

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ЗАТВЕРДЖУЮ  
Голова Вченої ради  
факультету/інституту

«26» березня 2018 р.

**ПРОГРАМА ДОДАТКОВОГО ВСТУПНОГО  
ВИПРОБУВАННЯ**

**третього (освітньо-наукового) рівня вищої освіти  
для здобуття наукового ступеня доктор філософії**

**121 «Інженерія програмного  
забезпечення»**

Ухвалено Вченою радою факультету/інституту  
(протокол від «26» березня 2018 р. № 8)

Київ  
КПІ ім. Ігоря Сікорського  
2018

## **РОЗРОБНИКИ:**

Дорогий Ярослав Юрійович, к.т.н., доц., доцент кафедри  
автоматики і управління в технічних системах

Катін Павло Юрійович, к.т.н., доц., доцент кафедри автоматки і  
управління в технічних системах

Теленик Сергій Федорович, д.т.н., проф., завідувач кафедри  
автоматики і управління в технічних системах

---

---

---

## ЗМІСТ ПРОГРАМИ

I ПОЯСНЮВАЛЬНА ЗАПИСКА

II СТРУКТУРА ПРОГРАМИ

III КРИТЕРІЇ ОЦІНЮВАННЯ

IV СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

## I ПОЯСНЮВАЛЬНА ЗАПИСКА

Додаткове фахове випробування для вступу за спеціальністю 121 – «Інженерія програмного забезпечення», третього рівня вищої освіти – доктор філософії, проводиться у формі усного екзамену, фаховою атестаційною комісією, що створюється за наказом ректору Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», відповідно з «Правилами прийому до НТУУ «КПІ ім. Ігоря Сікорського» в 2018 році» та «Положення про приймальну комісію НТУУ «КПІ ім. Ігоря Сікорського» для проведення фахових випробувань для вступу на навчання на базі раніше здобутого другого рівня вищої освіти.

Програма проведення додаткового фахового випробування для спеціальності 121 – «Інженерія програмного забезпечення», третього рівня вищої освіти – доктор філософії, охоплює предметне коло питань із основ програмної інженерії, технологій програмування та засобів створення інформаційних систем.

За своєю структурою зміст програми поділяється на такі розділи:

- 1) математична підготовка;
- 2) основи програмної інженерії;
- 3) мови та технології програмування;
- 4) засоби створення інформаційних систем.

Питання, що входять до тем розділів, розташовані у логічній послідовності та відповідають змісту навчальних дисциплін, що викладаються для студентів кваліфікаційного рівня «магістр» напряму підготовки 6.050103 – «Програмна інженерія».

Відповідь вступники фіксують письмово під час вступного випробування на аркуші усної відповіді зі штампом Приймальної комісії. Фахове випробування проводиться за екзаменаційним білетом, що містить чотири питання з цієї програми. Тривалість фахового випробування 2 астрономічних години.

II СТРУКТУРА ПРОГРАМИ  
РОЗДІЛ 1  
МАТЕМАТИЧНА ПІДГОТОВКА

1. Інтуїтивне визначення алгоритму, вирішуваність, обчислюваність, перераховність. Способи подання алгоритмів.
2. Сутність і особливості оцінювання складності алгоритмів. Теоретичні основи. Види оцінок складності.
3. Методи розроблення алгоритмів.
4. Динамічне програмування. Загальна схема. Умови застосування. Реалізація схеми динамічного програмування для задач множення матриць
5. Жадібні алгоритми. Загальна схема. Умови застосування. Теоретичні основи жадібних алгоритмів. Жадібні алгоритми для задач про вибір заявок і розклад, побудови кодів Хафмена,
6. Алгоритми паралельних обчислень. Моделі паралельних машин. Ефективна паралельна обробка префіксів.
7. Мережі і проблема визначення кістякового дерева мінімальної ваги. Алгоритм PRIM. Реалізація і аналіз складності алгоритму.
8. Алгоритми сортування і пошуку даних. Класифікація. Переваги і недоліки. Класичні ( «бульбашка», вставками, Шела та ін.) і сучасні алгоритми сортування.
9. Алгоритми пошуку підрядків. Алгоритми Рабіна-Карпа, Кнута-Моріса-Прата, Боєра-Мура. Алгоритми пошуку підрядків на основі скінчених автоматів.
10. Базові структури даних. Масиви. Зв'язані списки. Списки суміжності. Стеки, черги. Двійкові дерева пошуку. Випадкові двійкові дерева пошуку.
11. Розвиток уявлень про структури даних. Загальна характеристика хешування. Хеш-функції. Відкрита адресація. Організація переповнення ланцюжками. Червоно-чорні дерева: визначення; операції; оцінки; застосування. Б-дерева: визначення; операції; оцінки; застосування.

12. Уточнення інтуїтивного визначення алгоритму за Т'юрингом. Розв'язувані і не розв'язувані предикати, обчислювальні і не обчислювальні функції. Теорема Черча.

13. Рекурсивні функції. Визначення і властивості примітивно-рекурсивних, частково-рекурсивних і загально-рекурсивних функцій.

14. Рекурсивна обчислюваність. Рекурсивна вирішуваність, примітивно-рекурсивні множини, рекурсивні множини, частково-рекурсивні множини. Властивості рекурсивних і примітивно-рекурсивних множин. Рекурсивно-перераховні множини.

15. Теорія складності. Проблеми і кодування, обчислювальна модель і класи P, NP. Базові поняття теорії звідності, співвідношення класів P, NP і NP-повних проблем, звідність, теорема Кука.

16. Елементи теорії NP-повноти. Проблеми 3-виконуваність, «тривимірне сполучення», «вершинне покриття», «Кліка», розбиття, «Гамільтонів цикл», ієрархія NP-повних проблем.

17. Методи доведення NP-повноти. Загальна характеристика прийомів доведення NP-повноти, звуження проблеми, локальна заміна, побудова компонентів, Аналіз підпроблем. Проблеми з числовими параметрами і сильна NP-повнота.

18. Граматики породження: визначення і класифікація.

19. Властивості контекстно-вільних мов як базис теорії трансляції.

20. Алгебраїчні моделі формальних мов.

21. Регулярні множини і право лінійні граматики.

22. Автомати і граматики.

23. Синтаксичний аналіз.

24. Логіка і числення висловлювань.

25. Логіка і числення предикатів.

## РОЗДІЛ 2

### ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

1. Основні поняття програмної інженерії. Основні етапи життєвого циклу програмних систем. Варіанти (моделі) життєвого циклу. Загальна характеристика етапів життєвого циклу програмних систем.

2. Характеристика моделей життєвого циклу програмного продукту: каскадна, спіральна, ітеративна, інкрементна (пошагова), формальна, швидкого прототипування. Їх переваги, недоліки та використання.

3. Визначення та задачі аналізу вимог до програмного забезпечення. Вимоги до продукту та процесу. Функціональні та нефункціональні вимоги. Вимоги з кількісною оцінкою. Системні та програмні вимоги.

4. Моделі процесу визначення вимог до програмного забезпечення. Типи учасників процесу та їх роль. Джерела вимог та техніки отримання вимог.

5. Структура та архітектура програмного забезпечення. Архітектурні точки зору та представлення. Характеристика архітектур програмних продуктів. Поняття шаблонів проектування, фреймворки.

6. Загальні концепції проектування програмного забезпечення. Концепції та процес проектування програмного забезпечення. Загальні принципи (техніки) проектування.

7. Фундаментальні питання та проблеми проектування програмного забезпечення. Паралелізм та ефективна обробка програмних процесів. Контроль та обробка подій. Розподіл компонентів. Обробка помилок та виключних ситуацій. Взаємодія системи з користувачами та збереження даних.

8. Моделювання у програмній інженерії. Основні принципи моделювання програмних систем. Візуальне моделювання та CASE-технології. Уніфікована мова моделювання *UML*.

9. Аналіз якості та оцінювання програмного проектування. Показники якості розробки програмного забезпечення. Аналіз якості та техніки оцінювання. Виміри в проектуванні програмного забезпечення.

10. Призначення UML. Види діаграм UML. Діаграми структури та діаграми поведінки. Спрощена стратегія використання UML-діаграм при моделюванні програмного забезпечення.

11. Діаграми прецедентів та їх використання. Моделювання контексту та вимог до програмної системи. Прецеденти та потоки подій. Описи потоків подій. Потоки подій та сценарії як типи й екземпляри.

12. Специфікація прецедентів. Використання сценаріїв при плануванні версій. Зв'язки між акторами та прецедентами. Організація прецедентів (відношення залежності, включення та розширення). Відношення узагальнення для прецедентів та акторів.

13. Реалізація прецедентів та використання діаграм послідовностей. Побудова діаграм послідовності. Двохетапне розроблення діаграм послідовностей. Діаграми співробітництва.

14. Узгодженість (цілісність) моделей. Виявлення класів. Класи етапу аналізу: прикордонні (boundary) або інтерфейсні класи; класи-сутності (entity); управляючі (control) класи (класи-менеджери). Класи етапу проектування.

15. Відношення між класами (узагальнення, залежність, асоціація, агрегація, композиція) та їх виявлення. Проектування класів, відношень між класами. Пакетування класів.

16. Класифікація патернів: породжуючі патерни (патерни процесу створення об'єктів); структурні патерни (що ґрунтуються на композиціях – структурних об'єднаннях об'єктів чи класів); патерни поведінки (що характеризуються взаємодією об'єктів між собою).

17. Успадкування та композиції. Простір патернів проектування. Приклади патернів. Кодогенерація (інженірінг) та реінженірінг.

18. Стратегії та методи проектування програмного забезпечення. Загальні стратегії. Функціонально-орієнтоване та структурне проектування.

Об'єктно-орієнтоване проектування. Проектування на основі структур даних.  
Компонентне проектування.

19. Будова програми при використанні процедурного (імперативного) програмування. Проблеми, що виникають при використанні процедурного програмування та їх вирішення використанням об'єктно-орієнтованого програмування.

20. Характеристика структурного програмування як методології розробки програмного забезпечення. Базові конструкції програми. Поняття підпрограми (процедури або функції). Метод розробки програми. Мета та принципи модульного програмування.

21. Поняття якості програмного продукту. Характеристика основних критеріїв якості програмного продукту.

22. Методології розробки програмного забезпечення Microsoft Solution Framework, Rational Unified Process, Extreme Programming. Засоби розробки, що базуються на них.

23. Мета та задачі супроводу програмного забезпечення. Еволюція програмного забезпечення та потреба у його супроводі. Категорії та ключові питання супроводу програмного забезпечення.

24. Мета та задачі управління програмними проектами. Аналіз здійсненності програмного проекту: технічні, операційні, фінансові та соціальні аспекти. Процес оцінки та перегляду вимог до програмного забезпечення. Планування програмного проекту. Управління виконанням програмного проекту.

25. Поняття якості програмного забезпечення. Основні принципи та задачі керування якістю програмного забезпечення. Характеристика дефектів та методи управління якістю програмного забезпечення.

26. Основні принципи конструювання програмного забезпечення: мінімізація складності, очікування змін, конструювання з можливістю перевірки, стандарти конструювання.

27. Загальні принципи керування конструюванням програмного забезпечення. Планування та вимірювання в конструюванні програмного

забезпечення. Мови конструювання та програмне кодування. Тестування та якість конструювання. Загальні аспекти повторного використання.

28. Термінологія та ключові питання тестування програмного забезпечення. Зв'язок тестування з іншими видами діяльності процесу розробки програмного забезпечення. Мета та рівні тестування. Техніки тестування та їх комбінація. Вимір результатів тестування.

## РОЗДІЛ 3

### МОВИ ТА ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ

1. Структура програм на мовах програмування Python, C/C++ або Java. Прості та складені оператори, приклади використання операторів. Поняття константи та змінної в мовах програмування, принципи виділення пам'яті для розміщення констант та змінних. Оголошення та ініціалізація констант і змінних у програмах на мовах програмування Python, C/C++ або Java.

2. Прості типи даних. Розміщення даних різних типів в оперативній пам'яті. Приклади оголошення різних типів даних у програмах на мовах програмування Python, C/C++ або Java.

3. Сумісність типів та приведення типів даних. Типи даних користувача. Оголошення та використання таких типів в мовах програмування Python, C/C++ або Java.

4. Операції, вирази та оператори в мовах програмування Python, C/C++ або Java. Операції привласнення та арифметичні операції. Пріоритети операцій. Зміна пріоритетів.

5. Логічні операції та умовні оператори. Алгоритмічна структура розгалуження. Вибір із двох альтернатив. Блок схеми структур розгалуження. Вкладеність конструкцій вибору.

6. Алгоритмічні конструкції повторення. Типи циклів. Блок схеми циклів. Організація циклів.

7. Поняття функції (підпрограми) в мовах програмування. Оголошення та виклик функцій. Рекурсія функцій. Стандартні процедури та функції.

8. Поняття глобальних та локальних змінних та приклади їх використання в програмах на мовах Python, C/C++ або Java. Статичні та динамічні змінні.

9. Поняття масиву та його властивості, їх оголошення і використання. Одновимірні та багатовимірні масиви. Доступ до елементів

масиву.

10. Рядки. Операції над символами та рядками в мовах програмування Python, C/C++ або Java. Стандартні процедури та функції обробки рядків.

11. Записи та структури, їх оголошення і використання. Доступ до компонентів та операції над записами та структурами. Похідні типи даних.

12. Основні концепції об'єктно-орієнтованої методології програмування. Програмна модель об'єкту. Наведіть приклад програмної моделі довільного об'єкту на мові програмування Python, C++ або Java.

13. Оголошення класів на мові програмування Python, C++ або Java. Конструктори і деструктори класу. Призначення і основні правила використання. Наведіть приклад оголошення довільного класу та його використання в тексті програм на мові програмування.

14. Структура програми, що використовує класи, на мові програмування Python, C++ або Java. Наведіть приклади.

15. Поняття інкапсуляції та її застосування в мовах програмування. Методи які організують інтерфейси. Наведіть приклади визначення відкритих і закритих полів і методів класу.

16. Поняття спадкування та поліморфізму в об'єктно-орієнтованих мовах програмування. Поняття та приклади перевантажених функцій. Ієрархія класів. Одиночне і множинне спадкоємство.

17. Абстрактні класи. Призначення і використання. Наведіть приклад абстрактного класу.

18. Спеціальні методи класів. Перевантаження операторів. Поняття дружніх функцій і класів. Особливість доступу до закритих полів класу. Наведіть приклади організації доступу до закритих полів.

19. Організація обмінних операцій з використанням бібліотеки класів для введення - виведення. Поняття файлу і потоку, приклади роботи з ними. Маніпулятори потоків.

20. Обробка програмних помилок засобами мови програмування. Алгоритмічна конструкція типу `try catch`. Наведіть приклади використання

обробки помилок роботи програмного блоку.

21. Поняття шаблону функції та класу. Абстрактні алгоритми. Бібліотека стандартних шаблонів STL (Standard Template Library).

22. Компілятори і інтерпретатори. Опис їх роботи та порівняння переваг і недоліків. Основні задачі, що вирішуються компілятором. Вхідні і вихідні файли, їх логічний зміст.

23. Поняття про ресурси ЕОМ. Операційна система як програма управління ресурсами. Основні завдання управління програмними процесами, оперативною пам'яттю і пристроями введення - висновку в операційній системі.

24. Поняття багатозадачної операційної системи. Витисняюча багатозадачність. Схема витисняючої багатозадачності квантуванням за часом.

25. Управління процесами в багатозадачних ОС. Основні алгоритми управління.

26. Управління оперативної пам'яттю в багатозадачній ОС. Основні алгоритми управління. Віртуальна пам'ять.

27. Управління ресурсами. Монопольне та спільне використання ресурсів. Поняття моделювання ресурсу. «Спулінг».

28. Критичні секції в асинхронних процесах багатозадачних ОС. Основні алгоритми синхронізації. Завдання синхронізації процесів. Поняття критичного шляху. «Змагання» і «клінчі».

29. Основні методи організації зберігання файлів на зовнішніх пристроях прямого методу доступу. Поняття бібліотечної, індексної послідовної і кластерної структури.

30. Типи діалогів, порівняльні характеристики діалогів. Групи користувачів ОС. Основні характеристики діалогів типу меню та команда. Групи користувачів ОС.

31. Способи досягнення надійності програмного забезпечення. Привести практичні приклади з використанням базової мови програмування. Поняття про тестування. Навести простий приклад тестування.

32. Мова запитань до баз даних SQL. Особливості застосування.
33. Альтернативні мови запитання до баз даних. Особливості застосування.
34. Методологія розробки програмного забезпечення. Технологічний процес створення програмного забезпечення
35. Моделі життєвого циклу розробки програмного забезпечення. Ступенчаста і водоспадна модель. Пояснити на прикладі. Переваги і недоліки моделі.
36. Моделі життєвого циклу розробки програмного забезпечення. Спіральна модель. Пояснити на прикладі. Переваги і недоліки моделі.
37. Моделі життєвого циклу розробки програмного забезпечення. Раціональний уніфікований процес. Пояснити на прикладі. Переваги і недоліки моделі.
38. Методологія розробки програмного забезпечення. Екстремальне програмування.

## РОЗДІЛ 4

### ЗАСОБИ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

1. Загальна архітектура комп'ютера. Класифікація архітектур комп'ютерів за Флінном. Основні положення архітектури комп'ютера Фон Неймана.

2. Загальні принципи передачі файлів та повідомлень через комп'ютерні мережі, переваги пакетної передачі. Характеристика рівням моделі взаємодії відкритих систем (Open System Interconnect - OSI), приклади протоколів різних рівнів.

3. Формат MAC-адреси, поняття однопунктових (Unicast), багатопунктових (Multicast) та широкомовних (Broadcast) MAC-адреси. Приклади використання протоколами кадрів із такими типами MAC-адрес.

4. Поняття колізії при організації колективного доступу до середовища передачі в технології Ethernet. Алгоритм метода колективного доступу до середовища передачі з розпізнанням несучої та виявленням колізій (Carrier-Sense-Multiply-Access with Collision Detection — CSMA/CD).

5. Загальна характеристика технології Ethernet. Успадковані технології Fast Ethernet, Gigabit Ethernet і 10 Gigabit Ethernet, їх спільні риси з Ethernet та особливості.

6. Загальна характеристика протоколів мережного рівня. Формат IP-адреси та принцип виділення блоків IP-адрес за класовою схемою. Особливі IP-адреси та правила їх вживання. Планування IP-підмереж за допомогою масок підмереж.

7. Відображення IP-адрес на адреси канального рівня за допомогою протокола ARP. Модель роботи ARP. Відображення доменних імен на IP-адреси, етапи процесу визначення IP-адреси за доменним ім'ям.

8. Основні задачі протоколу IP. Формат заголовку та характеристика полів, що вирішують задачі IP.

9. Принципи маршрутизації IP-пакетів у складеній мережі, структура таблиць маршрутизації та команди задавання статичного маршруту до мереж.

10. Протоколи динамічної маршрутизації. Поняття, визначення, основні алгоритми роботи.

11. Загальна характеристика та основні задачі протоколу контрольних повідомлень Інтернету ICMP. Типи повідомлень ICMP. Принципи використання ICMP повідомлень в утилітах ping та traceroute.

12. Задачі протоколів транспортного рівня. Порівняльна характеристика протоколів TCP та UDP. Поняття портів та адрес сокетів, розподіл номерів портів. Номери портів популярних протоколів прикладного рівня.

13. Процедури встановлення з'єднання, передачі даних та завершення з'єднання протоколом TCP. Поля та прапорці заголовків пакетів, що використовуються при цих процедурах.

14. Формат заголовку TCP. Використання полів та прапорців для цілісної та гарантованої передачі даних. Спосіб керування потоком даних за допомогою "вікна".

15. Поняття телекомунікацій. Структура територіальної мережі. Приклади протоколів та технологій територіальних мереж, побудованих на комутації віртуальних каналів та комутації пакетів/комірок.

16. Поняття бази даних та системи управління базами даних (СУБД). Поняття сутності, атрибуту, зв'язку та домену та їх ілюстрація на довільному прикладі. Найбільш поширені СУБД.

17. Порівняльна характеристика реляційної та об'єктно-орієнтованої моделей даних. Приклади СУБД, що використовують такі моделі.

18. Проблема надлишкового дублювання та узгодження введення дубльованих даних. Вимоги до вибору ідентифікаторів записів. Аномалії вставки, видалення та модифікації даних. Поняття цілісності даних.

19. Базові типи зв'язків між даними. Формування зв'язків типів "один до одного", "один до багатьох" та "багато до багатьох". Приклади даних із такими типами зв'язків.

20. Елементи моделі сутність-зв'язок. Основні визначення ER-діаграм. Приклади ER-діаграм з різноманітними типами зв'язків.

21. Поняття нормалізації бази даних. Використання методу нормальних форм для оптимізації структури відношень. Залежності між атрибутами. Виявлення залежностей між атрибутами.

22. Критерій першої нормальної форми, поняття та приклади повторюваних груп та способу їх усунення. Критерій другої нормальної форми, поняття та приклади функціональної залежності атрибутів та визначника (детермінанта).

23. Критерій третьої нормальної форми, поняття та приклади транзитивної залежності між атрибутами відношення. Критерій нормальної форми Бойса-Кодда.

24. Команди маніпуляції із таблицями: команда додавання стовпців, команда модифікації характеристик стовпців, команда перейменування та команда видалення стовпців, команда видалення таблиць.

25. Синтаксис оператора SELECT. Приклади виразу SQL, що організує вибірку даних із двох таблиць для певного критерію.

26. Синтаксис операторів INSERT, DELETE, UPDATE. Приклади виразів SQL, що організують вставку нових даних, видалення та модифікацію даних, відповідно.

27. Використання вкладених підзапитів в мові SQL. Принцип роботи запиту зі зв'язаним підзапитом. Приклад організації запиту з підзапитом. Використання HAVING в зв'язаних підзапитах.

28. Оператор об'єднання UNION. Принцип роботи запиту при використанні оператора. Приклад використання оператора UNION. Усунення дублювання в UNION.

29. Операції поєднання таблиць шляхом посилальної цілісності. Різновиди та принципи роботи оператору об'єднання таблиць JOIN. Приклади використання оператора JOIN.

30. Використання індексування та розділення для оптимізації продуктивності баз даних. Приклад атрибутів відношень, для яких доцільна індексація та приклад виразу SQL для створення індексу.

### III КРИТЕРІЇ ОЦІНЮВАННЯ

Загальна оцінка за результатами додаткового фахового випробування визначається як середня арифметична з оцінок, отриманих вступником за відповіді на усі питання екзаменаційного білету.

Для одержання оцінки «відмінно» вступник повинен показати вільне володіння матеріалом з усіх питань білету та навести приклади практичного використання відповідних теоретичних положень.

Для одержання оцінки «добре» вступник повинен показати впевнене володіння основним матеріалом з теоретичних питань. Крім того, вступник повинен вказати галузі, де можуть бути використані відповідні теоретичні положення.

Для одержання оцінки «задовільно» вступник повинен орієнтуватися в розділах, що відповідають хоча б одному із теоретичних питань.

У разі, коли вступник не може дати навіть орієнтовної відповіді на жодне з теоретичних питань, не може продемонструвати розуміння відповідних теоретичних положень, вступник отримує оцінку «незадовільно».

#### IV СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Горбань О.М. Основи теорії систем та системного аналізу [Текст]: / Горбань О.М., Бахрушин В.Є. – Запоріжжя: ГУ "ЗІДМУ", 2004.
2. Стелтинг С. Применение шаблонов Java. Библиотека профессионала [Текст]: Пер.с англ. / Стелтинг С., Маасен О. - М.: Изд.дом «Вильямс», 2002.- 576 с.
3. Брауде Э. Технология разработки программного обеспечения. [Текст]: / Брауде Э. Пер.с англ. - СПб.: Питер, 2004. - 658с.
4. Буч Гр. Объектно-ориентированный анализ и проектирование с примерами приложений [Текст]: 3-е изд.: Пер. с англ. / Буч Гр., Максимчук Р., Энгл М. и др. - М.: ООО "И.Д. Вильямс", 2008. - 720 с.
5. Вигерс К.И. Разработка требований к программному обеспечению [Текст]: Пер. с англ. / Вигерс К.И. - М.: Русская редакция, 2004. - 575 с.
6. Липаев В.В. Программная инженерия. Методологические основы [Текст]: Учеб. / В.В.Липаев; Гос.ун-т — Высшая школа экономики. - М.: ТЕИС, 2006. - 608 с.
7. Ахо, Альфред, В., Хопкрофт, Джон, Ульман, Джеффри, Д. Структуры данных и алгоритмы. : Пер. с англ. : М. : ООО "И.Д. Вильямс", 2003. - 384 с.
8. Дж. Макконнелл Основы современных алгоритмов. 2-е дополненное издание. -М: Техносфера, 2004. - 368с.
9. Давыдов В.Г. Программирование и основы алгоритмизации: Учеб. пос. – М.: Высш. шк. 2003. – 447с.
10. Окулов С.М. Основы программирования. – М.: ЮНИМЕДИАСТАЙЛ, 2002. – 424 с.
11. Аляев Ю.А., Гладков В.П., Козлов О.А. Практикум по алгоритмизации и программированию на языке Python. – М.: Финансы и статистика, 2004. – 527 с.

12. Фараонов В.В. Turbo Pascal 7.0.: Учеб. пособие. -М.:КНОРУС, 2005. -576с.
13. Страуструп Б. Язык программирования С++. – СПб.: Невский Диалект, 2004. – 1104с.
14. Прата С. Язык программирования С++. Лекции и упражнения: Учебник. – СПб.: ООО «ДиаСофтЮП», 2005. - 1104 с.
15. Подбельский В.В. Язык Си++: Учеб.пособие. – М.: Финансы и статистика, 2003. – 560с.
16. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб.пособие. – М.: Финансы и статистика, 2004. – 600с.
17. Шилд Г. Искусство программирования на Java [Текст]: Пер.с англ. / Шилд Г., Холмс Дж. - М.: И.Д. Вильямс, 2005.- 336 с.
18. Таненбаум Э. Архитектура компьютера. -Спб.: Питер, 2002. - 704с.
19. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. - СПб.: Питер, 2000. - 672 с.
20. Таненбаум Э. Компьютерные сети. - СПб.: Питер, 2002. - 848 с.
21. TCP/IP: Архитектура, протоколы, реализация-2е изд. Д-р. Сидни Фейт McGraw-Hill, 2000. - 424с.
22. Конноли Т., Каролин Б., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика 2-е издание: Перевод с английского: Учебное пособие- М.:Издательский дом «Вильямс», 2000.