

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Силин Яков Петрович
Должность: Ректор
Дата подписания: 10.06.2026 09:52:34
Уникальный программный ключ:
24f866be2aca16484036a8cbb3c509a9531e605f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «Уральский государственный экономический университет»

09.12.2025 г.
протокол № 12
И.о. зав. кафедрой Кольева Н.С.

Утверждена
Советом по учебно-методическим
вопросам и качеству образования
16 декабря 2025 г.
протокол № 4
Председатель  Карх Д.А.



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Наименование дисциплины	Программирование
Направление подготовки	09.03.03 Прикладная информатика
Профиль	Инжиниринг предприятий и информационных систем
Форма обучения	заочная
Год набора	2026
Разработана:	
Доцент, к.п.н.	Кольева Н.С.
Ст. преподаватель	Панова М.В.

Екатеринбург
2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП	3
3. ОБЪЕМ ДИСЦИПЛИНЫ	3
4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП	3
5. ТЕМАТИЧЕСКИЙ ПЛАН	4
6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ	5
7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	6
8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ	9
9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ	9
10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	10
11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ	11

ВВЕДЕНИЕ

Рабочая программа дисциплины является частью основной профессиональной образовательной программы высшего образования - программы бакалавриата, разработанной в соответствии с ФГОС ВО

ФГОС ВО	Федеральный государственный образовательный стандарт высшего образования- бакалавриат по направлению подготовки 09.03.03 Прикладная информатика(приказ Минобрнауки России от 19.09.2017 г. №
---------	--

1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Формирование первоначальных знаний, умений и навыков разработки программных средств на языках высокого уровня.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к обязательной части учебного плана.

3. ОБЪЕМ ДИСЦИПЛИНЫ

Промежуточная аттестация	Часов					З.е.
	Всего за семестр	Контактная работа (поуч.зан.)			Самостоятельная работа в том числе подготовка контрольных и курсовых	
		Всего	Лекции	Лабораторные		
Семестр 2						
Экзамен, Контрольная работа	144	12	4	8	123	4

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ОПОП

В результате освоения ОПОП у выпускника должны быть сформированы компетенции, установленные в соответствии ФГОС ВО.

Шифр и наименование компетенции	Индикаторы достижения компетенций
ОПК-1 Способен применять естественнонаучные и общепрофессиональные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ИД-1.ОПК-1 Знать: основы высшей математики, физики, основы вычислительной техники и программирования.

<p>ОПК-1 Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;</p>	<p>ИД-2.ОПК-1 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования.</p>
	<p>ИД-3.ОПК-1 Иметь практический опыт: теоретического и экспериментального исследования объектов профессиональной деятельности</p>
<p>ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;</p>	<p>ИД-1.ОПК-7 Знать: основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.</p>
	<p>ИД-2.ОПК-7 Уметь: применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ</p>
	<p>ИД-3.ОПК-7 Иметь практический опыт: программирования, отладки и тестирования прототипов программно-технических комплексов задач</p>

5. ТЕМАТИЧЕСКИЙ ПЛАН

Тема	Наименование темы	Всего часов	Контактная работа (по уч.зан.)			Самост. работа	Контроль самостоятельной работы
			Лекции	Лабораторные	Практические занятия		
Семестр 2		135					
Тема 1.	Введение в язык программирования высокого	16	1	1		14	
Тема 2.	Компиляция и выполнение программы (ОПК-1, ОПК-	21	1	1		19	
Тема 3.	Управляющие операторы (условия, циклы) (ОПК-1,	18	1	1		16	
Тема 4.	Массивы (ОПК-1, ОПК-7)	18	1	1		16	
Тема 5.	Списки, строки и файлы (ОПК-1, ОПК-8)	17		1		16	

Тема 6.	Основы тестирования (ОПК-1, ОПК-7)	15		1		14	
Тема 7.	Рекурсия (ОПК-1, ОПК-7)	15		1		14	
Тема 8.	Поиск и сортировка (ОПК-1, ОПК-7)	15		1		14	

6. ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ШКАЛЫ ОЦЕНИВАНИЯ

Раздел/Тема	Вид оценочного средства	Описание оценочного средства	Критерию оценивания
Текущий контроль (Приложение 4)			
Темы 1-3	Тест (приложение 4)	Тест состоит из 31 вопроса	10 баллов
Темы 4-6	Контрольная работа (приложение 4)	Задания на программирование	10 баллов
Темы 7-8	Контрольная работа (приложение 4)	Задания на программирование	10 баллов
Промежуточная аттестация (Приложение 5)			
2 семестр (Эк)	Экзаменационный билет (приложение 5)	2 теоретических вопроса, 1 практическое задание	100 баллов

ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Показатель оценки освоения ОПОП формируется на основе объединения текущего контроля и промежуточной аттестации обучающегося.

Показатель рейтинга по каждой дисциплине выражается в процентах, который показывает уровень подготовки студента.

Текущий контроль. Используется 100-балльная система оценивания. Оценка работы студента в течение семестра осуществляется преподавателем в соответствии с разработанной им системой оценки учебных достижений в процессе обучения по данной дисциплине.

В рабочих программах дисциплин и практик закреплены виды текущего контроля, планируемые результаты контрольных мероприятий и критерии оценки учебных достижений.

В течение семестра преподавателем проводится не менее 3-х контрольных мероприятий, по оценке деятельности студента. Если посещения занятий по дисциплине включены в рейтинг, то данный показатель составляет не более 20% от максимального количества баллов по дисциплине.

Промежуточная аттестация. Используется 5-балльная система оценивания. Оценка работы студента по окончании дисциплины (части дисциплины) осуществляется преподавателем в соответствии с разработанной им системой оценки достижений студента в процессе обучения по данной дисциплине. Промежуточная аттестация также проводится по окончании формирования компетенций.

Порядок перевода рейтинга, предусмотренных системой оценивания, по дисциплине, в пятибалльную систему.

Высокий уровень – 100% - 70% - отлично, хорошо.

Средний уровень – 69% - 50% - удовлетворительно.

Показатель оценки	По 5-балльной системе	Характеристика показателя
100% - 85%	отлично	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на высоком уровне
84% - 70%	хорошо	обладают теоретическими знаниями в полном объеме, понимают, самостоятельно умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Могут быть допущены недочеты, исправленные студентом самостоятельно в процессе работы (ответаи т.д.)
69% - 50%	удовлетворительно	обладают общими теоретическими знаниями, умеют применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов на среднем уровне. Допускаются ошибки, которые студент затрудняется исправить самостоятельно.
49 % и менее	неудовлетворительно	обладают не полным объемом общих теоретическими знаниями, не умеют самостоятельно применять, исследовать, идентифицировать, анализировать, систематизировать, распределять по категориям, рассчитать показатели, классифицировать, разрабатывать модели, алгоритмизировать, управлять, организовать, планировать процессы исследования, осуществлять оценку результатов. Не сформированы умения и навыки для
100% - 50%	зачтено	характеристика показателя соответствует «отлично», «хорошо», «удовлетворительно»
49 % и менее	не зачтено	характеристика показателя соответствует «неудовлетворительно»

7. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

7.1. Содержание лекций

<p>Тема 1. Введение в язык программирования высокого уровня (ОПК-1, ОПК-7) Строки. Арифметические операции и var. Методы переменные.</p>
<p>Тема 2. Компиляция и выполнение программы (ОПК-1, ОПК-7) Стилистические ошибки. Рефакторинг кода. Константы и enum'ы. Выделение методов</p>
<p>Тема 3. Управляющие операторы (условия, циклы) (ОПК-1, ОПК-7) Цикл While. Цикл For. Сравнение циклов.</p>
<p>Тема 4. Массивы (ОПК-1, ОПК-7) Типы ссылки и типы значения. Передача массива в метод. Многомерные массивы. Массивы массивов.</p>
<p>Тема 5. Списки, строки и файлы (ОПК-1, ОПК-8) Специальные символы и форматированный вывод. Файлы и каталоги. Кодировка и работа с файлами.</p>
<p>Тема 6. Основы тестирования (ОПК-1, ОПК-7) Покрывание тестами. Функциональное тестирование.</p>
<p>Тема 7. Рекурсия (ОПК-1, ОПК-7) Рекурсия. Дерево рекурсии. Принцип "разделяй и властвуй". Перебор подмножеств. Перестановки. Размещения.</p>
<p>Тема 8. Поиск и сортировка (ОПК-1, ОПК-7) Сортировка пузырьком. Сортировка слиянием. Быстрая сортировка. Сравнение алгоритмов сортировки.</p>

7.2 Содержание практических занятий и лабораторных работ

<p>Тема 2. Компиляция и выполнение программы (ОПК-1, ОПК-7) Ошибки на этапе компиляции. Ошибки на этапе выполнения. Отладка.</p>
<p>Тема 3. Управляющие операторы (условия, циклы) (ОПК-1, ОПК-7) Операторы сравнения и логический тип данных. Сравнение чисел с плавающей точкой. Полные исокращенные операции. Операторы If и Else. Оператор Switch - Case.</p>
<p>Тема 4. Массивы (ОПК-1, ОПК-7) Массивы и цикл foreach. Короткая форма записи.</p>
<p>Тема 5. Списки, строки и файлы (ОПК-1, ОПК-8) Списки. Словари. Сравнение строк и массивов. Класс StringBuilder.</p>

<p>Тема 6. Основы тестирования (ОПК-1, ОПК-7)</p> <p>Введение в тестирование. Модульное тестирование.</p>
<p>Тема 7. Рекурсия (ОПК-1, ОПК-7)</p> <p>Сложность алгоритмов. Масштаб роста функций. O-символика. Оценка сложности. Сложность алгоритмов с числами. Классы сложности.</p>
<p>Тема 8. Поиск и сортировка (ОПК-1, ОПК-7)</p> <p>Линейный и бинарный поиск. Анализ линейного и бинарного поиска. Сравнение производительности. Создание графиков.</p>

7.3. Содержание самостоятельной работы

<p>Тема 2. Компиляция и выполнение программы (ОПК-1, ОПК-7)</p> <p>Решение задач "Angry Birds", "Бильярд", "Проценты".</p>
<p>Тема 3. Управляющие операторы (условия, циклы) (ОПК-1, ОПК-7)</p> <p>Решение задач "Рубли", "Два прямоугольника", "Расстояние до отрезка".</p> <p>Работа над проектом по построению путей выхода из лабиринта.</p>
<p>Тема 4. Массивы (ОПК-1, ОПК-7)</p> <p>Работа над проектом по построению гистограмм и тепловой карты.</p>
<p>Тема 5. Списки, строки и файлы (ОПК-1, ОПК-8)</p> <p>Работа над проектом по анализу текстов.</p>
<p>Тема 6. Основы тестирования (ОПК-1, ОПК-7)</p> <p>Работа над проектом по созданию парсера полей. Решение задач "Оттенки серого", "Медианный фильтр", "Пороговый фильтр", "Фильтр Собеля".</p>
<p>Тема 7. Рекурсия (ОПК-1, ОПК-7)</p> <p>Решение задач "Перебор паролей 2", "Хождение по чекпоинтам".</p>
<p>Тема 8. Поиск и сортировка (ОПК-1, ОПК-7)</p> <p>Решение задач "Левая граница", "Правая граница", "Автодополнение".</p> <p>Решение задач "Манипулятор", "Визуализация", "Поиск угла", "Решение манипулятора".</p>

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену
Приложение 1.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к зачету/экзамену
Приложение 2.

7.3.3. Перечень курсовых работ
Не предусмотрены.

7.4. Электронное портфолио обучающегося
размещаются контрольные работы

7.5. Методические рекомендации по выполнению контрольной работы
Приложение 6

7.6 Методические рекомендации по выполнению курсовой работы
Не предусмотрены.

8. ОСОБЕННОСТИ ОРГАНИЗАЦИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

По заявлению студента

В целях доступности освоения программы для лиц с ограниченными возможностями здоровья при необходимости кафедра обеспечивает следующие условия:

- особый порядок освоения дисциплины, с учетом состояния их здоровья;
- электронные образовательные ресурсы по дисциплине в формах, адаптированных к ограничениям их здоровья;
- изучение дисциплины по индивидуальному учебному плану (вне зависимости от формы обучения);
- электронное обучение и дистанционные образовательные технологии, которые предусматривают возможности приема-передачи информации в доступных для них формах.
- доступ (удаленный доступ), к современным профессиональным базам данных и информационным справочным системам, состав которых определен РПД.

9. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Сайт библиотеки УрГЭУ

<http://lib.usue.ru/>

Основная литература:

2. Корнеев В.И., Гагарина Л.Г., Корнеева М.В. Программирование графики на C++. Теория и примеры [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2024. - 517 – Режим доступа: <https://znanium.com/catalog/product/2111934>

3. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 248 – Режим доступа: <https://urait.ru/bcode/536903>

4. Швецкий М. В., Демидов М. В., Голанова А. В., Кудрявцева И. Программирование: математическая логика [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 675 – Режим доступа: <https://urait.ru/bcode/542164>

5. Кудрина Е. В., Огнева М. В. Основы алгоритмизации и программирования на языке С# [Электронный ресурс]: учебное пособие для вузов. - Москва: Юрайт, 2024. - 322 – Режим доступа: <https://urait.ru/bcode/541687>

6. Трофимов В. В., Павловская Т. А. Алгоритмизация и программирование [Электронный ресурс]: учебник для вузов. - Москва: Юрайт, 2025. - 108 – Режим доступа: <https://urait.ru/bcode/562040>

7. Палий И. А. Линейное программирование [Электронный ресурс]: учебник для вузов. - Москва: Юрайт, 2025. - 175 – Режим доступа: <https://urait.ru/bcode/563472>

Дополнительная литература:

2. Гуриков С. Р. Введение в программирование на языке Visual C# [Электронный ресурс]: Учебное пособие. - Москва: Издательство "ФОРУМ", 2020. - 447 – Режим доступа: <https://znanium.com/catalog/product/1092167>

3. Немцова Т.И., Голова С.Ю. Программирование на языке высокого уровня. Программирование на языке С++ [Электронный ресурс]: Учебное пособие. - Москва: Издательский Дом "ФОРУМ", 2021. - 512 – Режим доступа: <https://znanium.com/catalog/product/1172261>

4. Кислицын Е. В., Шишков Е. И. Разработка приложений на языке Java [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2017. - 86 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/17/p488938.pdf>

5. Плещев В. В., Шишков Е. И. Основы программирования на языках С++ и С# с примерами и упражнениями [Электронный ресурс]: учебное пособие. - Екатеринбург: [Издательство УрГЭУ], 2018. - 286 – Режим доступа: <http://lib.usue.ru/resource/limit/ump/18/p490708.pdf>

10. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ВКЛЮЧАЯ ПЕРЕЧЕНЬ ЛИЦЕНЗИОННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ, ОНЛАЙН КУРСОВ, ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Перечень лицензионного программного обеспечения:

Astra Linux Common Edition. Договор №0417-ПО/2019 от 08.05.2019, Акт №Sk000343 от 24.05.2019 и Контракт № 35-У/2018 от 13.06.2018, Акт № УТ213 от 17.12.2018. Срок действия лицензии - без ограничения срока.

Microsoft Visual Studio Community. Лицензия для образовательных учреждений. Срок действия лицензии - без ограничения срока.

Язык программирования Python. Python Software Foundation License (PSFL). Срок действия лицензии - без ограничения срока.

Microsoft Windows 10 .Договор № 52/223-ПО/2020 от 13.04.2020, Акт № Tr000523459 от 14.10.2020. Срок действия лицензии - Без ограничения срока.

Microsoft Office 2016. Договор № 52/223-ПО/2020 от 13.04.2020, Акт № Tr000523459 от 14.10.2020 Срок действия лицензии - Без ограничения срока.

МойОфис стандартный. Соглашение № СК-281 от 7 июня 2017. Дата заключения - 07.06.2017. Срок действия лицензии - без ограничения срока.

Перечень информационных справочных систем, ресурсов информационно-телекоммуникационной сети «Интернет»:

Справочно-правовая система Гарант. Договор № 58419 от 22 декабря 2015. Срок действия лицензии -без ограничения срока

Справочно-правовая система Консультант +. Договор № 143/223-У/2025 от 02.12.2025 Срок действия лицензии до 31.12.2026

Программирование и разработка веб-приложений

<https://openedu.ru/course/ITMOUniversity/PWADEV/>

Программирование на C#

<https://openedu.ru/course/urfu/CSHARP/>

11. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Реализация учебной дисциплины осуществляется с использованием материально-технической базы УрГЭУ, обеспечивающей проведение всех видов учебных занятий и научно-исследовательской и самостоятельной работы обучающихся:

Специальные помещения представляют собой учебные аудитории для проведения всех видов занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду УрГЭУ.

Все помещения укомплектованы специализированной мебелью и оснащены мультимедийным оборудованием спецоборудованием (информационно-телекоммуникационным, иным компьютерным), доступом к информационно-поисковым, справочно-правовым системам, электронным библиотечным системам, базам данных действующего законодательства, иным информационным ресурсам служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа презентации и другие учебно-наглядные пособия, обеспечивающие тематические иллюстрации.

7.3.1. Примерные вопросы для самостоятельной подготовки к зачету/экзамену

К экзамену

1. Понятие класса. Объявление класса.
2. Статические и динамические поля.
3. Методы. Статические и динамические методы.
4. Методы расширения.
5. Статические классы.
6. Наследование: понятие, иерархия.
7. Класс Array.
8. Понятие интерфейса.
9. Интерфейс IComparable. Реализация интерфейса.
10. Интерфейс IComparer. Реализация интерфейса.
11. Полиморфизм. Перегрузка и переопределение методов. Виртуальные методы. Ключевые слова virtual и override.
12. Ключевое слово private. Свойства (get, set).
13. Конструкторы.
14. Поля readonly и статические конструкторы.
15. Структуры. Инициализация полей структуры.
16. Ключевые слова ref и out.
17. Структуры данных: стек, очередь, дек.
18. Реализация очереди на связных списках.
19. Понятие «апкаст» и «даункаст».
20. Дженерик-классы.
21. Использование стека для анализа скобочных выражений.
22. Использование очереди для нахождения скользящего среднего.
23. Использование дженерик-методов для создания универсальной сортировки.
24. Дженерик-классы Tuple и Nullable.
25. Интерфейсы IEnumerable и IEnumerator. Реализация интерфейсов.
26. Реализация метода GetEnumerator с помощью конструкции yield return.
27. Списки. Индексация в списках.
28. Реализация методов Contains и Equals.
29. Перегрузка операторов.
30. Хеш-функции. Виртуальный метод GetHashCode.
31. Алгоритм нахождения подстроки в строке с помощью хеш-функции.
32. Делегаты: понятие и свойства.
33. Дженерик-делегаты.
34. Конструкции Func и Action. Анонимные делегаты.
35. Лямбда-выражения.
36. Язык LINQ. Реализация метода Where.
37. Язык LINQ. Реализация методов Select и ToList.
38. Язык LINQ. Методы фильтрации и преобразования.
39. Язык LINQ. Функции агрегирования и группировки.
40. События. Определение и вызов событий. Обработчики событий.
41. Использование делегатов для обработки событий.
42. Мультикаст-делегаты.
43. Назначение protected virtual OnTick, sender и EventArgs.
44. WindowsForm и WPF
45. Элементы управления.
46. Реализация резинового дизайна. Менеджеры компоновки.

47. Рисование.
48. Таймеры и анимация.
49. Паттерн MVC.
50. Потoki, домены и процессы. Класс Thread.
51. Методы BeginInvoke и EndInvoke для управления потоками.
52. Класс Parallel для асинхронного выполнения функций. Общие ресурсы и lock.
53. Блокирование потока с графическим интерфейсом. Асинхронные операции в приложениях с GUI.
54. Конструкции async и await. Класс BackgroundWorker.
55. Рефлексия типов.

7.3.2. Практические задания по дисциплине для самостоятельной подготовки к экзамену

Примерные практические задания к экзамену (ОПК-1, ОПК-7)

1. Разработать Windows-приложение, строящее диаграмму или график по заданным в файле данным. Создать меню с командами Input Data, Choose, Line, Bar, Quit. Команды Line и Bar недоступны. При выборе команды Input Data из файла читаются исходные данные. По команде Choose открывается диалоговое окно, содержащее: список для выбора цвета графика типа TListBox; группу из двух переключателей (Line, Bar) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода цвета и выбора режима: график или диаграмма. После указания параметров становится доступной соответствующая команда меню. По команде Line или Bar в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна.
2. Описать класс для работы с одномерным массивом целых чисел. Обеспечить следующие возможности: 1. задание произвольных целых границ; 2. обращение к отдельному элементу массива с контролем выхода за пределы массива; 3. Выполнение операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов; 4. Выполнение операций умножения и деления всех элементов массива на скаляр; 5. Вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую все разработанные элементы класса.
3. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение. Написать программу, демонстрирующую все разработанные элементы класса.
4. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую все разработанные элементы класса.
5. Составить описание класса для вектора, заданного координатами его концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую все разработанные элементы класса.
6. Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую все разработанные элементы класса.
7. Составить описание класса для представления времени. Предусмотреть возможности установки времени и изменения его отдельных полей (час, минута, секунда) с проверкой допустимости вводимых значений. В случае недопустимых значений полей выбрасываются

- ся исключения. Создать метод изменения времени на заданное количество часов, минут и секунд. Написать программу, демонстрирующую все разработанные элементы класса.
8. Описать класс, представляющий треугольник. Предусмотреть методы для создания объектов, вычисления площади, периметра и точки пересечения медиан. Описать свойства для получения состояния объекта. При невозможности построения треугольника выбрасывается исключение. Написать программу, демонстрирующую все разработанные элементы класса.
 9. Описать класс, представляющий круг. Предусмотреть методы для создания объектов, вычисления площади круга, длины окружности и проверки попадания заданной точки внутрь круга. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.
 10. Описать класс дробей — рациональных чисел, являющихся отношением двух целых чисел. Предусмотреть методы сложения, вычитания, умножения и деления дробей. Написать программу, демонстрирующую все разработанные элементы класса.
 11. Описать класс «комната», содержащий сведения о метраже, высоте потолков количестве окон. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать методы вычисления площади и объема комнаты и свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.
 12. Описать класс, представляющий квадратное уравнение вида $ax^2 + vx + c = 0$. Описать метод, вычисляющий решение этого уравнения и выбрасывающий исключение в случае отсутствия корней. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы класса.
 13. Описать класс «процессор», содержащий сведения о марке, тактовой частоте, объеме кэша и стоимости. Предусмотреть инициализацию с проверкой допустимости значений полей. В случае недопустимых значений полей выбрасываются исключения. Описать свойства для получения состояния объекта. Описать класс «материнская плата», включающий класс «процессор» и объем установленной оперативной памяти. Предусмотреть инициализацию с проверкой допустимости значений поля объема памяти. В случае недопустимых значений поля выбрасывается исключение. Описать свойства для получения состояния объекта. Написать программу, демонстрирующую все разработанные элементы классов.
 14. Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по фамилии и доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.
 15. Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, имени, дате рождения), добавления и удаления записей, сортировки по разным полям, доступа к записи по номеру. Написать программу, демонстрирующую все разработанные элементы класса.
 16. Описать класс «колода карт», включающий закрытый массив элементов класса «карта». В карте хранятся масть и номер. Обеспечить возможность вывода карты по номеру, вывода всех карт, перемешивания колоды и выдачи всех карт из колоды поодиночке и по 6 штук в

случайном порядке. Написать программу, демонстрирующую все разработанные элементы классов.

17. Описать класс «поезд», содержащий следующие закрытые поля: название пункта назначения; номер поезда (может содержать буквы и цифры); время отправления. Предусмотреть свойства для получения состояния объекта. Описать класс «вокзал», содержащий закрытый массив поездов. Обеспечить следующие возможности: вывод информации о поезде по номеру с помощью индекса; вывод информации о поездах, отправляющихся после введенного с клавиатуры времени; перегруженную операцию сравнения, выполняющую сравнение времени отправления двух поездов; вывод информации о поездах, отправляющихся в заданный пункт назначения. Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.
18. Описать класс «товар», содержащий следующие закрытые поля: название товара; название магазина, в котором продается товар; стоимость товара в рублях. Предусмотреть свойства для получения состояния объекта. Описать класс «склад», содержащий закрытый массив товаров. Обеспечить следующие возможности: вывод информации о товаре по номеру с помощью индекса; вывод на экран информации о товаре, название которого введено с клавиатуры; если таких товаров нет, выдать соответствующее сообщение; сортировку товаров по названию магазина, по наименованию и по цене; перегруженную операцию сложения товаров, выполняющую сложение их цен. Написать программу, демонстрирующую все разработанные элементы классов.
19. Описать класс «самолет», содержащий следующие закрытые поля: название пункта назначения; шестизначный номер рейса; время отправления. Предусмотреть свойства для получения состояния объекта. Описать класс «аэропорт», содержащий закрытый массив самолетов. Обеспечить следующие возможности: вывод информации о самолете по номеру рейса с помощью индекса; вывод информации о самолетах, отправляющихся в течение часа после введенного с клавиатуры времени; вывод информации о самолетах, отправляющихся в заданный пункт назначения; перегруженную операцию сравнения, выполняющую сравнение времени отправления двух самолетов. Информация должна быть отсортирована по времени отправления. Написать программу, демонстрирующую все разработанные элементы классов.
20. Описать класс «запись», содержащий следующие закрытые поля: фамилия, имя; номер телефона; дата рождения (массив из трех чисел). Предусмотреть свойства для получения состояния объекта. Описать класс «записная книжка», содержащий закрытый массив записей. Обеспечить следующие возможности: вывод на экран информации о человеке, номер телефона которого введен с клавиатуры; если такого нет, выдать на дисплей соответствующее сообщение; поиск людей, день рождения которых сегодня или в заданный день; поиск людей, день рождения которых будет на следующей неделе; поиск людей, номер телефона которых начинается на три заданных цифры. Написать программу, демонстрирующую все разработанные элементы классов.
21. Описать базовый класс «Строка». Обязательные поля класса: поле для хранения символов строки; значение типа `word` для хранения длины строки в байтах. Реализовать обязательные методы следующего назначения: конструктор без параметров; конструктор, принимающий в качестве параметра строковый литерал; конструктор, принимающий в качестве параметра символ; метод получения длины строки; метод очистки строки. Описать производный от «Строка» класс «Комплексное число». Строки данного класса состоят из двух

полей, разделенных символом i . Первое поле задает значение действительной части числа, второе – значение мнимой. Каждое из полей может содержать только символы десятичных цифр и символы $+$ и $-$, задающие знак числа. Символы $+$ и $-$ могут находиться только в первой позиции числа, причем символ $+$ может отсутствовать, в этом случае число считается положительным. Для класса «Комплексное_число» определить следующие методы: проверка на равенство; сложение чисел; умножение чисел.

22. Создать меню с командами Begin, Work, About. При выборе команды About открывается окно с информацией о разработчике. При выборе команды Begin открывается диалоговое окно, содержащее: два поля ввода; группу из двух переключателей (First letter, All letter) типа RadioButton; кнопку типа Button. Обеспечить возможность ввода предложения и выбора режима его преобразования: либо начинать с прописной буквы каждое слово (First letter), либо перевести все буквы в верхний регистр (All letter). При выборе команды Work открывается диалоговое окно с результатами преобразования.
23. Написать Windows-приложение – графическую иллюстрацию сортировки методом выбора. Создать меню с командами File, Animate, About, Exit. Команда Animate недоступна. Команда Exit завершает работу приложения. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда File) использовать объект класса OpenFileDialog. Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа). После чтения данных становится доступной команда Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.

7.5. Методические рекомендации по выполнению контрольной работы

**Приложение 6
к рабочей программе**

Федеральное государственное бюджетное образовательное учреждение высшего образования
УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ
КОНТРОЛЬНОЙ РАБОТЫ
по дисциплине
Программирование**

Методические рекомендации по выполнению контрольной работы для студентов заочной формы обучения

Темы контрольной работы (по выбору студента)

Разработать программу на любом языке программирования высокого уровня. Задание выбирать исходя из своего номера по списку (зачетной книжки).

1. 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.

2. Разработать программу для расчета площади фигуры, ограниченной осью ОХ, 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 , t задаются пользователем.

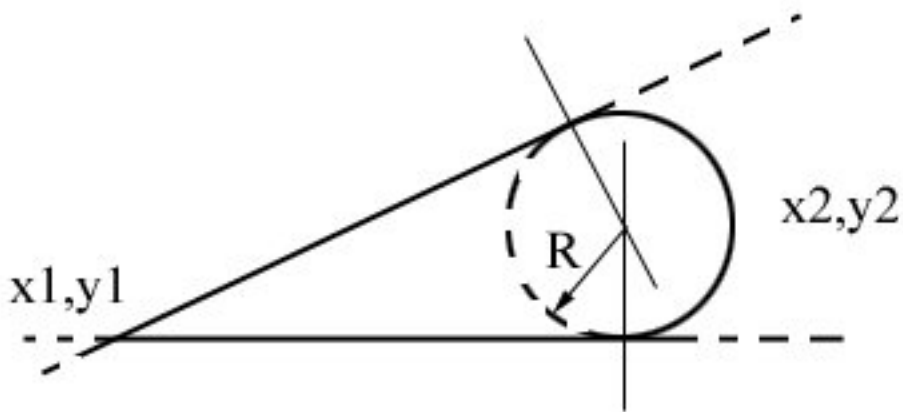
3. Разработать программу расчета размера выплат для погашения кредита в каждый месяц. Сумма выплат в месяц составляет сумму доли от суммы кредита (она неизменна в течение всего срока погашения кредита) и процентов на остаток от суммы кредита. Построить график выплат в течение всего срока. Выплаты и начисление процентов на оставшуюся сумму производятся ежемесячно. Исходные данные: размер кредита, срок выплат (от 1 до 5 лет), процентная ставка.

4. 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.

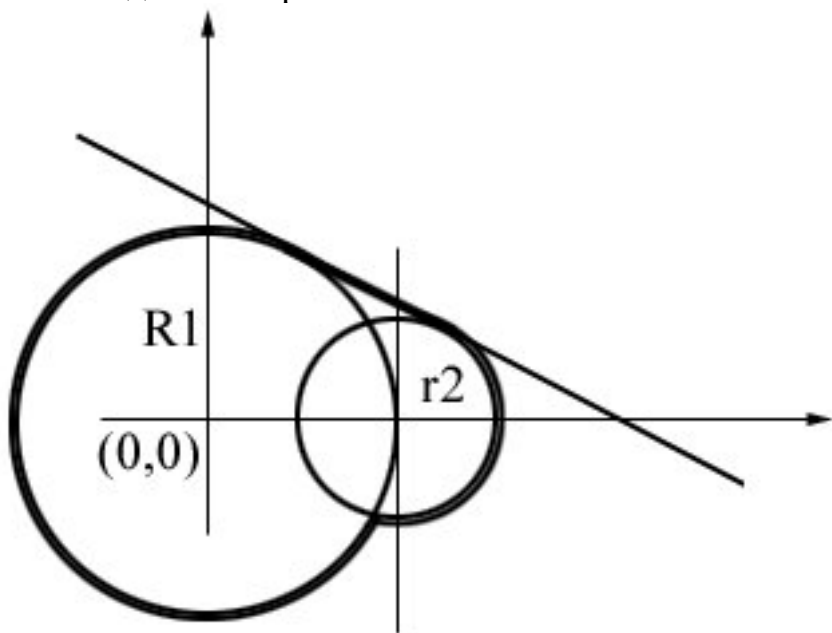
5. Разработать программу для решения квадратного уравнения и построения графика функции с осями координат. Коэффициенты уравнения считывать из файла, выбранного пользователем.

6. Разработать программу для определения координат точки пересечения двух прямых и построения их графиков с осями координат. Коэффициенты уравнений прямых считывать из файла, указанного пользователем.

7. 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.

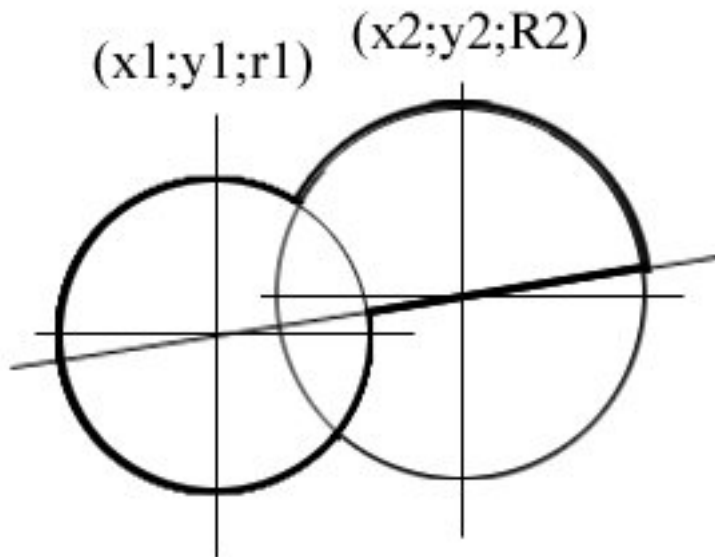


8. Разработать программу для построения изображения, приведенного на рисунке. Исходные данные (R_1, r_2) должны быть считаны из выбранного пользователем файла. Определить коэффициенты уравнения касательной к окружностям. Вывести их в диалоге приложения.

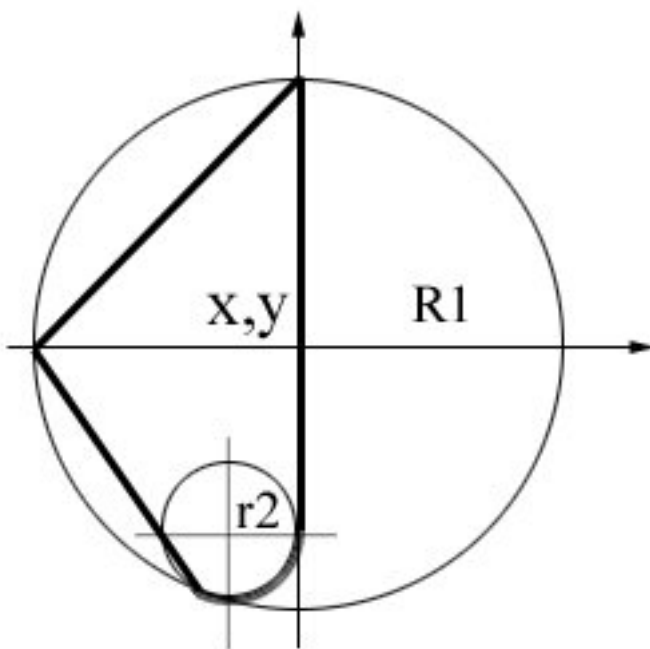


9. Разработать программу для построения графика функции $y=2*\sin(x)*e^{x/5}$. Считать из файла, указанного пользователем значения абсциссы X , рассчитать для них значения ординаты Y , записать результат вычисления в новый файл в формате $X_1 - Y_1$
 $X_2 - Y_2...$

10. 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)

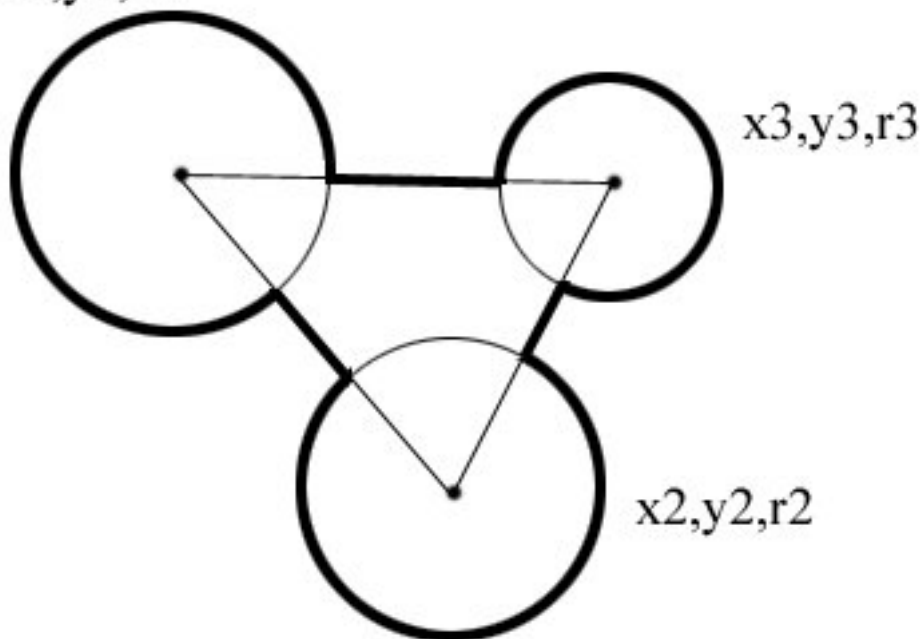


11. Разработать программу для построения изображения, приведенного на рисунке. Принять $x=0$, $y=0$. Исходные данные ($R1$, $r2$) должны быть считаны из выбранного пользователем файла. Программа должна осуществлять проверку правильности исходных данных (обязательное касание окружностей).



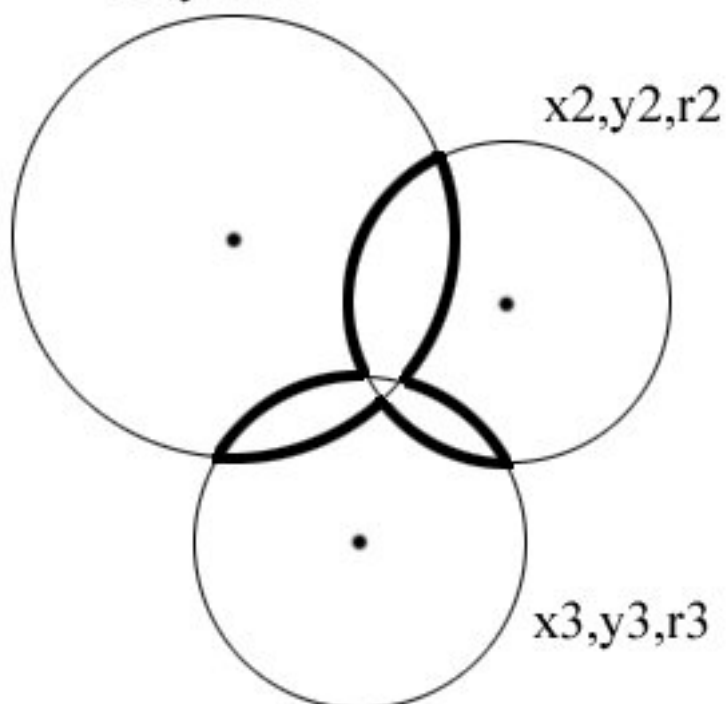
12. Разработать программу для расчета площади фигуры, изображенной на рисунке сплошной утолщенной линией. Исходные данные ($x1$, $y1$, $R1$, $x2$, $y2$, $R2$, $x3$, $y3$, $R3$) должны быть введены пользователем. Построить изображение полученной фигуры. Осуществлять проверку исходных данных на соответствие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

x_1, y_1, R_1



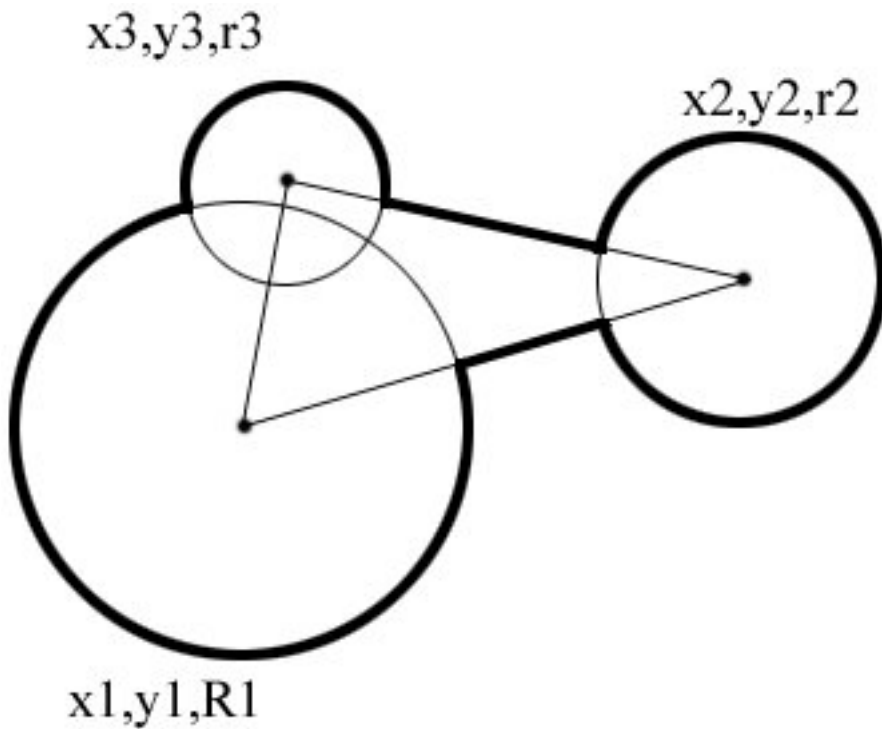
13. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные ($x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3$) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (все окружности пересекаются).

x_1, y_1, R_1

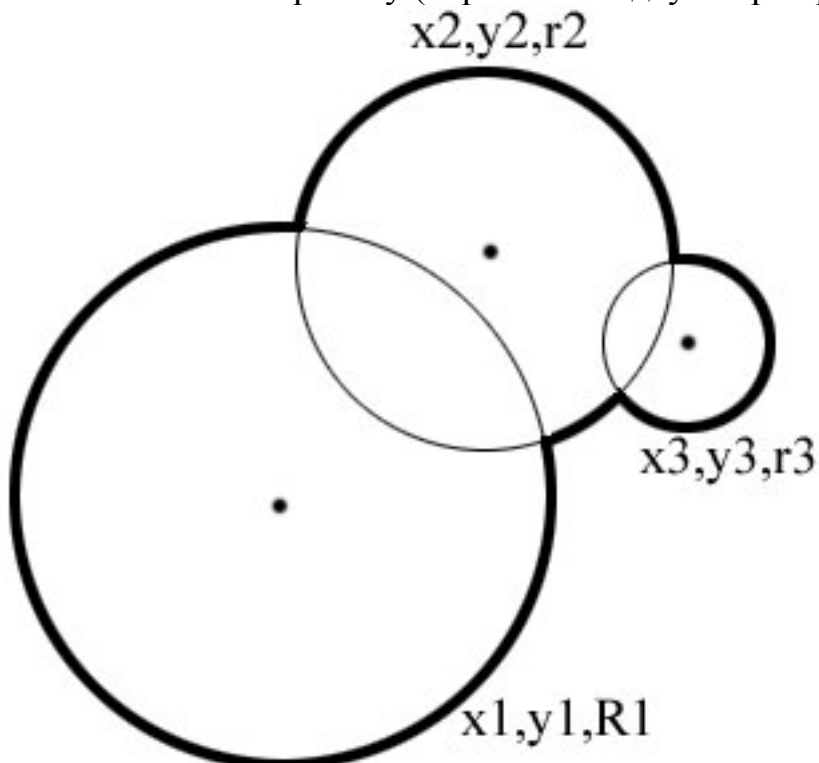


14. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные ($x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3$) должны быть считаны

из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (только две окружности пересекаются).

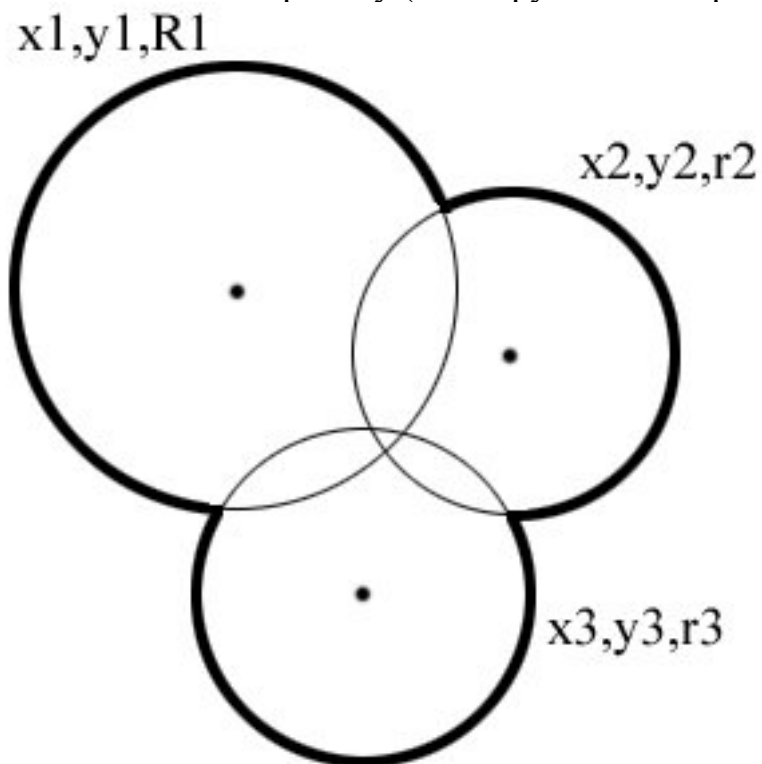


15. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные ($x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3$) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (пересечение двух пар окружностей).

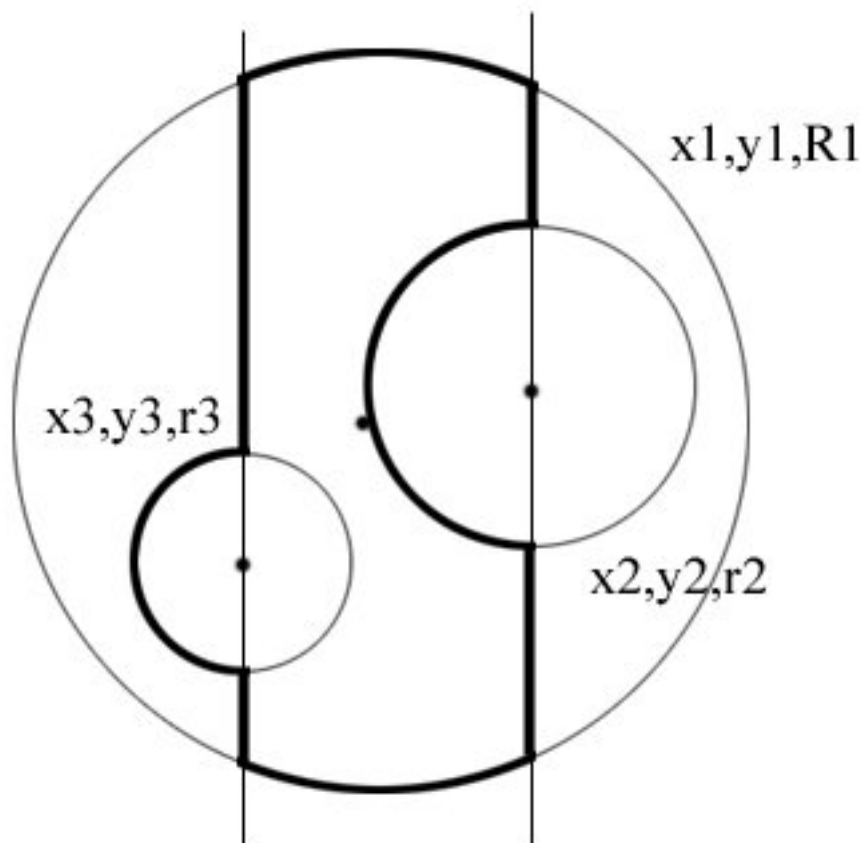


16. Разработать программу для игры в крестики-нолики для размещения 5 одинаковых значков в ряд на поле размером 10x10. Рассмотреть случай двух игроков.

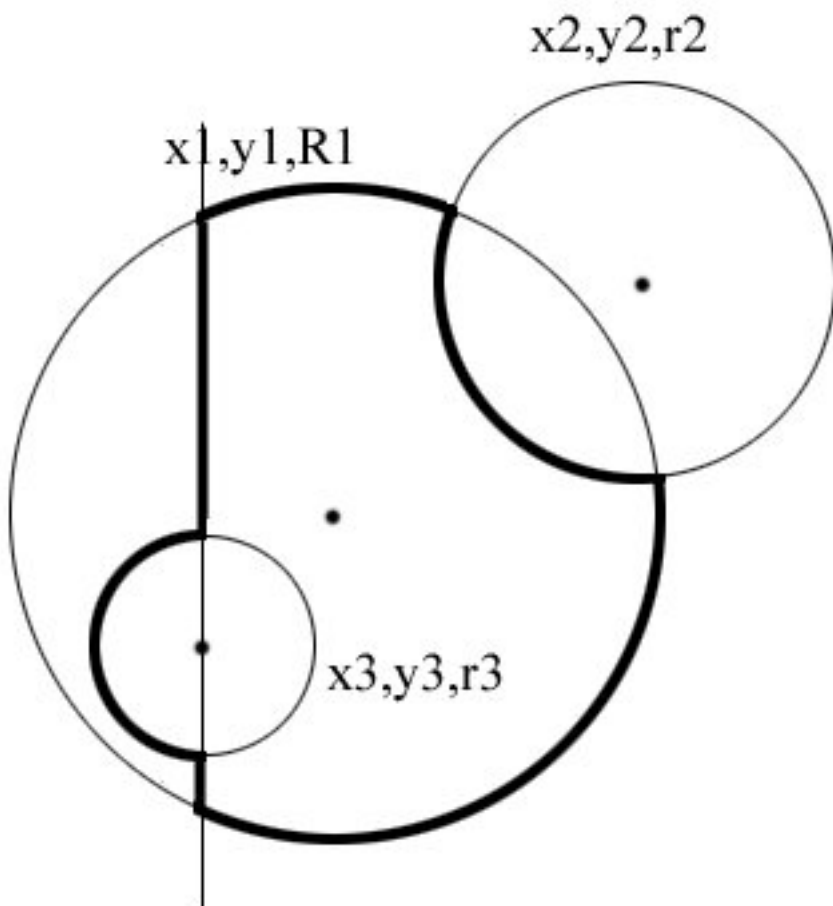
17. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (все окружности пересекаются).



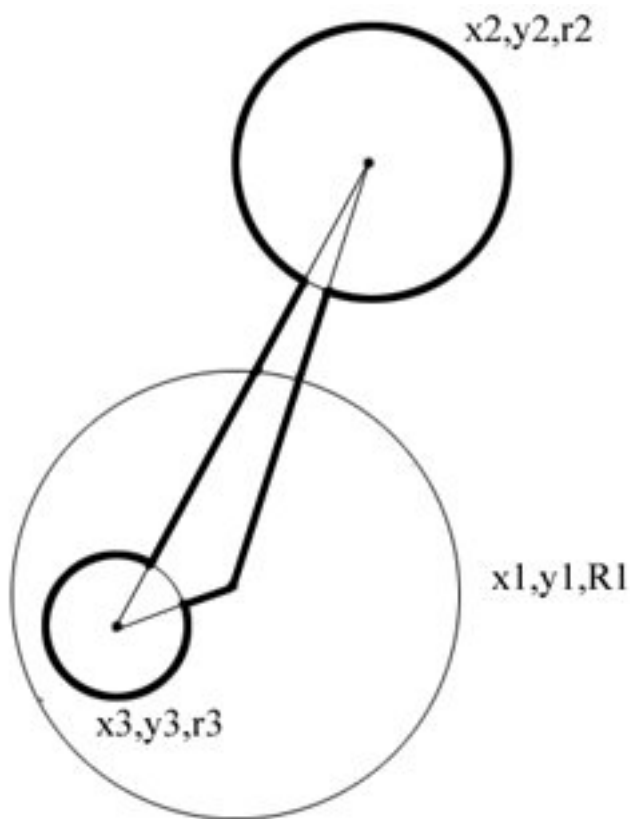
18. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (две окружности находятся внутри третьей).



19. 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 (две окружности пересекаются, третья находится внутри первой).



20. Разработать программу для расчета площади фигуры, изображенной на рисунке сплошной утолщенной линией. Исходные данные $(x_1, y_1, R_1, x_2, y_2, R_2, x_3, y_3, R_3)$ должны быть введены пользователем. Построить изображение полученной фигуры. Осуществить проверку исходных данных на соответствие варианту (окружности не пересекаются, третья окружность находится внутри первой).



21. 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.

22. 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.

23. 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.

24. Разработать программу, осуществляющую чтение текста из файла, сортировку слов текста по алфавиту методом выбора, запись полученного текста в новый файл.

25. 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.

26. 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.

27. 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	.-.

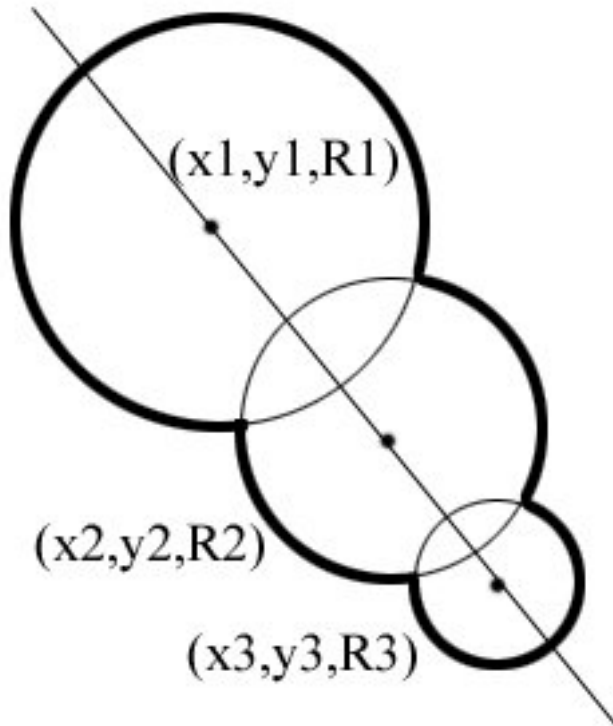
Азбука Морзе.

28. Разработать программу для работы с двумерными массивами. Ее функции:

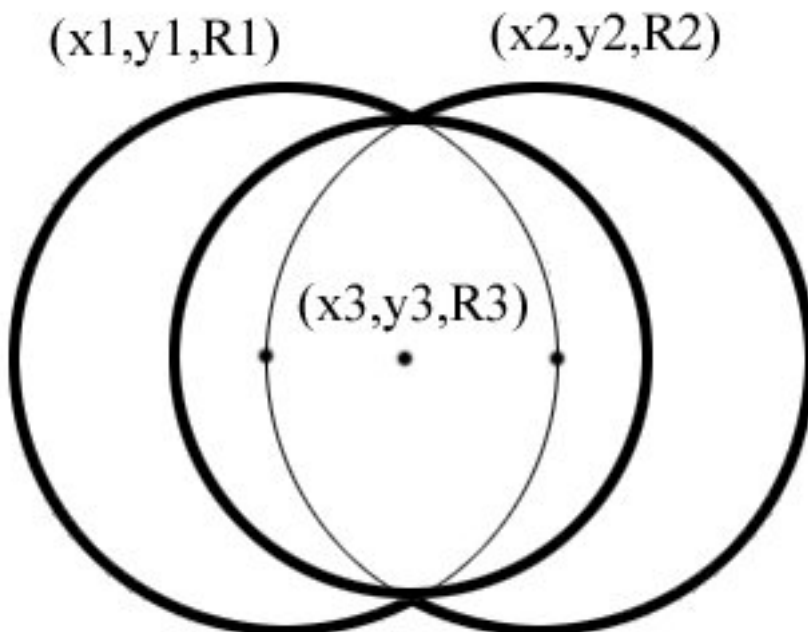
- считывание элементов двумерного массива из файла (txt), где в первой строке можно задавать размерность массива;
- нахождение индексов максимального и минимального элемента методом последовательного поиска;
- определение среднего арифметического элементов, суммы элементов по строкам и столбцам;
- вывод полученных данных на экран.

29. 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.

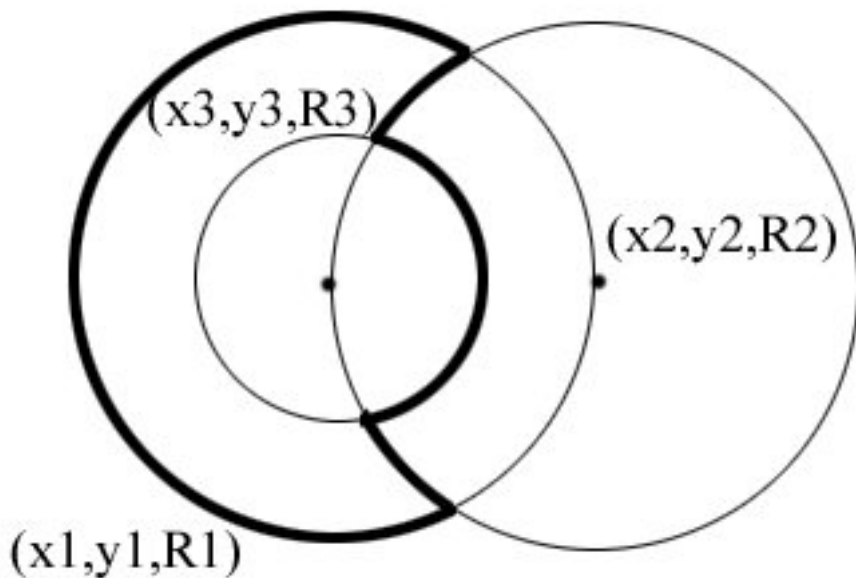
30. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту (окружности попарно пересекаются,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



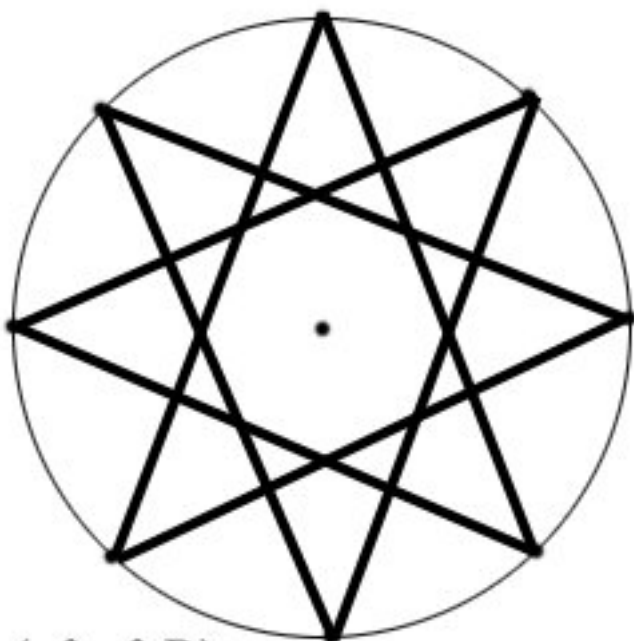
31. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x_1, y_1, R_1, x_2, y_2, r_2, x_3, y_3, r_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



32. 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.

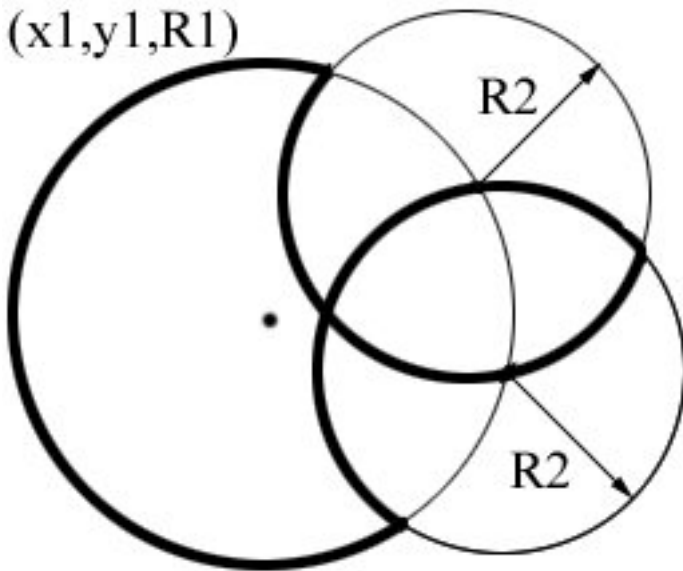


33. Разработать программу для построения изображения фигуры, приведенной на рисунке. Количество лучей может быть задано пользователем или считано из файла.

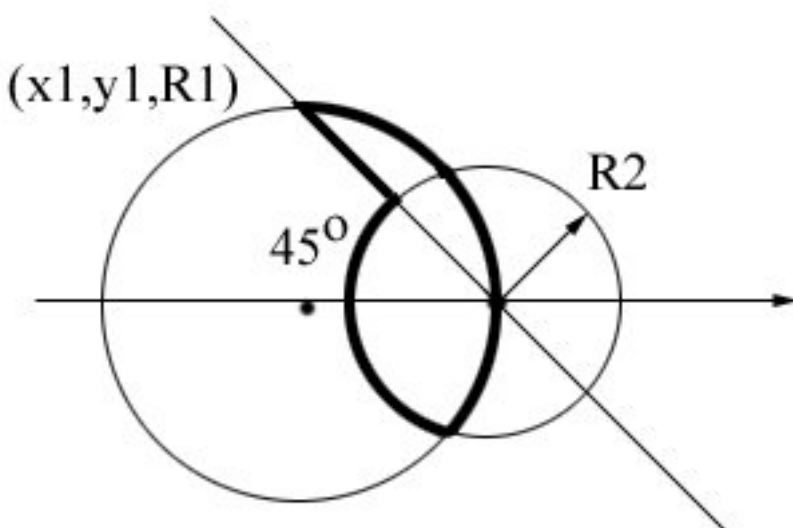


(x_0, y_0, R) ;
 $n = 1..12$ - количество лучей

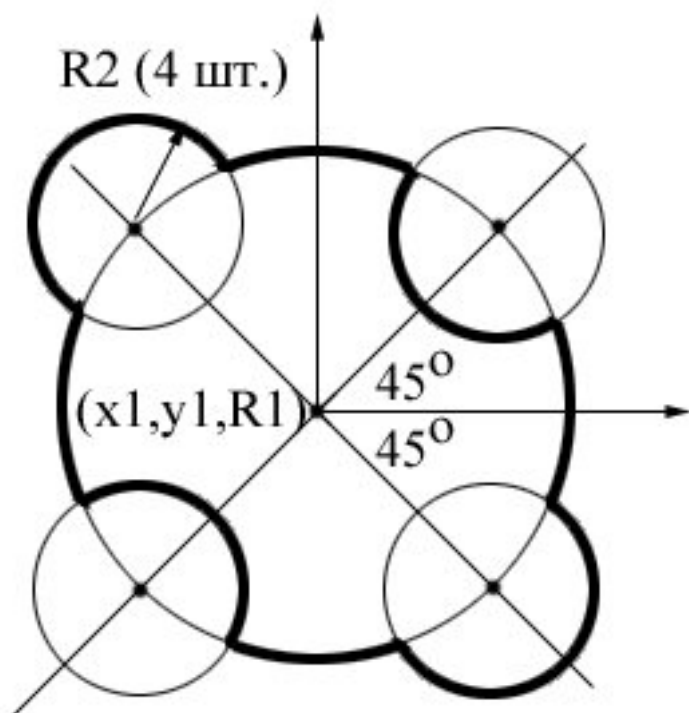
34. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные (x_1, y_1, R_1, R_2) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



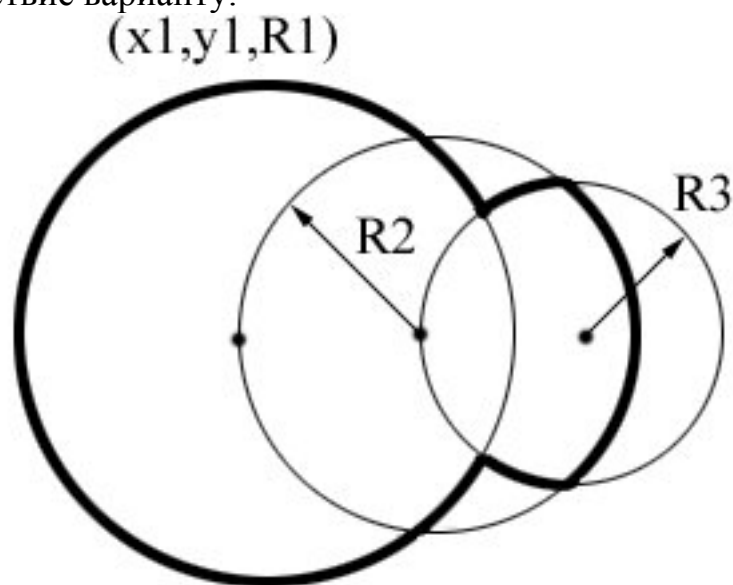
35. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные (x_1, y_1, R_1, R_2) должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



36. 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.



37. Разработать программу для построения изображения фигуры, приведенной на рисунке. Исходные данные $(x_1, y_1, R_1, R_2, R_3)$ должны быть считаны из выбранного пользователем файла. Осуществлять проверку исходных данных на соответствие варианту.



38. Разработать программу для построения графика функции с осями координат по заданным во внешнем файле точкам. В программе должен быть предусмотрен выбор файла в каталоге.

39. 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.

40. 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.

41. Разработать программу для работы с одномерным массивом. Ее функции:

- считывание элементов массива из файла (txt), где в первой строке можно задавать размерность массива;
- нахождение индексов максимального и минимального элемента методом последовательного поиска;
- определение среднего арифметического элементов, суммы элементов по строкам и столбцам;
- сортировка массива методом вставок;
- вывод полученных данных на экран.

42. 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.

43. 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.

44. Разработать программу для определения точек пересечения графика функции $y = a*x^4 + b*\sin x$ с осями координат. Также построить график этой функции в промежутке от -100 до 100 . Коэффициенты a и b задаются пользователем. Результаты вычислений записать в файл, указанный пользователем.

45. Разработать программу для построения графика изменения курса валют (доллар США и евро) и расчета величины изменения курса за последний период. Данные о курсах должны содержаться во внешнем файле. Предусмотреть функцию дополнения этого файла новыми данными.

46. Разработать программу для реализации движения объекта (закрашенного круга) по экрану. При запуске программы направление движения объекта произвольное, при достижении границы экрана направление движения меняется по законам отражения света. Предусмотреть возможность изменения пользователем скорости движения объекта.

47. Разработать программу для расчета площади выпуклого многогранника методом триангуляции (разбиения на треугольники) по формуле Герона (определение

площади треугольника по его сторонам и полупериметру). Координаты вершин многогранника должны быть считаны из указанного пользователем файла. В этом файле первым может быть число, равное количеству вершин многогранника. Построить изображение многогранника с линиями разбиения на треугольники.

48. Разработать программу для определения точек пересечения графика функции $y = a*x^4 + b*x^2 + c$ с осями координат. Также построить график этой функции в промежутке от -100 до 100 . Коэффициенты a , b , c задаются пользователем. Результаты вычислений записать в файл, указанный пользователем.

49. Разработать программу для управления базой данных «Домашняя библиотека». Реализовать функции создания в ней записи, ее редактирования, удаления и поиска записи по значению поля. Запись в базе данных содержит следующие поля: автор, название произведения, жанр, год издания.

50. Разработать программу для работы с двумя матрицами. Реализовать функции сложения, вычитания и умножения матриц. Элементы матриц задаются двумя файлами, указанными пользователями. Результат выводится на экран.

51. Разработать программу, которая подводит итоги Олимпийских игр. Программа должна получить от пользователя количество золотых, серебряных и бронзовых медалей, завоеванное каждой командой-участницей (принять количество команд равное 5). Далее по количеству медалей каждого достоинства необходимо вычислить соответствующее число очков команды. Количество очков вычисляется по следующему правилу: за золотую медаль команда получит семь очков, за серебряную — шесть, за бронзовую — пять очков. После таких расчетов список команд нужно упорядочить в соответствии с набранным количеством очков с помощью алгоритма сортировки обменом и вывести в диалоге. Построить также столбчатую диаграмму набранных очков с подписью названия каждой команды.

52. Разработать программу для построения графика розы ветров. Исходные данные считываются из указанного пользователем файла, который содержит количество дней в году для каждого из возможных направлений ветра (С, С-В, В, Ю-В, Ю, Ю-З, З, С-З). Также разработать функцию заполнения файла исходных данных случайными числами для случая невисокосного года.