class Menu {  
 constructor(options) {  
 // Обязательные параметры  
 this.button = *document*.querySelector(options.buttonSelector);  
 this.menu = *document*.querySelector(options.menuSelector);  
  
 // Настройки с значениями по умолчанию  
 this.animationDuration = options.animationDuration || 300;  
 this.breakpoint = options.breakpoint || 992; // Дефолтный breakpoint (px)  
 this.htmlElement = options.htmlElementSelector  
 ? *document*.querySelector(options.htmlElementSelector)  
 : *document*.body;  
  
 // Состояние  
 this.isAnimating = false;  
 this.isDesktop = this.checkViewport(); // Проверяем текущий размер экрана  
  
 this.init();  
 }  
  
 init() {  
 if (!this.button || !this.menu) {  
 *console*.error('Не найдены кнопка или меню!');  
 return;  
 }  
  
 // Применяем стили только для mobile  
 if (!this.isDesktop) {  
 this.setMobileStyles();  
 }  
  
 this.setupEvents();  
 this.addResizeListener();  
 }  
  
 // Устанавливает стили для мобильной версии  
 setMobileStyles() {  
 this.menu.style.transition = `  
 opacity ${this.animationDuration}ms ease,   
 transform ${this.animationDuration}ms ease  
 `;  
 this.menu.style.opacity = '0';  
 this.menu.style.transform = 'translateY(-10px)';  
 this.menu.style.display = 'none';  
 }  
  
 // Проверяет, соответствует ли viewport desktop-размеру  
 checkViewport() {  
 return *window*.matchMedia(`(min-width: ${this.breakpoint}px)`).matches;  
 }  
  
 // Добавляет обработчик изменения размера окна  
 addResizeListener() {  
 *window*.addEventListener('resize', () => {  
 const wasDesktop = this.isDesktop;  
 this.isDesktop = this.checkViewport();  
  
 // Если режим изменился (desktop ↔ mobile)  
 if (wasDesktop !== this.isDesktop) {  
 if (this.isDesktop) {  
 this.resetDesktopStyles();  
 } else {  
 this.setMobileStyles();  
 }  
 }  
 });  
 }  
  
 // Сбрасывает стили при переходе в desktop-режим  
 resetDesktopStyles() {  
 this.menu.style.removeProperty('transition');  
 this.menu.style.removeProperty('opacity');  
 this.menu.style.removeProperty('transform');  
 this.menu.style.removeProperty('display');  
 this.htmlElement.style.removeProperty('overflow');  
 }  
  
 setupEvents() {  
 this.button.addEventListener('click', (event) => {  
 event.stopPropagation();  
 if (!this.isDesktop) this.toggleMenu();  
 });  
  
 *document*.addEventListener('click', (event) => {  
 if (this.isDesktop) return;  
 if (!this.menu.contains(event.target) && !this.button.contains(event.target)) {  
 this.closeMenu();  
 }  
 });  
  
 *document*.addEventListener('keydown', (event) => {  
 if (this.isDesktop) return;  
 if (event.key === 'Escape') this.closeMenu();  
 });  
 }  
  
 toggleMenu() {  
 if (this.menu.style.display === 'none' || !this.menu.style.display) {  
 this.openMenu();  
 } else {  
 this.closeMenu();  
 }  
 }  
  
 openMenu() {  
 if (this.isAnimating) return;  
 this.isAnimating = true;  
  
 this.menu.style.display = 'flex';  
 this.htmlElement.style.overflow = 'hidden';  
  
 requestAnimationFrame(() => {  
 this.menu.style.opacity = '1';  
 this.menu.style.transform = 'translateY(0)';  
 });  
  
 setTimeout(() => {  
 this.isAnimating = false;  
 }, this.animationDuration);  
 }  
  
 closeMenu() {  
 if (this.isAnimating || this.menu.style.display === 'none') return;  
 this.isAnimating = true;  
  
 this.menu.style.opacity = '0';  
 this.menu.style.transform = 'translateY(-20px)';  
  
 setTimeout(() => {  
 this.menu.style.display = 'none';  
 this.htmlElement.style.overflow = '';  
 this.isAnimating = false;  
 }, this.animationDuration);  
 }  
}  
  
export default Menu;