Постановки задач

Лабораторная работа №1

1.1. Рациональное число

Разработать класс **Рациональное число**.

Класс должен хранить корректные дроби (знаменатель не равен 0) и выполнять с ними 4 стандартные арифметические операции. Дробь должна храниться в несократимом виде.

Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

1.2. Длинное число

Разработать класс **Длинное число**.

Класс должен хранить знаковые целые числа длиной 64 бита, используя для этого целый тип int (unsigned int) длиной 32 бита.

Класс должен предоставлять 5 стандартных арифметических операций для целых чисел, сохраняя ту же семантику обработки переполнения, что и для типа int.

Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

1.3. Время

Разработать класс **Время**.

Класс должен хранить время в формате: часы, минуты, секунды (в диапазоне от 0 часов 0 минут 0 секунд до 23 часа 59 минут 59 секунд).

Класс должен предоставлять операции: 1) установить время, 2) узнать время, 3) вычислить разницу между заданным временем и установленным (в часах, минутах и секундах), 4) сдвинуть время на заданное смещение (в часах, минутах и секундах) в меньшую и в большую сторону.

Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

1.4. Конвертер длин

Разработать класс **Конвертер длин**.

Класс должен хранить длину в метрах и предоставлять методы по ее преобразованию в другие единицы измерения (например, фут, ярд, аршин, сажень, …).

Класс должен предоставлять операции: 1) установить текущую длину в метрах, 2) узнать текущую длину в метрах, 3) узнать текущую длину в выбранной единице измерения (из списка поддерживаемых).

Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

1.5. Конвертер весов

Разработать класс **Конвертер весов**.

Класс должен хранить вес в килограммах и предоставлять методы по его преобразованию в другие единицы измерения (например, аптечный фунт, тройская унция, пуд, …).

Класс должен предоставлять операции: 1) установить текущий вес в килограммах, 2) узнать текущий вес в килограммах, 3) узнать текущий вес в выбранной единице измерения (из списка поддерживаемых).

Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

1.6. Конвертер температур

Разработать класс **Конвертер температур**.

Класс должен хранить температуру в градусах Цельсия и предоставлять методы по его преобразованию в другие единицы измерения (Фаренгейт, Кельвин, Ранкин, …).

Класс должен предоставлять операции: 1) установить текущую температуру в градусах Цельсия, 2) узнать текущую температуру в градусах Цельсия, 3) узнать текущую температуру в выбранной единице измерения (из списка поддерживаемых).

Класс должен содержать все необходимые конструкторы, оператор присваивания, а также «уметь» выводить себя на консоль.

Лабораторная работа №2

2.1. Календарь событий

Разработать класс **Календарь событий**.

Класс должен позволять сохранять даты заданных событий в формате: год, месяц, день (в диапазоне от 1 января 1 года до 31 декабря 2020 года), наименование события. На каждый день может приходиться только одно событие. Общее число событий – не более 30.

Класс должен предоставлять операции: 1) установить событие, 2) узнать дату выбранного события, 3) вычислить разницу между заданной датой и датой события (в годах, месяцах, днях), 4) сформировать новое событие, сдвинув выбранное существующее событие на заданное смещение (в годах, месяцах, днях) в меньшую и в большую сторону.

Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

2.2. Полином

Разработать класс **Полином**.

Класс должен хранить полином (многочлен) от одной переменной (x). Степень полинома n находится в диапазоне от 0 до 12.

Класс должен предоставлять следующие операции: 1) задать степень полинома, 2) задать коэффициенты мономов полинома, 3) узнать степень полинома, 4) узнать значение коэффициента по его номеру, 5) вычислить значение полинома в заданной точке x, 6) найти производную полинома.

Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

2.3. Вектор

Разработать класс **Вектор**.

Класс должен хранить вектор целых чисел заданного размера (от 1 до 20).

Класс должен предоставлять следующие операции: 1) задать размер вектора, 2) узнать размер вектора, 3) задать компоненту вектора по ее номеру, 4) узнать компоненту вектора по ее номеру, 5) вычислить длину вектора, 6) найти скалярное произведение двух векторов, 7) найти сумму двух векторов одного размера.

Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

2.4. Матрица

Разработать класс **Матрица**.

Класс должен хранить квадратную матрицу целых чисел заданного размера (от 2 до 8 строк).

Класс должен предоставлять следующие операции: 1) задать размер матрицы, 2) узнать размер матрицы, 3) задать элемент матрицы по его индексам, 4) узнать элемент матрицы по его индексам, 5) проверить, обладает ли матрица диагональным преобладанием, 6) вычислить сумму двух матриц одного размера.

Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

2.5. Динамический массив

Разработать класс **Динамический массив**.

Класс должен хранить заданное число вещественных элементов, размещаемых в динамической памяти.

Класс должен предоставлять следующие операции: 1) задать размер массива, 2) узнать размер массива, 3) задать элемент массива по его индексу, 4) узнать элемент массива по его индексу, 5) найти минимальный элемент массива, 6) проверить, является ли массив упорядоченным, 7) выделить из массива подмассив с элементами с нечетными индексами.

Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

2.6. Строка

Разработать класс **Строка**.

Класс должен хранить строку символов произвольной длины (от 0 до 40).

Класс должен предоставлять следующие операции: 1) задать строку, 2) узнать длину строки, 3) получить символ строки по его индексу, 4) изменить символ строки по заданному индексу, 5) выделить подстроку из строки, 6) проверить, является ли строка палиндромом, 7) найти, сколько разных символов латинского алфавита содержится в строке.

Класс должен содержать все необходимые конструкторы, деструктор, оператор присваивания, а также «уметь» выводить себя на консоль.

Лабораторная работа №3

3.1. Ряд Тейлора

Разработать класс **Ряд Тейлора**.

Класс должен формировать ряд Тейлора для выбранной функции из списка поддерживаемых с заданным числом членов ряда. Минимальный список функций: sin(x), cos(x), exp(x).

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции: 1) задать текущую функцию, 2) узнать текущую функцию, 3) задать текущее число членов ряда, 4) узнать текущее число членов ряда, 5) выдать формулу ряда для выбранной функции, 6) выдать значение заданного члена ряда, 7) рассчитать значение ряда в выбранной точке x, 8) вывести отклонение значения ряда в выбранной точке от эталонного значения текущей функции в данной точке (эталонное значение вычисляется, используя соответствующую функцию из стандартной библиотеки C++).

3.2. Табулятор функции

Разработать класс **Табулятор функции**.

Класс должен позволять выполнять табулирование произвольной функции одной переменной, заданной в виде функции языка C++.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции: 1) задать текущую функцию, 2) задать текущее число точек табулирования, 3) узнать текущее число точек табулирования, 4) задать отрезок табулирования, 5) узнать отрезок табулирования, 6) выполнить табулирование функции, 7) выдать результаты табулирования, 8) сохранить результаты табулирования в файл.

3.3. Расчет интегралов

Разработать класс **Расчет интегралов**.

Класс должен позволять вычислять приближенное значение интеграла от произвольной функции одной переменной, заданной в виде функции языка C++. Интеграл необходимо вычислять в заданных пределах интегрирования, используя методы левых, правых и средних прямоугольников.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции: 1) задать текущую функцию, 2) задать пределы интегрирования, 3) узнать пределы интегрирования, 4) задать число отрезков метода прямоугольников, 5) выбрать метод вычисления, 6) вычислить значение интеграла выбранным методом, 7) вывести результат вычисления на экран.

3.4. Пользовательское меню

Разработать класс **Пользовательское меню**.

Класс должен предоставлять одноуровневое меню с заданным числом команд в консольном режиме экрана.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции: 1) задать число команд меню, 2) узнать число команд меню, 3) задать название пункта меню с указанным номером, 4) вывести меню на экран в выбранной позиции окна консоли, 5) обеспечить выбор пользователем пункта меню с выдачей выбранного номера, 6) выдать номер последнего выбранного пользователем пункта меню.

3.5. Однострочный текстовый редактор

Разработать класс **Однострочный текстовый редактор**.

Класс должен предоставлять возможность разместить в выбранной позиции окна консоли поле заданной длины для ввода с клавиатуры последовательности символов. В минимальном варианте длина последовательности не должна превышать длину поля ввода.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции: 1) задать длину поля ввода, 2) узнать длину поля ввода, 3) задать позицию поля ввода в окне консоли, 4) узнать позицию поля ввода в окне консоли, 5) разместить однострочный текстовый редактор в окне консоли, 6) обеспечить ввод пользователем строки с длиной не больше длины поля ввода, 7) выдать введенную пользователем строку.

3.6. Словарь переводчика

Разработать класс **Словарь переводчика**.

Класс должен предоставлять возможность формировать англо-русский словарь. В минимальном варианте каждому английскому слову соответствует ровно одно русское слово-перевод.

Класс должен содержать необходимые служебные методы (конструкторы, деструктор и пр.).

Класс должен предоставлять следующие операции: 1) добавить в словарь слово и его перевод, 2) изменить перевод указанного слова, 3) узнать перевод выбранного слова, 4) проверить наличие слова в словаре, 5) узнать число слов в словаре, 6) сохранить словарь в файл, 7) считать словарь из файла.

Лабораторная работа №4

4.1. Термометр

Разработать класс **Термометр**.

Класс должен хранить историю наблюдений за температурой в течение одного календарного (не високосного) года. Наблюдения описываются датой (день, месяц, год) и временем (в часах). При поступлении нового наблюдения для уже существующих даты и времени старое наблюдение заменяется. Температура задается в градусах Цельсия.

Класс должен содержать необходимые служебные методы.

Класс должен предоставлять следующие операции: 1) установить начальные дату и время наблюдений, 2) узнать начальные дату и время наблюдений, 3) задать наблюдение, 4) узнать температуру в выбранном наблюдении, 5) задать серию наблюдений для выбранной даты, 6) найти среднюю температуру для выбранной даты, выбранного месяца, за всю историю наблюдений, 7) найти среднюю дневную или ночную температуру для выбранного месяца, 8) сохранить историю наблюдений в файл, 9) считать историю наблюдений из файла.

4.2. Весы напольные

Разработать класс **Весы напольные**.

Класс должен хранить историю наблюдений за показаниями веса членов семьи (до пяти человек). Показания описываются датой (день, месяц, год) и именем члена семьи. При поступлении нового наблюдения для уже существующей даты старое наблюдение заменяется. Вес задается в килограммах с точностью до 50 граммов.

Класс должен содержать необходимые служебные методы.

Класс должен предоставлять следующие операции: 1) установить начальную дату наблюдений, 2) узнать начальную дату наблюдений, 3) задать наблюдение, 4) узнать вес в выбранном наблюдении, 5) найти средний вес члена семьи в выбранном месяце или за всю историю наблюдений, 6) найти минимальный вес члена семьи в выбранном месяце или за всю историю наблюдений и дату, когда он наблюдался 7) найти максимальный вес члена семьи в выбранном месяце или за всю историю наблюдений и дату, когда он наблюдался, 8) сохранить историю наблюдений в файл, 9) считать историю наблюдений из файла.

4.3. Шагомер

Разработать класс **Шагомер**.

Класс должен хранить историю подсчета шагов владельца. Каждый подсчет описывается датой (день, месяц, год) и интервалом времени (час, минута начала движения, час, минута окончания движения). Подсчет ведется в шагах с точностью до единицы.

Класс должен содержать необходимые служебные методы.

Класс должен предоставлять следующие операции: 1) установить дату начала подсчетов, 2) узнать дату начала подсчетов, 3) задать подсчет, 4) получить информацию о выбранном подсчете, 5) найти среднее число шагов в выбранном месяце или за всю историю наблюдений, 6) найти среднее число шагов в выбранный день недели за всю историю наблюдений, 7) найти максимальное число шагов в день в выбранном месяце или за всю историю наблюдений и дату, когда оно было достигнуто, 8) сохранить историю подсчетов в файл, 9) считать историю подсчетов из файла.

4.4. Контакты

Разработать класс **Контакты**.

Класс должен хранить информацию о контактах владельца. Каждый контакт содержит следующие данные: фамилия; имя; отчество; телефон; день рождения (день, месяц, год); признак, относится ли контакт к избранным. Контакты хранятся упорядоченно по фамилии, имени, отчеству. Фамилия, имя, отчество (ФИО) являются обязательными полями. Данные вводятся на русском языке.

Класс должен содержать необходимые служебные методы.

Класс должен предоставлять следующие операции: 1) создать новый контакт, 2) изменить выбранный контакт, 3) найти контакт по ФИО, 4) найти контакт по телефону, 5) выдать все контакты на заданную букву, 6) узнать текущее число контактов, 7) внести контакт в список избранных, 8) удалить контакт из списка избранных, 9) выдать все избранные контакты, 10) удалить контакт, 11) сохранить контакты в файл, 12) считать контакты из файла.

4.5. Песенник

Разработать класс **Песенник**.

Класс должен хранить информацию о песенных композициях. Каждая песня описывается следующими данными: название, поэт (автор стихов), композитор (автор музыки), исполнитель, название альбома (если входит в какой-то альбом), дата выпуска (день, месяц, год). Песни хранятся упорядоченно по названию. Данные вводятся на русском языке.

Класс должен содержать необходимые служебные методы.

Класс должен предоставлять следующие операции: 1) добавить песню, 2) изменить данные выбранной песни, 3) найти песню по названию и исполнителю, 4) выдать все песни заданного поэта, 5) выдать все песни заданного композитора, 6) выдать все песни заданного исполнителя, 7) узнать текущее число песен, 8) удалить песню, 9) сохранить песенник в файл, 10) считать песенник из файла.

4.6. Фильмотека

Разработать класс **Фильмотека**.

Класс должен хранить информацию о фильмах. Каждый фильм описывается следующими данными: название, режиссер, сценарист, композитор, дата выхода в прокат (день, месяц, год), сборы (в рублях). Фильмы хранятся упорядоченно по названию и годам. Данные вводятся на русском языке.

Класс должен содержать необходимые служебные методы.

Класс должен предоставлять следующие операции: 1) добавить фильм, 2) изменить данные выбранного фильма, 3) найти фильм по названию и году, 4) выдать все фильмы заданного режиссера, 5) выдать все фильмы, вышедшие в прокат в выбранном году, 6) выдать заданное число фильмов с наибольшими сборами, 7) выдать заданное число фильмов с наибольшими сборами в выбранном году, 8) узнать текущее число фильмов, 9) удалить фильм, 10) сохранить фильмотеку в файл, 11) считать фильмотеку из файла.

Лабораторная работа №5

5.1. Банкомат

Разработать классы **Банкомат** и **Процессинговый центр**.

Класс **Банкомат** должен имитировать работу банкомата по приему и выдаче наличных денежных средств (в рублях). Класс должен поддерживать работу с купюрами в 100, 200, 500, 1000, 2000, 5000 рублей. Выдаваемая или принимаемая за одну операцию сумма ограничена 40 купюрами (независимо от их достоинства). Купюры каждого достоинства хранятся в отдельной кассете. Емкость каждой кассеты – 2000 тысяч купюр. Считать, что начальная загрузка банкомата составляет 80% для каждой кассеты. Считать, что клиент банка идентифицируется по номеру пластиковой карты (для упрощения – номер карты от «0001» до «9999»).

База клиентов хранится в классе **Процессинговый центр**. Номера выданных клиентам карт могут идти не подряд. Считать, что информация о клиенте состоит из номера карты, ФИО владельца, суммы на счету, (для упрощения – без копеек), PIN-кода (последовательность из 4-х цифр, каждая от 0 до 9).

Класс **Банкомат** должен предоставлять следующие операции: 1) принять карту клиента, 2) найти клиента по номеру карты, 3) проверить PIN-код, 4) распечатать состояние счета клиента, 5) выдать клиенту наличные (списав выданную сумму со счета), 6) принять от клиента наличные (зачислив полученную сумму на счет), 7) заблокировать карту клиента, если до ее возврата клиенту три раза подряд набран неверный PIN-код, 8) вернуть карту клиенту.

Все операции должны сопровождаться необходимыми проверками на указанные выше ограничения.

Класс **Процессинговый центр** должен использоваться для поддержки работы класса **Банкомат** и может быть разработан в минимально-необходимом варианте.

5.2. Депозит

Разработать классы **Депозит** и **Процессинговый центр**.

Класс **Депозит** должен имитировать работу интернет-банка в части управления депозитом для зарплатных клиентов банка (тех, чья зарплата перечисляется работодателем на счета работников, открытые в данном банке). Депозит может открываться клиентом на выбранный срок. Возможные варианты: 3 месяца, 6 месяцев, 1 год, 2 года, 3 года. Считать, что процентные ставки по депозитам зависят только от срока депозита и начальной суммы. Диапазоны начальных сумм: до 100 тыс. рублей, от 100 до 500 тыс. рублей, от 500 тыс. до 1 млн рублей, свыше 1 млн рублей. Процентные ставки указываются для срока 1 год (например, 6% годовых по вкладу на 6 месяцев от 100 до 500 тыс. рублей или 6.6% годовых по вкладу на 1 год от 500 тыс. до 1 млн рублей). Считать, что проценты начисляются раз в месяц на начальную сумму депозита и могут быть сняты клиентом в любой момент. Депозит может быть закрыт только по истечении срока, на который он был открыт.

База клиентов хранится в классе **Процессинговый центр**. Считать, что информация о клиенте состоит из номера зарплатного счета (для упрощения – номер счета от «0001» до «9999»), ФИО владельца, суммы на зарплатном счету, (для упрощения – без копеек), информации о депозите, пароля (произвольная строка, выбранная пользователем, с длиной больше 3 символов). Считать, что авторизация клиента в интернет-банке происходит по номеру зарплатного счета и паролю.

Класс **Депозит** должен предоставлять следующие операции: 1) авторизовать клиента, 2) показать информацию о доступных клиенту депозитах, исходя из суммы на его зарплатном счету, 3) проверить наличие открытого депозита, 4) открыть депозит (переведя указанную

сумму с зарплатного счета клиента на депозит), 5) показать состояние депозита, 6) снять проценты (переведя их на зарплатный счет клиента), 7) закрыть депозит (переведя всю накопленную сумму на зарплатный счет клиента).

Все операции должны сопровождаться необходимыми проверками на указанные выше ограничения.

Класс **Процессинговый центр** должен использоваться для поддержки работы класса **Депозит** и может быть разработан в минимально-необходимом варианте.

5.3. Кредит

Разработать классы **Кредит** и **Процессинговый центр**.

Класс **Кредит** должен имитировать работу интернет-банка в части управления кредитом для зарплатных клиентов банка (тех, чья зарплата перечисляется работодателем на счета работников, открытые в данном банке). Кредит может выдаваться клиенту на выбранный им срок. Возможные варианты: 1 год, 2 года, 3 года, 5 лет, 15 лет. Считать, что процентные ставки по кредитам зависят только от срока и суммы кредита. Диапазоны сумм: до 100 тыс. рублей, от 100 до 500 тыс. рублей, от 500 тыс. до 1 млн рублей, от 1 до 3 млн рублей. Процентные ставки указываются за 1 год (например, 12% годовых по кредиту на 3 года на сумму от 500 тыс. до 1 млн рублей или 15% годовых по кредиту на 1 год на сумму от 100 до 500 тыс. рублей). Выплаты по кредиту начисляются клиенту раз в месяц. Клиент может выплатить сумму больше или равную начисленной. Клиент может досрочно погасить кредит. Считать, что кредит одобряется банком клиенту, если текущая сумма на его зарплатном счету достаточна для шести ежемесячных выплат по кредиту. После одобрения банком сумма кредита перечисляется на зарплатный счет клиента.

База клиентов хранится в классе **Процессинговый центр**. Считать, что информация о клиенте состоит из номера зарплатного счета (для упрощения – номер счета от «0001» до «9999»), ФИО владельца, суммы на зарплатном счету, (для упрощения – без копеек), информации о кредите, пароля (произвольная строка, выбранная пользователем, с длиной больше 3 символов). Считать, что авторизация клиента в интернет-банке происходит по номеру зарплатного счета и паролю.

Класс **Кредит** должен предоставлять следующие операции: 1) авторизовать клиента, 2) показать информацию о доступных клиенту кредитах 3) проверить наличие взятого в банка кредита, 4) проверить возможность получения выбранного кредита, 5) получить выбранный кредит, 6) показать текущее состояние кредита, 7) погасить начисления по кредиту, 8) досрочно погасить кредит.

Все операции должны сопровождаться необходимыми проверками на указанные выше ограничения.

Класс **Процессинговый центр** должен использоваться для поддержки работы класса **Кредит** и может быть разработан в минимально-необходимом варианте.

5.4. Касса в магазине

Разработать классы **Касса** и **Склад**.

Класс **Касса** должен имитировать работу кассового аппарата по сканированию товаров и формированию чека за покупку. Каждый товар идентифицируется штрих-кодом (для упрощения – строка из четырех цифр от «0001» до «9999»). Один и тот же товар может сканироваться несколько раз, но в чек информация о каждом товаре входит в виде «наименование – стоимость за единицу (для упрощения в рублях без копеек) – количество – общая стоимость за товар». Чек состоит не менее чем из одной записи указанного вида. Чек

дополнительно включает общую стоимость товаров в покупке, суммарную скидку и итоговую сумму к оплате (все в рублях).

База товаров хранится в классе **Склад**. Товар описывается штрих-кодом, наименованием, стоимостью за единицу товара, скидкой в процентах от стоимости. Скидки устанавливаются на каждый товар независимо (в диапазоне от 1 до 50%).

Класс **Касса** должен предоставлять следующие операции: 1) «сканировать» очередной товар, 2) получить описание товара со склада, 3) добавить данные о товаре в чек, 4) сформировать чек за покупку, 5) рассчитать итоговую сумму к оплате, 6) удалить выбранный товар из покупки.

Класс **Склад** должен использоваться для поддержки работы класса **Касса** и может быть разработан в минимально-необходимом варианте.

5.5. Билетная касса

Разработать классы **Билетная касса** и **Кинотеатр**.

Класс **Билетная касса** должен имитировать работу кассы по продаже билетов на киносеансы в многозальном кинотеатре. Считать, что продажа билетов проводится на сеансы в пределах трех дней от текущей даты. Каждый сеанс описывается датой, временем начала сеанса, названием фильма, номером зала, стоимостью билета в зависимости от зоны (VIP и обычная). Для упрощения считать, что покупатель указывает тип зоны и требуемое число билетов, а места выделяются кассой автоматически (при наличии свободных). Зрительные места в каждом зале описываются номером ряда и номером в ряду. Для упрощения считать, что число мест во всех рядах в одном зале одинаково. Продажа билетов на сеанс прекращается через 10 минут после начала сеанса.

Информация о всех сеансах на ближайшие 30 дней проката хранится в классе **Кинотеатр**. Для каждого зала установлена базовая стоимость билетов (на дневные сеансы – от 12.00 до 18.00). Стоимость билетов на утренние сеансы (до 12.00) составляет 75% от базовой, стоимость билетов на вечерние сеансы (после 18.00) – 150% от базовой. Информация о зрительных местах (свободно/занято) в каждом зале на каждом сеансе также хранится в классе **Кинотеатр**.

Класс **Билетная касса** должен предоставлять следующие операции: 1) принять данные покупателя: дату, время сеанса, название фильма, номер зала, тип зоны, число мест, 2) проверить наличие требуемого количества свободных мест в требуемой зоне, 3) зарезервировать требуемое количество мест, 4) рассчитать общую стоимость билетов, 5) отменить заказ билетов, 6) сформировать билеты (каждый билет включает: дату, время сеанса, название фильма, номер зала, номер ряда, номер места в ряду).

Класс **Кинотеатр** должен использоваться для поддержки работы класса **Билетная касса** и может быть разработан в минимально-необходимом варианте.

5.6. Железнодорожная касса

Разработать классы **Железнодорожная касса** и **Горьковская железная дорога**.

Класс **Железнодорожная касса** должен имитировать работу кассы по продаже билетов на поезда Нижний Новгород – Москва. Считать, что продажа билетов проводится на поезда в пределах 30 дней от текущей даты. Считать, что по маршруту Нижний Новгород – Москва курсирует три скоростных поезда «Ласточка», один фирменный и один скорый поезд в сутки в каждом направлении. Все вагоны в поездах «Ласточка» однотипны и содержат по 100 сидячих мест. В фирменном и скором поездах вагоны трех типов: плацкартные (27 верхних, 27 нижних мест), купейные (18 верхних, 18 нижних мест), СВ (18 нижних мест). Число вагонов в поездах «Ласточка» – 8. В фирменном поезде – 2 вагона СВ, 6 купейных вагонов, 4 плацкартных вагона. В скором поезде – 4 купейных вагона, 8 плацкартных вагонов. Поезда идентифицируются

номерами (десять номеров из диапазона от 1 до 100), вагоны – номерами (целые числа от 1 до 12), места – номерами (целые числа от 1 до максимума для данного типа вагона).

Информация о всех поездах и всех проданных билетах хранится в классе **Горьковская железная дорога**. Для каждого поезда, каждого типа вагона и каждого типа места установлена стоимость билета. Считать, что все поезда не делают промежуточных остановок по маршруту.

Класс **Железнодорожная касса** должен предоставлять следующие операции: 1) принять данные покупателя: дату, поезд, тип вагона (если есть выбор), количество билетов каждого возможного вида (если есть выбор), ФИО пассажиров 2) проверить наличие свободных мест по запросу покупателя (при невозможности выдать все билеты в одном вагоне, считать заказ невыполнимым), 3) зарезервировать места, 4) рассчитать общую стоимость билетов, 5) отменить заказ билетов, 6) сформировать билеты (каждый билет включает: дату, номер поезда, номер вагона, номер места, ФИО пассажира, станция отправления, станция прибытия).

Класс **Горьковская железная дорога** должен использоваться для поддержки работы класса **Железнодорожная касса** и может быть разработан в минимально-необходимом варианте.

Лабораторная работа №6

6.1. Быки и коровы

Разработать систему классов и реализовать с ее помощью игру «Быки и коровы».

Требования (правила).

- Играют два игрока (человек и компьютер).
- Игрок выбирает длину загадываемого числа – n.
- Компьютер «задумывает» n-значное число с неповторяющимися цифрами.
- Игрок делает попытку отгадать число – вводит n-значное число с неповторяющимися цифрами.
- Компьютер сообщает, сколько цифр угадано без совпадения с их позициями в загаданном числе (то есть количество коров) и сколько угадано вплоть до позиции в загаданном числе (то есть количество быков).
- Игрок делает попытки, пока не отгадает всю последовательность.

Пример.

- Пусть n = 4.
- Пусть задумано тайное число «3219».
- Игрок ввел число «2310».
- Результат: две «коровы» (две цифры: «2» и «3» — угаданы на неверных позициях) и один «бык» (одна цифра «1» угадана вплоть до позиции).

6.2. Змейка

Разработать систему классов и реализовать с ее помощью игру «Змейка».

Требования (правила).

- Играет один игрок (человек), управляющий «змейкой».
- Игра идет на прямоугольном поле N x M клеток. Поле ограничено «стенами» так что вместе со стенами размер поля – (N + 2) x (M + 2) клеток.
- При старте игры змейка имеет длину 5 клеток, форму в виде горизонтального отрезка и располагается в произвольном месте поля, не пересекая и не касаясь стен.
- При старте игры «голова» змейки располагается слева, «хвост» справа. Голова змейки окрашена в цвет, отличный от цвета остальных клеток ее тела.
- При старте игры в произвольной клетке поля (не совпадающей с клетками, занятыми змейкой) возникает «пища».
- При запуске игрового процесса (по специальной команде или автоматически при старте игры) змейка начинает автоматическое движение влево с некоторой заданной скоростью.
- Движение заключается в том, что за каждый такт голова змейки перемещается на одну клетку в текущем направлении движения, а клетка, в которой располагался хвост, становится пустой.
- Игрок может сменить направление движения змейки с помощью клавиш-«стрелок» (вверх, вниз, влево, вправо).
- Если на текущем такте движения голова змейки должна будет занять клетку стены или клетку, которая уже занята любой из клеток ее тела, игра прекращается и считается проигранной.

- Задача игрока вырастить змейку до заданного при старте игрового процесса размера. Змейка вырастает в длину на одну клетку (с хвоста) при каждом поглощении пищи, т.е. в тот момент, когда ее голова на очередном такте движения занимает клетку, в которой расположена пища. На этом же такте в произвольном месте игрового поля (не совпадающей с клетками, занятыми змейкой) снова появляется пища.
- Если змейка выросла до заданной при старте игрового процесса длины, игра считается выигранной.

6.3. Морской бой

Разработать систему классов и реализовать с ее помощью игру «Морской бой».

Требования (правила).

- Играют два игрока (человек и компьютер).
- У каждого игрока два поля 10х10 клеток. В левом поле игрок расставляет свои корабли. В правом игрок пытается потопить чужие корабли.
- У каждого игрока имеются 4 «однопалубных» (из одной клетки) корабля, 3 «двухпалубных», 2 «трехпалубных» и 1 «четырехпалубный» корабль.
- Многопалубные корабли могут располагаться только по горизонтали или вертикали.
- Корабли не могут располагаться в соседних клетках. Соседними для каждой клетки считаются 8 окружающих ее клеток.
- Игра состоит из поочередных ходов игроков.
- Первый ход выполняет человек.
- Каждый ход состоит из следующих действий:
    - Игрок, выполняющий ход, «называет» выбранную клетку (координаты).
    - Соперник проверяет «попадание» на своей доске. Если в названной клетке расположен корабль, соперник оглашает попадание, иначе промах.
    - Игрок, выполняющий ход, ставит на своей правой доске по названным координатам отметку о результатах хода.
    - Если игрок, выполняющий ход, попал в корабль, ход остается у него, иначе переходит к сопернику.
- Выигрывает тот игрок, кто первым потопит все корабли противника.