МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГТУ»)

Факультет

Кафедра Систем управления и информационных технологий в строительстве

По дисциплине: Основы программирования и алгоритмизации

Тема: «Принципы построения алгоритмов»

Разработал студент Е.А. Ледовской

Подпись, дата Инициалы, фамилия

Руководитель Н.В. Акамсина

Подпись, дата Инициалы, фамилия

Нормоконтролер Н.В. Акамсина

Подпись, дата Инициалы, фамилия

Защищена \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата

Воронеж

2024

ПОСТАНОВКА ЗАДАЧИ

Цель задачи: научиться строить алгоритмы, написать программу в соответствии с заданием и оформить отчёт о проделанной работе.

Вариант задания: 7

Задание: *Мальчик может бегать в три раза быстрее, чем ходить. Скорость его ходьбы равна 4 км/час. Он принял участие в марафонском забеге, но сошел с дистанции, пробежав только х км. Сколько времени он затратил на преодоление этого расстояния?*

ВЫПОЛНЕНИЕ ЗАДАНИЯ

1. **Построение алгоритма**

Для начала построим алгоритм работы программы. Нужно найти время, которое пробежал мальчик на марафоне. Скорость ходьбы равна 4, а скорость бега в 3 раза больше скорости ходьбы. Дистанция, пройденная мальчиком, задается пользователем. Так для нахождения времени нам понадобится найти частное дистанции и скорости бега. Результатом будет вывод полученного времени.

На основе представленного в текстовом варианте алгоритма сделаем блок-схему. Для реализации схемы воспользуемся функциями Word. Ориентируясь на текстовый вариант, составим визуальный алгоритм. (Рис.1)

Начало

Начальные данные: скорость ходьбы

Получение пройденной дистанции

Нахождение скорости бега

Расчет времени, которое пробежал мальчик

Вывод полученного времени

Конец

Рисунок 1 – линейная блок-схема

1. **Написание программы**

Алгоритм работы программы продуман. Остается реализовать его в программе. Для этого нужно определиться со средой разработки. В данном случае это будет *Visual Studio.*

Открываем среду разработки, создаем файл «Task4.*c*» (Рис. 2) и начинаем работать. Подключаем нужные директивы через #include (в случае задания – *<stdio.h>* и *<locale.h>*). Объявляем функцию *main()*, отвечающую за работу всего кода, в ней подключим русский язык и создадим переменную walk с изначально заданным значением. (Рисунок 3)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 2 – создание проекта Рисунок 3 – объявление функции переменной

Добавляем переменные: дистанция, время, скорость бега и формулу для её нахождения. (Рис.4)

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Рисунок 4 – расчет бега и переменные

Сделаем запрос к пользователю при помощи двух функций: *puts()* (для вывода текста) и *scanf()* (для получения значения). Должно получиться следующим образом (Рисунок 5 и 6).



Рисунок 5 – запрос к пользователю для введения переменной.

Изображение выглядит как текст, снимок экрана, Шрифт, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 6 – запрос к пользователю для введения переменной.

Задаем переменной time формулу для расчета времени, затраченного на прохождение дистанции. (Рисунок 7).



Рисунок 7 – Расчет времени.

Теперь воспользуемся функцией printf() для вывода результата на экран.(Рис. 8)



Рисунок 8 – использование функции printf().

Приводим программу в подходящий вид и запускаем. (Рис. 9)

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 9 – итоговый код программы.

Зададим случайные числа при вводе в консоле. Например, 3, 56.3 и 108. Смотрим на получившийся результат. (Рис. 10, 11, 12)

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 10 – результат первого примера

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 11 – результат второго примера

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

Рисунок 12 – результат третьего примера

Сохраняем файл на GitHub для удаленного доступа в будущем.

ВЫВОД

В работе научились строить алгоритмы, написали программу в соответствии с заданием и оформили отчёт о проделанной работе. Проверили и отладили программу, а также сохранили её на GitHub.