Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники» филиал «Минский радиотехнический колледж»

**Задания**

на домашние контрольные работы № 1 и № 2

по дисциплине «Конструирование программ и языки программирования»

для учащихся заочной формы обучения

специальности 2- 40 01 01 «Программное обеспечение информационных технологий»

Составил преподаватель А.И. Назарова

Рассмотрено на заседании ЦК ПОИТ.

Рекомендовано к использованию на 20 17 - 2018 учебный год

Протокол № от\_\_\_\_\_\_\_\_\_

Председатель ЦК ПОИТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_И.Г. Смолер

**Указания по выполнению контрольных работ**

Учебным планом специальности для учащихся заочной формы обучения предусмотрено выполнение двух домашних контрольных работ по дисциплине “Конструирование программ и языки программирования”.

Домашние контрольные работы проводятсяс цельюруководства самостоятельной работой учащихся итекущего контроля заих работой над учебным материалом в период между установочной и экзаменационными сессиями.

Выполненные домашние контрольные работы высылаются учащимися в учреждение образования для рецензирования в сроки, установленные учебным графиком.

Главная цель контрольной работы – показать умения и навыки учащегося самостоятельно работать с литературой, проводить анализ и программирование задачи.

Выполнение домашней контрольной работы требует от учащихся заочной формы обучения самостоятельного изучения материала по пособиям, книгам, конспекту.

При необходимости учащийся может получить консультацию у преподавателя.

*При оформлении работы и выборе заданий следует руководствоваться следующими правилами:*

1. Контрольная работа должна быть выполнена строго в соответствии с предложенным вариантом.
2. Номер варианта зданий определяется по последней цифре шифра учащегося.
3. Титульный лист контрольной работы должен содержать фамилию, имя, отчество, номер группы и шифр учащегося, название дисциплины, номер варианта, фамилию преподавателя-рецензента.
4. Перед выполнением каждого задания полностью выписывается его условие, соответствующее варианту.
5. Задания надо располагать в порядке возрастания их номеров.
6. В конце контрольной работы обязательно наличие списка использованной литературы.
7. Программирование задач необходимо осуществлять в среде Microsoft Visual Studio на языке программирования С# c применением шаблонов Win32 Console Application или Windows Forms.

10. Задачи надо разбивать на подзадачи, которые необходимо реализовать в виде функций. Разработанные программы должны содержать комментарии, а так же должны быть правильно структурированы.

10. **На проверку необходимо представить отчет по контрольной работе на бумажном носителе, а также электронный вариант. Отчет должен быть подготовлен в текстовом редакторе Microsoft Word и содержать выполненные задания. В отчете необходимо представить также тексты разработанных программ и результаты их работы в виде копий экранных форм.**

Контрольная работа, содержащая все верно выполненные задания в соответствии с вариантом и правилами, представленными выше, зачитается.

«Не зачтено» выставляется в том случае, если задания содержат грубые ошибки, а результаты отсутствуют или неверные.

На проверку не принимаются контрольные работы, выполненные не в соответствии с вариантом.

**Контрольная работа № 1**

**Задание 1**

Значение аргумента х изменяется от а до b с шагом h. Для каждого х найдите значение функции Y(x), суммы S(x) и  и выведите в виде таблицы. Значения а, b, h введите с клавиатуры произвольно. Вычисление Y(x) и S(x) реализуйте в виде функций.

В основной программе организуйте ввод исходных данных, обращение к функциям и вывод результатов.

**Вариант 0**  .

**Вариант 1**  .

**Вариант** **2**  .

**Вариант** **3**  .

**Вариант** **4**  .

**Вариант** **5**  .

**Вариант** **6**

.

**Вариант** **7**  .

**Вариант** **8**  .

**Вариант 9**  .

**Задание 2**

Разработайте программу обработки массивов, результаты работы программы отобразите на экране.

**Вариант 0**

Дана целочисленная прямоугольная матрица. Определить:

- количество строк, не содержащих ни одного нулевого элемента;

- максимальное из чисел, встречающихся в заданной матрице более одного раза.

**Вариант 1**

Дана целочисленная прямоугольная матрица. Определить:

* количество столбцов, не содержащих ни одного нулевого элемента;

- переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик (характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов).

**Вариант 2**

Дана целочисленная прямоугольная матрица. Определить:

- количество столбцов, содержащих хотя бы один нулевой элемент;

- номер строки, в которой находится самая длинная серия одинаковых элементов.

**Вариант 3**

Дана целочисленная квадратная матрица. Определить:

- произведение элементов в тех строках, которые не содержат отрицательных

элементов;

- максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.

**Вариант 4**

Дана целочисленная квадратная матрица. Определить:

- сумму элементов в тех столбцах, которые не содержат отрицательных элементов;

- минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы.

**Вариант 5**

Дана целочисленная прямоугольная матрица. Определить:

- сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент;

- номера строк и столбцов всех седловых точек матрицы. (Матрица А имеет седловую точку Аij, если Аij является минимальным элементом в i-й строке и максимальным в j-м столбце и наоборот).

**Вариант 6**

Дана целочисленная матрица. Определить:

- такие k, что k-я строка матрицы совпадает с k-м столбцом (для матрицы размером 8х8).

- сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

**Вариант 7**

Дана целочисленная матрица.

- Переставляя столбцы заданной матрицы, расположить их в соответствии с ростом характеристик (характеристикой столбца целочисленной матрицы назовем сумму модулей его отрицательных нечетных элементов).

- Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

**Вариант 8**

Дана целочисленная матрица.

- Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером 10 на 10.

- Найти сумму модулей элементов, расположенных выше главной диагонали.

**Вариант 9**

Дана целочисленная прямоугольная матрица.

- Упорядочить строки матрицы по возрастанию количества одинаковых элементов в каждой строке.

-Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента.

**Задание 3**

Разработайте программу обработки строк. Используйте в задачах два класса строк: String и StringBuilder. Результаты отобразите на экране.

**Вариант 0**

Составьте программу, которая будет вводить строку в переменную String. Подсчитать, сколько различных символов встречаются в ней. Вывести их на экран.

**Вариант 1**

Составьте программу, которая будет вводить строку в переменную String. Найти в ней те слова, которые начинаются и оканчиваются одной и той же буквой.

**Вариант 2**

Составить программу, которая будет вводить строку в переменную String. Определить, сколько раз в строке встречается заданное слово.

**Вариант 3**

Строка, содержащая произвольный русский текст, состоит не более чем из 200 символов. Определите, какие буквы и сколько раз встречаются в этом тексте.

**Вариант 4**

Составьте программу, которая будет вводить строку в переменную String. Удалите из нее все лишние пробелы, оставив между словами не более одного. Результат поместите в новую строку.

**Вариант 5**

Составьте программу, которая будет вводить строку в переменную String. Напечатайте в алфавитном порядке все слова из данной строки, имеющие заданную длину n.

**Вариант 6**

Составить программу, которая будет вводить строку в переменную String. Найти слово, встречающееся в каждом предложении, или сообщить, что такого слова нет.

**Вариант 7**

Дана строка, содержащая текст на русском языке. В предложениях некоторые из слов записаны подряд несколько раз (предложение заканчивается точкой или знаком восклицания). Получить в новой строке отредактированный текст, в котором удалены подряд идущие вхождения слов в предложениях.

**Вариант 8**

Даны две строки А и B. Составьте программу, проверяющую, можно ли из букв, входящих в А, составить В (буквы можно использовать не более одного раза и можно переставлять).

Например, А: ИНТЕГРАЛ; В: АГЕНТ — составить можно; В: ГРАФ —нельзя.

**Вариант 9**

Дана строка, содержащая зашифрованный русский текст. Каждая буква заменяется на следующую за ней (буква я заменяется на а). Получить новую расшифрованную строку.

**Контрольная работа № 2**

**Задание**

Выполните задание согласно варианту. Продемонстрируйте работу программы. Каждый разрабатываемый класс должен содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы, свойства, индексаторы, перегруженные операции. Функциональные элементы класса должны обеспечивать непротиворечивый, полный и удобный интерфейс класса. При возникновении ошибок должны выбрасываться исключения. В программе должна выполняться проверка всех разработанных элементов класса.

**Вариант 0**

Создать абстрактный класс Prog с методами, позволяющими вывести на экран информацию о программном обеспечении, а также определить соответствие возможности использования (на момент текущей даты).

Создать производные классы: Свободное (название, производитель), Условно-бесплатное (название, производитель, дата установки, срок бесплатного использования), Коммерческое (название, производитель, цена, дата установки, срок использования) со своими методами вывода информации на экран, и определения возможности использования на текущую дату.

Создать базу (массив) из n видов программного обеспечения, вывести полную информацию из базы на экран, а также организовать поиск программного обеспечения, которое допустимо использовать на текущую дату.

**Вариант 1**

Создать абстрактный класс Stud с методами, позволяющим вывести на экран информацию о персоне, а также определить ее возраст (на момент текущей даты).

Создать производные классы: Абитуриент (фамилия, дата рождения, факультет), Студент (фамилия, дата рождения, факультет, курс), Преподаватель (фамилия, дата рождения, факультет, должность, стаж), со своими методами вывода информации на экран, и определения возраста.

Создать базу (массив) из n персон, вывести полную информацию из базы на экран, а также организовать поиск персон, чей возраст попадает в заданный диапазон.

**Вариант 2**

Создать абстрактный класс Издание с методами позволяющим вывести на экран информацию об издании, а также определить является ли данное издание искомым.

Создать производные классы: Книга (название, фамилия автора, год издания, издательство), Статья (название, фамилия автора, название журнала, его номер и год издания), Электронный ресурс (название, фамилия автора, ссылка, аннотация) со своими методами вывода информации на экран.

Создать каталог (массив) из n изданий, вывести полную информацию из каталога, а также организовать поиск изданий по фамилии автора.

**Вариант 3**

Создать абстрактный класс Avto с методами позволяющим вывести на экран информацию о транспортном средстве, а также определить грузоподъемность транспортного средства.

Создать производные классы: Легковая\_машина (марка, номер, скорость, грузоподъемность), Мотоцикл (марка, номер, скорость, грузоподъемность, наличие коляски, при этом если коляска отсутствует, то грузоподъемность равна 0), Грузовик (марка, номер, скорость, грузоподъемность, наличие прицепа, при этом если есть прицеп, то грузоподъемность увеличивается в два раза) со своими методами вывода информации на экран, и определения грузоподъемности.

Создать базу (массив) из n машин, вывести полную информацию из базы на экран, а также организовать поиск машин, удовлетворяющих требованиям грузоподъемности.

**Вариант 4**

Создать абстрактный класс Function с методом вычисления значения функции y=f(x) в заданной точке.

Создать производные классы: Line (y=кx+b), Kub (y=kx2+bx+c), Parabola (y=кx2 +c) со своими методами вычисления значения в заданной точке.

Создать массив n функций и вывести полную информацию о значении данных функций в точке х.

**Вариант 5**

Создать абстрактный класс Tovar с методами, позволяющим вывести на экран информацию о товаре, а также определить, соответствует ли она сроку годности на текущую дату.

Создать производные классы: Продукт (название, цена, дата производства, срок годности), Партия (название, цена, количество шт, дата производства, срок годности), Комплект (названия, цена, перечень продуктов) со своими методами вывода информации на экран, и определения соответствия сроку годности.

Создать базу (массив) из n товаров, вывести полную информацию из базы на экран, а также организовать поиск просроченного товара (на момент текущей даты).

**Вариант 6**

Создать абстрактный класс Tovar с методами, позволяющими вывести на экран информацию о товаре, а также определить, соответствует ли она искомому типу.

Создать производные классы: Игрушка (название, цена, производитель, материал, возраст, на который рассчитана), Книга (название, автор, цена, издательство, возраст, на который рассчитана), Спорт-инвентарь (название, цена, производитель, возраст, на который рассчитана), со своими методами вывода информации на экран, и определения соответствия искомому типу.

Создать базу (массив) из n товаров, вывести полную информацию из базы на экран, а также организовать поиск товаров определенного типа.

**Вариант 7**

Создать абстрактный класс Телефонный\_справочник с методами, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска.

Создать производные классы: Персона (фамилия, адрес, номер телефона), Организация (название, адрес, телефон, факс, контактное лицо), Друг (фамилия, адрес, номер телефона, дата рождения) со своими методами вывода информации на экран, и определения соответствия искомому типу.

Создать базу (массив) из n товаров, вывести полную информацию из базы на экран, а также организовать поиск в базе по фамилии.

**Вариант 8**

Создать абстрактный класс Figure с методами вычисления площади и периметра, а также методом, выводящим информацию о фигуре на экран.

Создать производные классы: Rectangle (прямоугольник), Circle (круг), Triangle (треугольник) со своими методами вычисления площади и периметра.

Создать массив n фигур и вывести полную информацию о фигурах на экран.

**Вариант 9**

Создать абстрактный класс Klient с методами, позволяющими вывести на экран информацию о клиентах банка, а также определить соответствие клиента критерию поиска.

Создать производные классы: Вкладчик (фамилия, дата открытия вклада, размер вклада, процент по вкладу), Кредитор (фамилия, дата выдачи кредита, размер кредита, процент по кредиту, остаток долга), Организация (название, дата открытия счета, номер счета, сумма на счету) со своими методами вывода информации на экран, и определения соответствия дате (открытия вклада, выдаче кредита, открытия счета).

Создать базу (массив) из n клиентов, вывести полную информацию из базы на экран, а также организовать поиск клиентов, начавших сотрудничать с банком в заданную дату.