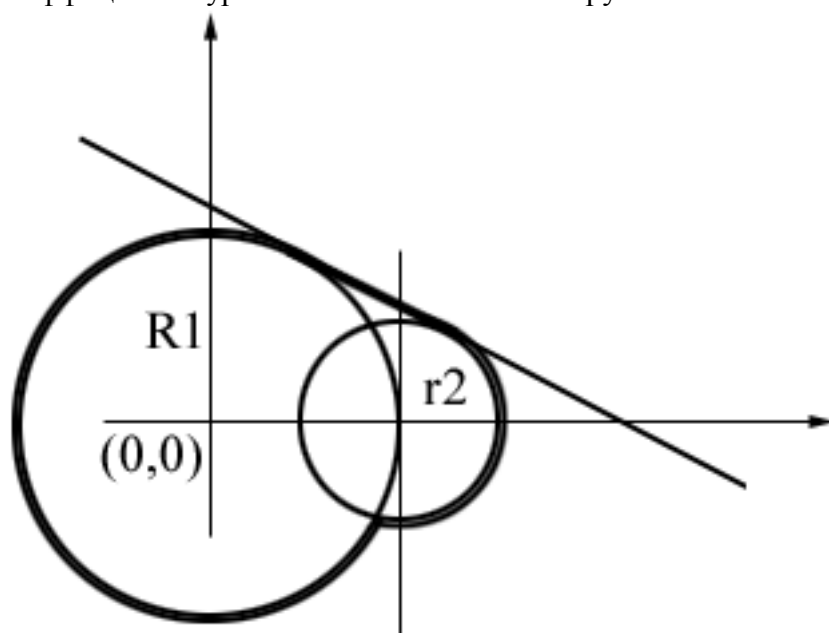
площадь. Значения a , b , k , m задаются пользователем.

18. Разработать программу расчета размера выплат для погашения кредита в каждый месяц. Сумма выплат в месяц составляет сумму доли от суммы кредита (она неизменна в течение всего срока погашения кредита) и процентов на остаток от суммы кредита. Построить график выплат в течение всего срока. Выплаты и начисление процентов на оставшуюся сумму производятся ежемесячно. Исходные данные: размер кредита, срок выплат (от 1 до 5 лет), процентная ставка.
19. Разработать программу для управления базой данных «Фонотека». Реализовать функции создания в ней записи, ее редактирования, удаления и поиска записи по значению поля. Запись в базе данных содержит следующие поля: исполнитель, название альбома, жанр, год выпуска.
20. Разработать программу для реализации алгоритма интерполяционного поиска числа. Для поиска программа должна сформировать массив случайных чисел, упорядоченных с помощью алгоритма сортировки пузырька. Размерность массива задается пользователем. Определить O -сложность алгоритма.
21. Разработать программу для реализации алгоритма последовательного поиска. Определить функцию изменения времени вычисления от объема исходных данных и O -сложность алгоритма. Для поиска использовать файл со случайными числами, для формирования которого предусмотреть функцию в программе и кнопку ее вызова.
22. Разработать программу для реализации алгоритма быстрого последовательного поиска. Определить функцию изменения времени вычисления от объема исходных данных и O -сложность алгоритма. Для поиска использовать файл со случайными числами, для формирования которого предусмотреть функцию в программе и кнопку ее вызова.
23. Разработать программу для реализации алгоритма бинарного поиска. Определить функцию изменения времени вычисления от объема исходных данных и O -сложность алгоритма. Для поиска использовать файл со случайными числами, для формирования которого предусмотреть функцию в программе и кнопку ее вызова. Случайные числа в файле должны быть отсортированы каким-либо из известных алгоритмов.
24. Разработать программу для реализации алгоритма точного поиска подстроки в строке Кнута-Морриса-Пратта. Определить функцию изменения времени вычисления от объема исходных данных и O -сложность алгоритма. (текст брать из готового файла, указанного пользователем, отображать его в диалоге, пользователь осуществляет ввод искомой подстроки)
25. Разработать программу для реализации алгоритма точного поиска подстроки в строке Боуера-Мура. Определить функцию изменения времени вычисления от объема исходных данных и O -сложность алгоритма (текст брать из готового файла, указанного пользователем, отображать его в диалоге, пользователь осуществляет ввод искомой подстроки).
26. Разработать программу для реализации алгоритма сортировки методом пирамиды. Вывести в диалоге столбчатую диаграмму зависимости времени работы алгоритма от объема исходных данных. Исходные данные брать из файла, для формирования которого использовать случайные числа. Разработать также функцию для создания такого файла с возможностью изменения количества записываемых целых чисел. Отсортированный массив записать в новый файл. Определить O -сложность алгоритма.
27. Разработать программу для решения квадратного уравнения и построения графика функции с осями координат. Коэффициенты уравнения считывать из файла, выбранного пользователем.
28. Разработать программу для определения координат точки пересечения двух прямых и построения их графиков с осями координат. Коэффициенты уравнений прямых считывать из файла, указанного пользователем.
29. Разработать программу для расчета площади фигуры, изображенной на рисунке сплошной линией. Принять $x_1=0$, $y_1=0$. $R=y_2-y_1=y_2-0=y_2$. Исходные данные (x_2, y_2) должны быть считаны из выбранного пользователем файла. Построить изображение полученной

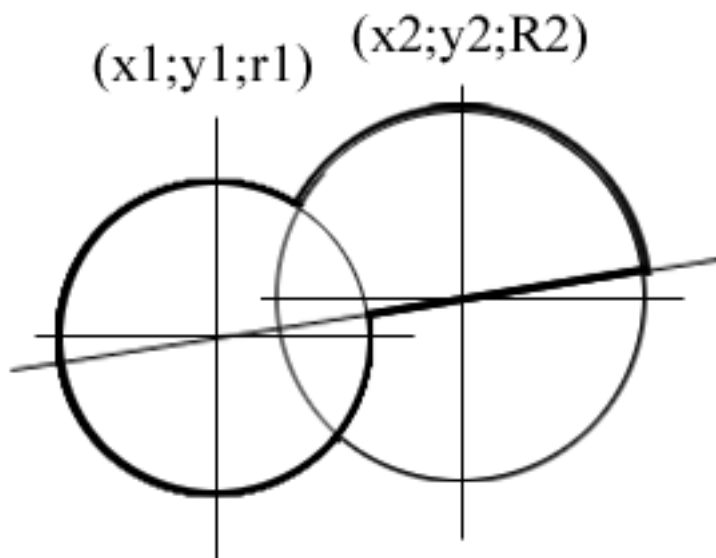
фигуры.



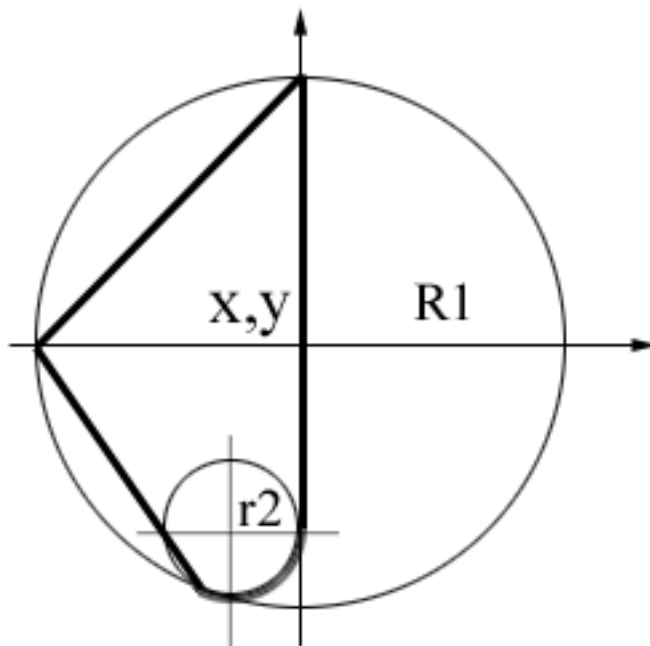
30. Разработать программу для построения изображения, приведенного на рисунке. Исходные данные (R_1 , r_2) должны быть считаны из выбранного пользователем файла. Определить коэффициенты уравнения касательной к окружностям. Вывести их в диалоге приложения.



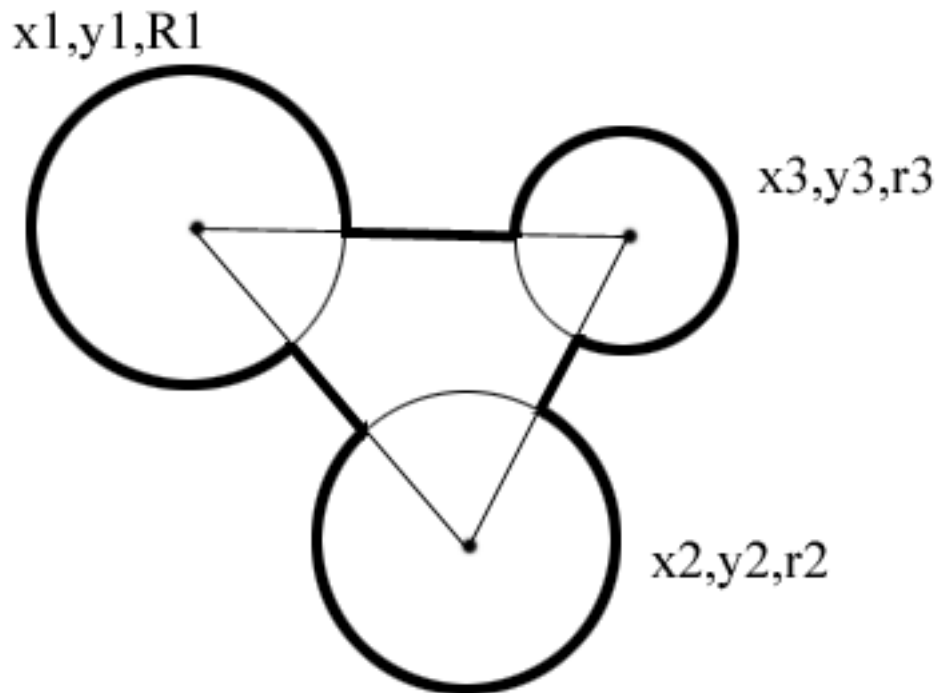
31. Разработать программу для реализации алгоритма сортировки методом Шелла. Вывести на экран график зависимости времени работы алгоритма от объема исходных данных. Исходные данные брать из файла, для формирования которого использовать случайные числа. Разработать также функцию для создания такого файла с возможностью изменения количества записываемых целых чисел. Отсортированный массив записать в новый файл. Определить O -сложность алгоритма.
32. Разработать программу для построения графика функции $y=2*\sin(x)*e^{x/5}$. Считать из файла, указанного пользователем значения абсциссы X , рассчитать для них значения ординаты Y , записать результат вычисления в новый файл в формате $X1 - Y1$
33. $X2 - Y2...$
34. Разработать программу для построения изображения, приведенного на рисунке. Исходные данные (x_1 , y_1 , x_2 , y_2 , r_1 , R_2) должны быть считаны из выбранного пользователем файла. Программа должна осуществлять проверку правильности исходных данных (обязательное пересечение окружностей)



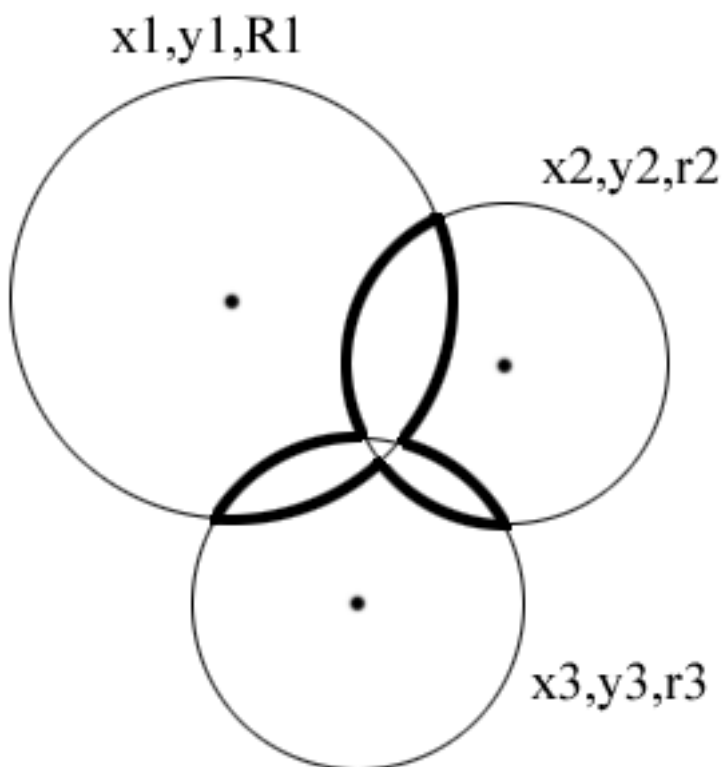
35. Разработать программу для построения изображения, приведенного на рисунке. Принять $x=0$, $y=0$. Исходные данные ($R1$, $r2$) должны быть считаны из выбранного пользователем файла. Программа должна осуществлять проверку правильности исходных данных (обязательное касание окружностей).



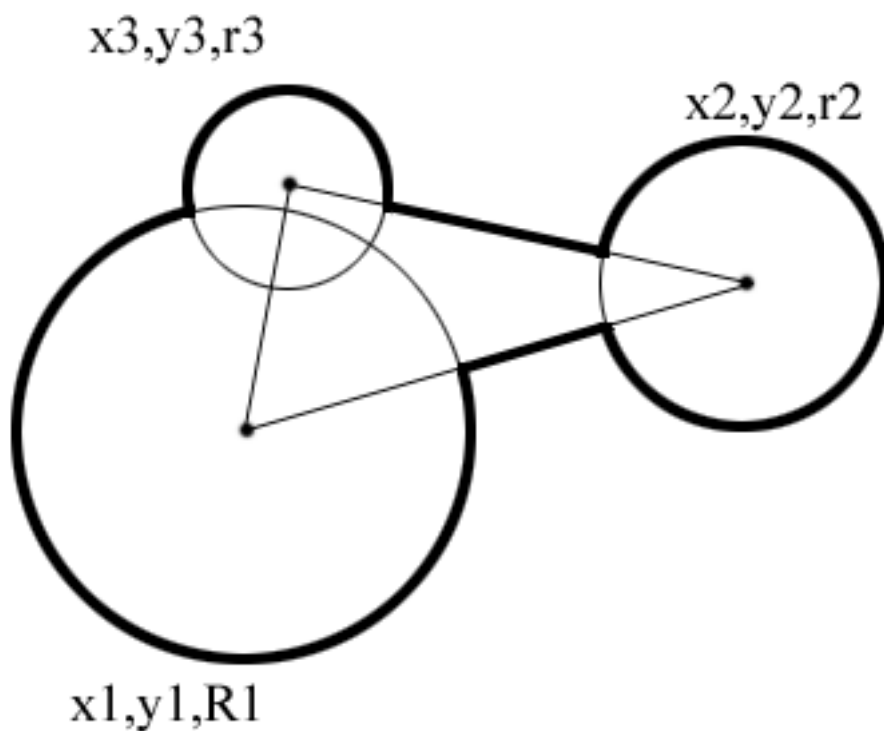
36. Разработать программу для расчета площади фигуры, изображенной на рисунке сплошной утолщенной линией. Исходные данные ($x1$, $y1$, $R1$, $x2$, $y2$, $R2$, $x3$, $y3$, $R3$) должны быть введены пользователем. Построить изображение полученной фигуры. Осуществлять проверку исходных данных на соответствие варианту (окружности не пересекаются).



37. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные ($x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3$) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (все окружности пересекаются).



38. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные ($x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3$) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (только две окружности пересекаются).

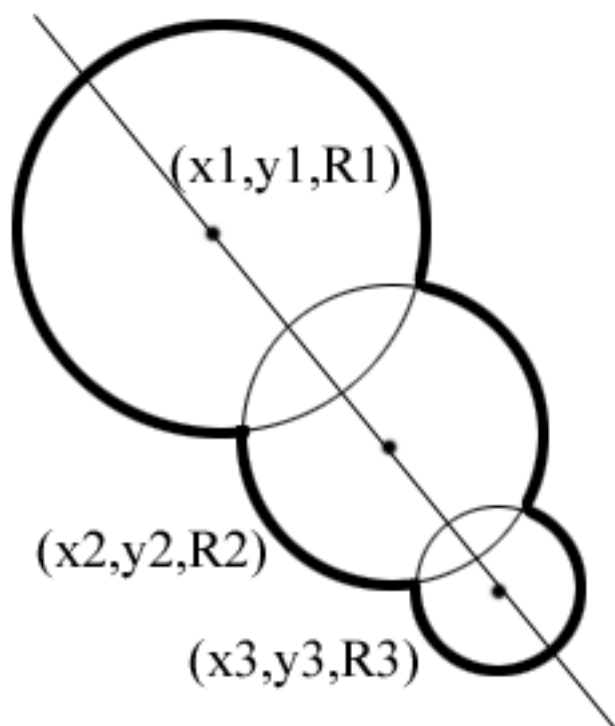


39. Разработать программу для реализации алгоритма внешней сортировки слиянием для 10 файлов максимум (количество файлов для слияния от 2 до 10 задается пользователем). Данные предварительно записать в n файлов. Пользователь задает имена файлов с исходными данными. Результаты сортировки записать в новый файл, указанный пользователем.
40. Разработать программу, которая будет кодировать текст файла (*.txt), например, следующим образом: ASCII-код каждого символа будет увеличиваться на одно и то же целое число (+1 или +28). Результирующий код программа должна записать в новый файл (*.txt). Предусмотреть также функцию для декодирования файлов. Имена файлов задаются пользователем.
41. Разработать программу для вычисления определенного интеграла методом трапеций для функции типа: $y = a * e^{b * \sin(c * x)} + d$. Коэффициенты a , b , c , d и границы интервала интегрирования задаются пользователем. В диалоге необходимо изобразить график функции, графики прямых, ограничивающих область интегрирования, оси координат.
42. Разработать программу, осуществляющую чтение текста из файла, сортировку слов текста по алфавиту методом выбора, запись полученного текста в новый файл.
43. Разработать программу для реализации игры в кости для двух игроков (пять костей, выпадают значения случайным образом; кости, значения которых отличны от 1 и 5 можно перекинуть; подсчет очков: 1 – 10 очков, 5 – 5 очков. Цель набрать ровно 300 или 500 очков. Побеждает игрок, первый набравший такую сумму очков). Кости реализовать графически.
44. Разработать программу, которая выводит на экран круговую диаграмму успеваемости в группе по итогам сессии. Исходные данные считываются из файла (количество оценок 5, 4, 3, 2 и количество неявившихся студентов). На диаграмме каждый столбец должен быть отмечен соответствующим цветом, расшифровка которых также должна быть приведена, также рассчитать процентное содержание каждого вида оценок от общего числа студентов.
45. Разработать программу, которая считывает текст из файла (*.txt), выводит его в этом виде на экран, переводит его согласно азбуке Морзе, изображает последовательность полученных символов на экране и записывает их в новый файл. Имена файлов задаются пользователем.

А	.-	Б	-...	В	.---	Г	--.
Д	-..	Е	----	Ж	...-	З	--..
И	...	Й	К	-.-	Л	.-..
М	--	Н	-.	О	---	П	.-.-
Р	.-.	С	...	Т	-	У	..-
Ф	..-.	Х	Ц	.-.-	Ч	---.
Ш	----	Щ	---.-	Ъ	-.-	Ы	-.--
Ь	-.--	Э	..-	Ю	..--	Я	.-.-

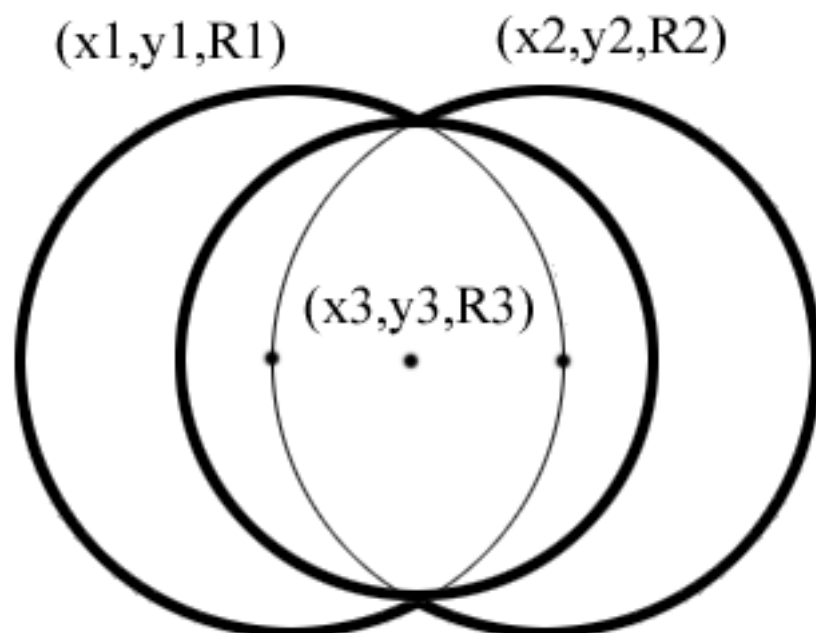
Азбука Морзе.

46. Разработать программу, решающую задачу о 8-ми ферзях с поиском всех возможных решений. Доска размером 8 x 8. Записать все возможные расстановки ферзей в файл и отобразить на экране.
47. Разработать программу для построения графика колебаний температуры. Исходные данные (температура в градусах в каждый день за отчетный период) считывается из файла, указанного пользователем. Разработать функции определения средней температуры за период, максимальный скачок температуры за день, день с максимальным отклонением температуры от среднего значения.
48. Разработать программную реализацию следующего алгоритма: дан текст из русских букв и знаков препинания, необходимо провести частотный анализ букв этого текста, т.е. напечатать каждую букву с указанием количества ее вхождений и процента от всего объема текста.
49. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные ($x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3$) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (окружности попарно пересекаются, центры окружностей лежат на одной прямой).

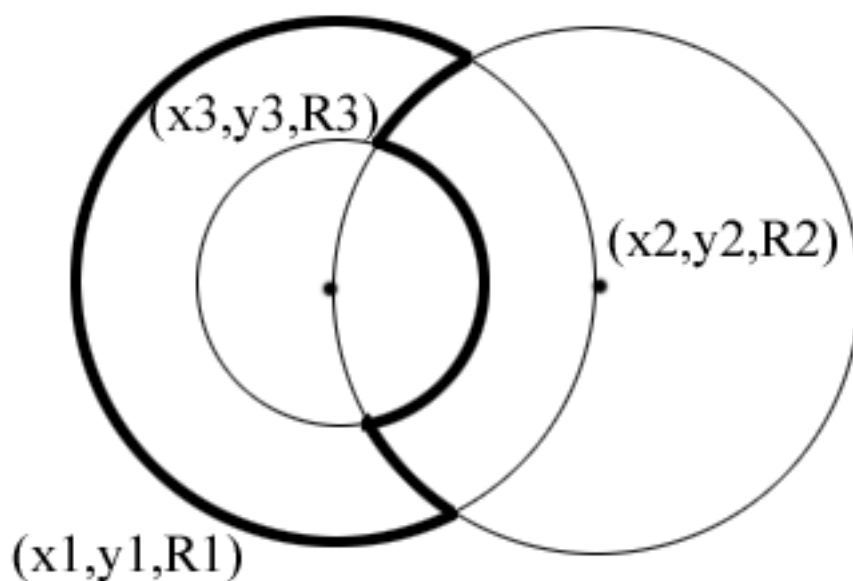


50. Разработать программу для построения изображения фигуры, приведенной на рисунке.

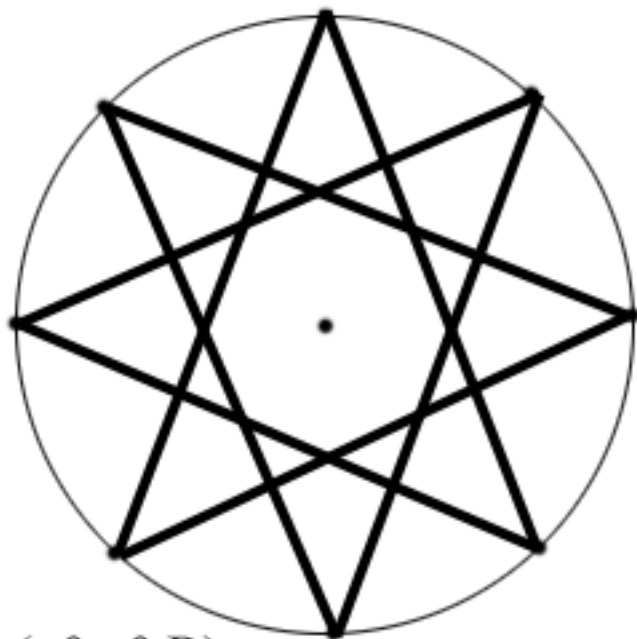
Исходные данные $(x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



51. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.

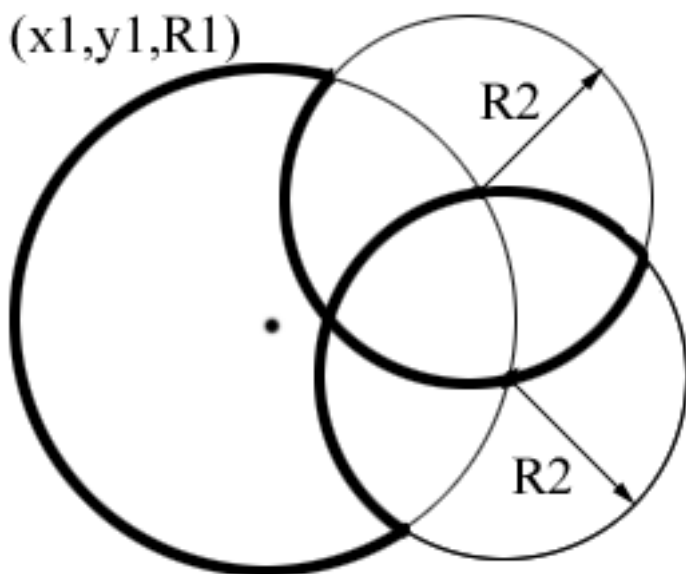


52. Разработать программу для построения изображения фигуры, приведенной на рисунке. Количество лучей может быть задано пользователем или считано из файла.

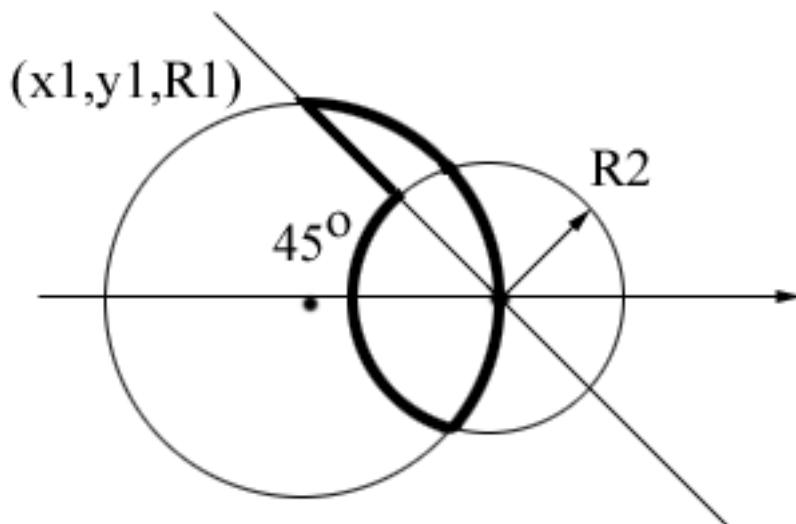


$(x_0, y_0, R);$
 $n = 1..12$ - количество лучей

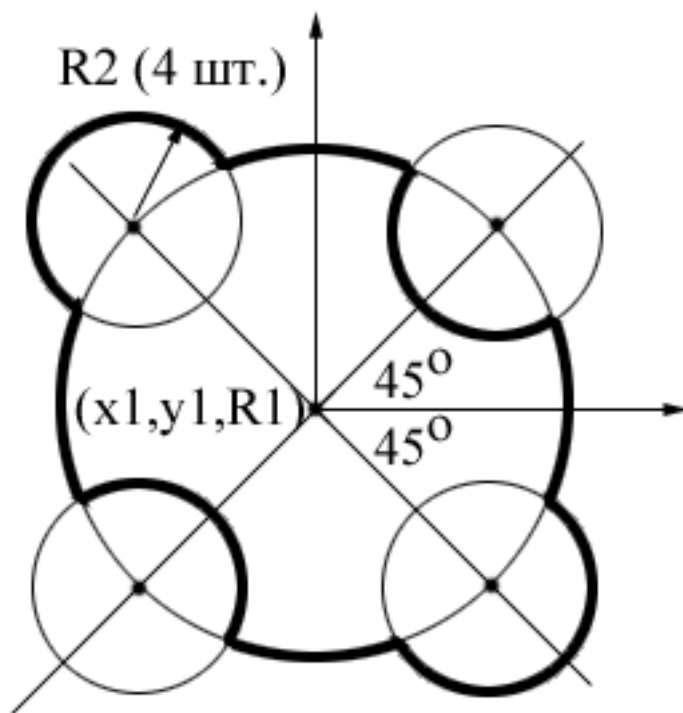
53. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные (x_1, y_1, R_1, R_2) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



54. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные (x_1, y_1, R_1, R_2) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



55. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x1, y1, R1, R2)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



56. Разработать программу для реализации алгоритма сортировки подсчетом распределений (в массиве данные должны часто повторяться). Сортируемую последовательность необходимо считывать из внешнего файла, указанного пользователем. Результат сортировки записать в новый файл. Оценить O -сложность алгоритма.
57. Разработать программу для реализации алгоритма сортировки подсчетом сравнений. Сортируемую последовательность необходимо генерировать из случайных чисел. Результат представить графически в виде двух наборов точек: первый – до сортировки, второй – после, по оси OX откладывать порядковый номер числа в последовательности, по оси OY – его значение. Оценить O -сложность алгоритма.
58. Разработать программу для реализации алгоритма сортировки вставками. Сортируемую последовательность необходимо генерировать из случайных чисел. Результат представить графически в виде двух наборов точек: первый – до сортировки, второй – после, по оси OX

- откладывать порядковый номер числа в последовательности, по оси ОУ – его значение; и записать в файл, указанный пользователем. Оценить О-сложность алгоритма.
59. Разработать программу для реализации алгоритма пузырьковой сортировки. Сортируемую последовательность необходимо генерировать из случайных чисел. Результат представить графически в виде двух наборов вертикальных линий: первый – до сортировки, второй – после, по оси ОХ откладывать порядковый номер числа в последовательности, по оси ОУ – его значение, проводить линию от оси ОХ до найденной точки; и записать в файл, указанный пользователем. Оценить О-сложность алгоритма.
 60. Разработать программу для реализации алгоритма быстрой сортировки. Сортируемую последовательность необходимо генерировать из случайных чисел. Результат сортировки записать в файл, указанный пользователем. Оценить О-сложность алгоритма.
 61. Разработать программу для реализации алгоритма сортировки выбором. Сортируемую последовательность необходимо генерировать из случайных чисел. Результат представить графически в виде двух наборов вертикальных линий: первый – до сортировки, второй – после, по оси ОХ откладывать порядковый номер числа в последовательности, по оси ОУ – его значение, проводить линию от оси ОХ до найденной точки; и записать в файл, указанный пользователем. Оценить О-сложность алгоритма.
 62. Разработать программу для итерационного поиска размера равномерных выплат по кредиту. Выплаты и начисление процентов на оставшуюся сумму производятся ежемесячно. Исходные данные: размер кредита, срок выплат (от 1 до 5 лет), процентная ставка. В диалоге изобразить график изменения остатка от суммы кредита.
 63. Разработать программу для построения графика функции с осями координат по заданным во внешнем файле точкам. В программе должен быть предусмотрен выбор файла в каталоге.
 64. Разработать программу для работы с базой данных, содержащей следующие поля: фамилия, имя, отчество, год рождения, паспортные данные. Предусмотреть возможность открытия указанного пользователем файла такой структуры, отображение содержимого файла в диалоге, ввод новых данных для дозаписи в файл, сортировку записей по году рождения методом подсчета распределений, отображение в диалоге отсортированной таким образом базы данных.
 65. Разработать программу для вычисления определенного интеграла методом средних прямоугольников для функции типа: $y=a*\cos(b*x)+x^d$. Коэффициенты a , b , c , d и границы интервала интегрирования считываются из файла. В диалоге необходимо изобразить график функции, графики прямых, ограничивающих область интегрирования, оси координат.
 66. Разработать программу для расчета длины полета тела массой m , брошенного под углом к горизонту α с начальной скоростью V_0 . Исходные данные считать из файла, указанного пользователем. Результат дозаписать в тот же файл.
 67. Разработать программу для перевода десятичных чисел в двоичные. Исходные данные считывать из файла, указанного пользователем. Результат отображать на экране, также должна быть возможность записи результата в новый файл.
 68. Разработать программу для расчета времени, необходимого для выполнения сортировки методом Шелла, массив данных должен быть считан из файла, указанного пользователем, и отображен в диалоге до и после сортировки.
 69. Разработать программу для расчета времени, необходимого для выполнения сортировки методом пирамиды, массив данных должен быть считан из файла, указанного пользователем, и отображен в диалоге до и после сортировки.
 70. Разработать программу для расчета времени, необходимого для выполнения сортировки методом подсчета сравнений, массив данных должен быть считан из файла, указанного пользователем, и отображен в диалоге до и после сортировки.