

В алгоритмическом языке нет понятия глобальных переменных. Это сознательное ограничение, которое нельзя снимать мимоходом. Глобальные переменные, по нашему мнению, – это принципиальное расширение класса функциональных ($f(\text{аргумент}) \rightarrow \text{результат}$) алгоритмов на так называемые алгоритмы работы во внешней обстановке (задаваемой глобальной переменной). Нам представляется, что алгоритмы работы в обстановке, хотя они с точки зрения житейского опыта даже предшествуют функциональным алгоритмам (например, алгоритм перехода улицы), более трудны для формализации и систематического изучения. В то же время в классе алгоритмов работы с величинами можно практически вплотную подойти к понятию глобальной переменной, разрешая некоторым величинам быть, одновременно, аргументом и результатом алгоритма.

Это особенно годится для алгоритмов работы с табличными величинами (например, алгоритмы сортировки).

Список использованных источников

1. Ершов, А. П. Два облика программирования / А. П. Ершов – Кибернетика, № 6. – 1982. – С. 122–123.
2. Ершов, А. П. Предварительные соображения о лексиконе программирования. – Проблемы кибернетики и вычислительной техники, вып. 1 / А. П. Ершов – М.: Наука. – 1985.
3. Основы информатики и вычислительной техники. Пробное учебное пособие для 9-го кл. средней школы. Под ред. А. П. Ершова и В. М. Монахова. – М.: Просвещение, 1985.

УДК 655.56

Инклюзивный дизайн приложений

Гурская Д. А., студент

Василевская В. А., студент

Белорусский национальный технический университет

Минск, Республика Беларусь

Научный руководитель: к.т.н., доцент Дробыш А. А.

Аннотация:

В данной статье рассматривается понятие инклюзивного дизайна приложений и его назначение. Также в данной статье выделены основные правила инклюзивного дизайна при создании приложений.

Инклюзивный, или универсальный дизайн – метод проектирования, который считается с потребностями и возможностями максимального количества людей. Вместо того чтобы ориентироваться на гипотетического пользователя, инклюзивный дизайн заботится о широком круге людей, создавая интерфейсы для совместного использования отдельным продуктом.

Инклюзивный дизайн не работает на определенную группу, он ориентируется на особые предпочтения и общие ситуации. Дизайн, который предназначен только для человека с повышенными потребностями, – не инклюзивный, потому что исключает других людей. Это лишь вспомогательная технология.

Люди с инвалидностью регулярно сталкиваются с трудностями: не все подъезды оборудованы пандусами, дверные проемы и двери лифтов зачастую слишком узкие для колясок, а рельефная плитка для незрячих людей – совсем редкость. Похожие проблемы есть и в цифровых продуктах: иногда пользователям с плохим зрением заказать еду в приложении так же трудно, как добраться до магазина и самостоятельно ее купить.

Основные правила инклюзивного дизайна:

1. Контраст. Проблемы со зрением в той или иной мере испытывают более 2 млрд человек. А для людей с ослабленным зрением важна контрастность. Для этого существует стандарт контрастности текста WCAG. Этот стандарт принят для текста, поэтому логотипы и иллюстрации не должны ему соответствовать.

Проверить, соответствует ли дизайн рекомендованному WCAG контрасту, можно в онлайн-сервисах Coolors или WebAIM или с помощью плагинов Figma – Contrast и Color Contrast Checker.

2. Размер шрифта. Оптимальные значения трудно определить точно, потому что при одинаковом кегле в одних шрифтах буквы будут крупными, а в других – мелкими. В гайдлайне «Сбера» рекомендовано брать не менее 16 пикселей для основного текста. В

стандартах WCAG есть только минимальный интерлиньяж – кегль умножить на 1,5 и никаких других требований.

3. Цвет. Некоторые пользователи хорошо различают текст любого размера, но имеют нарушения цветового восприятия. И если цвет в интерфейсе несет какую-то важную информацию, нужны текстовые подсказки. Например, для обозначения ошибки традиционно используют красный, который не воспринимают люди с дальтонизмом. Поэтому изменения цвета недостаточно – нужно явно написать, что именно пошло не так.

4. Скринридеры. Незрячим людям контраст, размер шрифта, подсказки и возможность увеличить масштаб не помогут. Для таких людей существуют скринридеры – программы и приложения, которые зачитывают текст на экране.

Для корректной работы скринридера важна работа не только дизайнера, но и разработчика. Типы элементов должны быть прописаны в коде – тогда скринридер сможет показать незрячим и слабовидящим людям, где кнопка, где подпись, а где заголовок.

Чтобы скринридер работал корректно, важно также сделать кнопки и заголовки информативными – чтобы они помогли без контекста разобраться в устройстве интерфейса.

5. Делать контент проще для восприятия. Тексты, напечатанные декоративными шрифтами, или плохо структурированные веб-страницы трудно воспринимать невыспавшимся людям и дислексикам. Упрощение дизайна улучшает доступность.

6. Помочь пользователям кликать и скроллить. Людям с двигательными нарушениями может быть трудно задействовать сразу две руки или точно попасть в маленькую кнопку. Если элемент неизбежно мелкий, вокруг него нужно больше пустого места.

Пространство вокруг элементов тоже должно быть кликабельным. Прицельно попасть пальцем или мышкой прямо в кнопку может быть трудно всем пользователям.

С точки зрения разработки важно корректно сделать управление интерфейсом с клавиатуры – начиная от скроллинга и заканчивая выбором кнопок для перехода по ссылкам. Нажимать на кнопки при нарушениях моторики проще, чем управлять мышкой и тачпадом.

7. Помнить про другие нарушения. У пользователя может быть полная или частичная потеря слуха. Обычно это не мешает работе с

сайтами и приложениями – многие люди добровольно отключают звук на своих гаджетах. Но в онлайн-игре или видео важно позаботиться о субтитрах. А если в интерфейсе какого-нибудь сервиса звуковой сигнал обозначает новое сообщение или успешную оплату, это должно еще и визуально отражаться на экране.

Активная анимация может спровоцировать приступ у людей с эпилепсией. В WCAG прописана возможность отключения анимации, которая не влияет на понимание интерфейса. В гайдлайне Google рекомендуется автоматически отключать анимацию контента спустя пять секунд, а если необходимо мигание, ограничить его до трех кадров в секунду.

Но всегда стоит помнить, что доступность для всех недостижима. Даже если дизайн сделан строго по принципам WCAG, нельзя с уверенностью сказать, что им сможет пользоваться любой. Все проблемы и неочевидные мелочи учесть невозможно – интерфейс может стать доступнее для одних и сложнее для других.

Список использованных источников

1. Курносова, С. А. Инклюзивная компетентность: концептуальное обоснование / С. А. Курносова – Типография ООО «Печатный двор». – Челябинск. – 2015.
2. Анищенко, Ю. WWW-редактор: больше, чем просто HTML / Ю. Анищенко. – 2012.
3. Шрейдер, Ю. А. Информационные процессы и информационная среда / Ю. А. Шрейдер. – Науч.-техн. информ. – 1976.

УДК 621.762.4

No-code разработка приложений

Каминская И. В., студент

Бабицкая Э. С., студент

Белорусский национальный технический университет,

г. Минск, Республика Беларусь

Научный руководитель: к.т.н., доцент Дробыш А. А.