

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

**Н. В. ЛЕВКОВИЧ
Н. В. СЕРИКОВА**

**ЗАДАНИЯ ПО КУРСУ
«ПРОГРАММИРОВАНИЕ НА C++»**

ВАРИАНТ А

**2024
МИНСК**

ОГЛАВЛЕНИЕ

3. Файлы	4
3.1. Текстовый файл чисел	4
3.2. Бинарный файл чисел	5
3.3. Бинарный файл структур.....	5
3.4. Текстовый файл структур.....	5
4. Классы	6
4.1. Класс «Слово»	6
4.2. Класс «Студент».....	7
4.3. Класс «База данных о Студентах».....	8
4.4. Класс «Вектор».....	8
4.5. Перегрузка операций	10
4.6. Перегрузка операций	10
5. Библиотека шаблонов STL.....	11
5.1. Строки string	11
5.2. Строки string	11
5.3. Строки string	11
5.4. Вектор.....	12
5.5. Матрица.....	13
5.6. Стек.....	13
5.7. Очередь	14
6. Наследование.....	15
6.1. Наследование.....	15
6.2. Полиморфизм. Виртуальные методы	28
6.3. «Фабрика класса»	28

16 занятий (64 час.)

оценка	количество задач
4	10
5	15
6	17
7	20

№	тема	№ задач			
		4	5	6	7
1	<u>3. Файлы</u>	<u>3.1</u>			
2		<u>3.2</u>	<u>3.3</u>		
3				<u>3.4</u>	
4	<u>4. Классы</u>	<u>4.1</u>			
5		<u>4.2</u>			
6		<u>4.4</u>			
7			<u>4.3</u>		
8			<u>4.5</u>		
9					<u>4.6</u>
10	<u>5. Библиотека STL</u>	<u>5.1</u>			
11		<u>5.2</u>	<u>5.3</u>		
12		<u>5.4</u>	<u>5.6</u>		
13		<u>5.5</u>		<u>5.7</u>	
14	<u>6. Наследование</u>	<u>6.1</u>			
15					<u>6.2</u>
16					<u>6.3</u>

3. ФАЙЛЫ

3.1. ТЕКСТОВЫЙ ФАЙЛ ЧИСЕЛ

*Выполнить задания для **текстового** файла чисел. Размер файла $\leq 64\text{GiB}$ (недостаточно ОП для хранения чисел файла).*

1. Компоненты файла f – вещественные числа. Определить и вывести на экран порядковый номер того из них, которое наиболее близко к соответственному целому числу.
2. Компоненты файла f – целые (отличные от нуля) числа: x, y_1, \dots, y_n . Вывести на экран два последовательных члена этой последовательности, среднее арифметическое которых ближе всего к x .
3. Компоненты файла f – вещественные числа. Записать в файл g наибольшее значение первых десяти компонент, затем следующих десяти и т. д.
4. Компоненты файла f – целые числа. Получить файл g , в котором записаны сначала все положительные числа, затем все отрицательные.
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, причём 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, чётных чисел столько же, сколько нечётных. Получить файл g из чисел исходного файла, в котором не было бы двух соседних чисел одинаковой чётности.
8. Компоненты файла f – целые числа, причём чётных чисел столько же, сколько нечётных. Получить файл g из чисел исходного файла, в котором записаны 2 чётных числа, затем 2 нечётных т. д.
9. Компоненты файла f – целые числа, причём десять идущих подряд положительных чисел чередуются с десятью отрицательными числами и т. д. Получить файл g из чисел исходного файла, в котором записано сначала пять положительных чисел, затем пять отрицательных и т.д.
10. Компоненты файла f – целые числа, причём десять идущих подряд положительных чисел чередуются с десятью отрицательными числами и т. д. Получить файл g из чисел исходного файла, в котором записано сначала двадцать положительных чисел, затем двадцать отрицательных и т. д.
11. Компоненты файла f – целые числа. Получить файл g из чисел исходного файла, в котором записаны сначала все отрицательные числа, затем все нули, затем все положительные числа.
12. Компоненты файла f – целые числа. Получить файл g из чисел исходного файла, в котором записаны сначала все четные положительные числа, затем все четные отрицательные, затем нечетные положительные, затем нечетные отрицательные.

3.2. БИНАРНЫЙ ФАЙЛ ЧИСЕЛ

Выполнить задания [3.1](#) для бинарного файла чисел.

Размер файла ≤ 64 GiB (недостаточно ОП для хранения чисел файла).

3.3. БИНАРНЫЙ ФАЙЛ СТРУКТУР

Выполнить задания для работы с большой базой данных, не помещающейся в оперативной памяти, храня записи в **бинарном** файле.

Написать программу для создания файла структур со сведениями о студентах (ФИО, возраст, курс, пол, успеваемость). Оформить заполнение и вывод файла отдельными функциями. Написать функцию, которая по заданному файлу определяет:

1. Определить ФИО самого старшего студента n курса.
2. Определить ФИО самого младшего студента n курса.
3. Определить средний возраст студентов n курса.
4. Определить количество студентов мужского пола на n курсе.
5. Определить средний балл успеваемости студентов n курса.
6. Определить средний балл успеваемости студентов по m предмету на n курсе.
7. Определить количество отличников на n курсе.
8. Определить количество неуспевающих студентов на n курсе.
9. Определить количество отличников по m предмету на n курсе.
10. Определить количество неуспевающих студентов по m предмету на n курсе.
11. Определить количество студентов на n курсе, имеющих средний балл успеваемости выше среднего балла по его курсу.
12. Определить количество студентов на n курсе, имеющих средний балл успеваемости ниже среднего балла по его курсу.

3.4. ТЕКСТОВЫЙ ФАЙЛ СТРУКТУР

Выполнить задания [3.3](#) для работы с большой базой данных, не помещающейся в оперативной памяти, храня записи в **текстовом** файле.

4. КЛАССЫ

4.1. КЛАСС «СЛОВО»

Разработать класс, содержащий методы:

- *Заполнение полей класса с клавиатуры*
- *Вывод на экран*

Реализовать в классе следующие виды конструкторов:

- *без параметров;*
- *инициализации слова заданным словом-инициализатором;*
- *инициализации слова заданным количеством повторов заданного символа;*
- *инициализации слова частью заданного слова-инициализатора (первые n символов, последние n символов);*
- *копирования;*
- *перемещения.*

Определить объекты класса так, чтобы вызывался конструктор каждого вида. Выполнить задания, используя объекты этого класса.

1. Проверить, есть ли во введённом слове удвоенные гласные буквы. Вывести эти буквы, если они есть, как значение нового объекта класса «Слово».
2. Вывести повторяющиеся в значении исходного объекта-слова буквы как значение нового объекта-слова.
3. Удвоить каждый символ введённого слова и сформировать новый объект-слово.
4. Во введённом значении объекта-слова поменять местами каждую пару соседствующих символов, начиная с первой. Слово дополнить пробелом справа, если в нем нечетное число символов. Сформировать новый объект-слово.
5. Вывести новое слово-объект, упорядочив по возрастанию в лексикографическом порядке буквы исходного слова.
6. Определить количество различных букв в написании введенного с клавиатуры слова и сформировать из них новое слово-объект.
7. Вычислить суммарный ASCII-код введённого с клавиатуры значения объекта-слова. Создать и вывести новое слово-объект, заменив каждую строчную букву исходного значения соответствующей прописной.
8. Заменить все строчные буквы введённого слова числом, равным порядковому номеру буквы по алфавиту. Сформировать новое слово-объект. Например: $Abc \rightarrow A23$.
9. Сравнить два слова и вывести новое слово-объект, значением которого является наибольшее из двух исходных.

10. Заменить все прописные буквы введенного английского слова числом, равным порядковому номеру буквы по алфавиту. Сформировать новое слово-объект. Например: aBC → a23.

11. Вычислить суммарный ASCII-код введенного с клавиатуры значения объекта-слова. Создать и вывести новое слово-объект, заменив каждый символ исходного слова, имеющего код больше, чем код символа 'n', символом 'f'.

12. Сравнить два слова и вывести новое слово-объект, значением которого являются 3 первых символа наибольшего из исходных слов.

4.2. КЛАСС «СТУДЕНТ»

Разработать класс «Студент» и определить его методы:

- Заполнение полей класса с клавиатуры
- Загрузка из бинарного файла
- Сохранение в бинарный файл
- Вывод на экран

Реализовать конструкторы четырех видов (без параметров, с параметрами, копирования, перемещения) и деструктор.

Написать программу для создания массива объектов класса «Студент» со сведениями о студентах (ФИО, возраст, курс, пол, успеваемость). По заданному массиву определить:

- 1.** Определить ФИО самого старшего студента n курса.
- 2.** Определить ФИО самого младшего студента n курса.
- 3.** Определить средний возраст студентов n курса.
- 4.** Определить количество студентов мужского пола на n курсе.
- 5.** Определить средний балл успеваемости студентов n курса.
- 6.** Определить средний балл успеваемости студентов по m предмету на n курсе.
- 7.** Определить количество отличников на n курсе.
- 8.** Определить количество неуспевающих студентов на n курсе.
- 9.** Определить количество отличников по m предмету на n курсе.
- 10.** Определить количество неуспевающих студентов по m предмету на n курсе.
- 11.** Определить количество студентов на n курсе, имеющих средний балл успеваемости выше среднего балла по его курсу.
- 12.** Определить количество студентов на n курсе, имеющих средний балл успеваемости ниже среднего балла по его курсу.

4.3. КЛАСС «БАЗА ДАННЫХ О СТУДЕНТАХ»

Используя класс «Студент» задания [4.2](#) разработать класс «База данных студентов» и определить его методы:

- Добавление студента в базу данных с вводом данных с клавиатуры
- Загрузка базы данных из бинарного файла
- Сохранение базы данных в бинарный файл
- Вывод на экран
- Поиска студентов в базе данных
- Выполнение задания из [4.2](#).

Реализовать конструкторы четырёх видов (без параметров, с параметрами, копирования, перемещения) и деструктор.

4.4. КЛАСС «ВЕКТОР»

Разработать класс «вектор», моделирующий математическое понятие «одномерный массив».

Реализовать конструкторы четырёх видов (без параметров, с параметрами, копирования, перемещения), деструктор.

Максимально возможный размер массива задать константой. В отдельном поле *size* должно храниться значение количества элементов (размерность) объекта-вектора.

Определить методы:

- задания размерности вектора,
- определения количества элементов заданного вектора;
- получения элемента вектора по заданному индексу, с контролем выхода за пределы размерности вектора
- занесения значения элемента вектора по заданному индексу, с контролем выхода за пределы размерности вектора,
- сложения, вычитания, умножения и деления всех элементов вектора на скаляр,
- определения длины вектора ($l = \sqrt{\sum_{i=1}^n x_i^2}$),
- вычисления поэлементного сложения (вычитания, умножения, деления) векторов с одинаковыми границами индексов,
- вывод значений элементов вектора на экран,
- ввода значений элементов вектора с клавиатуры,
- заполнения элементов вектора случайными числами.

Разместить описание класса в заголовочном файле, а определения методов и главную функцию программы – в отдельных файлах. Использовать объекты класса «вектор» при решении предложенной задачи:

1. $X(N)$ и $Y(M)$ - исходные векторы. Определить вектор с максимальной длиной.
2. $X(N)$ и $Y(M)$ - исходные векторы. Определить вектор с минимальной длиной.
3. Вычислить скалярное произведение векторов $X(N)$ и $Y(N)$: $k = \sum_{i=0}^{n-1} x_i y_i$. Получить вектор Z с элементами $z_i = x_i + k$.
4. Даны три вектора, содержащих n вещественных элементов. Определить, являются ли вектора ортогональными.
5. $X(N)$ и $Y(N)$ - исходные векторы. Получить вектор Z с элементами $z_0=1$; $z_i = x_i * y_i + x_{i-1} * y_{i-1}$, $i = 1, n-1$.
6. $X(N)$ и $Y(N)$ - исходные векторы. Получить вектор Z с элементами $z_0=1$; $z_i = x_i * y_i - x_{i-1} * y_{i-1}$, $i = 1, n-1$.
7. $X(N)$ и $Y(N)$ - исходные векторы. Получить вектор Z с элементами $z_i = a * x_i + b * y_i$, $i = 0, n-1$, a, b – максимальные значения векторов X, Y .
8. $X(N)$ и $Y(N)$ - исходные векторы. Получить вектор Z с элементами $z_i = a * x_i - b * y_i$, $i = 0, n-1$, a, b – минимальные значения векторов X, Y .
9. Найти среднее значение элементов вектора $X(N)$ и max - максимальное отклонение значений x_i от этого среднего значения. Получить вектор Z с элементами $z_i = x_i + max$.
10. Получить вектор $Y(N)$, упорядоченный по возрастанию значений элементов вектора $X(N)$. Вычислить скалярное произведение векторов $X(N)$ и $Y(N)$: $k = \sum_{i=0}^{n-1} x_i y_i$.
11. Включить в упорядоченный вектор $X(N)$ новый элемент z так, чтобы сохранилась упорядоченность его элементов. Реализовать алгоритм бинарного поиска для определения места вхождения включаемого элемента.
12. Получить вектор $Z(K)$, упорядоченный по возрастанию, путем слияния векторов $X(N)$ и $Y(M)$, упорядоченных по возрастанию ($K = N + M$).

4.5. ПЕРЕГРУЗКА ОПЕРАЦИЙ

Для предыдущего задания реализовать перегрузку заданной операции двумя способами: как метод класса, как независимую внешнюю функцию. Продемонстрировать выполнение этой операции над объектами типа «вектор».

1. «+» - сложения всех элементов вектора со скаляром,
2. «+» - скалярное произведение двух векторов,
3. «+» - поэлементного сложения двух векторов одинаковой размерности,
4. «-» - унарный минус для каждого элемента вектора,
5. «-» - вычитания из всех элементов вектора скаляр,
6. «-» - вычитания из всех элементов вектора скаляр,
7. «-» - поэлементного вычитания двух векторов одинаковой размерности,
8. «*» - умножения всех элементов вектора на скаляр,
9. «*» - поэлементного умножения двух векторов одинаковой размерности,
- 10.«/» - деления всех элементов вектора на скаляр,
11. «/» - поэлементного деления двух векторов одинаковой размерности.
- 12.«[]» - индексирования (обращения к отдельному элементу вектора).

4.6. ПЕРЕГРУЗКА ОПЕРАЦИЙ

Для задания [4.4](#) реализовать перегрузку операций ввода-вывода двумя способами: как метод класса, как независимую внешнюю функцию. Продемонстрировать выполнение этих операций над объектами типа «вектор».

5. БИБЛИОТЕКА ШАБЛОНОВ STL

5.1. СТРОКИ STRING

Выполнить задания **4.1 первого семестра**, используя класс `string` для представления данных. Задания выполнить через функции.

5.2. СТРОКИ STRING

Выполнить задания **4.2 первого семестра**, используя класс `string` для представления данных. Задания выполнить через функции.

5.3. СТРОКИ STRING

Выполнить задания **4.3 первого семестра**, используя класс `string` для представления данных. Задания выполнить через функции.

5.4. ВЕКТОР

Реализовать задание, используя объявление и методы соответствующего параметризованного класса `vector` из стандартной библиотеки шаблонов STL.

1. Дан массив целых чисел, содержащий n элементов. Элементы массива циклически сдвинуть на K позиций вправо.
2. Из двух массивов разной длины сформировать общий массив и вычислить сумму положительных элементов.
3. Дан массив целых чисел, содержащий n элементов. Получить массив из элементов, встречающихся в исходном массиве ровно один раз без повторений.
4. Дан массив целых чисел, содержащий n элементов. Получить массив из элементов, встречающихся в исходном массиве более одного раза без повторений.
5. Дан массив целых чисел, содержащий n элементов. Получить массив из элементов, встречающихся в исходном массиве ровно два раза без повторений.
6. Найти максимальную по длине монотонную неубывающую подпоследовательность элементов массива.
7. Из двух массивов разной длины сформировать общий массив и поменять местами его максимальный и минимальный элементы.
8. Получить массив $C(K)$, упорядоченный по возрастанию, путем слияния массивов $A(N)$ и $B(M)$, упорядоченных по возрастанию ($K = N + M$).
9. Даны два вектора (одномерных массива), содержащих n вещественных элементов. Если векторы различны, то получить вектор, являющийся суммой двух векторов, иначе переписать в него элементы исходного вектора.
10. Даны два вектора (одномерных массива), содержащих n вещественных элементов. Найти скалярное произведение двух векторов.
11. Даны три вектора (одномерных массива), содержащих n вещественных элементов. Определить, являются ли вектора ортогональными.
12. Дан массив целых чисел, содержащий n элементов. Элементы массива циклически сдвинуть на K позиций влево.

5.5. МАТРИЦА

*Реализовать задание, используя объявление и методы соответствующего параметризованного класса `vector` из стандартной библиотеки шаблонов `STL`.
Задана матрица (как вектор векторов).*

1. Вывести номера столбцов, все элементы, которых чётные.
2. Найти максимальный элемент среди стоящих на главной и побочной диагонали и поменять его местами с элементом, стоящим на пересечении этих диагоналей.
3. Среди строк заданной матрицы, содержащих только нечётные элементы, найти строку с максимальной по модулю суммой элементов.
4. Среди столбцов заданной матрицы, содержащих только такие элементы, которые по модулю не больше заданного натурального n , найти столбец с минимальным произведением элементов.
5. Найти все такие натуральные числа k , что k -я строка совпадает с k -м столбцом.
6. Матрица имеет седловую точку a_{ij} , если a_{ij} является минимальным в i -й строке и максимальным в j -м столбце. Найти все седловые точки заданной матрицы.
7. Для заданной матрицы A найти значение $\min_j (\sum_i |a_{ij}|)$.
8. Найти норму заданной матрицы A , определенную как $\max_i (\sum_j |a_{ij}|)$.
9. Определить, является ли заданная матрица ортонормированной, т. е. такой, в которой скалярное произведение каждой пары различных строк равно 0, а скалярное произведение каждой строки на себя равно 1.
10. Подсчитать количество строк заданной матрицы, которые составлены из различных чисел.
11. Подсчитать количество столбцов заданной матрицы, которые составлены из различных чисел.
12. Поменять местами строку, содержащую элемент с наибольшим значением в матрице, со строкой, содержащей элемент с наименьшим значением.

5.6. СТЕК

Разбить текст, хранящийся в текстовом файле, на слова и вывести их в обратном порядке, используя объявление и методы соответствующего параметризованного класса `vector` из стандартной библиотеки шаблонов `STL`.

Результат записать в новый текстовый файл.

5.7. ОЧЕРЕДЬ

Разбить текст, хранящийся в текстовом файле, на слова и вывести слова, удовлетворяющие условиям задания, используя объявление и методы соответствующего параметризованного класса `list` из стандартной библиотеки шаблонов `STL`.

Результат записать в новый текстовый файл.

Текст – непустая последовательность символов.

Слово – непустая последовательность любых символов, кроме символов-разделителей.

Предложение – последовательность слов, разделённых одним или несколькими символами-разделителями.

Символы-разделители: «пробел», «.», «,», «:», «;», «!», «?», «-», «(», «)».

1. Получить текст, составленный из слов исходного текста, заканчивающихся буквами 'xyz'.
2. Получить текст, составленный из слов исходного текста, начинающихся с букв 'rg'.
3. Получить текст, составленную из слов исходного текста, имеющих заданную длину *n*.
4. Получить текст, составленный из слов исходного текста, в которых нет одинаковых символов.
5. Получить текст, составленный из слов исходного текста, в которых нет символов-цифр.
6. Получить текст, составленный из слов исходного текста, в которых нечётное количество символов.
7. Получить текст, составленный из слов исходного текста, начинающихся с буквы 'a'.
8. Получить текст, составленный из слов исходного текста, начинающихся и заканчивающихся одной и той же буквой.
9. Получить текст, составленный из слов исходного текста, заканчивающихся на буквы 'td'.
10. Получить текст, составленный из слов исходного текста, состоящих только из латинских букв.
11. Получить текст, составленный из слов исходного текста, начинающихся с буквы 'A', заканчивающихся на букву 'S'.
12. Получить текст, составленный из слов исходного текста, начинающихся и заканчивающихся одной и той же введенной буквой (ввести с клавиатуры).

6. НАСЛЕДОВАНИЕ

6.1. 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. 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 выводющую свойства всех созданных объектов на экран. Первичную инициализацию свойств объектов осуществите с помощью конструкторов.*

Вариант 1

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)
- 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)
- количество остановок на маршруте

3. Метро

- координаты GPS точки начала сегмента маршрута (float долгота и широта)
- координаты GPS точки конца сегмента маршрута (float долгота и широта)
- кол-во станций на маршруте
- номер ветки

4. 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 2

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 (RGB)

2. Окружность

- координаты X, Y (центра окружности)
- длина радиуса
- цвет (RGB)

3. Многоугольник

- координаты X, Y (левого верхнего угла описывающего прямоугольника)
- длина радиуса описывающей окружности
- цвет (RGB)
- количество вершин

4. Текст

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

Вариант 3

Вас пригласили поучаствовать в разработке многопользовательской онлайн-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 монстра

- количество золотых монет, получаемых за выполнение задания
- идентификатор локации, где обитают монстры
- идентификатор типа монстров
- количество монстров указанного типа, которое необходимо уничтожить

2. Заказ на доставку предмета

- количество золотых монет, получаемых за выполнение задания
- идентификатор локации расположения предмета
- идентификатор локации расположения заказчика
- идентификатор заказываемого предмета

3. Задание на маршрут 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

4. Задание на охрану каравана

- количество золотых монет, получаемых за выполнение задания
- идентификатор локаций – начала маршрута
- количество охранников

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.

Вариант 4

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

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

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.*

Вариант 5

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

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

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

Вариант 6

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)
- 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)
- код типа продаваемых товаров (число от 1 до 100)
- количество продаваемых товаров

4. Игрок

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

Вариант 7

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. Астероид (подвижный мелкий объект)

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

2. Космический корабль (под управлением игрока)

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

3. 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)

4. Бонус

- координаты (X, Y, Z)
- скорость (V_x, V_y, V_z)
- количество боезапасов
- время существования

Вариант 8

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

Категории сущностей и свойства, которые должны присутствовать в программе:

1. Аспирант

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

2. Преподаватель

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

3. Учебно-вспомогательный персонал

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

4. Научный сотрудник

- ФИО
- степень (строка)
- номер договора

Вариант 9

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 (RGB)
- BorderColor – цвет контура (RGB)
- Координаты на экране X и Y
- Размеры высота и ширина элемента управления

2. HyperlinkLabel – при нажатии мышкой обеспечивает переход по гиперссылке

- Visible - отображается элемент управления или скрыт (bool)
- Координаты на экране 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 (RGB)
- Visible - отображается элемент управления или скрыт (bool)
- Координаты на экране X и Y
- Размеры высота и ширина элемента управления

4. Button – кнопка

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

Вариант 10

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. Пешеход

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

2. Велосипедист

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

3. Автомобиль

- координаты области в кадре (X_{min} , Y_{min} , X_{max} , Y_{max})
- цвет (строка)
- 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 (количество секунд после полуночи)

Вариант 11

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
- из какого металла (перечисление)

2. Холодное оружие

- наименование (строка)
- длина (см)
- номер века, в котором создано

3. Картины

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

4. Фотографии

- что изображено (строка)
- цветная (да, нет)
- приблизительная датировка (диапазон лет – два целочисленных года)

Вариант 12

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 использованы 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. Верхняя одежда

- наименование (строка)
- размер
- сезон (перечисление)
- наличие капюшона (да, нет)
- цена

2. Спортивная одежда

- наименование (строка)
- размер
- цена
- вид спорта (перечисление)

3. Нижнее бельё

- наименование (строка)
- размер
- цена
- материал (хлопок, синтетика, шёлк)

4. Обувь

- наименование (строка)
- размер
- цена
- назначение (муж., жен., детс.)
- наличие каблука (да, нет)

6.2. ПОЛИМОРФИЗМ. ВИРТУАЛЬНЫЕ МЕТОДЫ

Реализовать класс «одномерный динамический массив из потомков-объектов, описанных в [6.1](#)».

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 без параметров, **виртуальные методы**. Выполнить **тестирование** написанных методов (методы шаблоны, которые нигде не вызываются, никогда не компилируются).

6.3. «ФАБРИКА КЛАССА»

Реализовать персистентность объектов из задания [6.2](#), а именно, сохранение массива объектов в файл при выходе из программы и загрузку из файла при запуске.

Пояснение 1.

Персистентность в программировании означает способность состояния существовать дольше, чем процесс, создавший его.

То есть можно дать определение: *персистентность* - возможность длительного хранения состояния.

Без этой возможности, состояние может существовать только в оперативной памяти и теряется, когда оперативная память выключается, например, при выключении компьютера. Это достигается путём хранения состояния как данных на устройстве длительного хранения.

Пояснение 2.

Можно предложить следующий формат файла для выполнения задания.

1 строка: количество сохраняемых (или сохранённых) объектов;

2 строка: тип класса объекта 1 значение поля 1 значение поля 2

3 строка: тип класса объекта 2 значение поля 1 значение поля 2

...