**ЛАБОРАТОРНЫЕ РАБОТЫ 2 курс ООП (2020-2021)**

**ЛАБОРАТОРНАЯ РАБОТА № 10**  Классы, ссылки, аргументы по умолчанию

**Все на классах.**

**ЛАБОРАТОРНАЯ РАБОТА № 10А**

Программа расчета площади прямоугольника и площади квадрата.

Написать **одну** функцию расчета площади с использованием аргументов по умолчанию (для прямоугольника вводить 2 аргумента, для квадрата -1 аргумент).

**ЛАБОРАТОРНАЯ РАБОТА № 10Б**

Написать **две** функции, каждая из которых меняет местами минимальное и максимальное значения трех чисел. Осуществить передачу параметров в функции 2-мя методами (через ссылки, через указатели).

**ЛАБОРАТОРНАЯ РАБОТА № 10В**

Создать класс SUM, в котором описаны 2 целые переменные (x, y) и их сумма (s).

Написать методы класса:

– конструкторы с параметром и по умолчанию;

– функция инициализации x, y;

– функция вывода всех переменных класса;

– ф. подсчета суммы s, которую после вызова функции и вывести на экран.

Осуществить обращения к элементам класса через объекты и через указатели.

**ЛАБОРАТОРНАЯ РАБОТА № 10Г**

Создать класс TIMER (переменная, хранящая количество секунд), с тремя перегруженными конструкторами:

– К1 – инициализация переменной с помощью целого числа (количества секунд);

– К2 – с помощью двух целых переменных (количество минут и секунд);

– К3 – конструктор копирования.

В классе должна быть функция **вывода** количества секунд.

**ЗАДАНИЕ 10 Д.**

Создайте класс с именем train, содержащую поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа train, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массив по пункту назначения, причем поезда с одинаковыми пунктами назначения должны быть упорядочены по времени отправления.

**ЗАДАНИЕ 10 Е.**

Создать класс, описывающий двумерную матрицу. Реализовать функции:

– генерация массива (заполнение случайными числами);

– вывод массива на экран;

– поиск координат элемента с минимальными значениями;

– получение элемента по позиции и запись по заданным координатам;

– получение подматрицы по начальным позициям заданного размера.

Например, для матрицы размера **4х4 01 02 03 04**

**05 06 07 08**

**09 10 11 12**

**13 14 15 16**

выделить подматрицу размером **3х2 с позиции 1х1 06 07**

**10 11**

**14 15**

**ЛАБОРАТОРНАЯ РАБОТА № 1 (11) Классы**

Лабораторная работа **№ 1** (Лабораторные работы. Часть ІІ. Лабораторная работа №1. Простые классы. Доступ к элементам класса.)

**ЛАБОРАТОРНАЯ РАБОТА №11А**

Создать **класс Point** (2 переменные х и у). Реализовать функции класса:

– Конструктор по умолчанию;

– Конструктор с параметром;

– Конструктор копирования;

– функция инициализации переменных х и у (set());

– функция, которая возвращает сумму и произведение переменных (передача параметров через ссылки);

– вывод результатов.

**ЛАБОРАТОРНАЯ РАБОТА №11Б**

Создать **класс** **Figure** для вычисления площади фигур (круга и треугольника). В **конструкторе** использовать аргументы по умолчанию.

**ЛАБОРАТОРНАЯ РАБОТА №11В**

Создать **класс** **Complex**, представляющий комплексное число (вещественная (r) и мнимая (m) часть r + mі = 5 + 4і). Реализовать функции класса:

– инициализации r и m;

– добавления к r и m целых чисел;

– сложить два комплексных числа (второе комплексное число передать как аргумент функции);

– вывод комплексного числа.

**ЛАБОРАТОРНАЯ РАБОТА №11Г**

Разработать классы DOG и MASTER

1. DOG (Кличка, вес, возраст). MASTER (ФИО; ID клуба – **константа**; Собаки – **возможно несколько**!; count – **статический** счетчик создаваемых членов клуба).

Также реализовать следующие пункты.

1. Наличие множества конструкторов для инициализации объектов класса.
2. Реализовать инкапсуляцию (методы set/get) для изменения сведений о собаках и их владельцах.

**Для тестирования программы создать двух владельцев собак. У первого – две собаки, у второго – одна. Для каждого вывести** ФИО, ID клуба и информацию по всем собакам. А также – количество членов клуба.

**ЗАДАНИЕ 11 Д.**

Разработать класс для представления объекта множество чисел (**элементы не повторяются**).

– Определить конструктор с тремя параметрами (количество элементов, минимальный и максимальный в диапазоне генерируемых чисел), который задаёт элементы множества (*конструктор с параметрами*)*.*

*–* Определить конструктор, который генерирует множество из 10 элементов в диапазоне [-20; 20] (*конструктор по умолчанию*)*.*

– Определить конструктор, которые копирует в новое множество элементы другого множества, не превышающие заданное значение (*конструктор копирования*).

– Создать функцию, которая находит **максимальное** значение в множестве.

– Создать функции для построения **объединения**, **пересечения**, **разности** и **дизъюнктивной суммы** двух числовых множеств.

**ЗАДАНИЕ 11 Е.**

Создать класс Время (часы, минуты, секунды).

– Класс должен содержать гетеры и сеттеры (методы изменения времени на заданное количество часов, минут и секунд и вывода значений каждого поля) с проверкой допустимости вводимых значений. В случае недопустимых значений полей – корректно обработать ситуацию.

– Создать функцию вывода всего объекта время на экран.

– Создать функцию подсчета разности времен двух объектов.

– Создать функцию сравнения времен двух объектов.

– Определить угол между часовой и минутной стрелками.

**ЛАБОРАТОРНАЯ РАБОТА № 2** Перегрузка

Лабораторная работа **№ 2** из методички

**ЛАБОРАТОРНАЯ РАБОТА №2А**

Класс (по варианту) с перегруженными операциями ==, =, + как ф.-члены класса.

**ЛАБОРАТОРНАЯ РАБОТА №2Б**

Класс (по варианту) с перегруженными операциями ==, + как дружественные ф.

– **Перегрузить [ ]**, чтобы посчитать длину переменной типа char\*.

– **Перегрузить ( )**, чтобы проинициализировать переменные класса.

Отключить предупреждение об устаревших функциях (strcpy) -

1) свойства проекта 2)с/с++ -> Preprocessor -> Preprocessor Definition ->

\_CRT\_SECURE\_NO\_WARNINGS

**ЛАБОРАТОРНАЯ РАБОТА № 3** Наследование, множественное наследование, виртуальные функции и классы

Лабораторная работа **№ 3** из методички *(использовать виртуальные классы и функции).*

**ЛАБОРАТОРНАЯ РАБОТА №3А** *Наследование без передачи параметров*

Создать класс SOBAKA (вес и возраст – инициализация и печать этих параметров).

ПК от него SPANIEL (цвет – инициализация и печать).

В каждом классе – К и Д (просто печать, что это К и Д соответствующего класса),

– функция *golos* – печать "ГАВ" или "гав" в зависимости от веса (>= или < 10 кг).

**ЛАБОРАТОРНАЯ РАБОТА №3Б** *Наследование с передачей параметров через конструктор*

Создать класс BOOK (название книги, автор, количество страниц),

ПК – LIBRARY (номер каталога, метка о наличии (есть или нет)).

В каждом классе – К и функция вывода переменных.

Осуществить передачу параметров в конструктор для инициализации переменных БК через ПК.

Создать массив из 3-х книг и проверить:

– есть ли книга вообще (по названию);

– есть ли в библиотеке или "на руках" (по метке о наличии);

– смоделировать ситуацию, когда сначала книга была, а потом ее взяли почитать (т.е. изменилась метка о наличии).

**ЛАБОРАТОРНАЯ РАБОТА №3В**

Сгенерировать 10 случайных чисел.

БК – переменная x, ф. инициализации, ф. вывода.

ПК1 от БК – числа (из х), которые при делении на 7 дают в остатке 3 (т.е. в переменную класса сохранить остаток при делении на 7);

ПК2 от ПК1– числа (из х), которые при делении на 5 дают в остатке 2 (т.е. в переменную класса сохранить остаток при делении на 5);

В ПК2 распечатать числа (из х), которые удовлетворяют первому И второму условиям.

**ЛАБОРАТОРНАЯ РАБОТА №3Г**

Создать класс БК FIGURE, который описывает плоскую фигур, для вычисления площади которой достаточно 2-х измерений.

В этом БК создать **чистую виртуальную функцию** show() для печати значения площади и функцию set() для инициализации переменных.

Создать 3 ПК (в каждом определена своя функция show() для печати значения площади).

ПК1 – площадь треугольника;  
ПК2 – площадь прямоугольника;  
ПК3 – площадь круга (использовать **аргументы по умолчанию**).

Работать через объекты ПК1, ПК2 и ПК3 И через указатель на БК.  
**ЛАБОРАТОРНАЯ РАБОТА №3Д**

Создать иерархическую структуру студент, преподаватель, зав.кафедры.

Реализовать механизм наследования и создать **класс**, позволяющий вывести имена студентов, которые имеют хотя бы один долг и преподавателей, младше 50.

Создать иерархическую структуру преподаватель, зав.кафедрой, студент.

Разработать систему классов:

• Абстрактный класс **Persona** –

поля – фио, возраст;

виртуальные функции: print(), who(). ask(), которые будут реализованы в классах наследниках.

• Наследники класса **Persona**

Класс Prepod –

поля – дисциплина, которую преподает; степень (ктн – 1, дтн – 2, нет – 0);

функции: who() возвращает 1,

ask() возвращает степень,

print() печатает всю информацию о преподавателе.

Класс Student –

поля – оценки (массив из 5 оценок);

функции: who() возвращает 3,

ask() возвращает количество двоек,

print() печатает всю информацию о студенте.

• Наследник класса **Prepod**

Класс Zav\_kaf –

поля – должность;

функции: who() возвращает 2,

ask() возвращает возраст,

print() печатает всю информацию о зав.кафедрою.

Для работы с данными классами создать класс **VUZ**.

Организовать заполнение динамического массива Shtat персонами разных типов (два преподавателя, три студента, один зав.кафедрой). Распечатать преподавателей младше 50 лет, вывести имена студентов, которые имеют хотя бы один долг.

**ЛАБОРАТОРНАЯ РАБОТА № 4**

Лабораторная работа **№ 4** из методички.

**ЛАБОРАТОРНАЯ РАБОТА № 5**

Лабораторная работа **№ 5** из методички.

**ЛАБОРАТОРНАЯ РАБОТА № 6**

Лабораторная работа **№ 6** из методички.

**ЛАБОРАТОРНАЯ РАБОТА № 7**  Файлы в С++

**с 7й и далее с обработкой исключительных ситуаций**

**ЛАБОРАТОРНАЯ РАБОТА №7А**

Записать в текстовый файл четное количество вещественных чисел.

Считать из файла вещественные числа.

Создать класс COMP. Создать массив из объектов класса COMP.

Составить из пар считанных чисел комплексные, распечатать их на экран и сохранить в новый файл. Использовать функции **read(), write().**

**ЛАБОРАТОРНАЯ РАБОТА №7Б**

Перегрузить оператор << для класса, который будет выводить строку без цифр и больших букв в текстовый файл. В классе – переменная string, КсП, show(), перегрузка <<.

**ЛАБОРАТОРНАЯ РАБОТА №7В**

В бинарном файле из 10 чисел проверить, если 5 число – четное, заменить все числа до него числом 77, если нечетное – все числа после него числом 88.

**ЛАБОРАТОРНАЯ РАБОТА №7Г**

Общее задание: Используя наработки предыдущей лабораторной работы №1, создать **отдельный класс**, реализующий следующие функции:

* ввод исходных данных – массива объектов с клавиатуры (случайным образом или заданные явно в программе);
* вывод информации о массиве объектов класса на экран;
* сохранение одного объекта класса в файл (номер нужного объекта вводится с экрана). Функция должна иметь следующую сигнатуру:

*public void writeToFile(string filename, T data)* , где Т – тип объекта класса

* восстановление одного объекта класса из файла. Если в файле записано несколько объектов, то номер нужного объекта вводится с экрана (осуществить проверку на ввод неправильного номера). Функция должна иметь следующую сигнатуру:

*public T readFromFile (string filename)* , где Т – тип объекта класса

или

*public void readFromFile2 (string filename, T &data)*

* сохранение всего массива объектов в файл;
* чтение всего массива объектов класса из файла.

Разработать удобный и интуитивно понятный интерфейс для демонстрации работы всех реализованных функций!

**ЛАБОРАТОРНАЯ РАБОТА № 8** Шаблоны

**ЛАБОРАТОРНАЯ РАБОТА №8А**

Создать **шаблон** класса для работы с числами :

– К по У: – К с параметрами; – Ф. вывода;

– перегрузить операцию +.

Применить для целых и вещественных типов.

**ЛАБОРАТОРНАЯ РАБОТА №8Б**

Создать класс-**шаблон для работы с массивом:**

ПОЛЯ – динамический массив, максимальный размер массива (size), индекс (0 < ind < size для работы не со всеми элементами массива);

ФУНКЦИИ

– конструктор по умолчанию (выделяет память под массив и инициализирует переменные size=100, ind=0);

– функция заполнения одного элемента массива (в main() реализовать в цикле заполнение массива до значения ind);

– функция подсчета суммы значений в массиве размером ind;

– функция подсчета среднеарифметического значений в массиве размером ind;

– функция печати массива размером ind;

– перегрузить операцию [ ] для подсчета максимального значения в массиве любого размера (<=ind);

– деструктор для удаления динамического массива.

**ЛАБОРАТОРНАЯ РАБОТА № 9** Контейнеры

**ЛАБОРАТОРНАЯ РАБОТА №9А**

Использовать класс-шаблон VECTOR.

Создать класс (две целые переменные и их сумма):

– К по У;

– К с параметрами, для нахождения суммы двух переменных;

– Ф. вывода суммы (get()).

Создать и сравнить 2 вектора на равенство и неравенство сумм ПОПАРНО. Для этого перегрузить операции ==, >, < (или одну из них, но с проверкой на все).

**ЛАБОРАТОРНАЯ РАБОТА №9Б**

Создать ассоциативный массив (телефонная книжка) на основе двух классов (NAME -> first and NUMBER -> second).

Перегрузить оператор < в классе NAME для упорядочивания по имени.

В каждом классе:

– ф. вывода get();

Организовать map<> из 3-5 элементов.

Напечатать весь список.

Осуществить поиск по ключу (по имени).

Напечатать всех, у кого фамилия начинается на определенную букву(*попробовать*).

Удалить всех, у кого фамилия начинается на определенную букву(*попробовать*).

**ЛАБОРАТОРНАЯ РАБОТА №9В**

Использование **ФУНКТОРА** и контейнера **map**.

Создать класс PERSON (name, age):

– КпоУ, КсП для name, КсП для age (отдельно для каждого);

– get\_name(), get\_age();

– перегрузить оператор < – Ф. СРАВНЕНИЯ для упорядочивания по name;

– перегрузить оператор () – ФУНКТОР для упорядочивания по age.

Создать класс STAFF для работы с классом PERSON.

В классе STAFF:

– ф. заполнения (insert());

– ф. печати всех полей из PERSON;

– ф. печати одного поля из PERSON (только фамилию или только возраст);

– ф. удаления по фамилии;

– ф. вставки;

– ф. поиска по каждому полю.

В классе STAFF создать два контейнера map (два объекта) для сортировки по имени (используя ф. сравнения) и для сортировки по возрасту (используя функтор).

**ЛАБОРАТОРНАЯ РАБОТА № 10** Тестирование

Сделать 3 функции-шаблоны (шаблон для целых, м.б. *вещественных, символьных*) и 3 теста.

**Функции:**

а) min() – нахождения минимального;

б) sort() – сортировка массива по возрастанию;

в) sum() – нахождения в массиве суммы положительных.

**Тесты:**

а) для ф. min() – проверить результаты сравнения полож., отриц., и смешанных:

– с помощью ф. min() найти минимум двух чисел (символов);

– сравнить результаты (assert()).

б) для ф. sort():

– создать массив1 неотсортированный;

– создать массив2 ожидаемый отсортированный;

– отсортировать массив1, используя ф. sort();

– в цикле поэлементно сравнить результаты (assert()).

в) для ф. sum():

– создать массив;

– с помощью sum() найти сумму положительных;

– сравнить результаты ожидаемый и полученный (assert()).

**Смоделировать** ситуации, когда assert() выдает ошибки:  
– в ф. min() поменять знак в сравнении (знак > на знак <);

– в ф. sort()поменять знак при сортировке (т.е. будет сортировка по убыванию);

– в ф. sum() убрать условие нахождения положительных.