**Задание 13.**

В интегрированной среде разработки *Microsoft Visual Studio* на языке *Visual C#* в консольном режиме составить программное обеспечение для решения типовых задач программирования по тематике «Переменные перечислимого типа» («*Enum*»), заданных по варианту.

**Элементы рабочего задания:**

1. **Учесть все возможные ограничения**, накладываемые не только на переменные, но и на участвующие в расчёте функциональные зависимости**[[1]](#footnote-1)**.

2. **Подобрать корректные тестовые примеры в необходимом количестве[[2]](#footnote-2).**

3. Код составленного программного обеспечения **сопроводить комментариями** (элементы программистской этики)**[[3]](#footnote-3)**.

4. **Отчёт по решённой задаче составить** по схеме, изложенной в методических указаниях**[[4]](#footnote-4)**. Для удобства работы обучающихся далее представлена памятка основных разделов отчёта:

0. Титульный лист

1. Цель работы

2. Формулировка задачи

3. Блок-схема алгоритма

4. Подбор тестовых примеров

5. Листинг (код) программы

6. Расчёт тестовых примеров на ПК

7. Вывод по работе

5. **Схему алгоритма оформлять** согласно правилам**[[5]](#footnote-5)**. Помнить, что при работе с методами необходимо вводить нумерацию блоков и рассматривать содержимое методов в отдельных схемах алгоритма. Помнить, что оконечные блоки методов помимо надписей «Начало» и «Конец» должны содержать сигнатуры методов: либо краткие, в которых присутствует только наименование метода без перечисления формальных параметров (все параметры перечисляются далее в блоке «Данные» внутри метода); либо полные, в которых перечисляются все формальные параметры и их типы данных (в этом случае блок «Данные» внутри метода не размещается и не заполняется).

6. Материалы, подготавливаемые **к отправке по электронной почте**, оформляются согласно **Приложению 4** Сборника задач**[[6]](#footnote-6)**. Если материалы направляются на проверку через платформу *GitHub*, то **Приложением 4** разрешается не руководствоваться.

7. Настоятельно рекомендовано перед решением задачи **ознакомиться с рекомендациями**, собранными в **Приложении 3** Сборника задач**[[7]](#footnote-7)**.

8. Не забывать **давать значимые имена переменным**, проектам, решениям, файлам, методам, перечислимым переменным и элементам внутри них, а также другим разрабатываемым компонентам**[[8]](#footnote-8)**. При нежелании использовать значимые имена переменных в отчёте необходимо составить в разделе, содержащем формулировку задачи, таблицу соответствия вида:

Таблица 1 – Пример таблицы соответствия переменных

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование переменной в программе** | **Смысловое содержание используемой переменной** | **Тип данных** |
| 1 | *a* | Ускорение | Вещественное |
| 2 | *v* | Скорость | Вещественное |
| 3 | *S* | Пройдённый путь | Вещественное |
| 4 | *t* | Затраченное время | Вещественное |
| 5 | *x* | Абсцисса точки на Декартовой плоскости | Целое |
| 6 | *y* | Ордината точки на Декартовой плоскости | Целое |
| 7 | *z* | Аппликата точки в пространстве | Целое |

9. Строго следовать указанным по вариантам перечням служебных слов и конструкций.

**Внимание! При выявлении противоречивых условий среди вариантов индивидуального задания не предпринимать самостоятельных решений по устранению противоречий, а обратиться за консультацией к преподавателю, читающему курс «Программирования и основ алгоритмизации».**

**Задание состоит только из Индивидуальной части. Общая часть в нём отсутствует.**

**Вариант №1.** Наименованиядней недели: понедельник, вторник, среда, четверг, пятница, суббота и воскресенье, указанные на английском языке или на транслитерации записаны в переменную «*WeekDays*» перечислимого типа в естественном порядке следования. Идентификаторы, закреплённые за наименованиями дней недели из десятичного диапазона [110…710], распределены в обратной последовательности. При использовании циклической конструкции с предусловием подготовьте программное обеспечение, выводящее в консоль наименования дней недели с понедельника по воскресенье. При решении задачи оператор переключения не использовать.

**Вариант №2.** Статусы заказа в сети «Додо Пицца» с курьерской доставкой на дом, указанные на английском языке или на транслитерации, записаны в переменную перечислимого типа «*DodoStatus*» в хронологическом порядке: принят, готовится на кухне, приготовлен, передан курьеру, в пути, доставлен. Идентификаторы, закреплённые за статусами заказа, распределены в обратной последовательности из шестнадцатеричного диапазона [*0916*…*0E16*]. При использовании циклической конструкции по известному диапазону значений составьте с псевдослучайными значениями задержки по времени итераций циклической конструкции программное обеспечение, последовательно выводящее на экран консоли текущий статус заказа. По достижении статуса «Доставлен» предлагать пользователю возможность выставления оценки выполненному заказу. При решении задачи по варианту допускается выдача в консоль значений, содержащих наименования статусов заказа, кириллицей посредством использования оператора переключения.

**Вариант №3.** Каналы цветовой модели *RGB*: красный, зелёный и синий, записанные в переменную перечислимого типа «*RGBmodel*» на английском языке или на транслитерации, заданы своими шестнадцатеричными значениями: *FF000016*, *FF0016* и *FF16*, соответственно. Для перебора элементов в составе переменной перечислимого типа воспользоваться циклической конструкцией с постусловием. Не использовать оператор переключения. Вывести в консоль построчно только наименования цветов каналов (численных значений при выводе оказаться не должно). Для перемещения между элементами, записанными в переменную перечислимого типа, использовать битовые операции.

**Вариант №4.** Переменная перечислимого типа «*Months*» содержит наименования календарных месяцев: январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь; записанных на английском языке или при использовании транслитерации. Составить программное обеспечение для вывода в консоль количества дней для каждого месяца не високосного года на основе перечислимого типа данных при использовании оператора переключения. Перебирать позиции переменной перечислимого типа при использовании циклической конструкции по известному множеству значений. Месяцы закодированы в десятичном формате, в естественном порядке следования из диапазона [10010…11110].

**Вариант №5.** Позиции управления оборотами двигателя автомобиля с ручной коробкой передач записаны в переменную перечислимого типа «*Transmission*». Позиции управления ограничены множеством: первая, вторая, третья, четвёртая, пятая, реверс. Элементы множества должны быть названы на английском языке или на транслитерации, а также указаны в естественном порядке следования. За элементами закреплены десятичные значения согласно двоичному коду Грея. Для вывода наименований позиций управления не должен использоваться оператор переключения. Перебор элементов множества необходимо реализовать при использовании цикла по известному диапазону значений. В качестве оригинального двоичного нумератора элемента использовать параметр цикла, а при определении корректной, закодированной позиции управления, выполнять преобразование значения параметра цикла в код Грея посредством применения битовых операторов. На экран позиции управления должны оказаться выведены в естественной последовательности: первая, … пятая, а затем – реверс.

**Вариант №6.** Элементы цветовой модели *CMYK*: циановый, пурпурный, жёлтый, чёрный, записанные на английском языке или на транслитерации; будучи переведёнными в цветовую модель *RGB* и записанными в переменную перечислимого типа «*CMYKmodel*» представлены своими шестнадцатеричными значениями: *FFFF16*, *FF00FF16*, *FFFF0016*, *FFFFFF16*. Вывести их на экран в указанном порядке следования (циановый, пурпурный, жёлтый, чёрный) при использовании оператора переключения для обеспечения возможности просмотра кириллических наименований. Сформировать циклическую конструкцию с предусловием, внутри которой для перебора шестнадцатеричных значений использовать только битовые операции.

**Вариант №7.** Музыкальные ноты: «до», «ре», «ми», «фа», «соль», «ля», «си»; записанные в алфавитном порядке на английском языке или транслитерации в переменную перечислимого типа «*Notes*», содержат целочисленные связки, заданные в шестнадцатеричном формате «*0x*» в диапазоне [*216*, …, *1016*, …, *4016*] таким образом, что их можно вывести в консоль в естественном порядке следования посредством применения только операций битового сдвига. При составлении кода программы следует обойтись без использования оператора переключения для вывода музыкальных нот. Вывод элементов в составе переменной перечислимого типа организовать при использовании циклической конструкции с постусловием.

**Вариант №8.** Направления: влево, вправо, вперёд, назад; записаны на английском языке или на транслитерации и собраны в переменную перечислимого типа «*Directions*». Десятичные целочисленные связки для направлений заданы, соответственно, как [51210, 102410, 204810, 409610]. При использовании оператора переключения организуйте команды таким образом, что движение возможно только вперёд и назад, а повороты – влево и вправо. Смоделируйте задержку внутри циклической конструкции с предусловием между направлениями при использовании вещественного генератора псевдослучайных чисел с округлением результата до целого в большую сторону: в консоль выводятся команды только с наименованиями направлений и однократно. Как только все возможные команды окажутся продемонстрированными – циклическая конструкция прерывается.

**Вариант №9.** Наименования пальцев руки: большой, указательный, средний, безымянный, мизинец; записанные на английском языке или на транслитерации собраны в переменную перечислимого типа «*HandFingers*». Вместе с тем переменная перечислимого типа содержит несколько комбинаций из пальцев руки, такие как: «класс», «ОК», «пистолет», «телефон», «коза». Целочисленные десятичные связки представляют собой битовые флаги. Без использования оператора переключения организуйте последовательный вывод в консоль только наименований пальцев руки внутри циклической конструкции по известному множеству значений при использовании только битовых операций. После вывода наименований предложить оператору ввод в двоичном формате сочетания согнутых – «0» и разогнутых – «1» пальцев. При совпадении ввода с одной из комбинаций, заложенных в переменную перечислимого типа, – наблюдать в консоли её наименование, иначе – предлагать повторный ввод значения в двоичном формате.

**Вариант №10.** Наименования времени суток: утро, полдень, день, вечер, полночь, ночь; записаны на английском языке или на транслитерации в переменную перечислимого типа «*DayTime*» в хронологическом порядке. Целочисленные связки наименований времени суток в десятичном формате аналогичны битовым флагам из диапазона [12810, 25610, 51210, 102410, 204810, 409610] и закреплены за наименованиями времени суток в обратной последовательности. При использовании оператора переключения и циклической конструкции по известному множеству значений организовать доброжелательное и корректное приветствие для каждого из перечисленных наименований «*DayTime*» и вывести такие приветствия в консоль. Перемещение между значениями переменной перечислимого типа организовать при использовании только битовых операций.

**Вариант №11.** Наименования размеров верхней одежды: *XS*, *S*, *M*, *L*, *XL*, *XXL*; записаны в переменную перечислимого типа «*SizeNames*» и связаны с десятичным числовым диапазоном [-210…310]. Вывести без использования оператора переключения наименования размеров одежды в зависимости от строковой команды, выдаваемой пользователем, либо по возрастанию, либо по убыванию при использовании циклической конструкции по известному диапазону значений.

**Вариант №12.** Наименования состояний работоспособности технического устройства: включено, выключено, исправно, неисправно, на ремонте; собраны в переменной перечислимого типа «*WorkState*» и записаны либо на английском языке, либо на транслитерации. За наименованиями закреплены значения, записанные в коде Грея, указанные в обратной последовательности. Вывести наименования состояний работоспособности технического устройства в порядке, регламентированном постановкой задачи. К решению задачи вывода наименований состояний на кириллице применить оператор переключения. Вывод наименований в консоль выполнить при использовании циклической конструкции с постусловием.

**Вариант №13.** Переменная перечислимого типа «*KalininskayaLine*» содержит в качестве наименований станции Калининской линии Московского метрополитена, записанные на транслитерации или на английском языке. Предусмотреть вывод в консоль наименований стаций без использования оператора переключения. Реализовывать вывод наименований станций в естественном порядке их следования друг за другом при использовании циклической конструкции с постусловием по нажатии на «*Enter*». По нажатии на «*Esc*» – прерывать работу циклической конструкции. При совершении поездки между станциями в прямом направлении необходимо использовать операцию инкрементирования, при движении между станциями в обратном направлении – использовать операцию декрементирования. Смена направления происходит только по достижении конечной станции. Оператор на принудительную смену направления повлиять никак не может.

**Вариант №14.** Определить по названию дня недели «*Days*», выходной ли это или рабочий день. Нужно определить перечислимый тип со значениями дней недели, объявить переменную, относящуюся к этому типу, и в программе присвоить значение переменной. Затем проверять переменную: если значение суббота или воскресенье – выдавать сообщение «Выходной!», иначе «Работаем!».

**Вариант №15.** В зависимости от месяца «*Calendar*» года выдавать сообщение «холодно» – если зимний месяц, и «тепло» – если летний. Нужно определить перечислимый тип со значениями названий месяцев года, объявить переменную, относящуюся к этому типу, и в программе присвоить значение переменной (из значений типа). Затем проверять переменную: если значение относится к зимнему месяцу, – выдавать сообщение «холодно», иначе – «тепло».

1. Пример корректного подбора тестовых примеров в нужном количестве представлен в разделе 7 Сборника задач [Сафронов, А. И. Получение первичных профессиональных умений и навыков научно-исследовательской деятельности: Сборник задач для проведения аудиторных занятий по учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – Москва: РУТ. – 2019. – 91 с. – *EDN SXMWOD*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=46307421*](https://www.elibrary.ru/item.asp?id=46307421). [↑](#footnote-ref-1)
2. То же, что и сноска 1. [↑](#footnote-ref-2)
3. Пункт 6 на странице 391 [Сафронов, А. И. Способы проектирования эргономичных графических пользовательских интерфейсов для интеллектуальных транспортных систем / А. И. Сафронов // Интеллектуальные транспортные системы: Материалы III Международной научно-практической конференции, Москва, 30 мая 2024 года. – Москва: РУТ. – 2024. – С. 385-394. – *DOI 10.30932/9785002446094-2024-385-394*. – *EDN TGVERU*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=68533620*](https://www.elibrary.ru/item.asp?id=68533620). [↑](#footnote-ref-3)
4. Пример оформления отчётной документации [Сафронов, А. И. Составление отчётной документации по решённым задачам алгоритмизации и программирования: Учебно-методическое пособие для проведения аудиторных занятий по учебной практике / А. И. Сафронов, Н. Н. Зольникова, В. Г. Новиков. – Москва: РУТ. – 2018. – 83 с. – *EDN WNAIFP*]. Скачать можно на сайте Национальной Электронной Библиотеки (НЭБ) *elibrary.ru*: [*https://www.elibrary.ru/item.asp?id=46271697*](https://www.elibrary.ru/item.asp?id=46271697). [↑](#footnote-ref-4)
5. Правила изложены в разделе 8 того же учебно-методического пособия, что указано по сноске 4. [↑](#footnote-ref-5)
6. То же, что и сноска 1. [↑](#footnote-ref-6)
7. То же, что и сноска 1. [↑](#footnote-ref-7)
8. То же, что и сноска 3. [↑](#footnote-ref-8)