

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И КОМПЬЮТЕРНЫХ СИСТЕМ**

**Н.В. ЛЕВКОВИЧ
Н. В. СЕРИКОВА**

ЗАДАНИЯ ПО КУРСУ

«ОСНОВЫ ООП»

ВАРИАНТ В

**2021
МИНСК**

ОГЛАВЛЕНИЕ

1. Файлы	4
1.1. Текстовые файлы чисел	4
1.2. Бинарные файлы чисел	5
1.3. Бинарные файлы записей	5
1.4. Текстовые файлы записей	5
2. Классы.....	6
2.1. Класс «Вектор»	6
2.2. Класс «Линейный список».....	7
2.3. Класс «Двусвязный список».....	7
2.4. Перегрузка операций.....	8
2.5. Класс «Бинарное дерево».....	9
2.6. Класс «Многоразрядное число».....	10
2.7. Класс «Матрица»	12
3. Библиотека шаблонов STL	13
3.1. Строки string.....	13
3.2. Вектор	13
3.3. Список.....	13
3.4. Вычисление выражений с использованием обратной польской записи	13
3.5. Ассоциативные списки.....	14
4. Наследование. Полиморфизм.....	15
4.1. Наследование	15
4.2. Полиморфизм. Виртуальные методы	28
4.3. «Фабрика класса»	28
4.4. Полиморфизм. Виртуальные методы	28

16 занятий (64 час.)

оценка	количество задач
7	12
8	15
9	19
10	20

№	тема	№ задач			
		7	8	9	10
1	1. Файлы	1.1			
2		1.2			
3		1.3	1.4		
4	2. Классы	2.1			
5		2.2			
6		2.3	2.5		
7		2.4		2.6	
8				2.7	
9	3. Библиотека STL	3.1			
10		3.2			
11		3.3		3.4	
12					3.5
13	4. Наследование	4.1			
14		4.2			
15				4.3	
16			4.4		

1. ФАЙЛЫ

1.1. ТЕКСТОВЫЕ ФАЙЛЫ ЧИСЕЛ

Выполнить задания для **текстового** файла. Задания выполнить через функции. **Размер файлов $\leq 64\text{GiB}$.**

1. В файлах f и g содержатся вещественные числа, упорядоченные по возрастанию. Получить за один проход файл чисел, упорядоченный по возрастанию, путем слияния файлов f и g .

2. Компоненты файла f – целые числа. Получить файл g из чисел исходного файла, в котором записаны сначала все положительные числа, затем все отрицательные.

3. Компоненты файла f – целые числа, четных чисел столько же, сколько нечетных. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел одинаковой четности.

4. В файлах f и g содержатся вещественные числа, упорядоченные по убыванию. Получить за один проход файл чисел, упорядоченный по убыванию, путем слияния файлов f и g .

5. Компоненты файла f – целые числа, положительных чисел столько же, сколько отрицательных. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел с одинаковым знаком.

6. Компоненты файла f – целые числа, причем положительных чисел столько же, сколько отрицательных. Получить файл g из чисел исходного файла, в котором записаны 2 положительных числа, затем 2 отрицательных и т. д.

7. Компоненты файла f – целые числа, причем четных чисел столько же, сколько нечетных. Получить файл g из чисел исходного файла, в котором записаны 2 четных числа, затем 2 нечетных т. д.

8. В файлах f и g содержатся вещественные числа, упорядоченные по возрастанию. Получить за один проход файл чисел, встречающихся в обоих исходных файлах без повторений.

9. Компоненты файла f – целые числа. Получить файл g из чисел исходного файла, в котором записаны сначала все четные числа, затем все нечетные числа.

10. Компоненты файла f – положительные вещественные числа. Получить файл g из чисел исходного файла, в котором записаны сначала все числа больше 1, затем остальные числа.

11. В файлах f и g содержатся вещественные числа, упорядоченные по возрастанию. Получить за один проход файл чисел, встречающихся в файле f и не встречающихся в файле g .

12. Компоненты файла f – положительные вещественные числа, причем чисел $> \sqrt{2}$ столько же, сколько и меньших этого значения. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел $> \sqrt{2}$.

1.2. БИНАРНЫЕ ФАЙЛЫ ЧИСЕЛ

Выполнить задания 1.1 для бинарного файла. Задания выполнить через функции. Размер файлов $\leq 64\text{GiB}$.

1.3. БИНАРНЫЕ ФАЙЛЫ ЗАПИСЕЙ

Написать программу для работы с базой данных, содержащей записи со сведениями о студентах: ФИО, возраст, пол, курс, успеваемость, в которой должны выполняться следующие действия: создание файла, просмотр файла

Выполнить задания для бинарного файла. Размер файлов $\leq 64\text{GiB}$.

1. По исходному файлу определить ФИО самого младшего студента на каждом курсе. Сформировать файл из этих студентов.
2. По исходному файлу определить ФИО самого старшего студента на каждом курсе. Сформировать файл из этих студентов.
3. По исходному файлу определить всех студентов n курса, ФИО которых начинается на заданную букву. Сформировать файл из этих студентов.
4. По исходному списку определить всех студентов n курса, возраст которых превышает заданный. Сформировать файл из этих студентов.
5. По исходному файлу определить всех студентов n курса. Сформировать файл из этих студентов.
6. По исходному файлу определить всех отличников. Сформировать файл из этих студентов.
7. По исходному файлу определить всех отличников n курса. Сформировать файл из этих студентов.
8. По исходному файлу определить всех неуспевающих студентов. Сформировать файл из этих студентов.
9. По исходному файлу определить всех неуспевающих студентов n курса. Сформировать файл из этих студентов.
10. По исходному файлу определить студентов, имеющих средний балл успеваемости выше общего среднего бала. Сформировать файл из этих студентов.
11. По исходному файлу определить студентов на n курсе, имеющих средний балл успеваемости выше среднего бала по его курсу. Сформировать файл из этих студентов.
12. По исходному файлу определить студентов на n курсе, имеющих средний балл успеваемости ниже среднего бала по его курсу. Сформировать файл из этих студентов.

1.4. ТЕКСТОВЫЕ ФАЙЛЫ ЗАПИСЕЙ

Выполнить задания 1.3 для текстового файла. Размер файлов $\leq 64\text{GiB}$.

2. КЛАССЫ

2.1. КЛАСС «ВЕКТОР»

Разработать класс «вектор», моделирующий математическое понятие «одномерный массив» произвольного размера с возможностью изменения числа элементов.

Реализовать три вида конструктора (без аргументов, инициализации, копирования), деструктор.

Определить методы:

- задания размерности вектора,
- определения количества элементов заданного вектора;
- заполнения элементов вектора случайными числами.

Реализовать:

- перегрузку операторов индексирования `[]`, возвращающие изменяемое значение элемента (L-value) и не изменяемое (R-value) с контролем выхода за пределы размерности вектора (для получения элемента вектора по заданному индексу, занесения значения элемента вектора по заданному индексу),
- перегрузку операторов ввода/вывода значений элементов вектора,

Разместить описание класса в заголовочном файле, а определения методов и главную функцию программы – в отдельных файлах.

Используя объекты описанного класса «Вектор», реализовать алгоритм своего варианта задания 2.1 «Учебной вычислительной практики».

2.2. КЛАСС «ЛИНЕЙНЫЙ СПИСОК»

Объявить шаблон класса «Линейный список». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *добавления в структуру нового элемента;*
- *удаление элемента;*
- *поиск элемента по заданному ключу поиска (значению информационного поля);*
- *вывод элементов списка на экран;*
- *удаления списка.*

Используя объекты класса «Линейный список», реализовать алгоритмы своих вариантов заданий **2.4 «Учебной вычислительной практики».**

2.3. КЛАСС «ДВУСВЯЗНЫЙ СПИСОК»

Объявить шаблон класса «Двусвязный линейный список». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *запись в структуру нового элемента;*
- *удаление элемента;*
- *поиск элемента по заданному ключу поиска (значению информационного поля);*
- *вывод элементов списка на экран;*
- *удаления всех элементов списка.*

Используя объекты класса «Двусвязный линейный список», реализовать алгоритм своего варианта задания **2.6 «Учебной вычислительной практики».**

2.4. ПЕРЕГРУЗКА ОПЕРАЦИЙ

Объявить класс для решения задачи и определить его методы.

Реализовать:

- *три вида конструктора (без аргументов, инициализации, копирования),*
- *деструктор,*
- *перегрузить операции ввода/вывода данных,*
- *указанные в задании операции реализовать посредством перегрузки операций.*

Объявление класса разместить в заголовочном файле, определения методов – во втором файле, определение главной функций программы – в третьем файле. Продемонстрировать корректную работу всех операций и методов реализованного класса.

1. Определить класс для работы с объектом **«множество латинских букв»**. Реализовать: объединение двух множеств, пересечение двух множеств, разность двух множеств, добавление элемента во множество, удаление элемента из множества.

2. Определить класс для работы с объектом **«строка символов»**. Реализовать: сравнение строк, удаление из строки указанного символа. Удалить из заданной строки все символы, встречающиеся во второй заданной строке.

3. Определить класс для работы с объектом **«рациональная дробь»** (вида m/n). Реализовать: сложение, вычитание, умножение, деление двух дробей, приведение дроби к несократимому виду, вывод дроби в виде m/n , сравнение двух дробей.

4. Определить класс для работы с объектом **«полином»**. Ввод осуществить по возрастанию степеней членов полинома. Реализовать: сложение, вычитание, умножение, деление двух полиномов, умножение полинома на число, вычисление значения полинома в заданной точке, дифференцирование полинома.

5. Определить класс для работы с объектом **«комплексное число»**, которое задается своим модулем и углом. Реализовать: сложение, вычитание, произведение, деление двух чисел, возведение в целочисленную степень, извлечение квадратного корня.

6. Определить класс для работы с объектом **«строка символов»**. Реализовать: вставку в строку указанного символа в заданную позицию, удаление ведущих пробелов. Добавить в заданную строку отсутствующие символы из второй заданной строки, на позиции с такими же номерами.

7. Определить класс для работы с объектом **«множество букв кириллицы»**. Реализовать: объединение двух множеств, пересечение двух множеств, разность двух множеств, добавление элемента во множество, удаление элемента из множества.

8. Определить класс для работы с объектом **«строка символов»**. Реализовать: замену подстроки на подстроку, конкатенацию строк. Заменить в заданной строке заданную подстроку на другую заданную подстроку (столько раз, сколько она встречается в тексте).

9. Определить класс для работы с объектом **«множество игральных карт»** (колода 52 карты, 4 масти, 13 достоинств: «2»-«10», валет, дама, король, туз). Реализовать: объединение двух множеств, пересечение двух множеств, разность двух множеств, добавление элемента во множество, удаление элемента из множества. Замечание: не важно, как карты хранятся внутри класса – пользователь видит только ввод с клавиатуры и вывод на экран.

10. Определить класс для работы с объектом **«время»**, которое задается в формате час.минута.секунда. Реализовать (учесть переход через 24 ч.): добавления ко времени заданного количества секунд, вычитания из времени заданного количества секунд, сложения двух моментов времени, вычитание из одного момента времени другого, подсчет числа секунд между двумя моментами времени, лежащими в пределах одних суток.

11. Определить класс для работы с объектом **«строка символов»**. Реализовать: удаление подстроки из строки, проверку, является ли строка палиндромом. Удалить из заданной строки заданную подстроку (столько раз, сколько она встречается в тексте).

12. Определить класс для работы с объектом **«дата»**, которое задается в формате число.месяц.год. Реализовать (учесть високосные года): сложение даты с заданным количеством дней, вычитания из даты заданного количества дней, вычисление числа дней, прошедших между двумя датами.

2.5.КЛАСС «БИНАРНОЕ ДЕРЕВО»

Объявить шаблон класса «Бинарное дерево». Определить структуру класса и его методы. В классе должен быть конструктор по умолчанию, конструктор с формальными параметрами, конструктор копирования и деструктор.

Определить методы:

- *инициализация первого элемента структуры;*
- *запись в структуру нового элемента;*
- *удаление элемента (листа);*
- *поиск элемента по заданному ключу поиска (значению информационного поля);*
- *вывод элементов дерева на экран;*
- *удаления всех элементов дерева.*

Используя объекты класса «Бинарное дерево», реализовать алгоритм своего варианта задания 2.7 **«Учебной вычислительной практики»**.

2.6. КЛАСС «МНОГОРАЗРЯДНОЕ ЧИСЛО»

Объявить класс «многоразрядное целое число». Количество разрядов числа хранить в поле класса, значения разрядов числа хранить в динамической памяти, основание системы счисления задавать константой.

Реализовать:

- три вида конструктора (без аргументов, инициализации, копирования),
- деструктор,
- перегрузить операции ввода/вывода данных.
- перегрузить оператор индексирования [] для доступа к разрядам числа,
- операции в задании реализовать посредством перегрузки.

Объявление класса разместить в заголовочном файле, определения методов – во втором файле, определение главной функций программы – в третьем файле.

При переполнении старшего разряда лишние биты теряются.

Примечание.

Оператор умножения многоразрядных чисел удобно реализовать через более простые перегруженные операторы: умножение многоразрядного числа на одноразрядное число, сдвиг многоразрядного числа на целое число разрядов, сложение многоразрядных чисел.

Оператор деления многоразрядных чисел удобно реализовать через более простые перегруженные операторы: сдвиг многоразрядного числа на целое число разрядов, сравнение и вычитание многоразрядных чисел.

Выполнить задание, используя объекты этого класса.

1. Объявить класс «**32-разрядное целое положительное десятичное число**». Реализовать операции: сложения, умножения этих чисел. Вычислить $(1787109376_{10})^3$.
2. Объявить класс «**64-разрядное целое двоичное число со знаком**». Реализовать операции: сложения, умножения, операцию возведения в степень. Вычислить $3^{12} + (-3)^{11} + 3^{10} + (-3)^9$.
3. Объявить класс «**256-разрядное целое положительное двоичное число**». Реализовать операции: вычитания, сдвигов на n разрядов, целочисленного деления, получения остатка от деления. Создать метод для перевода «256-разрядного целого двоичного числа» в десятичное число (представленное в виде строки). Перевести в десятичную систему счисления число 2^{200} .
4. Объявить класс «**64-разрядное целое положительное двоичное число**». Реализовать операции: сложения, умножения, операцию возведения в степень. Вычислить $3^{12} + 3^{11} + 3^{10} + 3^9$.
5. Объявить класс «**64-разрядное целое положительное десятичное число**». Реализовать операции: сложения, умножения этих чисел. Вычислить $(8212890625_{10})^5$.

6. Объявить класс «**256-разрядное целое положительное двоичное число**». Реализовать операции: вычитания, сдвигов на n разрядов, целочисленного деления, получения остатка от деления. Найти НОД чисел $2^{249} - 2^6 + 7$ и $2^{108} - 1$.
7. Объявить класс « **n -разрядное целое положительное число в 10^9 -ричной с/с**» (каждый разряд числа представляет девять десятичных разрядов). Реализовать операции: сложения, умножения. Вычислить $100_{10}!$.
8. Объявить класс « **n -разрядное целое положительное число в 2^{32} -ричной с/с**» (каждый разряд числа представляет тридцать два двоичных разряда(int)). Реализовать операции: сложения, умножения. Вычислить $(5_{16})^{100}$. Ввод с клавиатуры и вывод на экран осуществить в шестнадцатеричной с/с.
9. Объявить класс « **n -разрядное целое положительное число в 100-ричной с/с**» (каждый разряд числа представляет два десятичных разряда). Реализовать операции: сложения, умножения. Вычислить $100_{10}!$.
10. Объявить класс « **n -разрядное целое положительное число в 10-ричной с/с**». Реализовать операции: сложения, умножения. Вычислить $100_{10}!$.
11. Объявить класс « **n -разрядное целое положительное число в 16-ричной с/с**». Реализовать операции: сложения, умножения. Вычислить $(5_{16})^{100}$. Ввод с клавиатуры и вывод на экран осуществить в шестнадцатеричной с/с.
12. Объявить класс « **n -разрядное целое положительное число в 256-ричной с/с**» (каждый разряд числа представляет два шестнадцатеричных разряда). Реализовать операции: сложения, умножения. Вычислить $(5_{16})^{100}$. Ввод с клавиатуры и вывод на экран осуществить в шестнадцатеричной с/с.

2.7.КЛАСС «МАТРИЦА»

Объявить класс «Матрица» и определить его методы.

Обеспечить представление матрицы произвольного размера с возможностью изменения числа строк и столбцов. **Каждая строка матрицы – объект класса «вектор» из задания 2.1.**

Обеспечить в конструкторе автоматическое генерирование имени матрицы («матрица 1», «матрица 2» и т.д.), используя для этого статическое поле – счетчик объектов класса.

Реализовать в классе метод вывода матрицы на экран с одновременным выводом ее имени. Алгоритм обработки, реализующий условие задачи, определить как функцию-член класса или как дружественную функцию класса.

Перегрузить оператор **индексирования []** для обращения к элементам матрицы с контролем выхода за пределы размерностей.

В случае выхода за пределы массива генерировать исключения с помощью ключевого слова **throw**.

Класс может реализовывать следующие операции над матрицами (в зависимости от условия задания):

- сложение, вычитание, умножение, деление на число
- сложение, вычитание, умножение на другую матрицу
- комбинированные операции присваивания ($+=$, $-=$, $*=$, $/=$);
- операции сравнения на равенство/неравенство;
- операции вычисления транспонированной матрицы;
- операцию возведения в степень;
- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметричная, верхняя треугольная, нижняя треугольная);
- операции ввода/вывода матрицы в стандартные потоки.

Обеспечить работу с безопасным массивом, т.е. контролировать выход индекса элемента массива за допустимый описанием объекта диапазон.

Используя объекты описанного класса «Матрица», реализовать алгоритм своего варианта задания 2.2 «Учебной вычислительной практики».

3. БИБЛИОТЕКА ШАБЛОНОВ STL

3.1. СТРОКИ STRING

Выполнить задания 4.1, 4.2, 4.3 курса «Основы и методологии программирования», используя string-строки для представления данных. Задания выполнить через функции.

3.2. ВЕКТОР

Реализовать задания 2.1, 2.2 «Учебной вычислительной практики», используя объявление и методы параметризованного класса **vector** из стандартной библиотеки шаблонов STL.

3.3. СПИСОК

Реализовать варианты заданий 2.4, 2.6 «Учебной вычислительной практики», используя объявление и методы параметризованного класса **list** из стандартной библиотеки шаблонов STL.

3.4. ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЙ

Выполнить задания с использованием подходящих связанных динамических структур данных. Оценить асимптотическую сложность алгоритма.

1. Написать программу, которая по заданному *алгебраическому* выражению (операнды – *целые числа*; операции: $- + * / \%$ и скобки $()$) вычисляет результат.
2. Написать программу, которая по заданному *алгебраическому* выражению (операнды – *целые и вещественные числа*; операции: $- + * /$ и скобки $()$) вычисляет результат.
3. Написать программу, которая по заданному *логическому* выражению (операнды – *целые числа и булевские константы*; операции отношения: $> < == != >= <=$, логические операции: $\&\& || !$ и скобки $()$) вычисляет результат.
4. Написать программу, которая по заданному выражению вычисляет результат (операнды – *целые числа*; битовые операции: $\& | ^ \sim >> <<$ и скобки $()$).

3.5. АССОЦИАТИВНЫЕ СПИСКИ

Дан текст программы на C++. Проанализировать программу за один просмотр и записать в выходной текстовый файл:

1. все **ключевые слова** в алфавитном порядке, встречающиеся в этой программе, указав количество появлений для каждого ключевого слова и номера строк программы, в которых они встречаются;
2. все **идентификаторы** в алфавитном порядке, встречающиеся в этой программе, указав количество появлений для каждого идентификатора и номера строк программы, в которых они встречаются;
3. все **неименованные константы**, встречающиеся в этой программе, указав количество появлений для каждой константы и номера строк программы, в которых они встречаются;
4. **имена всех определенных в этой программе функций** в алфавитном порядке, указав для каждой из них номера строк программы, в которых они вызываются;
5. все **циклы**, встречающиеся в этой программе, указав номера строк программы, определяющих начало и конец каждого цикла;
6. все **условные операторы**, встречающиеся в этой программе, указав номера строк программы, определяющих начало и конец каждого оператора.

Примечание.

Предусмотреть условие, при котором искомая информация не должна анализироваться в комментариях и строковых константах.

4. НАСЛЕДОВАНИЕ. ПОЛИМОРФИЗМ

4.1. НАСЛЕДОВАНИЕ

В предложенных ниже вариантах описывается конкретная задача. Для неё:

1. *Определить и описать иерархию классов: набор схожих по какому-либо критерию объектов, объединённых в подгруппы (родительские объекты).*

Для каждого класса описать поля, характеризующие объект, и методы, позволяющие заполнять и получать значения каждого из полей, выводить значения полей на экран, а также необходимые конструкторы. Общие свойства хранить в родительских классах.

- *Конструкторы и методы должны проверять параметры на допустимость.*
- *Вывод свойств родительских классов осуществить с использованием наследования и полиморфизма (виртуальный метод **Print()**, который вызывается из **operator<< базового класса**).*

Каждый класс должен уметь выводить в виде текста свой тип и набор своих свойств.

2. *Обсудить схему с преподавателем.*

3. *Реализовать иерархию в коде.*

4. *Создать программу для тестирования описанных объектов, создающую и заполняющую свойства нескольких отдельных объектов, а затем выводящую свойства всех созданных объектов на экран. Первичную инициализацию свойств объектов осуществите с помощью конструкторов.*

Примечание 1: *чтобы позднее такую иерархию классов было удобнее расширять, начинайте её с абстрактного базового класса.*

Примечание 2: *во всех вариантах предполагается не менее 3 уровней в иерархии наследования. Где удобно - добавляйте в иерархию не указанные в задании абстрактные или обычные классы.*

Примечание 3: *использование С-строк обязательно (чтобы ошибки в использовании «правила трёх» стали явными).*

Вариант 1

Вам заказали спроектировать программу для учёта банковских вкладов.

Разработайте иерархию наследования классов, позволяющую хранить информацию о всех депозитных счетах и их свойствах в программе:

1. Вклад до востребования (бессрочный вклад)

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов

2. Условный вклад (отличается от вклада «до востребования» наличием определённого условия для его закрытия)

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов
- условие закрытия вклада (срока, например, "до вступления в брак", "до достижения совершеннолетия", или другие условия из завещаний)

3. Срочный безотзывной вклад

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов
- срок вклада в месяцах
- период в течении которого допускаются дополнительные взносы во вклад (количество месяцев)

4. Срочный отзывной вклад

- номер счёта
- дата создания вклада
- годовая процентная ставка
- текущая сумма
- дата последнего начисления процентов
- срок вклада в месяцах
- годовая процентная ставка(пониженная) в случае досрочного расторжения договора
- - начальная сумма (для пересчёта процентов по пониженной ставке)

Вариант 2

Вы проектируете многопользовательскую 2D RPG игру с видом от третьего лица. В одной локации может находиться множество неподвижных предметов и игроков. Для реализации прототипа игры вам надо разработать иерархию наследования классов, позволяющую хранить все эти сущности и их свойства в программе. А именно:

1. Камень

- координаты X и Y (целые)
- размеры - ширина и высота (целые беззнаковые)
- идентификатор картинки, отображаемой на месте предмета (число от 1 до 10000)

2. Сундук

- координаты X и Y (целые)
- размеры - ширина и высота (целые беззнаковые)
- количество золотых монет в сундуке

3. Торговец

- текущие координаты X и Y (целые)
- имя
- код типа продаваемых товаров (число от 1 до 100)

4. Мирный житель

- текущие координаты X и Y (целые)
- имя

5. Игроки

- текущие координаты X и Y (целые)
- имя
- цвет одежды

Вариант 3

Вы проектируете 3D RPG игру на космическую тему.

В на игровой карте могут находиться одновременно разные сущности. Для реализации первой версии игры вам необходимо разработать иерархию наследования классов для представления в программе следующего минимального набора сущностей и их свойств:

1. Неподвижные (система координат в каждый момент времени отсчитывается относительно звезды или ближайшей планеты): звезда, планета

- название
- температура
- наличие твёрдой поверхности, на которую можно высадиться

2. Астероид (подвижный мелкий объект)

- координаты (X, Y, Z)
- скорость (V_x , V_y , V_z)
- максимальный размер в поперечнике

3. Космический корабль (под управлением игрока, или враждебный)

- имя
- координаты (X, Y, Z)
- скорость (V_x , V_y , V_z)
- максимальная скорость V_{\max}
- боезапас (сколько осталось выстрелов)

4. Ракета (выпущенная одним из кораблей)

- координаты (X, Y, Z)
- скорость (V_x , V_y , V_z)
- мощность взрывчатки
- время, на которое остался запас топлива (в секундах)

Вариант 4

Вы проектируете программу для оформления бухгалтерской отчётности факультета РФиКТ. Разработайте иерархию наследования классов, объединяющую всех людей на факультете, и позволяющую хранить всю необходимую информацию о них.

Категории сущностей и свойства, которые должны присутствовать в программе:

1. Студент

- ФИО
- год обучения
- номер студенческого билета (число до 9 цифр)
- средний балл

2. Магистрант

- ФИО
- год обучения
- номер студенческого билета (число до 9 цифр)
- средний балл

3. Аспирант

- ФИО
- год обучения
- номер студенческого билета (число до 9 цифр)
- средний балл

4. Преподаватель

- ФИО
- должность (строка)
- учёная степень (строка)

5. Учебно-вспомогательный персонал

- ФИО
- должность (строка)
- номер лаборатории (число от 1 до 100)

Вариант 5

Вы проектируете графический пользовательский интерфейс для новой операционной системы. Разработайте иерархию наследования классов для представления элементов управления и их свойств. Минимальный набор элементов, необходимый для реализации прототипа системы:

1. Ellipse – рисует вписанный в рамки элемента управления эллипс

- Visible - отображается элемент управления или скрыт (bool)
- FillColor – цвет заливки
- BorderColor – цвет контура
- Координаты X и Y
- Размеры высота и ширина элемента управления

2. HyperlinkLabel – при нажатии мышкой обеспечивает переход по гиперссылке

- Text - текст в текстовом поле
- Color - цвет текста элемента управления
- Visible - отображается элемент управления или скрыт (bool)
- Координаты X и Y
- Размеры высота и ширина элемента управления
- URL – адрес ссылки для перехода (строка)

3. TextBox - текстовое поле, позволяет пользователю вводить текст

- Text - текущий текст в текстовом поле
- Color - цвет текста элемента управления
- Readonly – запрещает редактирование текста пользователем (bool)
- Visible - отображается элемент управления или скрыт (bool)
- Координаты X и Y
- Размеры высота и ширина элемента управления

4. Button – кнопка

- Text - текущий текст на кнопке
- ToolTipText - текст, появляющийся в подсказке при наведении мышки на кнопку
- Visible - отображается элемент управления или скрыт (bool)
- Координаты X и Y
- Размеры высота и ширина элемента управления

Вариант 6

Вас пригласили поучаствовать в разработке системы видеонаблюдения для национального аэропорта «Минск». Система будет обладать элементами искусственного интеллекта, а именно: она будет выделять для каждого кадра в видеопотоке отдельные сущности и классифицировать их. В данный момент вам необходимо разработать иерархию наследования классов для представления результатов работы модуля распознавания – набора обнаруженных в кадре сущностей. Далее приведены сущности (и их свойства), которые выдаёт первая версия программы:

1. Человек

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- пол (М, Ж, не удалось определить)
- ребёнок (да, нет, не удалось определить)
- наличие очков (есть, нет, ракурс не позволяет определить)
- наличие бороды (есть, нет, ракурс не позволяет определить)

2. Велосипедист

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- пол (М, Ж, не удалось определить)
- ребёнок (да, нет, не удалось определить)
- наличие очков (есть, нет, ракурс не позволяет определить)
- наличие бороды (есть, нет, ракурс не позволяет определить)
- цвет велосипеда (строка)

3. Автомобиль

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- цвет
- номер (строка)
- такси (да, нет)

4. Бесхозная сумка

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- время суток, когда была первый раз замечена

Вариант 7

Краеведческий музей в деревне Простоквашино нуждается в современной системе учёта экспонатов. Разработайте иерархию наследования классов, которая будет использована в такой системе.

Следующие виды экспонатов хранятся в настоящий момент в музее (также приведены важные для учёта свойства):

1. Монеты

- наименование (строка)
- номинал (может быть как целым, так и долями от целого)
- годы использования (диапазон лет – два целочисленных года)

2. Холодное оружие

- наименование (строка)
- приблизительная датировка (диапазон лет – два целочисленных года)

3. Картины

- что изображено (строка)
- автор (строка)
- время создания (год)

4. Статуэтки

- что изображено (строка)
- автор (строка)
- время создания (год)

5. Фотографии

- что изображено (строка)
- время создания (год)

Вариант 8

Компания, имеющая большой товарный оборот на собственных складских площадях, и, ведущая вручную весь учёт, озабочена автоматизацией процесса движения товара на складах. Вам необходимо разработать иерархию наследования классов, объединяющую все товары, хранимые на складе. Эти классы будут использоваться для реализации функции складского учёта (постановка на учёт прибывающих товаров и снятие товаров с учёта при отправке заказов)

Виды товара, хранящиеся на складе в настоящий момент, и свойства, которые необходимо хранить в системе учёта:

1. Бытовая техника

- Наименование (строка)
- производитель (строка)
- цена
- остаток на складе (штук)

2. Верхняя одежда

- наименование (строка)
- размер
- материал (строка)
- сезон (строка)
- цена
- остаток на складе (штук)

3. Спортивная одежда

- наименование (строка)
- размер
- материал (строка)
- цена
- остаток на складе (штук)

4. Нижнее бельё

- наименование (строка)
- размер
- материал (строка)
- цена
- остаток на складе (штук)

Вариант 9

Одна известная транспортная компания планирует внедрить у себя на сайте функционал по построению маршрута из любой точки Минска в любую. Вам поручено разработать иерархию наследования классов для представления в программе сегментов такого маршрута. Каждый сегмент представляет часть маршрута, преодолеваемую на одном транспорте без пересадок. Виды транспорта и дополнительная информация, которую необходимо хранить в первой версии программы:

1. Автомобиль

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость аренды в рублях
- ссылка на страницу в интернет с арендой авто (строка)

2. Автобус

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

3. Троллейбус

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

5. Метро

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

6. Маршрутное такси

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время поездки в минутах
- стоимость проезда

7. Пешком

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- приблизительное время перемещения в минутах.

Вариант 10

Вас пригласили поучаствовать в создании программы 2D редактора векторной графики. Вам необходимо разработать иерархию наследования классов для представления в программе набора графических примитивов. Для первой версии продукта был отобран следующий минимальный набор фигур и их свойств:

1. Прямоугольник

- координаты X, Y левого верхнего угла
- размеры ширина и высота
- цвет (строка)

2. Эллипс

- координаты X, Y левого верхнего угла описывающего прямоугольника
- размеры ширина и высота
- цвет (строка)

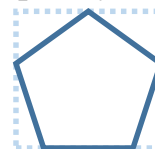
3. Звезда

- координаты X, Y левого верхнего угла описывающего прямоугольника
- размеры ширина и высота
- цвет (строка)
- количество вершин



4. Многоугольник

- координаты X, Y левого верхнего угла описывающего прямоугольника
- размеры ширина и высота
- цвет (строка)
- количество вершин



5. Текст

- координаты позиции с которой выводится текст X, Y
- размер шрифта в пт (float)
- текст (строка)
- цвет (строка)

Вариант 11

Вас пригласили поучаствовать в разработке многопользовательской онлайн-вой ролевой игры (MMORPG). Перед вами стоит задача разработать иерархию наследования классов для хранения в игре информации о наборе заданий на доске объявлений. Для минимальной рабочей версии игры вам предстоит обеспечить представление в игре следующих категорий заданий и их свойств:

1. Задание на отстрел монстров определённого вида в заданной локации (например, «крысы расплодились в подвале»)

- художественно оформленный текст задания (строка)
- количество золотых монет, получаемых за выполнение задания
- идентификатор локации, где обитают монстры
- идентификатор типа монстров
- количество монстров указанного типа, которое необходимо уничтожить

2. Задание на отстрел конкретного монстра

- художественно оформленный текст задания (строка)
- количество золотых монет, получаемых за выполнение задания
- идентификатор локации, где обитает монстр
- идентификатор типа монстра
- титул, присваиваемый победителю этого монстра (строка)

3. Заказ на доставку («принеси мне то, не знаю что...»)

- количество золотых монет, получаемых за выполнение задания
- идентификатор локации расположения предмета
- идентификатор локации расположения заказчика
- идентификатор заказываемого предмета
- отметка времени, до которой должна быть завершена доставка

4. Задание на охрану каравана

- количество золотых монет, получаемых за выполнение задания
- идентификатор локаций – начала маршрута
- идентификатор локаций – конец маршрута
- отметка времени, до которой караван должен прибыть в место назначения
- расчётное время движения каравана по маршруту в секундах

Примечания: Идентификатор локации - целое число от 1 до 10^6 .

- Идентификатор типа монстра - целое число от 1 до 200.
- Идентификатор предмета - целое число от 1 до 10^4 .
- Время измеряется в реальных секундах.
- Моменты времени хранятся в виде количества секунд с начала Unix эпохи (полночь первого января 1970 года).

Вариант 12

Для учёта имеющейся компьютерной техники на факультете РФиКТ необходимо разработать программу инвентаризации. В рамках этого проекта вам необходимо разработать иерархию наследования классов для представления параметров каждого компьютера. Классы разрабатываемой иерархии должны представлять только существенные для учёта параметры компьютера вместе с их свойствами:

1. HDD

- Ёмкость
- Год выпуска
- Производитель (строка)

2. SSD

- Ёмкость
- TBW, байт
- Фактическое количество записанных на накопитель байт по показаниям S.M.A.R.T. во время последней инвентаризации
- Год выпуска
- Производитель (строка)

3. Монитор

- Диагональ в дюймах
- Рабочее разрешение (ширина и высота)
- Частота обновления экрана
- Год выпуска

4. Процессор

- Год выпуска
- Модель (строка)
- Количество ядер
- Базовая частота, ГГц

Примечание: *TBW (Total Byte Written) - заявленный компанией-производителем параметр – это количество байт данных, которое можно гарантированно записать на диск прежде, чем ресурс чипов памяти будет исчерпан.*

4.2. ПОЛИМОРФИЗМ. ВИРТУАЛЬНЫЕ МЕТОДЫ

Реализовать класс «одномерный динамический массив из потомков-объектов, описанных в 4.1».

Реализовать следующие методы:

- создание массива;
- занесение объекта в массив;
- удаление объекта из массива;
- вывод на экран элементов массива с соответствующими полями для каждого объекта;
- удаление массива.

Определить: конструктор, конструктор копирования, оператор копирования, деструктор, виртуальные методы. Выполнить тестирование написанных методов.

Для сред разработки, поддерживающих стандарт C++11 и выше, определить конструктор перемещения и оператор перемещения.

4.3. «ФАБРИКА КЛАССА»

Реализовать персистентность объекта из задания 4.2, а именно, сохранение массива в файл при выходе из программы и загрузку из файла при запуске.

4.4. ПОЛИМОРФИЗМ. ВИРТУАЛЬНЫЕ МЕТОДЫ

Создать шаблон класса «односвязный список указателей с правом владения» (воспользовавшись шаблоном из задания 2.2). Использовать его для создания контейнера потомков-объектов, описанных в 4.1. Для данного списка реализовать следующие методы:

- инициализация списка;
- добавление объекта в список;
- исключение элемента (с удалением объекта, на который он ссылается);
- просмотр элементов списка и вывод на экран элементов списка с соответствующими полями для каждого объекта;
- удаление всех элементов, хранящихся в списке (с удалением объектов).
- создание копии списка (с копиями объектов).

Определить: конструктор, конструктор копирования, оператор копирования, деструктор, виртуальные методы. Выполнить тестирование написанных методов.

Для сред разработки, поддерживающих стандарт C++11 и выше, определить конструктор перемещения и оператор перемещения списка.