

## Основные требования

Курсовая работа выполняется на языке Java и может быть настольным или web – приложением (по желанию). Приложение обязательно имеет графический интерфейс пользователя. Программа должна сохранять результаты в файл, сериализовать состояние или записывать данные в БД. Балл оценки повышается, если реализованы многопоточность, модель клиент-сервер или использована база данных для хранения данных.

## Оформление записки

1. Техническое задание.
  - Введение (описание предметной области);
  - Назначение разработки, ее актуальность;
  - Требования к программе;
  - Постановка задачи (формулирование задач, подлежащих решению).
2. Проектирование (разработка диаграмм UML: диаграммы взаимодействия, диаграммы последовательности действий, диаграммы классов).
3. Структурное описание разработки.
  - Описание структуры приложения, его модулей.
  - Описание классов: спецификации данных, методов.
4. Функциональное описание разработки
  - Описание алгоритмов и методов решения.
5. Описание пользовательского интерфейса.
6. Тестирование программы (при необходимости).
7. Руководство пользователя.
8. Руководство программиста. (Установка программы, необходимые программные и аппаратные средства).
9. Заключение
10. Список литературы.
11. Приложение (текст программы с комментариями).

## Темы курсовых работ:

1. Реализация графических часов. Проект, автоматически загружает системное время. В программе должны быть отдельные минутная, часовая и секундная стрелки, а также шкала часов. Иметь две формы представления: часы со стрелками, электронные часы с цифрами. (Балл -3)
2. Игровая программа «Тетрис». Геометрические объекты спускаются в ограниченном пространстве вниз. Путем управления уложить плотно эти фигуры в ряд. Заполненные ряды удаляются и игроку записываются баллы. Сохранять в файл результаты игр в отсортированном виде. (Балл -3)
3. Игра «Сапер». (Балл -3)  
Разработать графический интерфейс. Иметь меню и элементы управления.
4. Сетевая игра "Шашки" . (Балл -5)

Разработать графический интерфейс. Иметь меню и элементы управления. Результаты игры сохранять в файл. Иметь возможность просмотреть файл.

#### 5. Разработка программы «Аквариум» (Балл -4)

- Определите объект TFish - аквариумная рыба. Рыба имеет координаты, скорость, размер, цвет, направление движения. Методами объекта являются:

Init - устанавливает значения полей объекта и рисует рыбу на экране методом Draw.

Draw - рисует рыбу в виде уголка, с острием в точке Coord и направленного острием по ходу движения рыбы.

Look - проверяет несколько точек на линии движения рыбы. Если хоть одна из них отличается по цвету от воды, возвращается её цвет и расстояние до рыбы.

Run - перемещает рыбу в текущем направлении на расстояние, зависящее от текущей скорости рыбы. Иногда случайным образом меняет направление движения рыбы. Если рыба видит препятствие, направление движения меняется, пока препятствие не исчезнет из поля зрения рыбы.

- Определите объект Taquarium, который является местом обитания рыб. Он представляет собой область экрана, наполненную водой. Рыбы живут в аквариуме, поэтому экземпляры объекта Tfish должны быть полями объекта Taquarium.

Методы:

Init - включает графический режим, заполняет аквариум водой, скалами и рыбами.

Run - организует бесконечный цикл, в котором выполняется метод Run всех обитателей аквариума.

Done - выключает графический режим.

- Определите два объекта Trike и Tкагр, которые наследуют объект Tfish. Оба они отличаются от Tfish тем, что по разному изображают себя на экране: Trike - в виде зеленой стрелки, а Tкагр - в виде красного треугольника. Воспользуйтесь виртуальными методами. Для этого вернитесь к определению Tfish и откорректируйте его, сделав Draw пустым и виртуальным.

- Объедините карпов и щук в две стаи. Стая - это связанный список рыб в динамической памяти. Для связи добавьте в объекты Trike и Tкагр поле Next - указатель на следующую рыбу в стае. Сделайте аквариум не владельцем рыб, а двух стай и позвольте пользователю пополнять стаи, вводя рыб с клавиатуры.

- Позвольте щукам проявить свой дурной характер и поедать карпов, как только они их увидят. Здесь возникнет проблема - установить, какого именно карпа видит щука. Она решается путем просмотра всей стаи карпов и поиска того, чьи координаты близки к координатам данной щуки. Найденный карп удаляется из стаи.

#### 6. Моделирующая игра «Жизнь» (Балл -4)

Это игра создана в 1970 г., ее автор - английский математик Дж. Конвей (Conway). В этой игре партнер не нужен - в неё можно играть одному. Возникающие в процессе игры ситуации очень похожи на реальные процессы, происходящие при зарождении, развитии и гибели колонии живых организмов.

*Правила игры.* Вообразите бесконечное поле, разделенное на клетки. На каждой клетке поля живет, рождается или погибает животное. Это зависит от условий Среды, т. е. от того, сколько соседей у него на ближайших восьми клетках (четырех по сторонам и четырех по углам).

Действуют три правила существования животных:

- Каждое животное, у которого два или три соседа, живет и сохраняется до следующего поколения.

- Животное погибает, если у него более нежеле три соседа (от недостатка места), совсем нет соседей или только один сосед (от одиночества).

- Когда рядом с какой-нибудь клеткой есть три животных (соседа), то на этой

клетке рождается новое животное.

- Важно понять, что животные погибают и рождаются одновременно. Они образуют одно поколение. За один ход в игре в соответствии с упомянутыми правилами осуществляется переход от одного поколения к другому.

Дж. Конвей рекомендует следующий способ осуществления ходов (при наличии клетчатой доски и косточек двух цветов):

- 1) начать с желаемой конфигурации (колонии животных), состоящей из черных косточек;
- 2) найти все косточки, которые должны *погибнуть*, и на каждую из них одеть по одной черной косточке;
- 3) найти все свободные клетки, на которых должно *родиться* животное, и положить на них по одной белой косточке;
- 4) удалить с доски всех *погибших* животных (т. е. столбики из двух косточек), а *новорожденных* (белые косточки) заменить черными. Выполнив эти операции, т. е. после первого хода, получим второе поколение. Аналогичным образом происходит и все остальные ходы в игре. Так получают все новые поколения.

Напишите программу, моделирующую колонию Жизни. Исходными данными служит начальное расположение животных (заданное пользователем или получаемое случайно - реализовать оба случая), а в качестве результата нужно получить вид сверху в графическом режиме всех поколений колонии.

Некоторые колонии разрастаются невероятным образом при весьма скромных начальных размерах. Есть другие колонии, которые медленно перемещаются по пустыне, переходя на все новые и новые территории. Ваша программа должна обрабатывать большие колонии без чрезмерной траты памяти или времени. Многократный просмотр большого массива для построения следующих поколений - это банальный подход; здесь хороший программист выбрал бы более экономичные структуры данных и алгоритмы. Вам, возможно, захочется испытать какой-либо метод, отслеживающий только занятые квадраты. В программе нельзя определить бесконечно большое поле. Должно хватать поля некоторой известной величины  $m \times n$ . Что делать, если эволюция достигает границ поля? Один из возможных выходов - прервать эволюцию. Однако эволюция могла бы продолжаться, если бы мы устранили границы поля: соединили бы любые два противоположных края поля, затем концы полученного цилиндра. Полученная фигура, имеющая форму бублика, называется тор. Используйте этот подход в вашей программе (для этого достаточно более аккуратно определить соседей клетки, находящейся на краю поля). История колонии Жизнь захаровывает, если её просматривать как фильм, но она будет еще более увлекательней, если предстанет в цвете. Каждой клетке при рождении может быть приписан некоторый цвет, определяемый, возможно, её поколением или генами, переданными ей родителями. Циклические, но при этом движущиеся колонии (а таких немало) великолепны в своем сверкающем многоцветном наряде.

#### 7. Сетевая игра «Морской бой» (Балл -5)

На поле 10 на 10 позиций стоят невидимые вражеские корабли: 4 корабля по одной клетке, три корабля по 2 клетки, 2 корабля по 3 клетки, 1 корабль в 4 клетки. Позиции указываются русскими буквами от А до К (по строкам) и цифрами от 1 до 10 (по столбцам). Конфигурация и положение кораблей на поле выбираются с помощью датчика случайных чисел. Если клетка корабля угадана играющим верно, она отмечается крестиком; в противном случае точкой.

Написать программу для игры против компьютера в односторонний морской бой.

#### 8. Разработать программу «Фотоальбом». (Балл -3)

Программа должна выполнять следующие функции:

- Иметь графический интерфейс.

- Добавление и удаление фотографий.
- Добавление, удаление и редактирование подписей к фото.
- Просмотр фотоальбома в оба конца.
- Сохранение фотоальбома в файл. Фотографии сохраняются с подписями.
- Загрузка выбранного фотоальбома для просмотра и редактирования.

9. Разработать многопоточную программу, моделирующую работу товарного склада. (Балл -5)

Программа, реализует модель работы склада, отвечающего за хранение и продажу некоторого товара (одного). Склад содержит N помещений, каждый из которых может хранить определённое количество единиц товара. Поступающий товар помещается в помещения специальным погрузчиком. За товаром прибыло K покупателей, каждому из которых требуется по Lk единиц товара. Площадка перед складом мала и на ней может в один момент времени находиться либо погрузчик, либо один из покупателей. Если покупателям требуется больше товара, чем имеется на складе, то они ждут новых поступлений, периодически проверяя склад. Время работы склада ограничено.

10. Разработать программу построения Линейных фракталов. (Балл -3)

Фракталы - это математические объекты, имеющие дробную размерность в отличие от традиционных геометрических фигур целой размерности (например, одномерных линий или двумерных поверхностей). Фракталы - это нечто больше, чем математический курьез. Они дают чрезвычайно компактный способ описания объектов и процессов. Многие структуры обладают фундаментальным свойством геометрической регулярности, известной как инвариантность по отношению к масштабу, или *самоподобие*. Если рассматривать эти объекты в различном масштабе, то постоянно обнаруживаются одни и те же фундаментальные элементы. Эти повторяющиеся закономерности определяют дробную, или фрактальную, размерность структуры. В природе все фрактально: облака, изрезанная линия побережья, кромка листа, нервные и кровяные сосуды и т. д.

Некоторые из фракталов называются линейными, потому что строятся с помощью линейных (аффинных) преобразований плоскости. Такие преобразования изображение перемещают, сжимают, отражают, вращают и трансформируют произвольным образом при условии, что прямые линии на изображении остаются прямыми после преобразования. Описывая фракталы посредством аффинных преобразований, мы можем значительно уменьшить количество данных, необходимых для передачи изображения по линиям связи или для хранения его в памяти компьютера. Сложная форма, подобная форме листа папоротника, может быть полностью описана линейным алгоритмом, основанным лишь на 28 числовых параметрах. Заметим, что представление того же листа в точечном виде, как телевизионное изображение, требует несколько сотен тысяч числовых величин.

Алгоритм построения линейных фракталов.

Аффинное преобразование евклидовых координат задается с помощью двух линейных уравнений:

$$x_2 = ax + by + e \quad y_2 = cx + dy + f$$

Для построения фрактала поступают следующим образом:

Выбирается несколько аффинных преобразований и каждому преобразованию сопоставляется некоторая вероятность его использования. Вид аффинных преобразований, их количество и вероятность зависят от конкретного фрактала.

Так, например, для листа папоротника используются 4 преобразования:

a	b	c	d	e	f	p
0.00	0.00	0.00	0.16	0.00	0.00	0.01
	0	-	0		1	
	-	0	0		1	

0 0 0 0

Берется начальная точка с координатами  $x = 0$  и  $y = 0$ . Затем применяется одно из данных преобразований координат для определения новых значений и, какое именно преобразование применить определяется случайным образом в соответствии с заданной вероятностью  $p$ . Полученная точка изображается на плоскости цветом связанным с примененным преобразованием. Этот процесс повторяется достаточное число раз, пока не построится достаточно реалистическое изображение. Чтобы изображение удачно было расположено на экране нужно задать логический экран - в случае листа папоротника :

по оси  $x$  от -4 до 0

по оси  $y$  от 6 до 10

Замена значений коэффициентов  $b$  и  $c$  для листа папоротника во втором уравнении соответственно на 0.06 и - 0.06 увеличивают кривизну стебля папоротника. Замена их на 0.02 и -0.02 - уменьшают кривизну. Если исключить первое уравнение, то пропадает стебель. Программа должна строить несколько фракталов.

#### 11. Разработать программу «Волчий остров». (Балл-4)

Волчий остров размером  $20 \times 20$  заселен дикими кроликами, волками и волчицами. Имеется по несколько представителей каждого вида. Кролики довольно глупы: в каждый момент времени они с одинаковой вероятностью  $1/9$  передвигаются в один из восьми соседних квадратов (за исключением участков, ограниченных береговой линией) или просто сидят неподвижно. Каждый кролик с вероятностью 0,2 превращается в двух кроликов. Каждая волчица передвигается случайным образом, пока в одном из соседних восьми квадратов не окажется кролик, за которым она охотится. Если волчица и кролик оказываются в одном квадрате, волчица съедает кролика и получает одно очко. В противном случае она теряет 0,1 очка. Волки и волчицы с нулевым количеством очков умирают.

В начальный момент времени все волки и волчицы имеют 1 очко. Волк ведет себя подобно волчице до тех пор, пока в соседних квадратах не исчезнут все кролики; тогда, если волчица находится в одном из восьми близлежащих квадратов, волк гонится за ней. Если волк и волчица окажутся в одном квадрате и там нет кролика, которого нужно съесть, они производят потомство случайного пола.

Запрограммировать предполагаемую экологическую модель и понаблюдать за изменением популяции в течение некоторого периода времени.

#### 12. Разработать игровую программу - Две лисы и 20 кур. (Балл - 4)

На поле, указанной на рисунке формы находятся две лисы и 20 кур. Куры могут перемещаться на один шаг вверх, влево или вправо, но не назад и не по диагонали. Лисы также могут перемещаться только на один шаг (вверх, вниз, влево и вправо).

Лиса может съесть курицу, как в игре в шашки: если в горизонтальном или вертикальном направлении за курицей на один шаг следует свободное поле, то лиса перепрыгивает через курицу и берет ее.

Лисы всегда обязаны есть, и, когда у них есть выбор, они обязаны осуществлять «наиболее длинное поедание». Если два приема пищи имеют одинаковую длину, осуществляется один из них — по выбору лисы.

Составить программу, которая играет за лис (лисы перемещаются вверх, вниз и в стороны, но не по диагонали). Игрок перемещает кур (куры могут двигаться вверх и в стороны, но не назад). Партнеры играют по очереди, причем куры начинают.

		Л		Л		
К	К	К	К	К	К	К
К	К	К	К	К	К	К
		К	К	К		
		К	К	К		

Они выигрывают партию, если девяти из них удастся занять 9 полей, образующих верхний квадрат поля.

Начальное положение кур и лис изображено на рисунке.

Лисы выигрывают, если им удастся съесть 12 кур, так как тогда оставшихся кур недостаточно, чтобы занять 9 верхних полей.

### 13. Сетевая программа «On-line конференция».

Разработать сетевую многопоточную программу «On-line конференция». Количество участников конференции неограниченно. Для каждого участника, создается свой рабочий поток. Сервер ведет учет участников конференций и пересылает сообщения всем подключенным к определенной конференции.

Участники могут создавать конференции. Остальные присоединяются по желанию (одновременно можно участвовать в нескольких). Закрывать конференцию может только ее владелец. Сообщения передаются только членам конференции.

### 14. Разработка программы «Чат».

Разработать программу для диалога нескольких пользователей в сети. Иметь возможность создания приватного и группового диалога.

### 15. Разработка программы «Страница преподавателя».

Программа реализует домашнюю страницу преподавателя кафедры. В окне введено несколько панелей, которые поочередно выходят на первый план. Первая панель - заглавная страница содержит фотографию преподавателя и его краткое резюме. Последующие панели содержат информацию о читаемых курсах и т.д. Информация хранится в базе данных.

Преподаватель может изменять информацию в панелях. Так же преподаватель может писать сообщения студентам, вывешивать задания и материалы для скачивания.

Студенты просматривают информацию и могут скачивать задания и материалы.

### 16. Программа «Контроль успеваемости студентов».

Разработать сетевую программу, реализующую страницу приема заданий по дисциплине.

Преподаватель – авторизованный пользователь. Задает дисциплину, список студентов, количество точек контроля. Далее, после авторизации он проставляет баллы по мере сдачи студентами заданий. Программа подсчитывает суммарный балл для каждого студента.

Студенты могут только посмотреть текущее состояние дел. Должна быть предусмотрена возможность сортировки списка студентов по алфавиту или по рейтингу.

Информация хранится в базе данных.

### 17. Программа «Электронная система учета работы пункта проката видео».

Реализовать клиент-серверное приложение для автоматизации работы пункта проката.

В базе данных хранится информация: Название фильма, стоимость проката, даты выдачи и возврата предмета, фамилия получателя. Несданные предметы – пустая дата возврата (или выделение красным цветом).

Подсчет дохода пункта за заданный год/месяц.

Поиски по фамилии, по названию фильма. Фильтрация данных по жанрам, по датам выдачи, по невозвращенным вовремя и т.д.

### 18. АРМ кассира кинотеатра.

Разработать клиент-серверное приложение, автоматизирующий продажу билетов в кинотеатре. Иметь схемы нескольких залов. Реализовать графический интерфейс выбора мест в зале.

Реализовать продажу билетов на несколько фильмов и сеансов.

Данные продаж сохранять в базе данных. Реализовать различные запросы к базе данных и отображение результатов в удобной форме.

19. Разработка игровой сетевой программы «Крестики-нолики».

Разработать сетевую версию игровой программы «Крестики нолики». Количество играющих пар неограниченно. Для каждого клиента, подключившегося к игре, создается свой рабочий поток. Кроме того, могут быть подключены клиенты, наблюдающие за выбранной игрой. Процесс игры отображается графически в окне клиентов. Другая часть окна выполняет роль простого чата.

20. Разработка «Интернет-магазина» (можно на 2-х)

Клиентская часть:

- Отображение каталога товаров по разделам и подразделам (в виде дерева).
- Реализация "корзины заказа" - отображение выбранных товаров и их количества с пересчетом суммы заказа.
- Регистрация и идентификация заказчика.
- Формирование счетов на оплату.

Административная часть:

- Администрирование каталога товаров: просмотр информации о товаре, добавление, удаление товаров и подразделов, редактирование характеристик и фотографий, изменение структуры каталога.
- Фиксация в базе и администрирование заказов: просмотр информации о заказе и заказчике, добавление, удаление заказов, редактирование параметров.
- Фиксация в базе и администрирование пользователей: просмотр информации о зарегистрированных пользователях, добавление, удаление пользователей, редактирование контактной информации о них.

21. Свободная тема. Сформулировать тему и составить небольшую постановку задачи, где описать программу и ее функционал.