1. Care din următoarele sunt evenimente într-un program pilotat de evenimente?
2. Click de mouse
3. Deplasarea mouse-ului (mouse move)
4. Acționarea barei de derulare
5. Nici unul
6. Testarea aplicațiilor pilotate de evenimente este mai simplă decât a programelor care folosesc alte paradigme.
7. Adevărat
8. Fals
9. Care din cele enumerate nu sunt controale?
10. Operatorul if
11. Acționare de tastă
12. Elementele interfeței grafice mai sunt cunoscute sub numele de
13. Controale
14. Handlere
15. Evenimente
16. Obiecte
17. Bifați variantele corecte pentru ID de elemente GUI
18. moveBtn
19. 3MOVE
20. Move btn
21. MOVEBTN
22. Bifați varianta cu cea mai corectă definiție pentru funcție de tipul callback
23. Funcție specificată ca parte a unui Event Listener; acesta este scris de programator, dar apelat de sistem ca rezultat al unui declanșator de eveniment
24. Activitatea de programare care să facă ca aplicația dvs să răspundă la evenimente prin declanșarea funcțiilor.
25. Program conceput pentru a rula blocuri de cod sau funcții ca răspuns la evenimentele specificate (ca de exemplu, un clic de mouse)
26. O comandă (onEvent în App Lab) care poate fi configurată pentru a declanșa o funcție atunci când un anumit tip de eveniment are loc pe un anumit element GUI.
27. Bifați varianta cu cea mai bună definiție a noțiunii de program pilotat de evenimente:
28. Funcție specificată ca parte a unui Event Listener; acesta este scris de programator, dar apelat de sistem ca rezultat al unui declanșator de eveniment
29. Activitatea de programare care să facă ca aplicația dvs să răspundă la evenimente prin declanșarea funcțiilor.
30. Program conceput pentru a rula blocuri de cod sau funcții ca răspuns la evenimentele specificate (ca de exemplu, un clic de mouse)
31. O comandă (onEvent în App Lab) care poate fi configurată pentru a declanșa o funcție atunci când un anumit tip de eveniment are loc pe un anumit element GUI.
32. Bifați dispozitivele de intrare:
33. Tastatura
34. Ploterul
35. Imprimanta
36. Mouse-ul
37. GUI este abrevierea pentru
38. Nici una
39. Interfața grafică uniformă
40. Interfața grafică unimodală
41. Interfața grafică a utilizatorului
42. Care din următoarele este/sunt exemplu/exemple de element al interfeței grafice:
43. Click de mouse
44. Nici una
45. Buton
46. Imagine
47. Ar fi correct să descriem un sistem de operare ca o aplicație event driven?
48. Adevărat
49. Fals
50. O aplicație pilotată de evenimente este obligată să proceseze toate evenimentele care sunt generate în timpul sesiunii de lucru cu această aplicație?
51. Doar cele care vrea programatorul
52. Doar cele care vrea beneficiarul
53. Toate variantele sunt incorecte
54. Beneficiarul împreună cu programatorul vor stabili care evenimentne trebuie tratate
55. Din care categorie de elemente sunt butoanele?
56. GUI
57. Control
58. Comandă
59. Meniu
60. Bifați variantele care sunt motivații pentru a scrie programe pilotate de evenimente.
61. Codul event driven este mai simplu și mai ușor de elaborate
62. Limbajele event driven sunt mai ieftine pentru dezvoltarea de cod
63. Programatorii scriu mai ușor cod folosind limbaje PPE
64. Folosind limbaje din categoria event driven este mai ușor de dezvoltat secții modulare de cod.
65. Care din caracteristicile enumerate este proprietate a programelor pilotate de evenimente?
66. Dinamism
67. Nici una
68. Bine structurate
69. Execuții impredictibile asociate cu anumite provocări
70. Este oare adevărată afirmația: Odată cu apariția PPE nu mai sunt necesare limbajele de programare imperativă.
71. Adevărat
72. Fals
73. Care este principalul obiect (în sens POO) al unei aplicații PPE?
74. Butonul start
75. Funcția WinMain
76. Meniul
77. Fereastra
78. Bifați varianta cu cea mai bună definiție a noțiunii de gestionare a unui eveniment (Event handling):
79. Funcție specificată ca parte a unui Event Listener; acesta este scris de programator, dar apelat de sistem ca rezultat al unui declanșator de eveniment
80. Activitatea de programare care să facă ca aplicația dvs să răspundă la evenimente prin declanșarea funcțiilor.
81. Program conceput pentru a rula blocuri de cod sau funcții ca răspuns la evenimentele specificate (ca de exemplu, un clic de mouse)
82. comandă (onEvent în App Lab) care poate fi configurată pentru a declanșa o funcție atunci când un anumit tip de eveniment are loc pe un anumit element GUI.
83. Bifați varianta cu cea mai bună definiție a noțiunii de gestionare a unui ascultător de eveniment (Event Listener):
84. Funcție specificată ca parte a unui Event Listener; acesta este scris de programator, dar apelat de sistem ca rezultat al unui declanșator de eveniment
85. Activitatea de programare care să facă ca aplicația dvs să răspundă la evenimente prin declanșarea funcțiilor.
86. Program conceput pentru a rula blocuri de cod sau funcții ca răspuns la evenimentele specificate (ca de exemplu, un clic de mouse)
87. comandă (onEvent în App Lab) care poate fi configurată pentru a declanșa o funcție atunci când un anumit tip de eveniment are loc pe un anumit element GUI.
88. Toate evenimentele într-un program provin de la utilizator:
89. Adevărat
90. Fals
91. Bifați afirmațiile **incorecte**:
92. Evenimentele sunt întreruperi generate de un periferic sau hardware de sistem
93. Evenimentele sunt mesaje generate de o altă aplicație
94. Nici unul din cele enumerate
95. Evenimentele sunt mesaje generate de sistemul de operare
96. Care informații sunt transmise handler-ului de eveniment de către dispecer?
97. Informațiile transmise handler-ului evenimentului de către dispecer vor varia, dar vor include date suficiente pentru a permite codului care tratează evenimentul să ia toate măsurile necesare.
98. Informațiile transmise handler-ului evenimentului de către dispecer vor specifica cine este stăpânul evenimentului
99. Informațiile transmise handler-ului evenimentului de către dispecer vor specifica cine este utilizatorul aplicației
100. Informațiile transmise handler-ului evenimentului de către dispecer vor specifica cine așteaptă rezultatul procesării evenimentului.
101. Pentru ce este concepută o aplicație bazată pe evenimente?
102. Pentru a dezvolta aplicații simple
103. Pentru a dezvolta aplicații ușor de testat
104. PPE este o extensie a programării controlate de întreruperi, de tipul celor folosite în SO sau în sistemele incorporate
105. O aplicație bazată pe evenimente este concepută pentru a recepționa evenimentele pe măsura apariției și a le procesa, folosind procedure adecvate de tratare.
106. Bifați caracteristicile unui eveniment:
107. Ceva ce are loc (se întîmplă)
108. Este un element al interfeței grafice
109. Poate fi declanșat în orice moment al execuției programului
110. Poate fi cauzat de utilizator sau calculator
111. Care este destinația funcției LoadIcon?
112. Încarcă o pictogramă
113. Încarcă o pictogramă în zona client
114. Permite procesarea unei iconițe
115. Dă posibilitatea să modificăm o pictogramă
116. Care opțiuni sunt caracteristici pentru funcția DefWindowProc?
117. Execută operațiile prestabilite
118. Dacă un mesaj nu are un event handler pentru tratare în procedura de fereastră, el este returnat acestei funcții.
119. Definește procedura de fereastră a aplicației
120. Împreună cu sistemul de operare tratează mesajele, pentru care programatorul nu a scris cod.
121. Bifați opțiunea corectă: Care este tipul funcției WinMain?
122. WINAPI
123. CALLBACK
124. Cîte cîmpuri are structura RECT?
125. 5
126. 6
127. 2
128. 4
129. Care este destinația funcției UpdateWindow?
130. Forțează aplicația să-și actualizeze zona client
131. Trimite un mesaj WM\_PAINT procedurii de fereastră
132. Toate răspunsurile sunt greșite
133. Actualizează mediul de execuție al ferestrei
134. Este recomandat să nu păstrați variabilele handle ale contextelor de dispozitiv de la un mesaj la altul. Totuși există excepții.
135. Cu excepția CreateDC
136. Cu excepția ClassDC
137. Cu excepția CreateCompatibleDC
138. Cu excepția OwnDC
139. Când este apelată funcția GetDC?
140. Obținem variabila handle contextului de dispozitiv apelând funcția GetDC atunci când avem nevoie de DC în alte scopuri, cum ar fi obținerea unoe informații despre contextual de dispozitiv.
141. Pentru a obține variabila handle a unui context de dispozitiv în timpul prelucrării altor mesaje decît WM\_PAINT apelând funcția GetDC
142. Este obligator să apelăm funcția ReleaseDC atunci cînd nu mai avem nevoie de variabila handle a contextului de dispozitiv.
143. Nu este obligator să apelăm funcția ReleaseDC atunci cînd nu mai aveți nevoie de variabila handle a contextului de dispozitiv, deoarece sistemul de operare oricum va elibera hdc
144. Cîte categorii de fonturi există din punctul de vedere al dimensiunii caracterelor?
145. 10
146. 2
147. 5
148. Mai mult de 100
149. Dacă valoarea lui cxChar este 10, care va fi valoarea lui cxCaps calculată folosind relația cxCaps=(tm.tmPitchAndFamily & 1 ? 3 : 2) \* cxChar/2; pentru un font cu dimensiune variabilă?
150. 20
151. 10
152. Toate sunt greșite
153. 15
154. Care este valoarea de adevăr a afirmației: Programul poate să valideze orice porțiune dreptunghiulară din zona client, apelând funcția ValidateRect. Dacă în urma acestui apel întreaga zonă invalidă este validată, toate mesajele WM\_PAINT aflate în coada de așteptare sunt șterse.
155. Adevărat
156. Fals
157. Din ce categorie fac parte funcțiile BeginPaint și EndPaint?
158. Nici una
159. Funcții care lucrează cu obiecte GDI
160. Funcții care obțin informații despre contextul de dispozitiv
161. Funcții care desenează ceva
162. Bifați afirmațiile corecte:
163. În mod prestabilit, desenarea se face folosind ca unități de măsură pixelii
164. Modurile de mapare GDI vă permit să desenați folosind ca unitate de măsură incii (sau fracțiuni de inci), milimetri sau orice altă unitate de măsură
165. Modurile de mapare GDI vă permit să desenați folosind ca unitate de măsură kilogramul
166. Modurile de mapare GDI vă permit să desenați folosind ca unitate de măsură metrul cub
167. Bifați opțiunile care reprezintă valori prestabilite de atribute ale DC.
168. MM\_TEXT
169. BLACK\_BRUSH
170. WHITE\_BRUSH
171. WHITE\_PEN
172. Bifați opțiunile care reprezintă stiluri iPenStyle corecte:
173. PS\_DOTDOT, PS\_NULL, PS\_INSIDEFRAME
174. PS\_SOLID, PS\_DASH, PS\_DOT, PS\_EXSIDEFRAME
175. PS\_SOLID, PS\_DASH, PS\_DOT, PS\_DASHDOT
176. PS\_DASHDOTDOT, PS\_NULL, PS\_INSIDEFRAME
177. Bifați afirmația/afirmațiile corectă/corecte referitoare la funcțiile CreatePen și CreatePenIndirect
178. Creează penițe logice care nu au nici o legătură cu contextual de dispozitiv până cînd nu apelați funcția SelectObject
179. Nu au nevoie de variabila handle a contextului de dispozitiv
180. Au nevoie de variabila handle a contextului de dispozitiv
181. Puteți să folosiți aceeași peniță logică pentru mai multe dispositive, cum ar fi imprimanta și ecranul
182. Bifați afirmațiile corecte:
183. Curbele Bezier au puncta de singularitate
184. Puteți să manipulați curba pînă cînd ajunge la o formă apropiată de cea dorită
185. Curba este întotdeauna tangent la linia trasată de la primul punct final, la primul punct de control și are întotdeauna aceeași direcție cu această linie
186. Curbele Bezier sunt întotdeauna ancorate în cele două puncte finale
187. Cum poate fi aflată starea curentă a unei taste?
188. Folosind funcția GetAsyncKeyState
189. Cu ajutorul lui IParam
190. Cu ajutorul GetKeyState
191. Cu ajutorul wParam
192. Bifați opțiunile care corespund afirmațiilor adevărate, despre tastele moarte (dead keys)
193. Nu pot crea singure caractere
194. Sunt folosite pentru adăugarea semnelor diacritice la o literă
195. Programul este obligat să prelucreze mesaju WM\_DEADCHAR
196. Atunci când utilizatorul apasă o tastă moartă, procedura de fereastră nu primește nici un mesaj
197. Ce reprezintă o apăsare de tastă legată de funcții de sistem?
198. Combinații cu tasta Ctrl
199. Combinații cu tasta Alt
200. Combinații cu tasta Shift
201. Altceva, lipsă aici
202. Bifați opțiunea care corespunde valorii de adevăr a afirmației: Preluarea codului de mouse pentru tratarea mesajelor de la barele de derulare este o soluție foarte bună, deoarece dacă vom dori vreodată să modificăm logica de funcționare a barelor de derulare vom face aceleași modificări și în codul de tratare a mesajului WM\_KEYDOWN
203. Adevărat
204. Fals
205. Care este valoarea de adevăr a afirmației: La creare, cursorul de editare nu este afișat.
206. Adevărat
207. Fals
208. Câte mesaje de mouse cu trei butoane sunt destinate pentru zona non-client a ferestrei?
209. 8
210. 9
211. 10
212. 11
213. Bifați afirmațiile corecte:
214. Valoarea parametrului wParam pentru un mesaj de mouse indică starea tastei Ctrl
215. Valoarea parametrului wParam pentru un mesaj de mouse indică starea tastei Shift
216. Valoarea parametrului wParam pentru un mesaj de mouse indică starea butoanelor mouse-ului
217. Valoarea parametrului wParam pentru un mesaj de mouse indică starea tastei Alt
218. Bifați pe valoarea de adevăr a afirmației: dacă în stilul ferestrei nu este inclus identificatorul CS\_DBLCLKS și utilizatorul execute de două ori click pe butonul din stînga al mouse-ului într-o succesiune rapidă, procedura de fereastră va recepționa următoarele mesaje: WM\_LBUTTONDOWN, WM\_LBUTTONUP, WM\_LBUTTONBLCLK și WM\_LBUTTONUP
219. Adevărat
220. Fals
221. Bifați pe valoarea de adevăr a afirmației: atunci cînd deplasați indicatorul mouse-ului peste zona client a unei ferestre se generează un mesaj WM\_MOUSEMOVE pentru fiecare pixel peste care trece indicatorul.
222. Adevărat
223. Fals
224. Bifați pe valoarea de adevăr a afirmației: Procedura de fereastră poate să primească un mesaj WM\_LBUTTONDOWN fără să primească mesajul WM\_LBUTTONUP corespunzător când butonul mouse-ului este eliberat după ce indicatorul a fost mutat în altă fereastră întotdeauna.
225. Adevărat
226. Fals
227. Care sunt obligațiile dispecerului atunci cînd are loc un eveniment?
228. Cînd are loc un eveniment, dispecerul trebuie să apeleze handler-ul corespunzător
229. Când are loc un eveniment, dispecerul trebuie să oprească finalizarea procesului de tratare a evenimentului anterior.
230. Când are loc un eveniment, dispecerul trebuie să determine tipul evenimentului
231. Când are loc un eveniment, dispecerul trebuie să accelereze finalizarea procesului de tratare a evenimentului anterior
232. Bifați pe resurse Windows
233. Penițe
234. Pictograme
235. Șiruri de caractere
236. Imagini bitmap
237. Cursoare
238. Ce se întîmplă în cazul în care dispecerul întîlnește un eveniment pentru care nu există un handler adecvat?
239. Dispecerul poate ignora un astfel de eveniment, lăsînd tratarea în seama SO
240. Utilizatorul în mod absolut ogligator se va revolta și va fi în drept să facă acest lucru
241. Poate fi generate o excepție, iar evenimentul să fie tratat de sistemul de operare
242. Compilatorul va semnaliza o eroare de compilare
243. Ce se întîmplă la execuția funcției CreateWindow?
244. Este create și afișată fereastra
245. Sistemul de operare atribuie o valoare variabilei hwnd
246. Este create fereastra
247. Primul mesaj WM\_CREATE este trimis procedurii de tratare a mesajelor
248. Bifați regulile care trebuie respectate pentru folosirea obiectelor GDI?
249. La sfîrșitul programului ștergeți toate obiectele GDI pe care le-ați creat.
250. Nu ștergeți obiectele GDI în timp ce sunt selectate într-un context de dispozitiv valid
251. Este interzisă crearea de obiecte pe care programul nu le folosește
252. Nu ștergeți obiectele de stoc
253. Din care cauză putem fi siguri că sosirea mesajului WM\_TIMER nu va întrerupe o operație de prelucrare a unui mesaj curent?
254. Deoarece asta nu se poate întîmpla
255. Întrebare incorectă
256. De-atîta
257. Deoarece programele windows preiau mesajele WM\_TIMER din coada de așteptare
258. Bifați continuările corecte pentru afirmația: O parte dintre valorile din contextual de dispozitiv sunt attribute grafice. Aceste attribute defines unele particularități privind modul de lucru al unor funcții de desenare din interfața GDI. În cazul funcției TextOut, de exemplu, atributele contextului de dispozitiv determină …
259. Culoarea textului
260. Modul de mapare a coordonatelor x și y în zona de client a ferestrei
261. Fontul folosit de windows pentru afișarea textului
262. Culoarea fondului zonei client
263. Bifați afirmațiile corecte:
264. Testarea în ppe este mai complicată fiindcă programatorul nu cunoaște fluxul de execuție a programului
265. Testarea în ppe este mai simplă fiindcă programatorul cunoaște fluxul de execuție a programului
266. Testarea în ppe este mai simplă fiindcă fluxul de execuție nu se schimbă de la o execuție la alta
267. Testarea în ppe este mai complicată fiindcă fluxul de execuție se schimbă de la o execuție la alta
268. Bifați afirmațiile corecte:
269. Handlerul de eveniment poate schimba starea sistemului
270. Handlerul de eveniment nu are nimic în comun cu operatorul de eveniment
271. Handlerul de eveniment produce un răspuns vizual pentru a informa sau direcționa utilizatorul
272. Handlerul de eveniment este un mic bloc de cod procedural care tratează un eveniment
273. Bifați opțiunea corectă pentru afirmația: windows se ocupă de modul de utilizare a mouse-ului și a tastaturii pentru barele de derulare.
274. Adevărat
275. Fals
276. Bifați valoarea de adevăr corectă a afirmației: dacă vreți ca tastele de deplasare să dubleze unele dintre funcțiile barelor de derulare trebuie să furnizați explicit o metodă de realizare a acestui lucru.
277. Adevărat
278. Fals
279. Care este valoarea de adevăr a afirmației : nu puteți crea propriile controale de tip fereastră descendent dacă folosim clasele de fereastră (și procedurile specifice) predefinite, clase pe care programele le folosesc pentru crearea tuturor controalelor de tip fereastră descendent.
280. Adevărat
281. Fals
282. Bifați mesajele de la barele de derulare:
283. WM\_VSCROLL
284. WM\_HSCROLLBAR
285. WM\_HSCROLL
286. WM\_VSCROLLBAR
287. Bifați opțiunile care reprezintă categorii de primitive GDI:
288. TEXT
289. Imagini bitmap
290. Suprafețe goale
291. Linii și curbe
292. Stilul WS\_OVERLAPPEDWINDOW … Bifați variantele care conduc la afirmații corecte
293. Este pentru stilul clasei de fereastră
294. Permite reacoperirea ferestrelor
295. Toate răspunsurile sunt corecte
296. Este stil de fereastră setat în al treilea parametru formal al funcției CreateWindow
297. Bifați opțiunile care conțin componente structurale ale GDI:
298. GDI este formată din cîteva sute de apeluri de funcții
299. Nici una
300. GDI este formată din cîteva tipuri de date și structuri associate
301. GDI este formată din cîteva macroinstrucțiuni
302. Care este valoarea de adevăr a afirmației: BLACK\_PEN desenează o linie compactă, de culoare neagră, cu grosimea de un milimetru, indifferent de modul de mapare.
303. Adevărat
304. Fals
305. Care este destinația funcției TranslateMessage?
306. Traduce mesajul obținut de GetMessage
307. Niciuna din celelalte opțiuni
308. Introduce în firul de așteptare mesaje WM\_CHAR pentru acționările de taste care conțin simboluri
309. Convertește unele dintre mesajele de la tastatură
310. Care este valoarea de adevăr a afirmației: Într-un program controlat de evenimente nu există un flux de control.
311. Adevărat
312. Fals
313. Care este domeniul de valori ale parametrului t din desemnarea parametrică a spline-urilor Bezier?
314. De la 0 pînă la 1 inclusiv capetele [0, 1]
315. De la 0 pînă la 1 exclusiv capetele (0, 1)
316. Nici o variantă
317. De la -ꝏ pînă la +ꝏ
318. Bifați opțiunile cu valoarea de adevăr ADEVĂRAT pentru afirmațiile referitoare la afișarea textului.
319. Funcția GetTextMetrics nu are nevoie de variabila handle a contextului de dispozitiv, deoarece returnează informații despre fontul selectat în contextul de dispozitiv, care este SYSTEM\_FONT
320. Puteți să stabiliți spațiul dintre liniile successive de text pe baza înălțimii unui character și spațiul dintre coloane pe baza lățimii medii a caracterelor din font
321. Dimensiunile caracterelor sunt obținute prin apelarea funcției GetTextMetrics
322. Pentru afișarea mai multor linii de text cu ajutorul funcției TextOut trebuie să determinați dimensiunile caracterelor din fontul folosit.
323. Bifați opțiunile care reprezintă caracteristici ale gestiunii evenimentelor (Event Handling)
324. Nici una în afară de aceasta
325. Acțiune sau set de acțiuni care trebuie executate atunci cînd eveniment are loc
326. A răspunde la un eveniment
327. Răspunsul la declanșarea evenimentelor
328. Programatorul ia în considerație toate evenimentele posibile
329. Bifați opțiunea care corespunde valorii de adevăr a afirmației: Cu ajutorul funcției SendMessage putem transforma mesajele WM\_KEYDOWN în mesaje echivalente WM\_VSCROLL sau WM\_HSCROLL
330. Adevărat
331. Fals
332. Nu întotdeauna
333. În condiții speciale
334. Bifați cauza din care se recomandă ca identificatorilor de meniu IDM să li se atribuie valori consecutive.
335. Managementul procesorului este mult mai efficient pentru codul care conține instrucțiuni switch sau case dacă valorilel atribuite identificatorilor sunt consecutive
336. Compilatorul C optimizează codul care conține instrucțiuni switch sau case mult mai bine dacă valorile atribuite identificatorilor sunt consecutive.
337. Managementul memoriei este mult mai efficient pentru codul care conține instrucțiuni switch sau case, dacă valorile atribuite identificatorilor sunt consecutive
338. Compilarea codului care conține instrucțiuni switch sau case este mult mai eficientă, dacă valorile atribuite identificatorilor sunt consecutive.
339. Bifați opțiunile care reprezintă cîmpuri ale structurii LOGPEN.
340. lopnStyle (UINT)
341. lopn Width (POINT) pentru x
342. lopn Width (POINT) pentru x și y
343. lopnColor (COLORREF)
344. Mesajul WM\_COMMAND indică faptul că utilizatorul a selectat un element permis din meniul ferestrei. Dar același mesaj este trimis și de ferestrele de control descendent. Cum facem diferența?
345. Folosim parametrul lParam
346. Folosind cuvântul cel mai semnificativ al parametrului lParam
347. Folosim valoarea parametrului lParam care este 0 atunci cînd este selectată o opțiune validă de meniu
348. Folosind cuvântul mai puțin semnificativ al parametrului lParam
349. Bifați exemplele de controale:
350. Bara de stare
351. Butoane
352. Bara de titlu
353. Casete de validare
354. Pentru codul de mai jos, bifați afirmațiile corecte

While (GetMessage (&msg, NULL, 0, 0))

{ TranslateMessage (&msg);

DispatchMessage (&msg); }

1. TranslateMessage traduce mesajul
2. DispatchMessage direcționează mesajul către fereastra, care deține Input Focus
3. Este fragmentul de cod care implementează funcțiile dispecerului unui program PPE realizat în C
4. GetMessage preia următorul mesaj din firul de așteptare
5. Este vreo diferență între noțiunile "tact de ceas" și "tact de cronometru"?
6. Sunt noțiuni absolut diferite
7. Este același lucru
8. Depinde de context
9. Nu sunt răspunsuri corecte
10. Bifați afirmațiile corecte:
11. În PPE programul este executat conform intenției programatorului
12. În PPE programatorul stabilește ordinea acțiunilor
13. Nici una din cele enunțate
14. În PPE programul centrat pe calculator
15. În PPE programatorul controlează fluxul de calcul
16. Cum putem afla dimensiunea zonei client? Bifați afirmațiile corecte
17. Prin apelarea funcției GetSystemMetrics cu parametrul SM\_CXFULLSCREEN
18. Prin despachetarea lui lParam, care însoțește mesajele WM\_SIZE
19. Prin apelarea funcției GetSystemMetrics cu parametrul SM\_CYFULLSCREEN
20. Prin prelucrarea meajului WM\_SIZE, care este trimis ferestrei de fiecare data cînd fereastra este redimensionată
21. Bifați afirmațiile adevărate.
22. O anumită resursă este încărcată în memorie numai atunci cînd este cerută de windows.
23. Atunci cînd sistemul de operare windows încarcă în memorie codul și datele unui program în vederea execuției, resursele sunt lăsate, de obicei, pe hard-disk
24. Resursele sunt accesibile imediat prin variabile definite în codul sursă al programului
25. Resursele sunt păstrate în zona de date a programelor executabile
26. Bifati optiunile care reprezinta categoriile de functii GDI
27. Funcții care obțin(sau creează) și eliberează (sau distrug) un context de dispozitiv.
28. Funcții care solicită informații despre contextual de dispozitiv.
29. Funcții care deseneză ceva.
30. Funcții care stabilesc sau obțin attribute ale contextului de dispozitiv.
31. Funcții care lucrează cu obiecte GDI.
32. Ce reprezinta un accelerator?
33. Altceva,lipsă aici
34. Combinații cu tasta Alt
35. Combinații cu tasta Ctrl
36. Combinații cu tasta Shift
37. Bifati consecintele faptului ca, cronometrul Windows are aceeasi rezolutie de 54,925 milisecunde ca si ceasul hardware al calculatorului, pe care de fapt se bazeaza
38. Intervalul de timp pe care îl specificati la apelarea funcției SetTimer este rotunjit la un multiplu întreg de tacturi de ceas.
39. O aplicație Windows nu poate sa primească mesaje WM\_TIMER cu o rată mai mare de 18,2 ori/secundă daca foloseste un singur cronometru.
40. Nici o consecință.
41. Există consecințe,dar sunt listate printer răspunsurile de aici.
42. Bifati raspunsurile corecte la intrebarea: **De ce depinde culoarea spatiilor (intreruperilor) intre liniutele unei linii intrerupte?**
43. De atributul modul de desenare a fondului
44. De atributul culoarea fondului
45. De atributul modul de mapare
46. De parametrul stilul peniței
47. Bifati afirmatiile corecte
48. Pentru toate mesajele de la mouse,parametrul wParam conține poziția mouse-ului.
49. Cuvîntul mai puțin semnificativ al lui IParam conține coordonata mouse-lui pe axa x.
50. Cuvîntul mai semnificativ al lui wParam conține coordonata mouse-lui pe axa y.
51. Pentru toate mesajele de la mouse,parametrul IParam conține poziția mouse-ului.
52. Bifati pe valoarea de adevar a afirmatiei: **Procedura de fereastra nu poate sa primeasca un mesaj WM\_LBUTTONUP fara sa fi primit mai intai un mesaj WM\_LBUTTONDOWN**
53. True
54. False
55. Bifati pe valoarea de adevar a afirmatiei**: Daca vreti ca procedura de fereastra sa primeasca mesaje generate de dublu clic, trebuie sa indicate identificatorul CS\_DBlCKS in stilul de fereastra**
56. True
57. False
58. Bifați pe valoare de adevăr a afirmației: **Poziția și dimensiunea controalelor de tip fereastră descendent sunt definite într-un șablon al casetei de dialog,conținut în fișierul de resurse al programului.**
59. True
60. False
61. Care tipuri de linii poate să deseneze Windows?
62. Linii drepte
63. Linii eliptice
64. Curbe Bezier
65. Linii paralele
66. Bifați opțiunile care reprezintă afirmații corecte referitor la parametrul rgbColor
67. Pentru toate stilurile de penițe ,exceptînd PS\_NULL atunci cand selectati penita în contextul de dispozitiv,Windows conversteste acest parametru la cea mai apropiata culoare pura pe care o poate reprezinta dispozitivul de afisare.
68. Pentru toate stilurile de penițe ,exceptînd PS\_DASH atunci cand selectati penita în contextul de dispozitiv,Windows conversteste acest parametru la cea mai apropiata culoare pura pe care o poate reprezinta dispozitivul de afisare.
69. Pentru toate stilurile de penițe ,exceptînd PS\_INSIDEFRAME atunci cand selectati penita în contextul de dispozitiv,Windows conversteste acest parametru la cea mai apropiata culoare pura pe care o poate reprezinta dispozitivul de afisare.
70. Parametrul rgbColor din functia CreatePan este un numar fară semn reprezintand culoarea penitei.
71. Bifați opțiunile care reprezintă utilizări ale cronometrului
72. Pentru gestiunea dispozitivelor periferice.
73. Închiderea versiunilor demonstrative ale unui program.
74. Deplasari succesive ale obiectelor grafice , de ex. Pentru animații.
75. Toate raspunsurile
76. Bifați opțiunile care reprezintă domenii de aplicare pentru PPE?
77. Programarea interfetelor grafice de utilizator
78. Programarea aplicatiilor simple pentru care limbajele PPE sunt foarte binevenite
79. Crearea aplicatiilor server
80. Programarea jocurilor
81. Bifați afirmația corectă
82. Cronometrul Windows este un dispozitiv de intrare,ce comunica periodic unei aplicatii trecerea unui anumit interval de timp.
83. Cronometrul Windows este un dispozitiv de evidenta,a timpului de sistem.
84. Cronometrul Windows este un dispozitiv de iesire,care comunica periodic cu aplicatia.
85. Cronometrul Windows este un dispozitiv de afisare a timpului real astronomic.
86. **Bifează pe valoarea de adevăr a afirmației:** Într-un program pot fi create mai multe meniuri principale.
87. True
88. False
89. **Bifați afirmația corectă**
90. În timpul dezvoltării programelor,resursele sunt definite într-un fișier txt de resurse, un fișier ASCII cu extensia .RC.
91. În timpul dezvoltării programelor,resursele sunt definite într-un fișier doc de resurse, un fișier ASCII cu extensia.RC.
92. În timpul dezvoltării programelor,resursele sunt definite într-un fișier script de resurse, un fișier ASCII cu extensia .RC.
93. În timpul dezvoltării programelor,resursele sunt definite într-un fișier obj de resurse ,un fișier ASCII cu extensia
94. **Bifați categoria prefixului CS pentru identificatorul cu majuscule respectiv**
95. Opțiune pentru stilul clasei
96. Stil de fereastră.
97. Număr de identificare pentru o pictogramă.
98. Opțiuni de creare a unei ferestre.
99. **Bifați afirmațiile adevărate**
100. Resursele sunt păstrare în zona de date a programelor executabile
101. Resursele sunt de fapt niște date și sunt stocate în fișierele .EXE ale programelor.
102. Resursele trebuie să fie explicit încărcate în memorie din fișierul executabil.
103. Resursele sunt accesibile imediat prin variabile definite în codul sursă a programului
104. Pictogramele ,cursoarele, meniurile și casetele de dialog fac partee din aceeași familie a “ resurselor Windows”.
105. Bifați afirmațiile corecte
106. Fereastra descendent prelucrează mesajele primite de la mouse și de la tastatură și înștiințează fereastra părinte atunci când starea proprie se modifică.
107. Fereastra descendent este un dispozitiv de introducere a datelor pentru fereastra părinte.
108. Fereastra descendent nu răspunde de datele introduse de utilizator și metodele de înștiințare a unei alte ferestre în momentul producerii unui eveniment important.
109. Fereastra descendent incapsulează funcționalități specifice legate de modul de afișare pe ecran.
110. **Care este destinația parametrilor initial x position și initial y position în apelul funcției CreateWindow.** Bifați toate răspunsurile corecte
111. Toate opțiunile sunt greșite.
112. Setează poziția ferestrei pe ecran
113. Reprezintă coordonatele colțului stânga-sus al ferestrei în pixeli pentru sistemul de coordonate al zonei client.
114. Reprezintă coordonatele colțului stânga-sus al ferestrei în pixeli pentru sistemul de coordonate al display-ului.
115. Bifați valoarea de adevăr a afirmației: **Pentru a corecta meniul, de obicei trebuie să corectați doar fișierul cu descrierea resurselor,nu și codul programului.**
116. True
117. False
118. Bifați opțiunile care corespund răspunsurile la întrebare: **Din care motiv un programator nu trebuie să transforme mesajele generate de acționarea tastelor în mesaje caracter?**
119. Trebuie să țineți cont de starea de modificare
120. Trebuie să țineți seama de configurația diferită a tastaturii de la o țară la alta
121. Nu este simplu
122. Nu este posibil
123. Bifați opțiune care reprezintă valoarea de adevăr a afirmației: **Nu exista vre-o diferenta între ,,actionarile de taste”(,,keystrokes”) și ,,caractere”.**
124. True
125. False
126. Bifați opțiunile care puncteaza pe domeniile de utilizare a regiunilor
127. Regiunile sunt folosite pentru contururi
128. Regiunile sunt folosite pentru umplere
129. Regiunile sunt folosite pentru decupare
130. Regiunile sunt folosite pentru desenare
131. Care probleme sunt rezolvate atunci cand utilizăm PPE pentru dezvoltarea aplicațiilor server?
132. Problema scalabilitații
133. Problema integrității
134. Problema confidențialității
135. Problema disponibilității
136. Bifați opțiunile care reprezintă afiramții corecte referitor la particularitațile stilului PS\_INSIDEFRAME
137. În cazul folosirii stilului PS\_INSIDEFRAME ,penița este centrată pe dreptunghiul de încadrare,daca grosimea liniei este mai mare de un pixel
138. PS\_INSIDEFRAME este singurul stil care poate folosi culori amesticate,pentru orice grosime ale liniei
139. ÎN cazul folosirii stilului PS\_INSIDEFRAME,linia este desenată in intregime în interiorul dreptunghiului de încadrare.
140. PS\_INSIDEFRAME este singurul stil care poate folosi culori amesticate,dar numai pentru grosimi mai mari de un pixel.
141. Bifați funcțiile din categoria : **Funcții care stabilesc sau obțin atribute ale contextului de dispozitiv**
142. SetTextOut
143. TextOut
144. SetTextAlign
145. DrawText
146. Care mesaj este considerat cel mai important mesaj pentru meniu?
147. WM\_INITMENU
148. WM\_INITMENUPOPUP
149. WM\_MENUSELECT
150. WM\_COMMAND
151. Bifați afirmațiile corecte
152. Toate afirmațiile sunt greșite
153. Variabila handle a contextului de dispozitiv este salvată în variabila hdc
154. Valoare returnată de funcția BeginPaint este variabila handle a contextului de dispozitiv
155. În timpul prelucrării mesajului WM\_PAINT, procedura de fereastră apelează mai întâi funcția BeginPaint ca să completeze câmpurile structurii ps.
156. Bifați opțiunile care reprezintă funcții din categoria *Funcții care desenează ceva*
157. PaintBitmap
158. TextOut
159. DrawText
160. LineTo
161. Bifați mesajele corecte
162. WM\_KEYUP
163. WM\_SYSTEMKEYUP
164. WM\_KEYDOWN
165. WM\_SYSTEMKEYDOWN
166. Bifați toate afirmațiile corecte
167. GDI permite lucrul doar la nivel de vectori
168. GDI este un sistem de afișare static ce permite numai animații limitate
169. GDI nu asigură un suport direct pentru afișarea tridimensională
170. GDI nu asigură un suport direct pentru rotirea obiectelor
171. Bifați afirmațiile corecte referitor la Funcția TextOut (hdc, x, y, psString, iLength) ...
172. Parametrul hdc este unul istoric rămas de la versiunile anterioare Windows
173. Funcția TextOut afișează pe ecran un șir de caractere
174. Parametrii x și y definesc poziția de început a șirului de caractere
175. Parametrul psString este un pointer la șirul de caractere iar iLength este lungimea acestui șir în caractere.
176. Bifați afirmațiile corecte
177. În majoritatea cazurilor, mesajele de la tastatură conțin mai multe informații despre tastatură decât aveți nevoie în program
178. Windows trimite programelor patru tipuri de mesaje prin care indică diferite evenimente de tastatură
179. O parte a sarcinii de manipulare a tastaturii constă în a ști ce mesaje sunt importante
180. Programul este obligat să trateze orice mesaj de la tastatură
181. Bifați opțiunea care reprezintă valoare de adevăr a afirmației: **Atunci când utilizatorul apasă și eliberează tastele, driverul de tastatură transmite aplicației informațiile legate de acțiunile asupra tastelor**
182. True
183. False
184. Bifați opțiunile care vom completa începutul de propoziție ”**Mesajele WM\_KEYDOWN sunt utile pentru ...**”, formând afirmații adevărate
185. Tastele de deplasare
186. Tastele funcționale
187. Tastele cu caractere
188. Tastele special
189. Câte câmpuri are structura MSG?
190. 6
191. 5
192. 8
193. 4
194. Care erau bibliotecile cu legături dinamice principale ale primelor versiuni WINDOWS?
195. KERNEL
196. GUI
197. USER
198. GDI
199. Care este punctul de intrare în programul HelloWin?
200. CreateWindow
201. WinMain
202. HWND hwnd
203. WndProc
204. Bifați afirmațiile corecte
205. Caracteristic pentru programarea imperativă este faptul că progamatorul controlează fluxul de calcul
206. Caracteristic pentru programarea imperativă este faptul că programatorul stabilește ordinea acțiunilor
207. Caracteristic pentru programarea imperativă este faptul că programul este executat conform intenției programatorului
208. Caracteristic pentru programarea imperativă este faptul că programul este centrat pe calculator
209. Toate celelalte afirmații sunt false
210. deCare din afirmațiile referitoare la mesajul WM\_PAINT din programul HELLOWIN sunt adevărate?
211. Este generat de execuția funcției UpdateWindow
212. Este un mesaj care are o prioritate foarte mare
213. În programul HELLOWIN este al doilea mesaj recepționat de procedura de fereastră, după mesajul WM\_CREATE
214. Mesajul PAINT este expediat aplicației de către sistemul de operare de fiecare dată când aplicația trebuie să-și actualizeze conținutul zonei client.
215. Bifați situațiile în care procedura ferestrei trebuie să fie pregătită să prelucreze orice mesaj WM\_PAINT.
216. Programul folosește funcțiile ScrollWindow sau ScrollDC ca să deruleze o parte din zona client a ferestrei
217. Utilizatorul redimensionează fereastra (chiar dacă stitlul clasei ferestrei nu include seturile de biți CS\_HREDRAW și CS\_VREDRAW)
218. Când o zonă anterior acoperită a ferestrei este adusă la suprafață atunci când utilizatorul mută o fereastră
219. Programul folosește funcțiile InvalidateRect sau InvalidateRgn pentru a genera în mod explicit un mesaj WM\_PAINT