УТВЕРЖДАЮ

Заведующий кафедрой ИСиТ

\_\_\_\_\_\_\_\_\_\_\_\_ В.Е.Казаков

30.08.2024 г.

ПЕРЕЧЕНЬ ТЕМ КУРСОВЫХ ПРОЕКТОВ

Факультет *информационных технологий и робототехники*  Группа *Ит-13,13а*

Кафедра *«Информационные системы и технологии»*

Дисциплина *«Объектно-ориентированное проектирование и программирование»*

Разработать информационную систему:

|  |
| --- |
| 1. **Система** «**Вектора**». В отдельном модуле создать класс для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы индексов, возможность задания произвольных границ индексов при создании объекта и выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, печати (вывода на экран) элементов массива по индексам и всего массива. При необходимости в класс можно добавлять поля, методы либо свойства. Создать форму для работы с классом. Предусмотреть проверку корректности обрабатываемых данных и операций над ними. |
| 1. **Система** **Фургон кофе.** Загрузить фургон с определенными объёмом и грузоподъёмностью упаковками с различными сортами кофе (молотый, растворимый, в зёрнах). Упаковки имеют разные веса и объёмы. Объём и грузоподъёмность фургона задаются пользователем. В фургон можно грузить одинаковые упаковки. Если по заданным параметрам загрузить фургон невозможно, попросить пользователя ввести другие параметры. Найти кофе в фургоне по его сорту (задаётся пользователем). Вывести количество данного товара (например, молотый – 5 упаковок). |
| 1. **Система** **Новогодний подарок.** Определить класс **Конфета**. Создать несколько объектов-конфет разного веса, состава и наименования. Собрать детский подарок определённого веса. Входные данные: вес подарка (целое), выходные данные: подарок - список конфет, с суммарным весом, равным заданному. Конфеты в подарке могут повторяться. Если на заданный вес подобрать подарок невозможно, сообщить об этом пользователю и попросить ввести другой вес. Найти конфеты в подарке, соответствующие заданному диапазону содержания сахара, вывести их список по возрастанию количества сахара. |
| 1. **Система** **Домашние электроприборы.** Определить иерархию электроприборов. Посчитать, какие приборы нужно включить, чтобы они суммарно потребляли заданную мощность (вводится пользователем). Если на такую мощность подобрать приборы невозможно, вывести список приборов, потребляющих максимально приближенно меньшую, чем заданная, мощность. Приборы в списке не должны повторяться. Найти приборы в квартире, соответствующие заданному диапазону параметров. |
| 1. **Система Звукозапись.** Определить иерархию музыкальных композиций. Записать на диск сборку. Посчитать продолжительность. Провести перестановку композиций диска на основе принадлежности к стилю; на основе их продолжительности. Найти композиции, соответствующие заданному диапазону длины треков. |
| 1. **Система драгоценные Камни.** Класс **Драгоценный камень**. Отобрать камни для ожерелья заданного веса (в каратах). Входные данные: вес ожерелья (целое), выходные данные: ожерелье - список камней, с суммарным весом, равным заданному. Камни в ожерелье могут повторяться. Если на заданный вес подобрать ожерелье невозможно, сообщить об этом пользователю и попросить ввести другой вес. Посчитать общую стоимость ожерелья. Составить список камней в ожерелье и их количеств (например, бриллиант – 10, рубин – 15,…). Отсортировать по убыванию список по суммарным стоимостям камней разных видов (например, если в ожерелье 20 изумрудов и 2 бриллианта, 20 изумрудов должны быть выше в списке, т. к. они вместе дороже, чем 2 бриллианта). |
| 1. **Система странствующий Рыцарь.** Определить иерархию амуниции рыцаря. Создать объекты – элементы амуниции разной стоимости (различные кирасы, шлемы, наколенники и пр.). Экипировать рыцаря на заданную стоимость. Входные данные: стоимость полной экипировки (целое), выходные данные: комплект брони с суммарной стоимостью, равной заданной. Учитывать, что некоторые элементы экипировки – парные. Если на заданную сумму подобрать броню невозможно, сообщить об этом пользователю и попросить ввести другую сумму. Найти элементы амуниции, соответствующие заданному диапазону параметров цены и вывести их в порядке убывания стоимости. |
| 1. **Система УПРАВЛЕНИЕ АВТОБУСНЫМ ПАРКОМ.**   Вся информация в приложении хранится в трех файлах:  автобусы (№ автобуса; марка; количество мест; водитель)  маршруты (№ маршрута; пункт отправления; пункт назначения; время отправления; время прибытия)  рейсы (идентификатор рейса; № автобуса; № маршрута; дата рейса)  Приложение должно поддерживать следующие возможности:   * добавление автобуса * изменение водителя на автобусе (если он не назначен на другой автобус) * удаление автобуса (если автобус не назначен ни на один рейс) * добавление маршрута * изменение времени отправления и прибытия маршрута * удаление маршрута (если по нему нет ни одного рейса) * добавление рейса * удаление рейса * поиск рейсов по водителю, пункту назначения и пункту отправления |
| 1. **Система** «Геометрические **фигуры**».   Создать:  Класс "Точка", имеющий 2 координаты.  Реализовать:  - конструктор для инициализации координат;  - процедуру изменения координат точки на величину step  Класс "Отрезок", наследующий от класса "Точка". Реализовать конструкторы:  - с параметрами 2 точки,  - с параметрами 2 пары координат.  Реализовать:  - функцию вычисления длины отрезка  - процедуру изменения начала и конца отрезка на величину step  Класс "Прямоугольник", наследующий от класса "Отрезок". Реализовать конструкторы:  - с параметрами 2 точки,  - с параметрами 2 пары координат.  - с параметрами 2 отрезка,  Реализовать:  - функцию вычисления площади прямоугольника;  - функцию вычисления периметра прямоугольника;  - процедуру перемещения всех вершин прямоугольника на величину step  Создать класс "Круг" , наследующий от класса "Отрезок". Реализовать конструкторы:  - с параметром отрезок (радиус)  - с параметрами 2 точки: центр и точка на границе круга  - с параметрами 2 пары координат: координаты центра и точки на границе круга  Реализовать:  - функцию вычисления площади круга;  - функцию вычисления периметра круга;  - процедуру перемещения центра круга на величину step  Класс "Эллипс", наследующий от класса "Круг". Реализовать конструкторы:  - с параметрами 2 отрезка (радиусы)  - с параметрами 3 точки: центр и точки на границе эллипса (концы малого и большого радиусов)  - с параметрами 3 пары координат: координаты центра и точек на границе эллипса (концов малого и большого радиусов)  Реализовать:  - функцию вычисления площади эллипса (площадь эллипса: Pi\*R\*r);  - функцию вычисления периметра эллипса (периметр эллипса: (4\*Pi\*D\*d+(D-d)^2) / (D + d) );  - процедуру перемещения центра эллипса на величину step  - возможность добавления фигур в список с сортировкой по площади, периметру (задаётся пользователем).  При необходимости в систему можно добавлять классы, поля, методы либо свойства. Предусмотреть проверку корректности обрабатываемых данных и операций над ними. |
| 1. **Система** **КВАДРАТНОЕ УРАВНЕНИЕ**. Создать класс «Квадратное уравнение». Поля класса – коэффициенты уравнения, корни уравнения; методы класса – расчет корней. Предусмотреть конструктор для инициализации полей класса. В конструкторе аналитически определить отрезки, на котором существуют корни, в методах реализовать получение корней методами дихотомии, Ньютона, простых итераций. Если корней не существует, после создания объекта класса должно отображаться соответствующее сообщение. При необходимости в класс можно добавлять поля, методы либо свойства. Предусмотреть проверку корректности обрабатываемых данных и операций над ними. |
| 1. **Система** **калькулятор** **для обработки математических операций** с вещественными числами ("-", "+", "/", "\*", "(", ")", "0"-"9",".", «корень», «степень»). Определить абстрактный класс «Operation». Каждую математическую операцию реализовать, как наследник класса Operation. Пользователь вводит математическое выражение с клавиатуры. |
| 1. **Система** **калькулятор** **для обработки математических операций с матрицами** вещественных чисел (сложение, умножение, умножение на число, транспонирование, вычисление определителя). Пользователь вводит с клавиатуры одну или две матрицы, в зависимости от требуемой операции, и саму операцию. |
| 1. **Система** **Игра «крестики-нолики»**. Пользователь вводит с клавиатуры координаты своего хода, программа выводит координаты своего и (или) сообщение о выиграше/проиграше. Предусмотреть возможность выбора, кто будет делать первый ход. |
| 1. **Система меню столовой.** В меню включаются не менее 5 классов продуктов (задаются в текстовом файле). Реализовать для пользователя возможность выбора продуктов на завтрак. Реализовать процедуру подсчёта в завтраке продуктов заданного класса (класс вводится с клавиатуры). Реализовать интерфейс Nutritious (Питательный), содержащий метод calculateCalories(). Каждый из классов продуктов должен его реализовать.   Реализовать обработку специальных параметров (начинающихся с дефиса):  -calories вычислить и напечатать общую калорийность завтрака;  -sortByNameUp отсортировать продукты в завтраке по возрастанию (использовать название продуктов);  -sortByNameDown отсортировать продукты в завтраке по убыванию (использовать название продуктов);  -sortByCountUp отсортировать продукты в завтраке по возрастанию (использовать количество продуктов);  -sortByCountDown отсортировать продукты в завтраке по убыванию (использовать количество продуктов);  -sortByCaloryUp отсортировать продукты в завтраке по возрастанию (использовать общую калорийность продуктов);  -sortByCaloryDown отсортировать продукты в завтраке по убыванию (использовать общую калорийность продуктов). |
| 1. **Система пункт проката видеодисков.** Вся информация в приложении хранится в трех файлах:   – пользователи проката (фамилия; имя; отчество; адрес; телефон);  – видеодиски (идентификатор диска; название; жанр; год выпуска; режиссер);  – прокат (идентификатор проката; телефон пользователя; идентификатор диска; дата выдачи; предполагаемая дата возврата; возвращен ли диск).  Приложение должно поддерживать следующие возможности:  – добавление пользователя  – изменение информации о пользователе  – удаление пользователя (если у него нет невозвращенных дисков и нет просроченных дисков)  – добавление диска  – удаление диска (если он не в прокате)  – выдача диска в прокат (если диск доступен и у пользователя нет просроченных дисков)  – возврат диска  – поиск клиента по фамилии и телефону  – поиск диска по названию, жанру, году выпуска, режиссеру (задаётся пользователем) |
| 1. **Система БОЛЬНИЦА.** Пациенту назначается лечащий Врач. Врач может сделать Назначение Пациенту (процедуры, лекарства, операции). Медсестра или другой Врач выполняют назначение. Пациент может быть выписан из Больницы по окончании лечения, при нарушении режима или иных обстоятельствах. После выписки информация о Пациенте сохраняется в Архиве. |
| 1. **Система ВСТУПИТЕЛЬНЫЕ ЭКЗАМЕНЫ.** Абитуриент регистрируется на Специальность, сдает экзамены. Преподаватель выставляет оценку. Результаты Абитуриентов хранятся в Списке. Система подсчитывает средний балл и определяет Абитуриентов, зачисленных в учебное заведение на основании плана приема на Специальность, *а также Абитуриентов, имеющих полупроходной балл.* |
| 1. **Система СТРОИТЕЛЬНАЯ ФИРМА.** Заказчик представляет Техническое Задание (ТЗ) на проектирование дома. Главный конструктор регистрирует ТЗ, определяет стоимость проектирования и строительства, выставляет Заказчику Счет за проектирование и после оплаты создает Бригаду Конструкторов для выполнения Проекта. |
| 1. **Система ТЕЛЕФОННАЯ СТАНЦИЯ.** Абонент оплачивает Счет за Услуги, может попросить Администратора сменить номер и отказаться от услуг. Администратор изменяет номер или временно отключает Абонента за неуплату. Телефонный оператор ежемесячно вносит информацию об Услугах, оказанных Абоненту. |
| 1. **Система ИНТЕРНЕТ-МАГАЗИН.** Администратор добавляет информацию о Товаре в список. Клиент делает Заказ на Товары. Администратор регистрирует продажу после оплаты Товаров Клиентом или может занести неплательщиков в "черный список" (при просрочке оплаты). Курьер доставляет Товар, делая об этом соответствующую пометку в списке. |
| 1. **Система ФАКУЛЬТАТИВ.** Преподаватель объявляет запись на Курс. Студент записывается на Курс, обучается и по окончании Преподаватель выставляет Оценку, которая сохраняется в Архиве. Студентов, Преподавателей и Курсов при обучении может быть несколько. |
| 1. **Система БИБЛИОТЕКА.** Читатель оформляет Заказ на Книгу. Система осуществляет поиск в Каталоге. Библиотекарь выдает Читателю Книгу на абонемент или в читальный зал. |
| 1. **Система АВТОБАЗА.** Диспетчер распределяет заявки на Рейсы между Водителями и назначает для этого Автомобиль. Водитель может сделать заявку на ремонт. Врач может отстранить Водителя от работы. Техник делает отметку о выполнении Рейса и состоянии Автомобиля |
| 1. **Система АЭРОФЛОТ.** Администратор формирует летную Бригаду (пилоты, штурман, радист, стюардессы) на Рейс. Каждый Рейс выполняется Самолетом с определенной вместимостью и дальностью полета. Рейс может быть отменен Диспетчером из-за погодных условий или назначения или из-за технических неисправностей. |
| 1. **Система ЗАКАЗ ГОСТИНИЦЫ.** Клиент оставляет Заявку на Номер, указав класс апартаментов, количество мест и время пребывания. Администратор рассматривает Заявку, подтверждает или отклоняет. Результат просматривает Клиент. В случае подтверждения Заявки Клиент оплачивает услуги. По истечении срока пребывания, Администратор делает отметку об освобождении Номера. |
| 1. **Система ЖИЛИЩНО-КОММУНАЛЬНЫЕ УСЛУГИ.** Квартиросъемщик отправляет Заявку, в которой указывает вид работ, желаемое время выполнения. Диспетчер подбирает необходимого Исполнителя и регистрирует его в Плане работ. Диспетчер может отклонить Заявку, в случае занятости всех Исполнителей. Квартиросъемщик может просмотреть информацию о сформированных им Заявках и сделать отметку об удовлетворении Заявки. Исполнители могут просмотреть список Заявок, назначенных для них. |
| 1. **Система ПРОКАТ АВТОМОБИЛЕЙ.** Клиент выбирает Автомобиль из списка доступных. Заполняет форму Заказа, указывая паспортные данные, срок аренды. При подтверждении Заказа, система выписывает сумму и Клиент оплачивает Заказ. В случае повреждения Автомобиля Клиентом, Администратор вносит соответствующие пометки |
| 1. **Система СКАЧКИ.** Клиент делает Ставку на Лошадь в Забеге. Букмекер устанавливает уровень выигрыша на Лошадь в Забеге. Администратор фиксирует список, состав и результаты Забегов на день. |
| 1. **Система УПРАВЛЕНИЯ ПРОГРАММНЫМИ ПРОЕКТАМИ.**   Вся информация приложения хранится в трех файлах:   * работники (табельный № работника; фамилия; специальность; опыт работы) * проекты (идентификатор проекта; название; описание; стадия проекта) * назначения (идентификатор назначения; идентификатор проекта; табельный № работника)   Приложение должно поддерживать следующие возможности:   * добавление работника * изменение данных работника * добавление проекта * переход проекта на следующую стадию (проект проходит по порядку следующие стадии: анализ → проектирование → разработка → тестирование → внедрение → сдача проекта) * провал проекта (установка стадии в «провален») * добавление назначения (при условии, что работник не может быть назначении одновременно на несколько проектов, исключения – только проекты со стадиями «провален» и завершен») * поиск проектов по названию, стадии * поиск работника по специальности, специальности и опыту * статистика – процент проваленных и завершенных проектов (всего и для каждого работника в отдельности). |
| 1. **Система КООРДИНАЦИИ ВЫПУСКА ГАЗЕТ.**   Вся информация приложения хранится в трех файлах:   * авторы (фамилия; специальность: журналист или фотограф; телефон) * номер (идентификатор номера; описание; сдан ли в набор) * материал (идентификатор материала; телефон автора; идентификатор номера; описание; статус: готовиться, сдан редактору, корректура, готов)   Приложение должно поддерживать следующие возможности:   * добавление работника * изменение данных работника * добавление номера газеты * запуск номера в набор (установка поля «сдан ли в набор» при условии, что готовы все материалы номера) * добавление материала (при условии, что у автора нет готовящихся материалов) * изменение статуса материала * поиск всех и только свободных авторов нужной специальности * поиск всех материалов по статусу * поиск всех отпечатанных материалов данного автора (материалов, входящих в номера, сданные в набор). |
| 1. **Система УПРАВЛЕНИЕ СРЕДСТВАМИ НА БАНКОВСКИХ СЧЕТАХ.**   Вся информация приложения хранится в двух файлах:   * клиенты (идентификатор клиента; фамилия; телефон) * счета (идентификатор счета; идентификатор клиента-владельца; сумма)   Приложение должно поддерживать следующие возможности:   * добавление клиента * изменение данных о клиенте * удаление клиента (при отсутствующих открытых счетах) * открытие счета * закрытие (удаление) счета (при нулевой сумме счета) * пополнение баланса счета * выдача средств (при достаточной сумме) * перевод средств на другой счет (при достаточной сумме с вычетом 5% комиссии) * поиск клиентов по фамилии с суммарным балансом всех его счетов. |
| 1. **Система УЧЕТ ОРГТЕХНИКИ В ОРГАНИЗАЦИИ.**   Вся информация приложения хранится в двух файлах:   * отделы (идентификатор отдела; название; материально-ответственное лицо) * техника (идентификатор устройства; идентификатор отдела; название; описание; стоимость; год поступления)   Приложение должно поддерживать следующие возможности:   * добавление отдела * изменение материально-ответственного лица * расформирование отдела (при отсутствии в нем оргтехники) * поступление нового устройства * списание устройства * поиск оргтехники, закрепленной за материально-ответственным лицом (с итоговой стоимостью) * поиск оргтехники по названию (с итоговой стоимостью) * поиск оргтехники по отделу (с итоговой стоимостью) * поиск морально-устаревшей техники (старше 5-ти лет). |
| 1. **Система КООРДИНАЦИЯ РАБОТЫ АДВОКАТСКОЙ КОНТОРЫ.**   Вся информация приложения хранится в двух файлах:   * юристы (идентификатор юриста; фамилия; специализация; стоимость часа) * процессы (идентификатор дела; идентификатор юриста; роль юриста: адвокат, помощник адвоката, консультант; описание; статус: обсуждается, в процессе, выиграно, проиграно)   Приложение должно поддерживать следующие возможности:   * добавление юриста * изменение стоимости часа работы юриста * новый процесс (переходит в статус «обсуждается», при добавлении рассчитывается стоимость для клиента одного дня ведения процесса, учитывая, что за один день адвокат работает 8, помощник адвоката 6, а консультант – 4 часа, при этом один юрист не может вести несколько дел сразу) * перевод дела в статус «в процессе» * закрытие дела (установки статуса «выиграно» или «проиграно») * поиск свободного юриста по специализации (с указанием процента выигранных дел) * вывод всех текущих дел (со списком юристов и стоимостью дня по каждому делу) * поиск юриста по фамилии (с указанием дела, над которым он работает) * вывод всех проигранных дел данного юриста. |
| 1. **Система СОЦИАЛЬНЫЕ УСЛУГИ.**   В текстовом файле хранится список учета нуждающихся в улучшении жилищных условий. Каждая запись этого списка содержит: порядковый номер, Ф.И.О., величину жилплощади на одного члена семьи, наличие льгот (да/нет) и дату постановки на учет (число, месяц, год). Рассчитать ожидаемую дату получения квартиры на сегодняшний день с учетом того, что максимальный срок ожидания – 3 года. При наличии льгот срок ожидания уменьшается на 1 год. Если величина жилплощади на одного члена семьи меньше 20 кв. м, срок ожидания также уменьшается на 1 год. По заданному году постановки на учет (вводится с клавиатуры), вывести список нуждающихся с указанием ожидаемой даты получения квартиры. Реализовать возможность добавления в файл записи о новом нуждающемся. Реализовать возможность удаления в архив (отдельный текстовый файл) сведений о нуждающемся, получившем квартиру. |
| 1. **Система ТРАНСПОРТНОЕ СРЕДСТВО.** В отдельном модуле cоздать класс «транспортное средство». Определить поля «максимальная скорость», «расход топлива на 100 км» и «ёмкость бака». Предусмотреть конструктор для создания объекта, методы вычисления расхода топлива на заданное расстояние (если полученный расход превышает ёмкость бака, вывести значение ёмкости бака) и вычисления скорости по заданному расстоянию и времени в пути (если полученная скорость превышает максимальную, вывести зна-чение максимальной скорости). Создать класс «легковой автомобиль», наследующий от класса «транспортное средство». Определить поля «модель», «цвет», «количество посадочных мест». Предусмотреть для их инициализации конструктор, а для работы с ними – соответствующие свойства и методы. Если количество посадочных мест превышает 9, объект класса не может быть создан. Создать класс «грузовой автомобиль», наследующий от класса «транспортное средство». Опреде-лить поля «тип автомобиля» (тягач, самосвал, для перевозки людей и пр.), «грузоподъёмность». Предусмотреть для их инициализации конструктор, а для работы с ними – соответствующие свойства и методы. При необходимости в классы можно добавлять поля, методы либо свойства. Создать форму для работы с классами. Предусмотреть проверку коррект-ности обрабатываемых данных и операций над ними. Реализовать возможность добавления транспортных средств в список с сортировкой по максимальной скорости, ёмкости бензобака, расходу топлива на 100 км (задаётся пользователем). |
| 1. **Система** **Цветочница.** Создать класс **Цветок**. Создать объекты-цветы разного наименования и стоимости. Собрать букет из цветов на определённую сумму (задаётся пользователем с клавиатуры). Входные данные: стоимость букета (целое), выходные данные: букет - список цветов, с суммарной стоимостью, равной заданной. Цветы в букете могут повторяться. Если на заданную сумму подобрать букет невозможно, сообщить об этом пользователю и попросить ввести другую сумму. Провести сортировку цветов в букете по их стоимости; по их названиям (тип сортировки вводится пользователем). Найти количество цветов заданного типа в букете (розы, гвоздики и пр. – вводится пользователем). |

ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ ПРОЕКТА

1. На основе сущностей предметной области создать классы их описывающие.
2. Типы данных должны соответствовать их сущности (например, для описания даты использовать тип Date)
3. Классы и методы должны иметь отражающую их функциональность названия.
4. Для хранения списков или наборов классов, использовать классы-коллекции.
5. Использовать возможности ООП: классы, наследование, полиморфизм, инкапсуляция.
6. Каждый класс должен иметь исчерпывающее смысл название и информативный состав.
7. Наследование должно применяться только тогда, когда это имеет смысл.
8. При кодировании должны быть использованы соглашения об оформлении кода java code convention.
9. Классы должны быть грамотно разложены по пакетам.
10. Для хранения параметров инициализации использовать файлы (Если нет требования консольного ввода).
11. Предусмотреть корректную обработку неправильных действий пользователя.
12. Везде, где это возможно, использовать лямбда-выражения и Stream-API.
13. Предпочтительно создание собственных классов-исключений

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ст.преп. Деркаченко П.Г.