

Перелік завдань для формування білетів та тестів у ВНС
з навчальної дисципліни
«Алгоритми та моделі обчислень»

1. Історія поняття та варіанти визначення алгоритму.
2. Основні властивості та параметри алгоритму.
3. Базові структури алгоритмів (*алгоритмічні конструкції*). Теорема Бьома-Якопіні.
4. Рекурсивні алгоритми.
5. Паралельні алгоритми.
6. Недетерміновані алгоритми.
7. Імовірнісні алгоритми (*Probabilistic algorithms*).
8. Абстрактні моделі алгоритму.
9. Формальні алгоритмічні системи (*ФАС*).
10. Детермінований скінченний автомат (*DFA*).
11. Недетермінований скінченний автомат (*NFA*) та імовірнісний автомат (*PA*).
12. ω -автомат.
13. Автомат з однобуквеними переходами. Формування автомату з однобуквеними переходами за заданим недетермінованим автоматом.
14. Видалення непродуктивних та недосяжних станів скінченного автомату.
15. Видалення λ -переходів та ε -переходів у недетермінованому скінченному автоматі.
16. Детермінізація квазидетермінованого скінченного автомату.
17. Детермінізація недетермінованого скінченного автомату.
18. Мінімізація скінченного детермінованого автомату.
19. Перетворювачі (*трансдуктори*) на основі детермінованого скінченного автомату. Автомат Мура (*Moore machine*) та автомат Мілі (*Mealy machine*).
20. Автомат з магазинною пам'яттю (*МП-автомат, PDA*).
21. Машина Тюрінга та її варіанти.
22. Числення Поста.
23. Нормальні алгоритми Маркова.
24. Регістрова машина.
25. РАМ-машина. ПРАМ-машина та варіанти пам'яті із впорядкованим доступом.
26. Властивості граматик регулярних мов. Автоматні граматики. Доповнення автоматної мови.
27. Лема про накачку для регулярних мов.
28. Знаходження мови для заданої регулярної граматики.
29. Регулярні вирази.
30. Формування регулярного виразу для заданого недетермінованого скінченного автомату.

31. Формування недетермінованого скінченного автомату для заданої регулярної граматики.
32. Формування недетермінованого скінченного автомату для заданого регулярного виразу.
33. Властивості граматик КВ-мов (*контекстно-вільні мови*).
34. Лема про накачку для КВ-мов.
35. Деревя розбору КВ-грамматик.
36. Створення МП-автомату для заданої КВ-грамматики.
37. Нотації БНФ та РБНФ.
38. Нормальна форма Хомського для КВ-грамматик.
39. Алгоритм Кока-Касамі-Янгера для КВ-грамматик у нормальній формі Хомського.
40. Складність по часу виконання алгоритму. Оцінка розміру вхідних даних.
41. Складність по часу виконання алгоритму. Залежність часу виконання від розміру вхідних даних.
42. Складність по часу виконання алгоритму. Аналіз нерекурсивних алгоритмів.
43. Складність по часу виконання алгоритму. Аналіз рекурсивних алгоритмів.
44. Складність по часу виконання алгоритму. Аналіз алгоритму для найкращого, середнього та найгіршого випадку.
45. Асимптотичний аналіз. O - нотація.
46. Асимптотичний аналіз. o - нотація.
47. Асимптотичний аналіз. Ω - нотація.
48. Асимптотичний аналіз. ω - нотація.
49. Асимптотичний аналіз. Θ - нотація.
50. Детермінований (DTIME) та недетермінований (NTIME) ресурси часу виконання.
51. Здійсненні по часу виконання класи складності алгоритмів. DLOGTIME-клас складності (*логарифмічний*).
52. Здійсненні по часу виконання класи складності алгоритмів. PolylogTIME-клас складності (*полілогарифмічний*).
53. Здійсненні по часу виконання класи складності алгоритмів. P-клас складності (*поліноміальний*).
54. Здійсненні по часу виконання класи складності алгоритмів. P-повні задачі.
55. Здійсненні по часу виконання класи складності алгоритмів. RP-клас та coRP-клас складності.
56. Здійсненні по часу виконання класи складності алгоритмів. ZPP-клас складності.
57. Здійсненні по часу виконання класи складності алгоритмів. BPP-клас складності.
58. Здійсненні по часу виконання класи складності алгоритмів. BQP-клас складності.
59. Проблематичні по часу виконання класи складності алгоритмів. NP- та coNP- класи складності.
60. Проблематичні по часу виконання класи складності алгоритмів. Співвідношення між P- та NP- класами складності.

61. Проблематичні по часу виконання класи складності алгоритмів. NP- та coNP- повні задачі. Теорема Кука-Левіна. Стандартні NP-повні задачі.
62. Проблематичні по часу виконання класи складності алгоритмів. NP-складні, еквівалентні, проміжні та прості задачі
63. Проблематичні по часу виконання класи складності алгоритмів. RP-клас складності.
64. Проблематичні по часу виконання класи складності алгоритмів. UP-клас складності.
65. Проблематичні по часу виконання класи складності алгоритмів. #P-клас (вимовляється як «шарп P») складності та #P-повні задачі.
66. Проблематичні по часу виконання класи складності алгоритмів. \oplus P-клас (вимовляється як «паритет P») складності.
67. Нездійсненні по часу виконання класи складності алгоритмів. EXPTIME-клас складності (експоненціальний клас складності).
68. Нездійсненні по часу виконання класи складності алгоритмів. NEXPTIME-клас складності.
69. Нездійсненні по часу виконання класи складності алгоритмів. 2-EXPTIME-клас складності.
70. Нездійсненні по часу виконання класи складності алгоритмів. ELEMENTARY-клас складності.
71. Нездійсненні по часу виконання класи складності алгоритмів. R-клас складності.
72. Нездійсненні по часу виконання класи складності алгоритмів. PR-клас складності.
73. Нездійсненні по часу виконання класи складності алгоритмів. RE-клас складності та coRE-клас складності.
74. Нездійсненні по часу виконання класи складності алгоритмів. ALL-клас складності.
75. Аналіз паралельних алгоритмів. Особливості аналізу паралельних алгоритмів.
76. Аналіз паралельних алгоритмів. Теорема Брента. Закон Густавсона–Барсіса. Закон Амдала.
77. Аналіз паралельних алгоритмів. NC-клас складності.
78. Ємнісна(просторова) складність алгоритму (складність по об'єму пам'яті).
79. Детермінований (DSPACE) та недетермінований (NSPACE) просторові (ємнісні) ресурси.
80. Теорема Севіча.
81. Здійсненні по об'єму пам'яті класи складності алгоритмів. L-клас складності.
82. Здійсненні по об'єму пам'яті класи складності алгоритмів. PolyL-клас складності (полілогарифмічний).
83. Здійсненні по об'єму пам'яті класи складності алгоритмів. SL-клас складності.
84. Здійсненні по об'єму пам'яті класи складності алгоритмів. NL-клас складності.
85. Здійсненні по об'єму пам'яті класи складності алгоритмів. NL-повні задачі.

86. Здійсненні по об'єму пам'яті класи складності алгоритмів. Еквівалентність класів NL та coNL.
87. Здійсненні по об'єму пам'яті класи складності алгоритмів. RL-клас складності.
88. Проблематичні по об'єму пам'яті класи складності алгоритмів. PSPACE-клас складності.
89. Проблематичні по об'єму пам'яті класи складності алгоритмів. PSPACE-повні задачі.
90. Нездійсненні по об'єму пам'яті класи складності алгоритмів. EXPSPACE-клас складності(*експоненціальний клас складності по пам'яті*).
91. Структурна складність. Цикломатична складність графу потоку керування та графу алгоритму.
92. Структурна складність. Структурна складність обчислень поданих структурною матрицею потокового графу алгоритму.
93. Структурна складність. AC^i -класи складності.
94. Структурна складність. ACC^0 -клас складності.
95. Структурна складність. TC^i -класи складності.
96. Структурна складність. CC-клас складності.
97. Ієрархії класів складності. Теорема про ієрархії класів часової складності.
98. Ієрархії класів складності. Теорема про ієрархії класів просторової складності.
99. Ієрархії класів складності. Поліноміальна ієрархія та PH-клас складності.
100. Ієрархії класів складності. Експоненціальна ієрархія.
101. Ієрархії класів складності. Ієрархія Гжегорчика.
102. Ієрархії класів складності. Арифметична ієрархія.
103. Ієрархії класів складності. Булева ієрархія.
104. Вербальне та аналітичне подання алгоритму.
105. Подання алгоритму псевдокодом або з використанням формальних мов.
106. Блок-схема алгоритму.
107. Граф потоку керування.
108. Граф алгоритму.
109. Поточковий граф алгоритму.
110. Діаграма Нассі-Шнайдермана.
111. Діаграми UML. Діаграма прецедентів(*діаграма варіантів використання*).
112. Діаграми UML. Діаграма послідовності.
113. Діаграми UML. Діаграма комунікації.
114. Діаграми UML. Узагальнена діаграма взаємодій.
115. Діаграми UML. Діаграма стану.
116. Діаграми UML. Діаграма діяльності.
117. Діаграми UML. Діаграма класів.
118. Діаграми UML. Діаграма компонентів.
119. Діаграми UML. Діаграма розгортання.
120. Подання алгоритму структурною матрицею потокового графу алгоритму.
121. Класифікація типів даних. Базові та похідні типи даних.
122. Поняття Абстрактного типу даних(АТД). Контейнери та колекції.
123. АТД Стек.

124. АТД Черга. Черга з пріоритетом. Двобічна черга та двобічна черга з пріоритетом.
125. АТД Список.
126. Одно- та двобічнозв'язні списки. Список з пропусками. Розгорнутий зв'язаний список.
127. АТД Граф.
128. АТД Дерево.
129. Бінарне дерево пошуку. Збалансоване дерево. АВЛ-дерево.
130. Бінарне дерево пошуку. Збалансоване дерево. Червоно-чорне дерево.
131. Б-дерево(*англ. B-tree*) та R-дерево.
132. Купа. Двійкова купа. Біноміальна купа. Фібоначчієва купа.
133. Множина. Мультимножина.
134. Асоціативний масив(Словник). Мультисловник.
135. Геш-таблиця.
136. Покрокове проектування алгоритмів.
137. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Повний перебір.
138. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Метод зменшення розміру задачі.
139. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Метод декомпозиції.
140. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Метод перетворень.
141. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Динамічне програмування.
142. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Дерево розв'язків(*дерево рішень*).
143. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Бектрекінг(*перебір з поверненням*).
144. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Метод гілок і границь.
145. Підходи при синтезі алгоритмів(*алгоритмічні стратегії*). Альфа-бета відсікання.
146. Евристичні алгоритми. Метод спроб і помилок(*Trial and error*).
147. Евристичні алгоритми. Скупі(*жадібні*) алгоритми та локальний пошук.
148. Ітераційне вдосконалення алгоритму.
149. Просторово-часовий компроміс(*просторово-часове балансування*) при проектуванні алгоритмів.
150. Прогнозування складності алгоритму під час застосування відповідних стратегій.
151. Послідовний пошук та послідовний пошук з бар'єром.
152. Бінарний пошук.
153. Порозрядний пошук.
154. Зовнішній пошук.
155. Застосування геш-таблиць для пошуку. Розв'язання колізій при гешуванні відкритою адресацією та методом ланцюжків.
156. Сорткування вибором.
157. Сорткування вставками.

158. Сортування обміном.
159. Сортування злиттям.
160. Сортування Шелла.
161. Швидке сортування.
162. Пірамідальне сортування.
163. Порозрядне сортування.
164. Мережі сортування.
165. Зовнішнє сортування.
166. Примітивний алгоритм пошуку підрядка.
167. Алгоритм Рабіна-Карпа.
168. Алгоритм Кнута-Морріса-Пратта.
169. Алгоритм Бойєра-Мура.
170. Пошук підрядків за допомогою скінчених автоматів.
171. Наближене порівняння рядків.
172. Матриці та дії з ними. Алгоритм Копперсміта-Вінограда та алгоритм Штрассена.
173. Робота з довгими числами.
174. Многочлени та швидке перетворення Фур'є.
175. Системи алгебраїчних рівнянь.
176. Алгоритми розв'язання систем лінійних та нелінійних рівнянь.
177. Алгоритми апроксимації і інтерполяція чисельних функцій.
178. Пошук у графі.
179. Породження всіх каркасів графа.
180. Каркас мінімальної ваги. Метод Дж. Крускала. Метод Р. Пріма.
181. Досяжність в графі. Визначення зв'язності. Двозв'язність.
182. Ейлерові цикли.
183. Гамільтонові цикли.
184. Алгоритм Дейкстри.
185. Алгоритм Флойда.
186. Метод генерації всіх максимальних незалежних множин графа. Задача про найменше покриття.
187. Задача про найменше розбиття.
188. Розфарбування графа.
189. Пошук мінімального розфарбування вершин графа.
190. Потоки в мережах.
191. Метод побудови максимального потоку в мережі.
192. Методи наближеного рішення задачі комівояжера(метод локальної оптимізації, алгоритм Ейлера, алгоритм Крістофідеса).
193. Аналіз алгоритмів на графах.
194. Методи паралельного виконання програми за допомогою спільної пам'яті або за допомогою передачі повідомлень.
195. Організація паралельних обчислень відповідно до принципу консенсусу і на основі вибору.
196. Методи визначення завершення паралельних обчислень.
197. Паралельний пошук, паралельне сортування, паралельні чисельні алгоритми, паралельні алгоритми на графах.
198. STL(Standard Template Library). Базові типи бібліотеки.

199. STL(Standard Template Library). Засоби бібліотеки для роботи з стрічками та вводом/виводом.
200. STL(Standard Template Library). Послідовні контейнери бібліотеки.
201. STL(Standard Template Library). Асоціативні контейнери бібліотеки.
202. STL(Standard Template Library). Ітератори.
203. STL(Standard Template Library). Арифметичні функціональні об'єкти.
204. STL(Standard Template Library). Предикати.
205. STL(Standard Template Library). Адаптери-заперечувачі.
206. STL(Standard Template Library). Адаптери-зв'язувачі.
207. STL(Standard Template Library). Адаптери вказівників на функції.
208. STL(Standard Template Library). Адаптери методів.
209. STL(Standard Template Library). Алгоритми сортування та пошуку бібліотеки.
210. STL(Standard Template Library). Чисельні алгоритми бібліотеки.
211. STL(Standard Template Library). Модифікуючі та немодифікуючі алгоритми бібліотеки.
212. STL(Standard Template Library). Розподільники пам'яті для контейнерних класів бібліотеки.
213. Набір бібліотек Boost. Застосування `boost::assign`.
214. Набір бібліотек Boost. Застосування `boost::function`.
215. Набір бібліотек Boost. Застосування `boost::bind`.
216. Набір бібліотек Boost. Застосування `boost::optional`.
217. Набір бібліотек Boost. Застосування `boost::variant`.
218. Набір бібліотек Boost. Застосування `boost::lexical_cast`.
219. Набір бібліотек Boost. Застосування `boost::spirit`.
220. Набір бібліотек Boost. Застосування `boost::filesystem`.
221. Набір бібліотек Boost. Застосування `boost::asio`.
222. Набір бібліотек Boost. Застосування `boost::static_assert`.
223. Набір бібліотек Boost. Метапрограмування за допомогою `boost::mpl`.
224. JCL (Java Class Library). Застарілий інтерфейс Enumeration.
225. JCL (Java Class Library). Інтерфейс Iterator.
226. JCL (Java Class Library). Інтерфейс Iterable.
227. JCL (Java Class Library). Інтерфейс Collection.
228. JCL (Java Class Library). Інтерфейси Set, SortedSet та NavigableSet.
229. JCL (Java Class Library). Класи HashSet, LinkedHashSet та TreeSet.
230. JCL (Java Class Library). Інтерфейси Queue та Deque.
231. JCL (Java Class Library). Класи LinkedList, ArrayDeque та PriorityQueue.
232. JCL (Java Class Library). Інтерфейс List.
233. JCL (Java Class Library). Класи Vector, Stack, ArrayList та LinkedList.
234. JCL (Java Class Library). Інтерфейс ListIterator.
235. JCL (Java Class Library). Інтерфейси Map, SortedMap та NavigableMap.
236. JCL (Java Class Library). Класи HashMap, TreeMap, LinkedHashMap, ArrayList та WeakHashMap.
237. JCL (Java Class Library). Алгоритми з допоміжного класу Collections.
238. JCL (Java Class Library). Використання бібліотеки для конкурентних обчислень. Застосування ключового слова `synchronized`.

239. JCL (Java Class Library). Використання бібліотеки для конкурентних обчислень. Огляд засобів пакету `java.util.concurrent`.
240. FCL (Framework Class Library). Узагальнений та неузагальнений інтерфейси `IEnumerator`.
241. FCL (Framework Class Library). Узагальнений та неузагальнений інтерфейси `IEnumerable`.
242. FCL (Framework Class Library). Узагальнений та неузагальнений інтерфейси `ICollection`.
243. FCL (Framework Class Library). Узагальнений та неузагальнений інтерфейси `IList`.
244. FCL (Framework Class Library). Узагальнені класи `HashSet`, `List` та `Collection`.
245. FCL (Framework Class Library). Неузагальнені класи `ArrayList` та `Array`.
246. FCL (Framework Class Library). Узагальнений та неузагальнений класи `Stack`.
247. FCL (Framework Class Library). Узагальнений та неузагальнений класи `Queue`.
248. FCL (Framework Class Library). Неузагальнений клас `BitArray`.
249. FCL (Framework Class Library). Узагальнений та неузагальнений інтерфейси `IDictionary`.
250. FCL (Framework Class Library). Узагальнені класи `Dictionary`, `SortedDictionary` та `SortedList`.
251. FCL (Framework Class Library). Неузагальнені класи `ListDictionary`, `HashTable` та `SortedList`.
252. FCL (Framework Class Library). Властивості `ICollection.IsSynchronized` та `ICollection.SyncRoot`.
253. FCL (Framework Class Library). Огляд засобів простору імен `System.Collections.Concurrent`.
254. Застосування алгоритмів лінійної алгебри при програмуванні на C++ за допомогою стандарту BLAS.
255. Застосування алгоритмів обробки сигналів при програмуванні на C++ та Python засобами OpenCV (*Open Source Computer Vision Library*).
256. Елементи теорії моделей. Моделі в теорії моделей.
257. Елементи теорії моделей. Інтерпретації формальних мов.
258. Елементи теорії моделей. Теорія повноти моделей (*для логіки першого порядку*).
259. Елементи теорії моделей. Принцип перенесення моделей (*для логіки першого порядку*).
260. Елементи теорії обчислюваності. Поняття обчислюваної функції.
261. Елементи теорії обчислюваності. Примітивно-рекурсивні функції.
262. Елементи теорії обчислюваності. Теза Черча-Тюрінга.
263. Елементи теорії обчислюваності. Задача про прийняття рішень.
264. Елементи теорії обчислюваності. Необчислюваність. Теореми Геделя про неповноту.
265. Елементи теорії обчислюваності. Необчислюваність. Поняття необчислюваної функції (*Uncomputable function*).

266. Елементи теорії обчислюваності. Необчислюваність. Задача про зупинку машини Тюрінга.
267. Елементи теорії обчислюваності. Необчислюваність. Нерозв'язувана задача про прийняття рішень та степінь Тюрінга.
268. Елементи теорії обчислюваності. Обчислення з оракулом.
269. Елементи теорії обчислюваності. Зведення однієї задачі до іншої за поліноміальний час. Зведення Карпа (*Karp reductions, Many-one reductions*).
270. Елементи теорії обчислюваності. Зведення однієї задачі до іншої за поліноміальний час. Зведення Кука (*Cook reduction*).
271. Елементи теорії обчислюваності. Зведення однієї задачі до іншої за поліноміальний час. Зведення Левіна (*Levin reduction*).
272. Елементи теорії обчислюваності. Зведення однієї задачі до іншої за поліноміальний час. Зведення через таблицю істинності(*Truth-table reduction*).
273. Елементи теорії обчислюваності. Зведення однієї задачі до іншої за поліноміальний час. Зведення Тюрінга (*Turing reduction*).
274. Елементи теорії категорій. Приклади категорій.
275. Елементи теорії категорій. Дуальна категорія.
276. Елементи теорії категорій. Морфізми в теорії категорій.
277. Елементи теорії категорій. Початковий та термінальний об'єкти в теорії категорій.
278. Елементи теорії категорій. Функтори в теорії категорій.
279. Елементи теорії категорій. Натуральне перетворення в теорії категорій.
280. Елементи теорії категорій. Моноїдальна категорія(*тензорна категорія*).
281. Імперативний та декларативний підходи до програмування.
282. Функційні моделі обчислень. Лямбда числення. Підстановки та перетворення при застосуванні лямбда числення.
283. Функційні моделі обчислень. Лямбда числення. Розширення чистого лямбда числення.
284. Функційні моделі обчислень. Лямбда числення. Теорема про нерухому точку.
285. Функційні моделі обчислень. Лямбда числення. Редекси і нормальна форма.
286. Функційні моделі обчислень. Лямбда числення. Теорема Черча-Россера.
287. Функційні моделі обчислень. Лямбда числення. Редукція термів в лямбда численні.
288. Функційні моделі обчислень. Типізоване лямбда числення. Поняття типу в типізованому лямбда численні.
289. Функційні моделі обчислень. Типізоване лямбда числення. Просте типізоване лямбда числення.
290. Функційні моделі обчислень. Типізоване лямбда числення. Підстановка типу та уніфікація.
291. Функційні моделі обчислень. Комбінаторна логіка (*як варіант лямбда числення*).
292. Функційні моделі обчислень. Комбінаційна логіка (*як функційна модель обчислень*).
293. Функційні моделі обчислень. Абстрактна система переписувань (*Abstract rewriting system*).

294. Парадигма функційного програмування.
295. Функційне програмування за допомогою Haskell. Базові типи в Haskell.
296. Функційне програмування за допомогою Haskell. Системи модулів в Haskell.
297. Функційне програмування за допомогою Haskell. Списки в Haskell.
298. Функційне програмування за допомогою Haskell. Види поліморфізму. Параметричний та спеціальний поліморфізм в Haskell.
299. Функційне програмування за допомогою Haskell. Класи типів в Haskell.
300. Функційне програмування за допомогою Haskell. Стандартні класи типів в Haskell.
301. Функційне програмування за допомогою Haskell. Реалізації класів типів в Haskell.
302. Функційне програмування за допомогою Haskell. Згортки в Haskell.
303. Функційне програмування за допомогою Haskell. Моноїди в Haskell.
304. Функційне програмування за допомогою Haskell. Клас типів Foldable.
305. Функційне програмування за допомогою Haskell. Функтори в Haskell.
306. Функційне програмування за допомогою Haskell. Клас типів Pointed.
307. Функційне програмування за допомогою Haskell. Аплікативні функтори.
308. Функційне програмування за допомогою Haskell. Клас типів Traversable.
309. Функційне програмування за допомогою Haskell. Клас типів монад.
310. Функційне програмування за допомогою Haskell. Монада Maybe.
311. Функційне програмування за допомогою Haskell. Монада IO.
312. Функційне програмування за допомогою Haskell. Монада Reader та монада Writer.
313. Функційне програмування за допомогою Haskell. Монада State.
314. Функційне програмування на C++.
315. Функційне програмування мовами Erlang та Elixir.
316. Мережа процесів Кана.
317. Мережа Петрі.
318. Мережа взаємодій.
319. Синхронний потік даних.
320. Парадигма реактивного програмування.
321. Функційне реактивне програмування.
322. Реактивне програмування за допомогою бібліотеки ReactiveX. Шаблон спостерігача (*Observer pattern*) та його розширення у ReactiveX.
323. Реактивне програмування за допомогою бібліотеки ReactiveX. Сутності для спостереження (*Observable*).
324. Реактивне програмування за допомогою бібліотеки ReactiveX. Оператори бібліотеки ReactiveX.
325. Реактивне програмування за допомогою бібліотеки ReactiveX. Сутність Single.
326. Реактивне програмування за допомогою бібліотеки ReactiveX. Сутність суб'єкту (*Subject*).
327. Реактивне програмування за допомогою бібліотеки ReactiveX. Планувальник (*Scheduler*) рушія бібліотеки ReactiveX.
328. Реактивне програмування за допомогою бібліотеки ReactiveX. Реалізації ReactiveX для популярних мов програмування.

- 329. Реактивне програмування за допомогою фреймворку Spring WebFlux з набору фреймворків Spring.
- 330. Використання шаблонів при проектуванні програмного забезпечення.
- 331. GoF-шаблони. Твірні шаблони (*Creational pattern*).
- 332. GoF-шаблони. Структурні шаблони (*Structural pattern*).
- 333. GoF-шаблони. Поведінкові шаблони (*Behavioral pattern*).
- 334. GRASP-шаблони.
- 335. Шаблони рівночасних обчислень (*Concurrency pattern*).
- 336. Шаблони архітектури програмного забезпечення (*Architectural pattern*).