

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ
(повна назва інституту/факультету)

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри БМК

_____ Є.А. Настенко
(підпис) (ініціали, прізвище)

“ ” _____ 2016 р.

Дипломна робота

на здобуття освітньо-кваліфікаційного рівня «спеціаліст»

зі спеціальності 7.05010101 «Інформаційні управляючі системи та технології»
(код і назва)

на тему: Інформаційна система аналізу фізичного стану людини до та після реабілітації опорно-рухової системи (технічна частина)

Виконав: студент VI курсу, групи БС-42с
(шифр групи)

Новак Руслан Валерійович
(прізвище, ім'я, по батькові)

(підпис)

Керівник ст. викл. Маріц Наталія Олександрівна
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з розділів ДР доц. к.т.н. Антонова-Рафі Юлія Валеріївна
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Консультант з охорони праці доц., к.т.н. Демчук Г.В.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Консультант з економічних питань проф., д.е.н., Семенченко Н.В.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент доцент, к.б.н., доцент каф. БМІ Калашнікова Лариса Євгенівна
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Київ – 2016 року

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Дипломної роботи	Антонова-Рафі Ю.В., доц.		
Охорони праці	Демчук Г. В., доц.		
Економічних питань	Семенченко Н.В., проф		

7. Дата видачі завдання 2 листопада 2015 року

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримати завдання на ДР	02 листопада 2015р.	
2	Аналіз можливостей та методів вирішення задачі	02-15 листопада 2015р.	
3	Розробка теоретичної частини	15-30 листопада 2015р.	
4	Виконання розрахункової частини	1-31 грудня 2015р.	
5	Аналіз результатів та формування висновків на основі проведених робіт	1-15 січня 2016р.	
6	Оформлення розділу охорони праці	15-20 січня 2016р.	
7	Оформлення розділу економічних питань	20-25 січня 2016р.	
8	Подання ДР рецензенту. Отримання рецензії.	25 січня 2016р.	
9	Подання в електронному вигляді ДР та анотації до неї на сайт кафедри.	25 січня 2016р.	
10	Подання пакету документів по ДР до захисту в ЕК	26 січня 2016р.	
11	Захист ДР в ЕК	05 лютого 2016р.	

Студент

(підпис)

Новак Р.В.

(ініціали, прізвище)

Керівник роботи

(підпис)

Маріц Н.О.

(ініціали, прізвище)

АНОТАЦІЯ

Диплому роботу викладено на 84 сторінках, вони містять 5 розділів, 7 ілюстрацій, 30 таблиці, та 49 джерел в переліку посилань.

Метою роботи є розробка інформаційної системи аналізу фізичного стану людини до та після реабілітації опорно-рухової системи.

У першому розділі дані загальні визначення та завдання до предметної області. У другому розділі наведено огляд існуючих програмних рішень. У третьому розділі подано етапи розробки програмного забезпечення. В розділі 4 подано рекомендації з охорони праці в приміщенні де проводиться телеконференція. П'ятий розділ призначений розрахунку економічної ефективності програмного продукту.

Робота виконана на замовлення ТОВ «Артіфішіал Кор» та складено акт впровадження.

Робота апробована на конференціях:

- «Научные исследования и их практическое применение. Современное состояние и пути развития '2015», тема доповіді: «Клиометрия. Анализ скоринг-систем при реабилитации опорно-двигательного аппарата »;

- «Inżynieria i technologia. Priorytetowe obszary nauki», тема доповіді: «Клиометрия. Анализ скоринг-систем при реабилитации опорно-двигательного аппарата »;

Ключові слова: опорно-руховий апарат, система аналізу, Java, Firebird.

ABSTRACT

The Guidelines are written on 84 pages, consist of 5 sections, 7 pictures, 30 tables and 49 references.

The objective is to develop a framework for analysis of the physical condition of the person before and after rehabilitation of the musculoskeletal system.

The first section details the general definitions and tasks to the domain. The second section provides an overview of existing software solutions. The third section presents the stages of software development. Section 4 provides guidance on health and safety at the premises where the teleconference. The fifth section is calculating the economic efficiency of the software.

This work was commissioned by the TOB «Артiфiцiал Кор» and the successful implementation in organization.

Work approved at conferences:

- «Scientific research and its practical application. Current status and the development of "2015», the theme of Reports of «Cliometrics. An analysis of scoring systems in the rehabilitation of the musculoskeletal system»;

- «Inżynieria i technologia. Priorytetowe obszary nauki», theme of the report: «Cliometrics. An analysis of scoring systems in the rehabilitation of the musculoskeletal system»;

Keywords: musculoskeletal system analysis, Java, Firebird.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	8
ВСТУП	9
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Опорно-рухова система людини	11
1.1.1 Будова скелета.....	12
1.1.2 М'язи, їх будова та функції.	17
1.2 Java.....	20
1.2.1 Історія створення.....	20
1.2.2 Безпечність.....	22
1.3.3 Ефективність.....	23
1.3.4 Об'єктно-орієнтована спрямованість.....	24
1.3.5 Стійкість до помилок.....	26
1.3. Перспективи застосування.	27
2 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ ПРОЕКТІВ.....	29
2.1 АРМ Лікаря.....	29
2.4 Висновки	30
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	31
3.1 Обґрунтування вибору мов програмування	31
3.2 Визначення моделі життєвого циклу розробки програмного продукту	33
3.3 Технічне завдання	34

БС42с.4108.1300.3166.ПЗ				
Вим	ЛИСТ	№ докум.	ПІДПИС	Дата
Розробила	Новак Р.В.			Інформаційна система аналізу фізичного стану людини до та після реабілітації опорно-рухової системи (технічна частина)
Перевірів	Антонова-Рафі Ю.В.			
Реценз.				
Н. Контр.	Кисляк С.В.			
Затвердив	Настенко Є.А.			
				Лім.
				ЛИСТ
				ЛИСТІВ
				5
				104
НТУУ "КПІ" ФБМІ БС-42с				

3.3.1 Назва програми.....	34
3.3.2 Призначення та область застосування	34
3.3.3 Вимоги до програми	34
3.3.4 Умови експлуатації	35
3.3.5 Техніко-економічні показники	37
3.3.6 Стадії і етапи розробки.....	37
3.3.7 Порядок контролю і приймання	39
4 ОХОРОНА ПРАЦІ ПРИ НАДЗВИЧАЙНИХ СИТУАЦІЯХ	40
4.1 Вступ	40
4.2 Характеристика робочого приміщення	40
4.3 Аналіз потенційних небезпек, що створюються в процесі експлуатації комплексу та розробка заходів по покращенню умов праці	43
4.3.1 Механічні чинники	43
4.3.2 Мікроклімат	44
4.3.3 Освітлення.....	45
4.3.4 Шум	46
4.3.5 Вібрація	47
4.3.6 Електробезпека.....	48
4.3.7 Пожежна безпека при надзвичайних ситуаціях.....	49
4.4 Висновок	50
5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ПРОДУКТУ	51
5.1 Вступна частина	51
5.2 Постановка задачі техніко-економічного аналізу	52
5.3 Обґрунтування функцій програмного продукту.....	52
5.3.1 Формування варіантів функцій.....	52
5.3.2 Варіанти реалізації основних функцій.....	53

					БС42с.4108.1300.3166.ПЗ			
Вим	ЛИСТ	№ докум.	ПІДПИС	Дата				
Розробила	Новак Р.В.				<i>Інформаційна система аналізу фізичного стану людини до та після реабілітації опорно-рухової системи (технічна частина)</i>	Лім.	ЛИСТ	ЛИСТІВ
Перевірів	Антонова-Рафі Ю.В.						6	104
Реценз.						<i>НТУУ "КПІ" ФБМІ БС-42с</i>		
Н. Контр.	Кисляк С.В.							
Затвердив	Настенко Є.А.							

5.4 Обґрунтування системи параметрів програмного продукту	56
5.4.1 Опис параметрів	56
5.4.2 Кількісна оцінка параметрів	56
5.5 Аналіз рівня якості варіантів реалізації функцій.....	60
5.6 Економічний аналіз варіантів розробки програмного продукту	62
5.6.1 Визначення трудомісткості.....	62
5.6.2 Розрахунок заробітної плати розробників.....	64
5.6.3 Розрахунок собівартості ПП	64
5.6.4 Розрахунок показників економічної ефективності.....	66
5.7 Висновки	67
ЗАГАЛЬНІ ВИСНОВКИ	68
ПЕРЕЛІК ПОСИЛАНЬ.....	70
Додаток А.....	75

					БС42с.4108.1300.3166.ПЗ			
<i>Вим</i>	<i>ЛИСТ</i>	<i>№ докум.</i>	<i>ПІДПИС</i>	<i>Дата</i>				
<i>Розробила</i>	<i>Новак Р.В.</i>				<i>Інформаційна система аналізу фізичного стану людини до та після реабілітації опорно-рухової системи (технічна частина)</i>	<i>Літ.</i>	<i>ЛИСТ</i>	<i>ЛИСТІВ</i>
<i>Перевірів</i>	<i>Антонова-Рафі Ю.В.</i>						7	104
<i>Реценз.</i>						<i>НТУУ "КПІ" ФБМІ БС-42с</i>		
<i>Н. Контр.</i>	<i>Кисляк С.В.</i>							
<i>Затвердив</i>	<i>Настенко Є.А.</i>							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ПЗ – програмне забезпечення

Java - об'єктно-орієнтована мова програмування, випущена компанією Sun Microsystems

СУБД – система управління базою даних

Firebird – компактна, крос-платформова, вільна реляційна система керування базами даних

JDBC – Java DataBase Connectivity — прикладний програмний інтерфейс Java, який визначає методи, з допомогою яких програмне забезпечення на Java здійснює доступ до бази даних

Бібліотека – набір готового функціоналу певної мови програмування

Фреймворк – інфраструктура програмних рішень, що полегшує розробку складних систем

НС – навколишнє середовище;

ГДК – гранично допустима концентрація;

СН – санітарні норми;

ДСТУ – Державний стандарт України

ЗІЗ – засоби індивідуального захисту.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		8

ВСТУП

Щодня у нашому світі розробляються і створюються новітні інформаційні технології. Вони дають можливість розвиватися, при цьому додають все більше комфорту в нашому житті.

Сучасні медичні організації працюють і накопичують величезні обсяги даних. Від того, наскільки ефективно ця інформація використовується лікарями, керівниками, що управляють органами, залежить якість медичної допомоги, загальний рівень життя населення, рівень розвитку країни в цілому і кожного її територіального суб'єкта зокрема.

При цьому необхідно мати на увазі, що сьогодні акцент поступово зміщується з вирішення завдань по інформатизації адміністративних управлінських структур на реалізацію проектів інформатизації медичної діяльності. Тобто пріоритети почали змінюватися у бік таких напрямів, як автоматизація робочого місця лікаря, створення єдиної електронної медичної карти, електронного обліку пацієнтів, поширення інтелектуальних систем підтримки ухвалення лікарських рішень, застосування і широке поширення новітніх медичних технологій, розвиток телемедицини

Актуальність. Створення інформаційної системи аналізу обумовлена необхідністю використання постійно зростаючих обсягів інформації при вирішенні діагностичних, завдань. Зазвичай більша частина робочого часу медичного персоналу витрачається на оформлення талонів, записів в амбулаторній картці, призначень консультацій або обстежень, а також отримання необхідної звітності.

Використання сучасних комп'ютерних технологій надасть можливість полегшити та підвищити ефективність роботи медичних працівників.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		9

Створення інформаційної системи аналізу стану пацієнта до та після реабілітації опорно-рухової системи спрямовано на розробку автоматизації робочого місця лікаря реабілітолога в частині введення та обробки даних медичної документації й одержання потрібної звітності. Автоматизація здатна не просто скоротити робочий час, але й створити принципово новий інструмент у сфері обслуговування населення в медичних установах.

Мета роботи: аналіз існуючих рішень та розробка інформаційної системи аналізу фізичного стану пацієнта до та після реабілітації.

Об'єкт. Інформаційна система аналізу фізичного стану пацієнта до та після реабілітації..

Предмет. Розробка програмного забезпечення системи аналізу фізичного стану пацієнта до та після реабілітації.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		10

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опорно-рухова система людини

Переміщення в просторі і зміна положення тіла - одна з основних функцій організму. Цю функцію за-безпечує опорно-руховий апарат, який складається з активної і пасивної частин. Пасивна частина - це кістки і з'єднання між ними. Активна частина опор-но-рухового апарата представлена скелетними поперечнопосмугованими м'язами[1].

З кісток утворений скелет (skeleton, від грецько-го skeletos - висушений), що виконує різноманітні функції. Це в першу чергу опора і пересування. Фор-ма тіла людини також залежить від скелета, який має бічну симетрію і сегментну будову.

Кістки є важелями, що приводяться в рух м'язами. При скороченні м'язів частини тіла змінюють своє положення і переміщують тіло в просторі. До кісток прикріплюються зв'язки, фасції та інші спо-лучнотканинні утвори. Крім цього, кістки скелета утворюють стінки порожнин (порожнини черепа, грудної клітки, таза, хребтовий канал), які надійно захищають розташовані в них органи від ушкоджень. Так, у порожнині черепа розташований головний мозок, у хребтовому каналі - спинний мозок, у по-рожнині грудної клітки - серце і великі судини, ле-гені, стравохід тощо, у порожнині таза - сечостатеві органи. Кістки беруть участь у мінеральному обміні, депонують солі кальцію, фосфору тощо. Жива кістка містить вітаміни А, D, С та інші[3].

Кістки побудовані з кісткової та хрящової тканини, яка складається з клітин і щільної міжклітинної речовини. Хрящ і кістка об'єднані між собою спільністю походження, будови і функції. Так, попередником більшості кісток є хрящ. Хрящовий скелет у зарод-ка людини складає приблизно 60 % від маси всього тіла. Поступово хрящ замінюється кісткою. Хрящові моделі довгих кісток, хребців, кісток основи черепа, кінцівок замінюються кісткою. Ріст цих

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		11

кісток за-безпечується проліферативною активністю хряща. Кістки лицевого черепа і склепіння мозкового черепа розвиваються з перетинчастого черепа, минаючи стадію хряща. Такі ж стадії розвитку проходить ключиця. Хрящі вушної раковини і повітроносних шляхів не замінюються кісткою.

Одним з важливих моментів пристосування організму до навколишнього середовища є рух. Він здійснюється системою органів, до яких належать кістки, їх з'єднання і м'язи, що об'єднані в єдине ціле — апарат руху, або опорно-рухову систему. Усі кістки, з'єднані між собою різними видами сполучної тканини, утворюють скелет — пасивну частину апарату руху, а прикріплені до кісток скелетні м'язи — його активну частину.

Кістково-м'язова система підтримує людину у вертикальному положенні (кістки хребта і нижніх кінцівок). Захисну функцію виконують кістки черепа, хребта, грудної клітки, таза, а також м'язи живота, захищаючи від впливу зовнішніх факторів головний і спинний мозок, легені, серце та органи черевної порожнини. Рухову функцію виконують кістки і м'язи кінцівок, хребет, м'язи спини і грудей[2].

Кровотворну функцію виконують переважно плоскі кістки, в яких міститься червоний кістковий мозок. Як депо деяких мінеральних солей і мікроелементів, скелет бере участь у мінеральному обміні (солей кальцію, заліза, фосфору), а м'язи, депонуючи глікоген, — у вуглеводному обміні.

1.1.1 Будова скелета

Скелет людини, хоча й подібний до скелета тварин, має певні особливості, що пов'язано з прямоходінням та трудовою діяльністю. Скелет людини поділяють на скелет тулуба, скелет верхніх і нижніх кінцівок, скелет голови — череп (Рисунок 1.1).

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		12

Скелет тулуба складається з хребта і грудної клітки. Хребет поділяють на п'ять відділів: шийний, що складається з 7 хребців, грудний — з 12, поперековий — з 5, крижовий — з 5 і куприковий — з 4—5 хребців. Останні 9—10 хребців зрослися між собою. Отже, в хребті налічується 33—34 хребці. Хребці — короткі кістки, розміщені одна над одною. Хребець складається з тіла, дуги і семи відростків. Задній відросток (непарний) можна промацати на спині у вигляді горбка. Між дугою й тілом хребця є отвір. Отвори усіх хребців утворюють хребетний канал, в якому розміщений спинний мозок. Розміри тіл хребців залежать від навантаження на них: вони найменші в шийному відділі і найбільші — в поперековому. Хребці, з'єднуючись один з одним за допомогою хрящів і зв'язок, утворюють гнучкий і еластичний хребетний стовп, якому властива значна рухливість.

Для хребта людини характерні вигини, які надають йому форми, що в профіль нагадує латинську літеру S. Шийний і поперековий вигини спрямовані вперед (лордоз), грудний і крижовий — назад (кіфоз). Така форма хребта є результатом прямоходіння, вона надає хребту пружності. У новонародженого хребет прямий (крім невеликого крижового вигину) і тільки з розвитком прямоходіння утворюються всі вигини: шийний, коли дитина починає тримати голову, грудний, коли починає сидіти, поперековий і крижовий, коли починає стояти. Вигини хребта збільшують розміри грудної й тазової порожнин, полегшують утримання рівноваги і забезпечують пом'якшення поштовхів та струсів тіла під час стрибків і бігу[4].

Сильні вигини грудного відділу (сутулість, горб) або поперекового ("внутрішні горби"), а також більш-менш виражені вигини вбік (сколіоз) розвиваються при різних захворюваннях (рахіт) або внаслідок неправильної пози під час роботи (наприклад, під час тривалого сидіння за партою, столом).

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		13

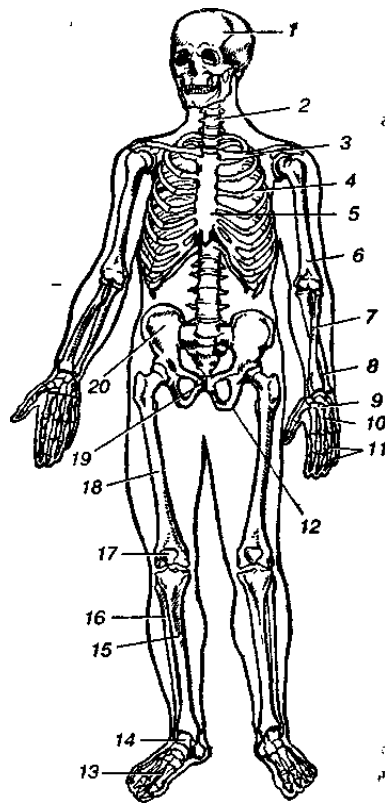


Рисунок 1.1 Скелет людини:

1 — череп; 2 — хребет; 3 — ключиця; 4 — ребро; 5 — груднина; 6 — плечова кістка; 7 — променева кістка; 8 — ліктьова кістка; 9 — кістки зап'ястка; 10 — кістки п'ястка; 11, 13 — фаланги пальців; 12, 19, 20 — тазові кістки (12 — сіднична, 19 — лобкова, 20 — клубова); 14 — кістки зап'ястка і плесна; 15 — великогомілкова кістка; 16 — малогомілкова кістка; 17 — надколінок; 18 — стегнова кістка

Грудна клітка утворена непарною грудною кісткою (грудниною), 12 парами ребер і грудними хребцями. Ребра рухомо-з'єднані з хребцями (у них є суглобові ямки для зчленування з голівками ребер) і напіврухомо (за допомогою хрящів) з грудною кісткою. Грудна кістка, або груднина, належить до плоских кісток. З нею з'єднуються передніми кінцями 7 пар ребер (згори вниз). Наступні 3 пари з'єднуються одна з одною своїми хрящами (нижня до верхньої), утворюючи реберну дугу. Передні кінці 11ї і 12ї пар ребер не досягають дуги і вільно закінчуються в м'язах черевної стінки. Грудна клітка обмежує грудну порожнину, в якій розміщені серце, легені, стравохід, трахея, судини і нерви. Вона бере участь у дихальних рухах завдяки ритмічним

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		14

скороченням міжреберних м'язів. Форма грудної клітки зазвичай схожа на конус, зрізана верхівка якого звернута вгору, а основа — вниз, і залежить від статі, віку, статури й фізичного розвитку людини. Вона може бути широкою і короткою, довгою і вузькою. У зв'язку з прямоходінням передньозадній розмір грудної клітки людини, на відміну від тварин, менший від поперечного. Добре розвинена грудна клітка в людей, які займаються фізичною працею, фізкультурою і спортом.

Скелет верхніх кінцівок складається із скелета плечового поясу, за допомогою якого кінцівки з'єднуються з тулубом (лопатка, ключиця), і скелета вільних верхніх кінцівок.

Лопатка — плоска кістка трикутної форми, яка передньою поверхнею прилягає до задньої стінки грудної клітки. Її зовнішній кут має суглобову ямку, до якої приєднується голівка плечової кістки.

Ключиця має сподібну форму, її контури добре видно під шкірою і її завжди можна промацати. Груднинний кінець ключиці з'єднується з грудниною, зовнішній — з відростком лопатки. Функція ключиці полягає в тому, що вона відсуває плечовий суглоб від грудної клітки, забезпечуючи значну свободу рухів верхніх кінцівок.

Скелет вільної верхньої кінцівки утворюють плечова кістка, дві кістки передпліччя — ліктьова і променева та кістки кисті. Кисть утворена кістками зап'ястка, п'ястка та фалангами пальців. Зап'ясток складається з двох рядів дрібних кісток, що з'єднуються між собою, з передпліччям і кістками п'ястка (до складу зап'ястка входить вісім кісток). П'ять п'ясткових кісток входить до складу кисті. Фаланги утворюють скелет пальців. Перший, або великий, палець має дві фаланги, решта пальців — по три[6].

Скелет нижніх кінцівок утворений тазовим поясом і скелетом вільних нижніх кінцівок.

До складу тазового поясу входять дві тазові кістки, які спереду з'єднані одна з одною, а ззаду утворюють з крижовим відділом хребта малорухомий

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		15

суглоб. У дітей тазова кістка складається з трьох кісток (клубової, сідничної і лобкової), відокремлених одна від одної хрящем. Тільки після 16 років вони зростаються в одну. На місці їх зростання є вертлюжна суглобова западина, куди входить голівка стегнової кістки. Складна будова тазової кістки зумовлена її функцією. З'єднуючись із стегном і крижовим відділом хребта, тазові кістки переносять масу тіла на нижні кінцівки і забезпечують функцію руху, опори і захисту. У зв'язку з вертикальним положенням тіла людини таз у неї ширший і масивніший, ніж у тварин. Він має вигляд чаші і підтримує внутрішні органи черевної порожнини. Таз жінки ширший і коротший, ніж у чоловіка. Це зумовлено значенням таза жінки як родового каналу.

До кісток вільної нижньої кінцівки належать стегнова кістка, надколінок, великогомілкова, малогомілкова і кістки стопи. Великогомілкова й малогомілкова кістки утворюють гомілку. Стопу утворюють кістки заплесна, плесна і фаланги пальців. Заплесно має 7 кісток, з яких найбільшими є п'яткова й надп'яткова. З надп'ятковою кісткою гомілкові кістки утворюють суглоб. Плесно складається з п'яти довгих кісток. Фаланги утворюють скелет пальців, їх кількість така сама, як і в руці. Стопа як єдине ціле виконує опорну функцію під час стояння і ходіння. З'єднуючись між собою, кістки стопи утворюють склепіння, яке опуклою частиною звернене вгору. Така будова пов'язана з "вертикальним положенням тіла людини, із збільшенням навантаження на стопу при прямоходінні. Склепінчаста стопа значно пом'якшує поштовхи під час ходіння.

Кістки голови, або череп, міцно з'єднані між собою швами. Винятком є нижня щелепа. Череп утворює коробку для мозку, оточує отвори, де починаються органи дихання, травлення і органи чуття. Кістки черепа також виконують опорну функцію для дихальних шляхів (порожнина носа) і травної системи (порожнини рота, глотки). Скелет голови поділяють на мозковий і лицевий

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		16

У скелеті всі кістки з'єднуються між собою різними способами залежно від ролі певної кістки чи групи кісток. З'єднання кісток бувають неперервні, або нерухомі чи малорухомі (без щілин), і переривчасті, або рухомі (з порожнинами між кістками). Є й проміжні форми з'єднання — напівсуглоби (наприклад, з'єднання тазових кісток між собою). Неперервно кістки з'єднуються зрощенням (наприклад, кістки таза), зв'язками (хребці), хрящем (ребра) або швами (кістки мозкового черепа).

Переривчасте з'єднання кісток, або суглоб (мал. 104), — більш молодий утвір організму. Всі суглоби мають загальний план будови, який включає суглобову порожнину, суглобову сумку і суглобові поверхні. Суглобову порожнину виділяють умовно, оскільки в нормі між суглобовою сумкою і суглобовими кінцями кісток порожнини не існує, а міститься рідина.

Суглобова сумка охоплює суглобові поверхні кісток, утворюючи герметичну капсулу. Суглобова сумка складається з двох шарів. Зовнішній шар переходить в окістя, внутрішній шар виділяє в порожнину суглоба рідину, що відіграє роль мастила, забезпечуючи вільне ковзання суглобових поверхонь[5].

Структура кісток є динамічною реактивною системою, що змінюється упродовж усього життя людини. Відомо, що у людей, які займаються важкою фізичною працею, компактний шар кістки досягає відносно більшого розміру. Залежно від зміни навантаження на окремі частини тіла можуть змінюватися розміщення кісткових перекладок і структура кістки загалом.

1.1.2 М'язи, їх будова та функції.

Скелетні м'язи — це активна частина опорнорухової системи, скорочення якої зумовлює переміщення частин тіла і всього тіла в просторі.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		17

М'яз, як і всі інші органи, має складну будову (мал. 106). До його складу входить кілька тканин. Основу скелетного м'яза становить посмугована м'язова тканина. В кожному м'язі розрізняють активну (м'язове черевце, або тіло) і пасивну (сухожилки) частини. Як правило, м'яз має два сухожилки, якими він прикріплений до кісток (місце прикріплення сухожилка до кістки позбавлене окістя і називається горбистістю).

М'язове черевце складається з багатьох м'язових волокон, що утворюють пучки різних розмірів і забезпечують скорочення. В кожному пучку м'язові волокна зв'язані одне з одним пухкою волокнистою сполучною тканиною у вигляді тонкої сітки. Пучки м'язових волокон сполучені між собою прошарками сполучної тканини. Весь м'яз зовні також вкритий сполучною тканиною.

Сухожилки м'яза побудовані з щільної сформованої сполучної тканини. Колагенові волокна сухожилків проникають у м'язове черевце і там навколо посмугованих м'язових волокон утворюють кілька шарів, які міцно сполучають черевце із сухожилком. Сукупність сполучної тканини утворює так званий м'який скелет м'яза.

До м'яза, як і до інших органів, підходять нерви і кровоносні судини. В складі нервів проходять рухові (відцентрові, еферентні) і чутливі (доцентрові, аферентні) волокна. Нервові імпульси, що передаються по рухових нервових волокнах від мозку до м'яза, зумовлюють його скорочення. По чутливих нервових волокнах у мозок надходить інформація з м'язових рецепторів про стан м'яза. Як орган з інтенсивним обміном речовин, м'яз має добре кровопостачання, інтенсивність якого регулюється автономною (вегетативною) нервовою системою. Найчастіше один і той самий м'яз отримує кров, а з нею і поживні речовини з кількох артерій[4].

Скорочуючись, м'яз стає коротшим і товщим, при цьому він виконує певну механічну роботу. Величина виконуваної роботи залежить від сили його скорочення і довжини, на яку м'яз коротшає. Сила м'яза пропорційна кількості

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		18

м'язових волокон, з яких він побудований, а точніше — площі поперечного перерізу всіх м'язових волокон м'яза.

Скелетні м'язи, перекидаючись через суглоб, а іноді через два і більше суглобів, прикріплюються своїми кінцями до різних кісток. Вкорочення м'яза під час скорочення супроводжується зближенням його кінців і кісток, до яких прикріплений м'яз. При цьому кістки й суглоби, в яких здійснюється рух кістки, виконують функцію важелів.

У здійсненні кожного руху бере участь, як правило, кілька груп м'язів, причому м'язи однієї групи, наприклад передні м'язи плеча, скорочуються, а м'язи протилежної групи (задні) в цей час розслабляються. Завдяки одночасному скороченню і розслабленню протилежних груп м'язів забезпечується плавність рухів. М'язи, що виконують одну й ту саму роботу — один і той самий рух у певному суглобі, називають синергістами, а м'язи, що діють у протилежних напрямках, — антагоністами. Наприклад, усі м'язи, які здійснюють згинання плечового суглоба, складатимуть групу синергістів цього руху, синергістами між собою будуть і м'язи, що зумовлюють розгинання цього суглоба. Однак ці дві групи м'язів — згиначі і розгиначі — одна відносно одної будуть антагоністами. Антагоністична дія м'язів — важливе пристосування в роботі рухового апарату. Під час кожного руху напружуються не лише м'язи, що його здійснюють, а й їхні антагоністи, протидіючи тязі і тим самим надаючи рухові точності й плавності.

Узгоджене чергування скорочення і розслаблення різних груп м'язів та координація всіх рухів здійснюються нервовою системою і мають рефлекторний характер. Якщо ми, наприклад, наступили на щонебудь гостре або доторкнулись до чогось гарячого, то руку чи ногу ми відсмикнемо ще до того, як виникне відчуття болю. Гаряче тіло або колючий предмет подразнюють рецептори шкіри. Збудження, що виникло в них, по доцентрових нервових волокнах передається в центральну нервову систему, де відбувається передавання збудження на відцентровий нейрон, і до м'яза надходить імпульс,

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		19

який викликає його скорочення і відсмикування ноги чи руки. В той же час у м'язах антагоністах виникає гальмування, і вони розслабляються. Інколи м'язи згиначі і м'язи розгиначі можуть одночасно перебувати в розслабленому (рука вільно провисає вздовж тіла) або скороченому (рука зафіксована зігнутою в ліктьовому суглобі) стані.

Часто м'язові рефлекси виникають у відповідь на подразнення рецепторів, що містяться в самих м'язах або сухожилках. Прикладом може бути колінний рефлекс. Складні дії нашого повсякденного життя, наприклад ходіння, здійснюються внаслідок узгодженої дії не окремого органа, а цілої групи органів опорно-рухового апарату, що зумовлено регулювальною функцією нервової системи.

Під час скорочення м'яз виконує роботу, яку можна виміряти. Для цього значення маси вантажу, який підіймає м'яз, множать на висоту його підняття. При збільшенні вантажу робота зростає, а потім, досягши певного рівня (максимуму для кожного окремого м'яза), поступово знижуватиметься. Якщо вантаж досить великий і м'яз не здатний його підняти, робота дорівнюватиме нулю. Якщо взяти середній для певного м'яза вантаж і підіймати його з різною частотою, то виявиться, що найбільша робота м'яза спостерігатиметься при середньому ритмі рухів[7].

1.2 Java

1.2.1 Історія створення

Мова програмування Java зародилася в 1991 р. в лабораторіях компанії Sun Microsystems. Розробку проекту започаткував Джеймс Гослінг, сам проект мав назву «Green» (Зелений). Створення першої робочої версії, яка мала назву «Oak» (дуб), зайняло 18 місяців. Оскільки виявилось, що ім'я Oak уже використовувалось іншою фірмою, то в результаті тривалих суперечок

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		20

навколо назви нової мови з поміж ряду запропонованих було вибрано назву Java, у 1995 р. мову було офіційно перейменовано[8].

Головним мотивом створення Java була потреба в мові програмування, яка б не залежала від платформи (тобто від архітектури) і яку можна було б використовувати для створення програмного забезпечення, яке вбудовується в різноманітні побутові електронні прилади, такі як мобільні засоби зв'язку, пристрої дистанційного керування тощо.

Досить скоро майже всі найпопулярніші тогочасні веб-оглядачі отримали можливість запускати «безпечні» для системи Java аплети всередині веб-сторінок. У грудні 1998 р. Sun Microsystems випустила Java 2 (спершу під назвою J2SE 1.2), де було реалізовано декілька конфігурацій для різних типів платформ. Наприклад, J2EE призначалася для створення корпоративних застосунків, а значно урізана J2ME для приладів з обмеженими ресурсами, таких як мобільні телефони. У 2006 році в маркетингових цілях, Версії J2 було перейменовано у JavaEE, JavaME та JavaSE, відповідно.

13 листопада 2006 року Sun випустили більшу частину Java в якості вільного та відкритого програмного забезпечення згідно з умовами GNU General Public License (GPL). 8 травня 2007 корпорація закінчила процес, в результаті якого всі початкові коди Java були випущенні під GPL, за винятком невеликої частини коду, на який Sun не мала авторського права.

Період становлення Java збігся у часі з розквітом міжнародної інформаційної служби WorldWideWeb. Ця обставина відіграла вирішальну роль у майбутньому Java, оскільки Web теж вимагала платформи-незалежних програм. Як наслідок, були зміщені акценти в розробці Sun з побутової електроніки на програмування для Інтернет. [9]

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		21

1.2.2 Безпечність

Один із ключових принципів розробки мови Java полягав у забезпеченні захисту від несанкціонованого доступу. Програми на Java не можуть викликати глобальні функції й одержувати доступ до довільних системних ресурсів, що забезпечує в Java рівень безпеки, недоступний для інших мов. Даний рівень безпеки виконання Java-програм забезпечує віртуальна машина Java, котра вбудована в операційну систему. Об'єктна модель у Java проста і легко розширюється, у той же час, заради підвищення продуктивності прості типи даних Java не є об'єктами.

WorldWideWeb висунула Java на передній план програмування, і Java, в свою чергу, сильно вплинула і навіть змінила обличчя Internet, розширивши спектр об'єктів, які можуть розповсюджуватись у кіберпросторі. Програми нової форми - аплети - завантажуються з віддаленого сервера і можуть запускатися динамічно, тобто без участі користувача. До появи Java такий підхід був неприпустимий з міркувань безпеки та переносимості. В архітектурі аплетів зроблено ряд штучних обмежень, які роблять їх цілком безпечними. Перш за все, Java є інтерпретованою мовою і простір ресурсів Java-програми обмежений так званою віртуальною Java-машиною (VJM), яка може контролювати поведінку програми і захищати систему від побічних ефектів, які можуть виникати з вини аплету. Крім того, в мові Java є додаткові обмеження, які не дозволяють аплету стати «троянським конем». Зокрема, Java-аплет не може отримати доступ до локального жорсткого диску. При такій спробі генерується виключна ситуація.[8]

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		22

1.3.3 Ефективність

Оскільки аплети Java інтерпретуються, а не компілюються, то їх виконання на різних платформах значно полегшується. В цьому випадку достатньо створити для кожної платформи виконуючу Java-систему. Якщо існує така система для даної операційної системи, то будь-яка Java-програма може виконуватись в даному середовищі без додаткової компіляції на цій платформі. Проте Java не є інтерпретованою мовою в чистому розумінні. Програма на Java компілюється. Результатом роботи компілятора Java є байткод (bytecode). Байткод - це оптимізований набір команд, призначений для виконання уявним пристроєм - віртуальною Java-машиною. В такий спосіб витрати на інтерпретацію зводяться до мінімуму, оскільки байткод вже є оптимізованим, і досягається досить висока продуктивність Java-програм. Наведені вище особливості дають підставу розглядати Java не як ще одну мову програмування, а як окрему інформаційну технологію. Таким чином, інтерпретація - це найлегший шлях до перенесення програм, реалізований в Java технології. Незважаючи на те, що мова Java була розроблена в розрахунок на інтерпретацію, технічно немає нічого такого, що б перешкоджало компіляції байткоду в виконуваний код. До байткоду, який пересилається по мережі, застосовується динамічна компіляція, але це ніяк не впливає на переносимість та безпеку, оскільки роботу програми все ще контролює виконуюча система. Такий підхід застосовано в багатьох виконуючих системах Java, що забезпечує продуктивність на рівні оптимізованого коду C++.

Мова Java є однією з наймолодших в сімействі мов програмування і була розроблена з розрахунку на те, щоб професійний програміст міг легко її опанувати та ефективно використовувати. За основу Java взятий синтаксис C++ - безсумнівно однієї з найбільш популярних мов програмування

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		23

сучасності. Проте, Java - це цілком самостійна мова програмування, і при її створенні не йшлося про будь-яку сумісність з C++. Тому деякі механізми реалізовані в Java інакше, а деякі зовсім відсутні. Ідеологічно ж Java побудована дещо інакше ніж C++. Розробники Java ґрунтувалися на досвіді розробки програм на C++ і прагнули позбутися можливостей, які зарекомендували себе непевними. Так, в Java відсутня перегрузка операторів а також автоматичне приведення несумісних типів - конструкції, які при неуважному використанні є джерелом важких для виявлення помилок. Взагалі, інтерфейси Java більш прості та прозорі для розуміння. Написати на Java програму з графічним інтерфейсом значно легше. Звичайно, простота інтерфейсів компенсується меншою гнучкістю, бібліотека Java не така багата, як стандартні бібліотеки C/C++. Але згадаймо, що Java задуманий для використання на різних платформах і тому реалізує в собі найбільш стандартні можливості задля легшої адаптації під конкретне середовище. [10]

1.3.4 Об'єктно-орієнтована спрямованість

Від C++ Java успадкувала потужний механізм об'єктно-орієнтованого програмування. Оскільки Java розроблювався «на пустому місці», тобто не було потреби забезпечувати сумісність з попередніми версіями, розробники мали повну свободу мислення. В результаті був сформований ясний і прагматичний підхід до об'єктів. Вільно переймаючи ідеї, які реалізовувалися протягом останніх десятиріч, мові Java вдалося знайти рівновагу між парадигмою «все є об'єктом» і прагматичним підходом. Об'єктна модель Java проста і легко розширюється, в той час як просі типи, як цілі, зберігаються як дані, що не є об'єктами, що дозволяє значно підвищити швидкість при їх обробці.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		24

В Java вбудований набір ключових класів, що містять основні абстракції реального світу, з яким прийдеться мати справу вашим програмам. Основою популярності Java є вбудовані класи-абстракції, що зробили його мовою, дійсно незалежним від платформи.

Фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, подібний із C и C++. У Java використовуються практично ідентичні вимоги для оголошення змінних, передачі параметрів, операторів і для керування потоком виконання коду. У Java додані всі гарні риси C++, але виключені недоліки останнього.

Вказівники або адреси в пам'яті — найбільш могутня і найбільш небезпечна риса C++. Причиною більшості помилок у сьогоденньому коді є саме неправильна робота з вказівниками. Наприклад, одна з типових помилок — прорахуватися на одиницю в розмірі масиву і зіпсувати вміст комірки пам'яті, розташованої слідом за ним.

Хоча в Java дескриптори об'єктів і реалізовані у вигляді вказівників, у ній відсутні можливості працювати безпосередньо з ними. Ви не можете перетворити ціле число в вказівник, а також звернутися до довільної адреси пам'яті.

Проте на мові Java можна створювати не тільки апплети, а й консольні додатки, GUI-додатки, сервлети та JSP.

Компанія Sun раніше пропонувала безкоштовну еталонну реалізацію Javaweb-сервера додатків під назвою JServ. Після виходу технології J2EE весь код був переданий компанії ApacheSoftwareFoundation, а продукт змінив свою назву на Tomcat.

На даний момент Tomcat є еталонною реалізацією Javaweb-сервера і входить в групу проектів Apache під назвою Jakarta.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		25

1.3.5 Стійкість до помилок.

Багатофункціональність середовища Web висуває надзвичайно високі вимоги до надійності програм. Як наслідок, при розробці Java пріоритет був відданий можливості створення стійких до помилок програм. Java звільняє програміста від хвилювань з приводу багатьох поширених причин, які викликають помилки програмування. Як вже згадувалося, Java є строго типізованою мовою програмування. Ще виконуюча система Java бере на себе «прибирання сміття», тобто автоматично звільняє пам'ять, яка була розподілена динамічно. Звичайно, це дещо знижує ефективність коду, але запобігає типовим помилкам, коли програміст забуває звільнити виділену пам'ять, або, навпаки, звільняє пам'ять, яка ще використовується. Java підтримує об'єктно-орієнтовану обробку виключних ситуацій подібно до C++. Але на відміну від C++ в Java обробка виключних ситуацій є обов'язковою. Тобто неможливо скомпілювати програму, яка відкриває файл, не обробивши можливі помилки типу «файл не знайдено», які виникають при цьому. Добре написана Java-програма може сама обробляти всі помилки часу виконання. [6]

1.3.6 Підтримка багатозадачності.

Java розроблялася з орієнтацією на вимоги до створення інтерактивних програм, які працюють з мережею. З цією метою Java підтримує багатозадачність програмування, яке дозволяє легко розробляти програми, що викинуть багато процесів одночасно. Виконання Java-програми засновано на елегантному, але в той самий час високоорганізованому рішенні багатопроцесової синхронізації, яке дозволяє вам створювати високоефективні інтерактивні системи.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		26

У Java реалізовано кілька цікавих рішень, що дозволяють писати код, що виконує одночасно масу різних функцій і не забуває при цьому стежити за тим, що і коли повинно відбутися. У мові Java для рішення проблеми синхронізації процесів застосований найбільш елегантний із усіх коли-небудь, винайдених методів, що дозволяє конструювати прекрасні інтерактивні системи. Прості в звертанні витончені підпроцеси Java дають можливість реалізації в програмі конкретної поведінки, не відволікаючись при цьому на побудову глобальної циклічної обробки подій.

1.3.7 Незалежність від архітектури.

Основним питанням для розробників Java стало питання довготривалості та переносимості. Одна з головних проблем, із якою зустрілися програмісти, полягала в відсутності гарантій того, що написана сьогодні програма завтра працюватиме з тим же успіхом, причому на тій самій машині. Оновлення операційної системи, модернізація процесора та зміна об'єму оперативної пам'яті можуть призвести до збою програми. Розробники Java, прагнули змінити цю ситуацію і прийняли декілька важких рішень відносно мови Java та процесу виконання Java-програми. Їх мета полягала в тому, щоб «одного разу написане працювало всюди, в любий час і завжди». Внаслідок цього Java є системою, яка легко розширюється за рахунок створення нових стандартних класів та бібліотек.

1.3. Перспективи застосування.

Програми на Java можуть знайти різне застосування в навчальному процесі: інтерактивні навчаючі програми (HTML в поєднанні з Java),

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		27

програми-тести і особливо ділові ігри. Додаткові переваги можна отримати, якщо пистати ці програми у вигляді аплетів, які ініціалізуються з Web сервера внутрішньої мережі Intranet. В такий спосіб можна уникнути інсталяції програми на багатьох комп'ютерах - користувач просто запускає Web-браузер і завантажує потрібну сторінку. Для тестових програм, написаних на Java з використанням архітектури клієнт/сервер можна підвищити ступінь конфіденційності. База даних тестових запитань знаходиться на сервері в каталозі з обмеженим доступом. Коли користувач завантажує аплет, він автоматично підключається до програми-сервера, яка виконується на сервері і може видавати запитання з бази даних у відповідь на запит користувача. В такий спосіб унеможлиблюється викрадення бази даних, за умови відсутності фізичного доступу до серверу у користувачів[11].

Окреме питання - навчальні ділові ігри. Під такою грою розумітимемо гру, за участю кількох користувачів, в якій мається на увазі динамічний обмін інформацією між ними. Система безпеки Java накладає обмеження, внаслідок якого аплет може встановлювати з'єднання лише з хостом, з якого він був загрузений і ні з яким більше. Але це обмеження легко обходиться: на сервері виконується програма-сервер, з якою з'єднуються усі клієнти і через яку здійснюється обмін інформацією. Таким чином така програма повинна мати архітектуру клієнт/сервер.

2 ОГЛЯД І АНАЛІЗ ІСНУЮЧИХ ПРОЕКТІВ

2.1 АРМ Лікаря

Опис. Автоматизація діяльності амбулаторно-поліклінічних і стаціонарних служб медичних установ.

Основні можливості:

- швидкий доступ до медичної інформації пацієнта по локальній мережі або через інтернет (амбулаторна карта, стаціонарна карта, результати аналізів/обстежень, первинний огляд, щоденники, лікування, виписка, рекомендації, електронна історія хвороби)
- єдина база амбулаторних і стаціонарних хворих
- робота з медичними протоколами
- робота з довідниками (класифікатори хвороб МКХ-10, лікарських препаратів та ін.)
- створення і використання власних довідників (шаблонів) для швидкого та уніфікованого введення інформації в протоколи лікарів
- формування лікарем реєстрів відібраних пацієнтів (по довільним критеріям групування)
- зберігання медичних зображень (МРТ , КТ , УЗД тощо)
- ведення архіву пацієнтів.
- імпорт звітних форм в документи формату: *. Pdf[17]

Переваги :

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		29

2.4 Висновки

У даному розділі було проаналізовано існуючі рішення та були зроблені висновки, які недостатки є в сучасних методах, що можна покращити в розробці нашого продукту.

В цілому, ми чітко бачимо перспективи масового і обґрунтованого застосування нашого продукту. Ми розуміємо, що, навіть незважаючи на різні об'єктивні складнощі, наше програмне забезпечення буде активно розвиватися і, в тому числі, звичайно ж, і на ринку програмного забезпечення.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		30

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору мов програмування

Всім відомо, принципи Java - простота мови, безпека і можливість перенесення коду. Ці принципи не завжди виконуються. Як, наприклад, складність синтаксису вкладених класів. Але в цілому їх вплив на мову здобуло для Java перші ряди, зробило Java досить надійним інструментом, в таких областях як космічна промисловість та для фінансової сфери.

Однією з найбільших переваг мови є наявність багатой бібліотеки. З одного боку величезність бібліотеки є однією з перешкод до вивчення мови, і, як наслідок, високий поріг входження. Але завдяки високому порогу входження популярність мови знизилася, в той час як затребуваність мови залишилася колишньою.

Багата бібліотека зводить тривіальну частина роботи програміста до пошуку готового рішення. Що звільняє програміста від рутини і надає йому можливість більш серйозно вкластися в творчий аспект, заощадити час і сили.

Багатослівність і громіздкість – одна з проблем Java. І причини тут можна виділити дві: жорстка політика об'єктно-орієнтованого підходу (далі ООП), відсутність функцій першого класу.

Політика ООП введена в Java з метою позбавлення від деяких небезпек, що виникають при створенні великого проекту. У Java ж неможливо створювати функції, які не є методами якого-небудь класу[20].

Використання функцій «не методів» призводить до того, що:

- може виникнути конфлікт при іменуванні функцій, коли ім'я функції з таким же набором параметрів і такого ж типу даних вже є;

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		31

- проект в цілому може ставати більш непервороткий;
- проект стає важчий для осмислення, не тільки для нового програмісту, але й для того, хто давно в проекті.

Java, незважаючи на різні способи оптимізації, все ж досить ресурсомістка і повільна.

Причини в наступному:

- авто збір сміття;
- компіляція "на льоту" (Just In Time компіляція);
- відмова від таких небезпечних механізмів як арифметика покажчиків, неявне перетворення типів з втратою точності, функції першого класу.

Але це все робить мову більш платформо незалежною і безпечною. Авто збір сміття з одного боку звільняє програміста від турботи звільнення пам'яті, з іншого боку відома проблема, коли авто збір сміття не спрацював і відбувався серйозний витік пам'яті.

Компіляція на льоту, мабуть, одне з найбільш вдалих рішень щодо реалізації.

Веб-розробка на мові програмування Java - це даність сьогодення. Поки, основна маса додатків, розроблених на Java, це корпоративні портали і сайти банків. Тобто, ті ресурси, на яких треба поєднати безпеку з широкими функціональними можливостями. І пов'язано це з багатьма причинами: високий поріг входження при освоєнні Java, відсутність послуг хостингу з підтримкою сервера додатків, брак і неповнота можливостей мови стосовно інтернет-сайтів, відсутність російськомовної інформації з даного питання.

Але час не стоїть на місці і інформаційні технології стрімко розвиваються, удосконалюються і уніфікуються. Мови програмування

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		32

копіюють один у одного відсутні і затребувані аспекти, невблаганно рухаючись до формування якоїсь "універсальної мови", можливості якої будуть покривати всі потреби розробника.

Універсальність Java, з точки зору розробки додатків і для настільних додатків, і для інтернет-додатків, - незаперечна. Активність ІТ індустрії в створенні зручних середовищ розробки - на висоті. Ставка найбільших виробників інформаційних продуктів на мову Java - стабільна. Значить, майбутнє, в якому використання Java при створенні сайтів буде масовим явищем, зовсім не за горами.

3.2 Визначення моделі життєвого циклу розробки програмного продукту

Проект - це унікальний процес, у ході виконання якого отримують унікальний продукт. Таким чином, для розробки продукту в проекті, швидше за все повинен застосовуватися унікальний процес. Замість створення кожного проекту «з нуля», менеджер проекту може скористатися узагальненою, перевіреною на практиці методикою, адаптувавши її для конкретного проекту. Як правило, завжди є можливість вибору серед кількох «початкових» життєвих циклів.

Вибір та адаптація життєвого циклу розробки проекту впливає на методики розробки продукту. На вибір методів і інструментальних засобів також може впливати вибір життєвого циклу.

На основі аналізу вимог та вимог до вхідних і вихідних даних ПП обирається спіральна модель ЖЦ розробки ПП. Ця модель найбільш реально відображає розробку ПП і дозволяє враховувати ризики на кожному витку спіралі.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		33

3.3 Технічне завдання

3.3.1 Назва програми

Назва програми: «Система аналізу фізичного стану людини до та після реабілітації опорно-рухової системи».

3.3.2 Призначення та область застосування

Програмний продукт «Система аналізу фізичного стану людини до та після реабілітації опорно-рухової системи» повинен забезпечувати можливість виконання таких функцій:

- 1) можливість входу для лікаря та пацієнта;
- 2) реєстрація нового користувача;
- 3) зберігання результатів тренувань;
- 4) можливість дистанційного підключення.

3.3.3 Вимоги до програми

3.3.3.1 Вимоги до функціональних характеристик

Програма повинна забезпечувати можливість виконання таких функцій:

- 1) вхід під унікальним логіном та паролем користувача;
- 2) пошук пацієнта;
- 3) створення нового пацієнта;
- 4) внесення результатів тренувань;
- 5) перегляд історії тренувань;

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		34

3.3.3.2 Вимоги до надійності

Вимоги до забезпечення надійного функціонування програми

Надійне (стійке) функціонування програми має бути забезпечене виконанням Замовником сукупності таких організаційно-технічних заходів:

- а) організацією безперебійного живлення технічних засобів;
- б) використанням ліцензійного програмного забезпечення.

Час відновлення після відмови

Час відновлення після відмови, викликаний збоєм електроживлення технічних засобів, не повинен перевищувати 30-ти хвилин за умови дотримання правил експлуатації технічних та програмних засобів.

3.3.4 Умови експлуатації

3.3.4.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, за яких забезпечуються задані характеристики, мають задовольняти вимогам до технічних засобів.

3.3.4.2 Вимоги до кваліфікації та чисельності персоналу

Мінімальна кількість персоналу, що вимагається для роботи програми, має складати не менше 2 штатних одиниць — системний адміністратор та

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		35

кінцевий користувач програми — пацієнт або лікар. Системний адміністратор повинен мати вищу профільну освіту. До переліку задач, що виконуються системним адміністратором входять:

- а) задача підтримки дієздатності технічних засобів;
- б) задачі встановлення (інсталяції) і підтримка дієздатності системних програмних засобів — операційної системи та розробленої програми;
- в) задача встановлення (інсталяції) програми.

3.3.4.3 Вимоги до складу і параметрів технічних засобів

До складу технічних засобів має входити ІВМ-сумісний персональний комп'ютер з такими характеристиками:

- 1) процесор Intel CORE DUO i3;
- 2) оперативна пам'ять об'ємом 2 Гбайт, не менше;
- 3) HDD об'ємом 200 Гбайт, не менше;
- 4) операційна система Windows 7/8;
- 5) веб-переглядач.

3.3.4.4 Вимоги до інформаційної і програмної сумісності

Програма реалізується за допомогою окремого додатка засобами JAVA
Вимоги до програмних засобів, що використовуються програмою
Системні програмні засоби, що використовуються програмою, повинні бути представлені ліцензійною локалізованою версією операційної системи Windows 10.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		36

3.3.5 Техніко-економічні показники

Орієнтовна економічна ефективність не розраховується.

3.3.6 Стадії і етапи розробки

3.3.6.1 Стадії розробки

Розробка повинна бути проведена за дев'ять стадій:

- 1) формування ідеї дипломної роботи (ДР);
- 2) аналіз організаційного оточення, існуючих системи;
- 3) вибір життєвого циклу розробки, обґрунтування вибору мов програмування;
- 4) формування технічного завдання;
- 5) проектування;
- 6) реалізація програмного продукту;
- 7) тестування програмного продукту;
- 8) формування рекомендацій щодо супроводу програмного продукту;
- 9) визначення норм і заходів з охорони праці.

3.3.6.2 Етапи розробки

На стадії розробки технічного завдання має бути виконаний етап розробки, узгодження і затвердження технічного завдання.

На стадії попереднього проектування мають бути виконані такі етапи робіт:

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		37

- 1) вибір ЖЦ, його обґрунтування;
- 2) розробка структурної схеми програми.

На стадії проектування мають бути виконані такі етапи робіт:

- 1) розробка загального інтерфейсу програми;
- 2) розробка модулів програми;
- 3) реалізація об'єднання модулів в одну програму.

На стадії тестування роботи програми виконується випробування програмного продукту на наявність помилок, достовірність роботи всіх підсистем і системи в цілому.

На стадії оформлення дипломної роботи виконується підбиття підсумків і опис результатів усіх виконаних робіт за етапами.

3.3.6.3 Зміст робіт за етапами

На етапі розробки технічного завдання повинні бути виконані такі роботи:

- 1) постановка задачі;
- 2) визначення і уточнення вимог до технічних засобів;
- 3) визначення вимог до програми;
- 4) визначення стадій, етапів і термінів розробки програми;
- 5) узгодження і затвердження технічного завдання.

На етапі попереднього проектування мають бути виконані такі роботи:

- б) розробка структурної схеми програми;

На етапі проектування програми повинна бути виконана робота по кодуванню програми.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		38

На етапі тестування роботи програми повинна бути виконана перевірка достовірності отриманих у програмі результатів.

На етапі розробки програмної документації повинна бути виконана розробка програмних документів у відповідності до вимог до складу документації.

3.3.7 Порядок контролю і приймання

3.3.7.1 Види випробувань

Згідно з вимогами до дипломної роботи в НТУУ «КП».

3.3.7.2 Загальні вимоги до приймання роботи

Дипломна робота має містити всі пункти завдання.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		39

4 ОХОРОНА ПРАЦІ ПРИ НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Вступ

Розділ «Охорона праці» у даній дипломній роботі розглядається, та розробляється на базі кабінету реабілітації для пацієнтів з травмами опорно рухового апарату.

Метою проектного завдання з охорони праці є виявлення потенційно шкідливих та небезпечних факторів, що виникають під час роботи з обладнанням, з урахуванням умов довкілля, котрі можуть негативно вплинути на організм користувача в процесі експлуатації. Розглянуті норми й заходи охорони праці та техніки безпеки, котрі й будуть спрямовані на зменшення, усунення негативних виробничих факторів.

4.2 Характеристика робочого приміщення

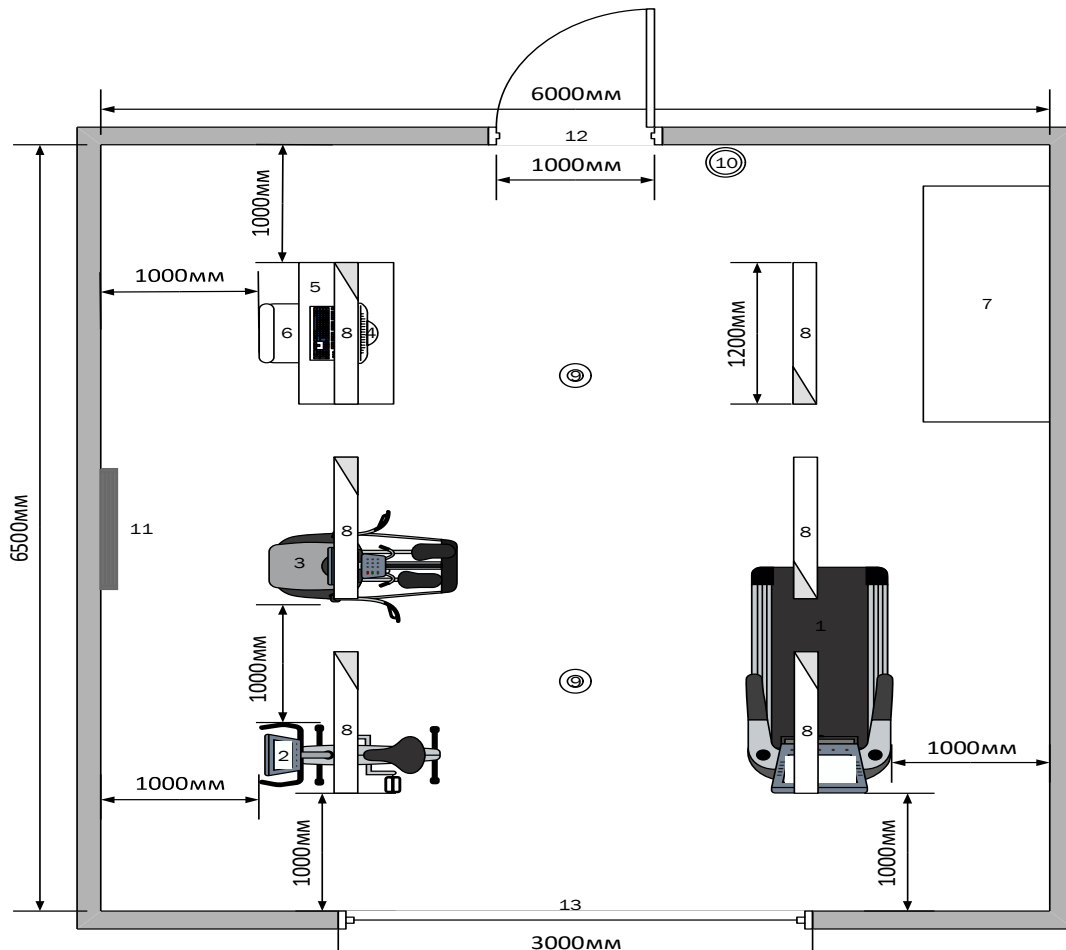
План та розміри приміщення вказані на рисунку 4.1. та таблиці 4.1.

Таблиця 4.1 – Характеристики приміщення

Параметр	Характеристика
Розмір приміщення	6000(мм) × 6500(мм) × 3000(мм)
Кількість працюючих	2 чол.
Площа кімнати	6(м) × 6,5(м) = 39(м ²)
Об'єм кімнати	6(м) × 6,5(м) × 3(м) = 117(м ³)
Природне освітлення	вікно: 3(м) × 1.5(м)
Штучне освітлення	6 люмінесцентних ламп
Опалення та охолодження	1 кондиціонер
Підлога	паркет
Стіни	цегляні, обклеєні шпалерами

Таблиця 4.2 – Характеристики обладнання

Параметр	Характеристика
Бігова доріжка Fitex RT 700	
Габаритні розміри	2,1 (м) × 0.9 (м) × 1.5 (м)
Розмір полотна	1,52 (м) × 56 (м)
Максимальна швидкість	20 (км/год)
Максимальний кут нахилу	15%
Потужність двигуна	4 (к.с.)
Максимальне вага пацієнта	200 (кг)
Велотренажер SportsArt C575U	
Габаритні розміри	1,2 (м) × 0.6 (м) × 1.5 (м)
Вага тренажеру	58,5 (кг)
Кількість рівнів навантаження	20
Максимальне вага пацієнта	227 (кг)
Орбітрек Reebok One Series GX60	
Габаритні розміри	1,45 (м) × 0.65 (м) × 1.7 (м)
Вага тренажеру	53 (кг)
Кількість рівнів навантаження	32
Максимальне вага пацієнта	180 (кг)
Персональний комп'ютер Lenovo ThinkCentre Edge 62z AIO (моноблок)	
Процесор	Intel Core i3-3240 (3.4 ГГц)
Відеокарта	Intel HD Graphics
Оперативна пам'ять	4 (ГБ)
Об'єм HDD	500 (ГБ)
Розмір екрану	18.5"
Роздільна здатність	1366x768
Браслет для вимірювання показників Jawbone Up3	
Габаритні розміри	210 (мм) x 12,2 (мм) x 3,0–9,3 (мм)
Вага	29(г)
Матеріал ремінця	Гіпоалергічний термополіуритан медичного класу
Матеріал корпусу та датчиків	Алюміній
Датчики	Крокомір, пульсометр, дихання, шкірно-гальванічної реакції, температури тіла
Максимальне розмір	140-190(мм)
Джерело живлення	Li-Ion акумуляторна батарея, 38(мАч)



1 - бігова доріжка, 2 - велотренажер, 3 - орбітрек, 4 - комп'ютер,
 5 - робочий стіл, 6 - крісло, 7 - кушетка, 8 - світильник з люмінесцентними лампами типу
 ЛБ, 9 - датчик теплової пожежної сигналізації, 10 - вогнегасник, 11 - кондиціонер,
 12 - двері, 13 - вікно

Рисунок 4.1 – План приміщення.

Відповідно до стандарту ДСанПІН 3.3.2.007-98 показники площі та об'єму, що припадають на одного співробітника знаходяться в межах норми. На основі стандарту, ширина дверей на 100 мм перевищує мінімальне значення. Ширина проходу, також знаходиться в межах норми.

У робочому приміщенні також не розміщено трубопроводів, дренажної системи, що могло б спричинити до протікання води та поганих наслідків для збереження даних та роботу приладів.

Змін.	Лист	№ докум.	Підпис	Дата

БС42с.4208.1300.3166.ПЗ

Лист

42

4.3 Аналіз потенційних небезпек, що створюються в процесі експлуатації комплексу та розробка заходів по покращенню умов праці

Потенційним шкідливим фактором при експлуатації обладнання персонального комп'ютера вважається небезпека враження людини електричним струмом. Можлива й пожежна небезпека в аварійній ситуації, але вона менш ймовірна.

Таблиця 4.3 – Небезпечні та шкідливі виробничі чинники

Фізичні	Хімічні	Біологічні	Психофізичні
- механічні чинники; мікроклімат; освітлення; шум; вібрація; електробезпека; пожежна безпека	- пил	- відсутні	- відсутні

4.3.1 Механічні чинники

Таблиця 4.4 – Джерела і наслідки

Джерела	Наслідки
Бігова доріжка	
Електричний двигун	Потрапляння одягу і як наслідок отримання травм
Вал	
Проміжок між біговм полотном і корпусом	
Бігове полотно	Падіння
Швидкість руху бігового полотна	
Кут нахилу	
Велотренажер	
Маховик	Потрапляння одягу і як наслідок отримання травм
Ремінна передача	
Педалі	Травми ступні та гомілки
Орбітрек	
Маховик	Потрапляння одягу і як наслідок отримання травм
Педалі	

В зв'язку з тим, що в даному кабінеті знімаються показники з людей які проходять реабілітацію після травм опорно рухового апарату особливих навантажень не передбачено.

Таблиця 4.5 – Заходи для попередження травмувань

Вид захисту		Заходи	
Технічні заходи	На біговій доріжці	Електричний двигун	Встановлені захисні кожухи з мінімальним зазором
		Вал	
		Проміжок між полотном і корпусом	
		Бігове полотно	Гумове покриття, поперечні лінії для сигналізації руху полотна.
		Швидкість руху бігового полотна	Встановлена максимальна безпечна швидкість 20 км/год
		Кут нахилу	Встановлено обмеження на кут нахилу 15°
	На велотренажері	Маховик	Встановлені захисні кожухи з мінімальним зазором
		Ремінна передача	
		Педалі	Гумове покриття та фіксатори з ремінцями
	На орбітреку	Маховик	Встановлено захисний кожух з мінімальним зазором
Педалі		Гумове покриття та фіксатори з ремінцями	
Організаційні заходи		Інструктаж перед початком занять	
ЗІЗ		Не передбачені	

4.3.2 Мікроклімат

Робота, яка виконується в робочому приміщенні відноситься до категорії 1б: виконуються сидячи, стоячи та пов'язані з ходьбою і супроводжуються деяким фізичним напруженням. Енергозатрати при виконанні робіт становлять 121-150 ккал/год.

Таблиця 4.6 – Джерела і наслідки

Джерела	Наслідки
Температура повітря	
Спортивні тренажери та комп'ютер	Підвищена температура знижує працездатність людини
Вологість	
Погана вентиляція	Можливе накопичення вологи в обладнанні
Швидкість переміщення повітря	
Відкриті двері та вікно	Не має

Таблиця 4.7 – Порівняння реальних і нормативних значень

	Період року	Температура повітря, t, °С	Відносна вологість, %	Швидкість переміщення повітря, м/с
Нормативні значення	Холодний та теплий	15 – 32	20-80	не більше 0,2
Реальні значення	Холодний	18 – 22	40 – 60	0,1
	Теплий	22 – 25	40 – 60	0,2

Таблиця 4.8 – Заходи для нормалізації параметрів мікроклімату

Вид захисту		Заходи
Технічні заходи	У технологічному обладнанні	Вільна циркуляція повітря в захисних кожухах.
	У приміщенні	У холодну пору року: Опалення (Mitsubishi Electric MS-GF25VA). У теплу пору року: Кондиціонер типу спліт-системи, марки Mitsubishi Electric MS-GF25VA (режим осушення/охолодження/обігрів; P = 2,5 (кВт)). Система штучної вентиляції марки Mitsubishi LGH-50RX4-E (повітрообмін 500 м ³ /год)
Організаційні заходи		Вологе прибирання, кондиціювання, зволоження або осушення повітря
ЗІЗ		Не передбачені

4.3.3 Освітлення

Таблиця 4.9 – Джерела і наслідки

Джерела	Наслідки
Недостатнє освітлення	Погіршення зору працівника
Надмірне освітлення	Зниження рівня працездатності

Під час роботи зображення спостерігається на моніторах комп'ютера. Характеристика зорової роботи наведена у таблиці 10.

Таблиця 4.10 – Характеристика зорової роботи

Характеристика зорової роботи	Найменший розмір об'єкту розрізнення, (мм)	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фону	Шрифт	Система загального освітлення
Середньої точності	Понад 0,5 до 1	IV	в	Малий Середній Великий	Світлий Світлий Середній	1 (мм) 2-3 (мм) 4-5 (мм)	200 лк

Таблиця 4.11 – Порівняння реальних і нормативних значень

	Реальні значення	Нормативні значення
Освітленість, лк	271	200

Основні характеристики освітлення приміщення відповідають встановленим нормам ДБН В.2.5-28-2006 та ДСанПіН 3.3.2.007-98.

Таблиця 4.12 – Заходи для нормалізації параметрів освітлення

Вид захисту	Заходи
Технічні заходи	Природне освітлення: бокове (однобічне), 1-не вікно 3(м)х1,5(м) Штучне освітлення – комбіноване (6 світильників ЛПО-01, 12 ламп Р = 40 (Вт), висота 3 (м))
Організаційні заходи	Вологе прибирання, підтримання чистоти вікон та світильників
ЗІЗ	Не передбачені

4.3.4 Шум

Таблиця 4.13 – Джерела та наслідки

Джерела шуму	Наслідки
Бігова доріжка	Втомленість та роздратованість. Зниження рівня працездатності. Порушення слуху.
Велотренажер	
Орбітрек	
Моноблок	
Кондиціонер	
Система штучної вентиляції	
Зовнішній шум	

Таблиця 4.14 – Порівняння реальних і нормативних значень

Джерела шуму	Рівень шуму, дБА	Час впливу, год	Реальне значення, дБА	Нормативне значення, дБА
Бігова доріжка	24	2	31,6	60
Велотренажер	21	2		
Орбітрек	10	2		
Моноблок	10	8		
Кондиціонер	25	8		
Система штучної вентиляції	23	8		
Зовнішній шум	10	8		

Середній рівень шуму у не перевищує встановлені норми за ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку.

Таблиця 4.15 – Заходи і засоби захисту від шуму

Вид захисту		Заходи
Технічні заходи	У технологічному обладнанні	Джерела шуму знаходяться в захисному кожусі.
	У приміщенні	Для захисту від зовнішнього шуму встановлена шумоізоляція.
Організаційні заходи		Режим праці і відпочинку, дотримуються правила технічної експлуатації, проведення планово-попереджувальних оглядів та ремонтів
ЗІЗ		Не передбачені

4.3.5 Вібрація

Таблиця 4.16 – Джерела та наслідки

Джерела вібрації	Наслідки
Електродвигун бігової доріжки	Головний біль, запаморочення, нудота. Зниження рівня працездатності. Зміни у центральній нервовій системі, вестибулярному апараті, серцево-судинній системі.
Маховик велотренажера	
Маховик орбітрека	
Електродвигун кондиціонера	

Оскільки безпосереднього контакту з джерелами вібрації не відбувається, то вібрація, яка спостерігається в робочому приміщенні, відноситься до типу загальна вібрація.

Загальна вібрація в робочому приміщенні відноситься до категорії 3: «технологічна вібрація, яка діє на людину на робочих місцях стаціонарних машин чи передається на робочі місця, які не мають джерел вібрації»,

типу в: «на робочих місцях заводоуправлінь, конструкторських бюро, лабораторій, учбових пунктів, обчислювальних центрів, медпунктів, конторських приміщень, робочих кімнат та інших приміщень для працівників розумової праці.»

Таблиця 4.17 – Заходи і засоби захисту від шуму

Вид захисту	Заходи
Технічні заходи	На ніжках тренажерів встановлені гумові віброізолятори.
Організаційні заходи	Своєчасні планово-попереджувальні ремонти, режим праці і відпочинку, дотримуються правила технічної експлуатації.
ЗІЗ	Не передбачені

4.3.6 Електробезпека

Відповідно до ОНТП24-86 та ПУЕ-87 приміщення відноситься до класу приміщень без підвищеної небезпечності ураження персоналу електричним струмом. Використовується мережа однофазного струму 220В.

Таблиця 4.18 – Джерела та наслідки

Джерело	Наслідки
Бігова доріжка	Враження електричним струмом. Опіки.
Моноблок	
Кондиціонер	
Система штучної вентиляції	
Освітлення	

Таблиця 4.19 – Порівняння реальних і нормативних значень

Джерело	Реальне значення	Нормативне значення	Потужність Вт/год
Бігова доріжка	Мережа змінного струму напругою 220±20 В, частотою 50 Гц.	Мережа змінного струму напругою 220±20 В, частотою 50 Гц.	2500
Моноблок			150
Кондиціонер			2500
Система штучної вентиляції			200
Освітлення			480

Мережа відповідає нормі згідно ДСТУ EN 50160:2014.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		48

Таблиця 4.20 – Заходи і засоби захисту від ураження електричним струмом

Вид захисту		Заходи подолання небезпеки
Технічні заходи	норма	Прихована та ізольована проводка. Світильники встановлені на висоті 3м. Електричні блоки безпеки.
	аварійний	Техніка ввімкнена в мережу через заземлені фільтри. В коридорі знаходиться автоматичний вимикач S203-C 6kA (ABB) на 21A з захисними характеристиками С ГОСТ Р50345-99 (МЭК 60898-95).
Організаційні заходи		Всі працюючі ознайомлені з правилами техніки безпеки, своєчасне навчання і перевірка знань працівників з питань електробезпеки
ЗІЗ		Не передбачені

4.3.7 Пожежна безпека при надзвичайних ситуаціях

В приміщенні присутні наступні горючі речовини: папір, дерево, пластмаса, тканина (оббивка крісел).

Таблиця 4.21 – Джерела та наслідки

Джерела	Небезпечний фактор	Наслідки
Електроприлади	Коротке замикання	Пожежа. Опіки.
Горючі матеріали	Займання матеріалів	

Таблиця 4.22 – Параметри пожежної безпеки

Категорія кабінету за вибухонебезпечністю	В (пожежонебезпечна)
Категорія кабінету за пожежною безпекою	П-Па - у приміщенні знаходяться тверді горючі речовини та матеріали.
Пожежний датчик	ДТЛ (ефективною площею захисту 20м ²), 1 шт.

Вимоги СНіП 2.01.02-85 та СНіП 2.09.02-85 щодо вогнетривкості будівлі, часу евакуації у випадку пожежі та ширини евакуаційних проходів дотримуються в повній мірі. Також виконано вимоги з пожежної безпеки відповідно до НАПБ А.0.001-95 «Правила пожежної безпеки в Україні».

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		49

Таблиця 4.23 – Заходи з пожежної безпеки

Вид захисту	Заходи подолання небезпеки
Технічні заходи	Суворе дотримання правил і норм, визначених чинними нормативними документами, при реконструкції приміщень, технічному переоснащенні виробництва, експлуатації електромереж, опалення, освітлення та ін. Вогнегасник.
Організаційні	Організація пожежної охорони, навчань, інструктажів та ін.
ЗІЗ	Не передбачені

Кабінет реабілітації для людей з травмами опорно рухового апарату знаходиться в кабінеті №3 (Рисунок 2).

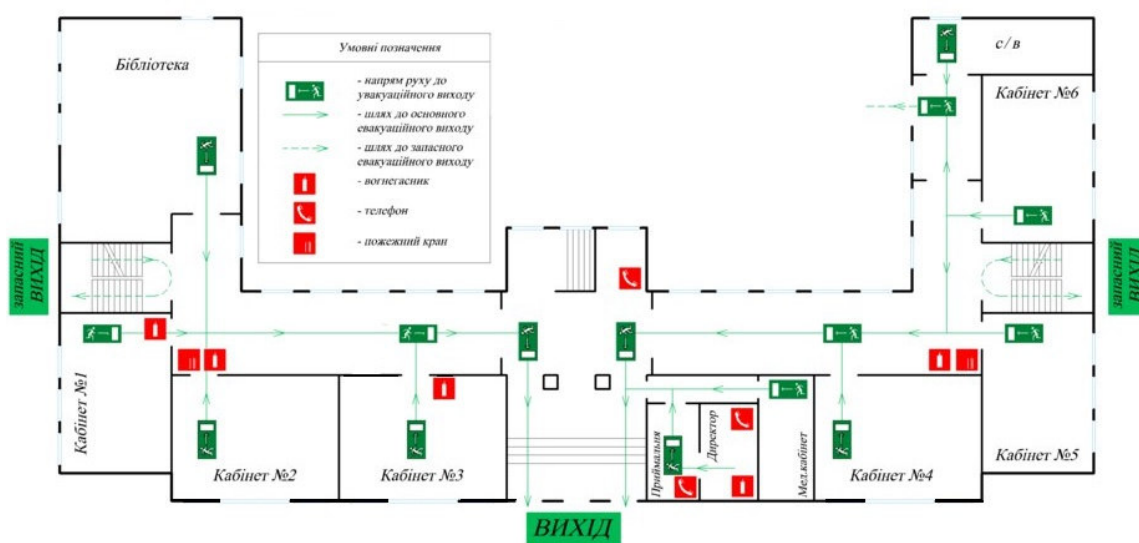


Рисунок 2. План евакуації людей з приміщення

4.4 Висновок

Розглянуте робоче приміщення має належні параметри площі та об'єму. Підтримка мікроклімату відповідає нормам завдяки кондиціонеру та системі штучної вентиляції. Освітлення, рівень шуму та вібрації відповідають нормам.

Завдяки добре організованому розташуванню приладів та їх захисту від перенавантаження, в приміщенні дотримані норми пожежної та електробезпеки.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		50

5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ПРОДУКТУ

5.1 Вступна частина

Задача полягає в розробці програмного продукту, призначеного для аналізу стану пацієнта до та після реабілітації. Програмний продукт та інтерфейс користувача був розроблений за допомогою мови програмування Java використовуючи середовище розробки IntelliJ IDEA. Програмний продукт призначено для використання на персональних комп'ютерах під управлінням будь-якої операційної системи сімейства Windows.

Оскільки результатом дипломної роботи є програмний продукт, що не має аналогів, для аналізу можливих шляхів розробки програмного продукту було обрано один з найбільш ефективних методів, що використовуються для вибору кращих варіантів рішення - функціонально-вартісного аналізу (ФВА).

В даному розділі проводиться аналіз варіантів реалізації модулю з метою вибору оптимальної, з економічної точки зору.

Функціонально-вартісний аналіз — це метод комплексного техніко-економічного дослідження об'єкта з метою розвитку його корисних функцій при оптимальному співвідношенні між їхньою значимістю для споживача і витратами на їхнє здійснення. Є одним з основних методів оцінки вартості науково-дослідної роботи, оскільки ФВА враховує як технічну оцінку продукту, що розробляється, так і економічну частину розробки. Крім того, даний метод дозволяє вибрати оптимальний, як з погляду розробника, так і з точки зору покупця варіант розв'язання будь-якої задачі, а також дозволяє оптимізувати витрати й час виконання робіт.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		51

5.2 Постановка задачі техніко-економічного аналізу

У даному розділі застосовується метод ФВА для проведення техніко-економічного аналізу розробки програмного продукту для аналізу стану пацієнта до та після реабілітації, за допомогою мови програмування Java.

Технічні вимоги до продукту наступні:

- програмний продукт повинен функціонувати на персональних комп'ютерах ОС Windows із стандартним набором компонент;
- забезпечувати зручність та простоту взаємодії з користувачем або з розробником програмного забезпечення у випадку використання його як модуля;
- передбачати мінімальні витрати на впровадження та підтримку програмного продукту.

5.3 Обґрунтування функцій програмного продукту

5.3.1 Формування варіантів функцій

Головна функція F_0 – розробка програмного продукту, який дає найменшу похибку при порівнянні реальних значень цільової змінної та результуючого значення побудованого багаточлену у відповідній точці. Виходячи з конкретної мети, можна виділити наступні основні функції програмного продукту:

F_1 – вибір мови програмування;

F_2 – кількість вхідних матеріалів(даних);

F_3 – представлення вихідних даних;

F_4 – функціонал;

F_5 – графічний інтерфейс користувача;

Кожна з основних функцій може мати декілька варіантів реалізації.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		52

Функція F1:

- а) мова програмування C#;
- б) мова програмування Python;
- в) мова програмування Java.

Функція F2:

- а) мінімум вхідних даних;
- б) максимум вхідних даних;

Функція F3:

- а) збереження інформації у базу даних;
- б) вивід інформації на екран.

Функція F4:

- а) мінімальний функціонал (тільки розрахункова частина);
- б) ілюстративний функціонал.

Функція F5:

- а) інтерфейс користувача, створений на базі WinForms;
- б) інтерфейс користувача, створений на базі JavaFX.

5.3.2 Варіанти реалізації основних функцій

Варіанти реалізації основних функцій наведено у морфологічній карті системи (Рисунок 5.1). На основі цієї карти побудовано позитивно-негативну матрицю варіантів основних функцій (таблиця 5.1)

Морфологічна карта відображає всі можливі комбінації варіантів реалізації функцій, які складають повну множину варіантів програмного продукту.

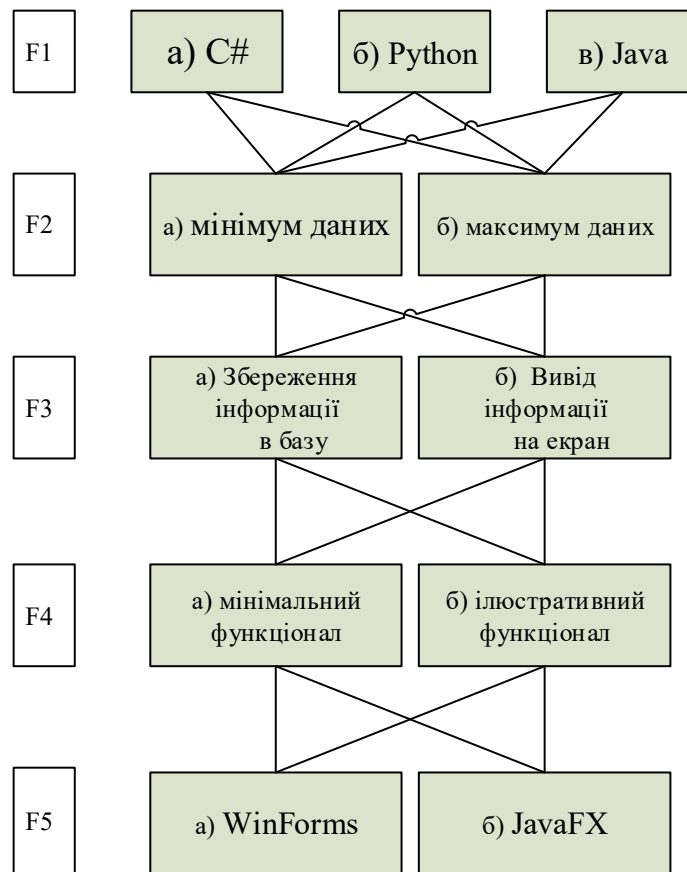


Рисунок 5.1 Морфологічна карта.

Таблиця 5.1. Позитивно-негативна матриця

Основні функції	Варіанти реалізації	Переваги	Недоліки
F1	а	Займає менше часу при написанні коду	Не кросплатформний
	б	Код швидко виконується, кросплатформний	Займає більше часу при написанні коду
	в	Займає менше часу при написанні коду, код швидко виконується	Кросплатформний
F2	а	Займає менше часу при написанні коду	Підійде тільки для спеціалістів в даній галузі
	б	Можливість використання без глибоких знань у галузі	Більш трудомісткий
F3	а	Можливість послідуочого редагування та друку в будь-якому редакторі	Неспівпадіння кодування українських символів в DOS та Windows

Продовження таблиці 5.1

Основні функції	Варіанти реалізації	Переваги	Недоліки
	б	Отримання твердої копії результатів	Необхідність наявності принтеру на комп'ютері
F4	а	Простота реалізації	Складність у сприйнятті інформації
	б	Простота у сприйнятті інформації	Складність реалізації
F5	а	Легкий у створенні	Відсутність кросплатформеності
	б	Стабільний у використанні, проста інтеграція з Python	Складний у створенні

На основі аналізу позитивно-негативної матриці робимо висновок, що при розробці програмного продукту деякі варіанти реалізації функцій варто відкинути, тому що вони не відповідають поставленим перед програмним продуктам задачам.

Функція F1: оскільки розрахунки проводяться з великими об'ємами вхідних даних, то час написання та виконання програмного коду є найважливішими факторами, отже варіант а) і б) потрібно відкинути.

Функція F2: оскільки програма повинна бути призначена для широкого класу користувачів, то потрібно відкинути варіант а).

Функція F3: оскільки користувачеві потрібна інформація в електронному вигляді, то необхідно відкинути варіант б).

Функція F5: інтерфейс користувача відіграє велику роль у даному програмному продукту, тому варіант а) було відкинуто.

Таким чином будемо використовувати наступні варіанти реалізації програмного продукту:

1. F1в – F2б – F3а – F4а – F5б;
2. F1в – F2б – F3а – F4б – F5б;

5.4 Обґрунтування системи параметрів програмного продукту

5.4.1 Опис параметрів

Маючи вимоги щодо основних функцій, які реалізуються в програмному продукті, визначають основні параметри виробу, які надалі використовуватимуться для розрахунку коефіцієнта технічного (Таблиця 4.4.1.1).

Таблиця 5.2 Основні параметри виробу

Параметр	Функція	Опис
X1	швидкодія мови програмування	відображає легкість розуміння та супроводу коду програмного продукту
X2	рівень складності алгоритму	відображає легкість розуміння та супроводу коду програмного продукту
X3	швидкість обробки та виводу даних	відображає час, який витрачається на розрахунки
X4	простота реалізації інтерфейсу	відображає на скільки легко користуватися програмним продуктом.

5.4.2 Кількісна оцінка параметрів

Будемо розглядати 3 типи варіантів значення параметрів.

Таблиця 5.3 Основні параметри програмного продукту

Назва параметра	Умовні позначення	Одиниці виміру	Значення параметра		
			гірші	середні	кращі
Швидкодія мови програмування	X1	Кб/с	100	300	550
Рівень складності алгоритму	X2	%	50	57	60
Швидкість обробки	X3	мс	8000	500	60
Простота реалізації інтерфейсу	X4	%	5	80	100

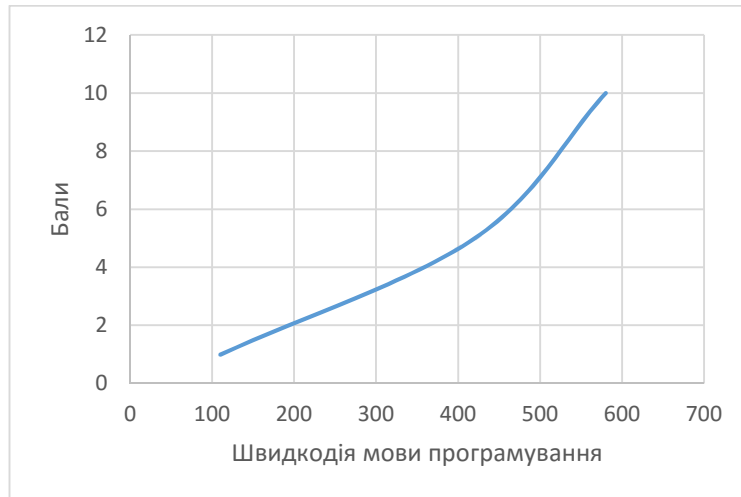


Рисунок 5.2 Бальна оцінка швидкодії мови програмування

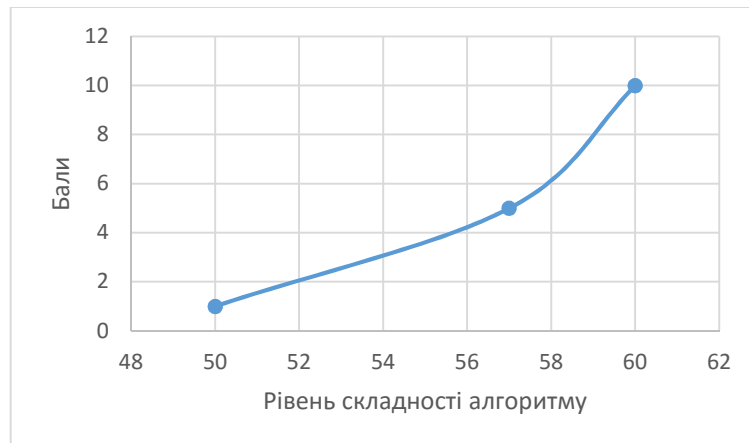


Рисунок 5.3 Бальна оцінка складності алгоритму програмування

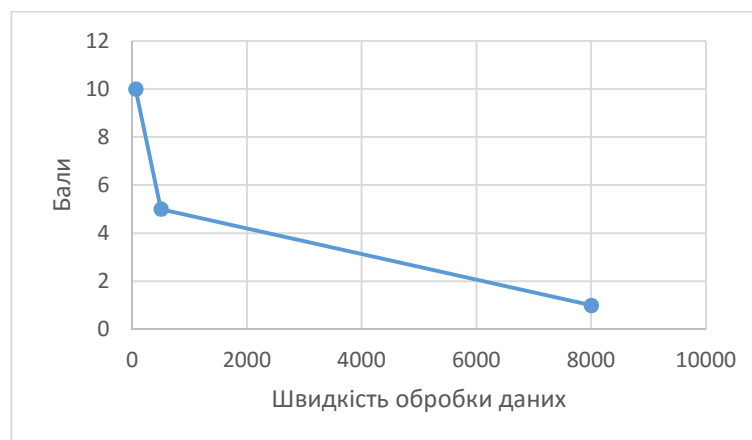


Рисунок 5.4 Бальна оцінка швидкості обробки даних

Змін.	Лист	№ докум.	Підпис	Дата

БС42с.4208.1300.3166.ПЗ

Лист

57

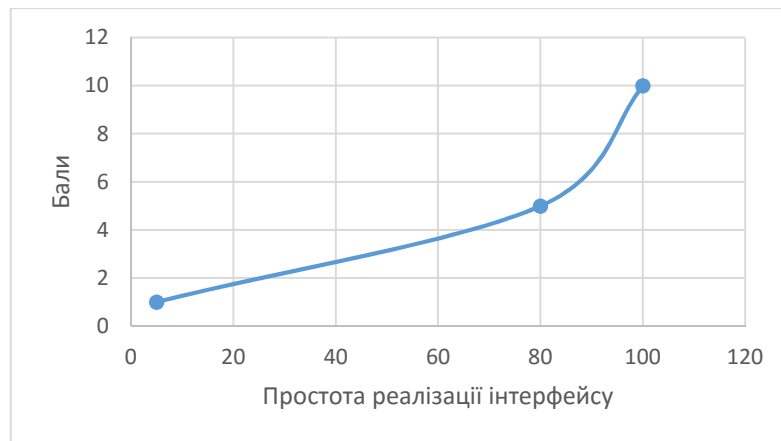


Рисунок 5.5 Бальна оцінка простоти реалізації інтерфейсу

Аналіз експертного оцінювання параметрів

Кожен експерт оцінює ступінь важливості кожного параметру для конкретно поставленої цілі – розробка програмного продукту, який дає найбільш точні результати при знаходженні апроксимуючого багаточлену

Таблиця 5.4 Результати ранжування параметрів

Параметри	Експерти							Підсумкова оцінка	Числове значення
	1	2	3	4	5	6	7		
X1,X2	<	<	>	<	<	<	<	<	0,5
X1,X3	<	>	>	<	>	<	>	>	1,5
X1,X4	<	=	>	<	=	<	<	<	0,5
X2,X3	>	>	>	>	>	>	>	>	1,5
X2,X4	>	>	>	>	>	=	>	>	1,5
X3,X4	>	<	>	<	<	<	>	<	0,5

Розрахуємо коефіцієнт конкордації (узгодженості) експертних оцінок.

Таблиця 5.5 Результати ранжування показників

Познач. параметра	Ранг параметра за оцінкою експерта							Сума рангів R_i	Відхилення Δ_i	Δ_i^2
	1	2	3	4	5	6	7			
X1	4	3	4	4	3	3	3	24	6,5	42,25
X2	1	1	2	1	2	1	2	10	-7,5	56,25
X3	2	2	1	2	1	2	1	11	-6,5	42,25
X4	3	4	3	3	4	4	4	25	7,5	56,25
Разом	10	10	10	10	10	10	10	70	0	197

Порахуємо коефіцієнт узгодженості:

Для перевірки степені достовірності експертних оцінок, визначимо наступні параметри:

а) сума рангів кожного з параметрів і загальна сума рангів:

$$R_i = \sum_j^N r_{ij} R_{ij} = \frac{Nn(n+1)}{2} = 70$$

де N – кількість експертів, n – кількість параметрів;

б) середня сума рангів:

$$T = \frac{1}{n} R_{ij} = 17,5$$

в) відхилення суми рангів кожного параметра від середньої суми рангів:

$$\Delta_i = R_i - T$$

Сума відхилень по всім параметрам повинна дорівнювати 0;

г) загальна сума квадратів відхилення:

$$S = \sum_{i=1}^n \Delta_i^2 = 197$$

$$W = \frac{12S}{N^2(n^3 - n)}, W = \frac{12 * 197}{(7^2 * (4^3 - 4))} = 0,80 > W_k = 0,67$$

Ранжирування можна вважати достовірним, тому що знайдений коефіцієнт узгодженості перевищує нормативний, котрий дорівнює 0,67.

Скориставшись результатами ранжирування, проведемо попарне порівняння всіх параметрів і результати занесемо у таблицю.

Числове значення, що визначає ступінь переваги i -го параметра над j -тим, a_{ij} визначається по формулі:

$$a_{ij} = \begin{cases} 1.5, & X_i > X_j \\ 1, & X_i = X_j \\ 0.5, & X_i < X_j \end{cases}$$

З отриманих числових оцінок переваги складемо матрицю $A = \| a_{ij} \|$.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		59

Для кожного параметра зробимо розрахунок вагомості K_{Bi} за наступними формулами:

$$\hat{E}_{\hat{a}^3} = \frac{b_i}{\sum_{i=1}^n b_i}, \text{ де } b_i = \sum_{i=1}^n a_{ij}$$

Відносні оцінки розраховуються декілька разів доти, поки наступні значення не будуть незначно відрізнятись від попередніх (менше 2%). На другому і наступних кроках відносні оцінки розраховуються за наступними формулами:

$$\hat{E}_{\hat{a}^3} = \frac{b'_i}{\sum_{i=1}^n b'_i}, \text{ де } b'_i = \sum_{j=1}^n a_{ij} b_j$$

Таблиця 5.6 Розрахунок вагомості параметрів

Параметри	Параметри				Перший крок		Другий крок	
	X1	X2	X3	X4	b_i	K_{Bi}	b_i^1	$\hat{E}_{\hat{a}^3}^1$
X1	1,0	0,5	1,5	1,5	4,5	0,281	16,25	0,28
X2	1,5	1,0	1,5	1,5	5,5	0,343	21,25	0,36
X3	0,5	0,5	1,0	0,5	2,5	0,156	9,25	0,16
X4	0,5	0,5	1,5	1,0	3,5	0,218	12,25	0,20
Всього:					16	1	59	1

Як видно з таблиці, різниця значень коефіцієнтів вагомості не перевищує 2%, тому більшої кількості ітерацій не потрібно.

5.5 Аналіз рівня якості варіантів реалізації функцій

Визначаємо рівень якості кожного варіанту виконання основних функцій окремо.

Абсолютні значення параметрів X1 (складність алгоритму) та X4 (інтуїтивність користування) відповідають технічним вимогам умов функціонування даного ПП.

Абсолютне значення параметра X3 (швидкість обробки та виводу даних) обрано не найгіршим (не максимальним), тобто це значення відповідає або варіанту а) 8000 мс або варіанту б) 500мс.

Абсолютні значення параметра X2 (об'єм пам'яті для збереження даних в базу даних) обрані з міркувань, що об'єм пам'ті буде усередненим, тобто це значення буде в проміжку від 6000 до 500.

Коефіцієнт технічного рівня для кожного варіанта реалізації ПП розраховується за формулою:

$$K_{TP} = \sum_i K_{ei} \cdot B_i, i = 1 \dots n,$$

де n – кількість параметрів, K_{ei} – коефіцієнт вагомості i -го параметра, B_i – оцінка i -го параметра в балах.

Таблиця 5.7 Розрахунок показників рівня якості варіантів реалізації

Основні функції	Варіант реалізації функції	Абсолютне значення параметра	Бальна оцінка параметра	Коефіцієнт вагомості параметра	Коефіцієнт рівня якості
F1	В	10	3,6	0,28	1,008
F2	Б	16	2,4	0,36	0,864
F3	А	8000	4,4	0,16	0,704
F4	А	80	4.2	0,36	1,512
	Б	5	2	0,2	0,4
F5	Б	320	5	0,2	1

За цими даними визначаємо рівень якості кожного з варіантів:

$$K_{TEP1} = 1,008 + 0,864 + 0,704 + 1,512 + 0,4 = 4,488$$

$$K_{TEP2} = 1,008 + 0,864 + 0,704 + 0,4 + 0,4 = 3,376$$

Отже, найкращим є перший варіант, для якого коефіцієнт технічного рівня має найбільше значення.

5.6 Економічний аналіз варіантів розробки програмного продукту

5.6.1 Визначення трудомісткості

Два варіанти включають в себе по 2 завдання. Причому, для кожного варіанта одне завдання буде спільним.

Перше завдання:

- 1) складність алгоритму 1;
- 2) ступінь новизни задачі Б;

$K_{ст.п} = 1.2$ (відповідно до пунктів 1 та 2).

- 3) складність організації контролю вхідної і вихідної інформації 2:

$K_{ск} = 1$;

- 4) Мова програмування – Java високого рівня: $K_M = 1$;

- 5) стандартні модулі не використовуються $K_{ст} = 1$;

- 6) трудомісткість $T_p = 64$ людино-днів

Об'єм кожної з функцій визначається відповідно “Каталогу функцій ПС ВТ”.

$$T_o = T_p * K_{п} * K_{ск} * K_M * K_{ст} * K_{ст.п} ;$$

$$K_{п} = \frac{K_1 * m + K_2 * n + K_3 * p}{m + n + p}$$

де K_1, K_2, K_3 , - поправкові коефіцієнти;

m, n, p - кількість наборів даних ЗІ, ПДІ, БД відповідно.

В нашому випадку $K_1 = 1,62$; (ПІ група складності 1):

$$K_{п} = \frac{K_1 * m}{m} = 1,62$$

Тоді загальна трудомісткість:

$$T_o = 64 * 1,62 * 1 * 1 * 1 * 1,2 = 124,45 \text{ людино-днів}$$

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		62

Друге завдання (1-й варіант):

1) складність алгоритму 1;

2) ступінь новизни задачі В;

$K_{СТ.П} = 1.2$ (відповідно до пунктів 1 та 2).

3) складність організації контролю вхідної і вихідної інформації 2:

$K_{СК} = 1$;

4) Мова програмування – Java високого рівня: $K_M = 1$;

5) стандартні модулі використовуються $K_{СТ} = 0,7$;

6) трудомісткість $T_p = 30$ людино-днів;

$K_{П} = 1,2$

$T_o = 30 * 1,2 * 1 * 1 * 0,7 * 1,2 = 30,24$ людино-днів

Друге завдання (2-й варіант):

1) складність алгоритму 1;

2) ступінь новизни задачі Б;

$K_{СТ.П} = 1.2$ (відповідно до пунктів 1 та 2).

3) складність організації контролю вхідної і вихідної інформації 2:

$K_{СК} = 1$;

4) Мова програмування – Java високого рівня: $K_M = 1$;

5) стандартні модулі використовуються $K_{СТ} = 0,7$;

6) трудомісткість $T_p = 30$ людино-днів;

$K_{П} = 1,62$

$T_o = 30 * 1,62 * 1 * 1 * 0,7 * 1,2 = 40,82$ людино-днів

Складемо трудомісткість відповідних завдань, щоб отримати їх трудомісткість:

$T_I = (124,45 + 30,24) * 8 = 1237,52$ людино-годин

$T_{II} = (124,45 + 40,82) * 8 = 1322,16$ людино-годин

Очевидно, що вищу трудомісткість має варіанти II.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		63

5.6.2 Розрахунок заробітної плати розробників.

Витрати на оплату праці розробників ПП:

$$C_3 = C_r \cdot T \cdot (1 + H_d / 100),$$

де C_r – розмір погодинної оплати праці програміста, грн.;

T – трудомісткість розробки ПП, людино-година;

H_d – норматив, який враховує додаткову заробітну плату, $H_d = 20\%$.

У розробці бере участь один програміст з окладом 10000 грн. Визначимо зарплату програміста за годину при 21 робочому дні на місяць:

$$C_r = \frac{10000}{21 \cdot 8} = 59,52 \text{ грн.}$$

Тоді зарплата розробника за варіантами відповідно рівні:

$$I : C_3 = 59,52 \cdot 1237,52 \cdot (1 + 0,2) = 88388,63 \text{ грн.}$$

$$II : C_3 = 59,52 \cdot 1322,16 \cdot (1 + 0,2) = 94433,96 \text{ грн.}$$

Відрахування на єдиний соціальний внесок (22%):

$$I : C_{\text{від}} = 88388,63 \cdot 0,22 = 19445,5 \text{ грн.}$$

$$II : C_{\text{від}} = 94433,96 \cdot 0,22 = 20775,47 \text{ грн.}$$

5.6.3 Розрахунок собівартості ПП

Для обслуговування ПК залучений спеціаліст із ЗП 4500 грн./міс.

Коефіцієнт його зайнятості при обслуговуванні – 0,2.

Вартість комп'ютера – 15000 грн.

Для однієї машини отримаємо:

$$C_r = 1 \cdot 4500 \cdot 0,2 = 900 \text{ грн.}$$

З урахуванням додаткової заробітної плати

$$C_{\text{об}} = 900 \cdot (1 + 0,2) = 1080 \text{ грн.}$$

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		64

Відрахування на єдиний соціальний внесок (22% від фонду оплати праці):

$$C_{\text{від}} = C_{\text{зп}} * 0.22 = 1080 * 0,22 = 237,6 \text{ грн.}$$

Амортизаційні відрахування обчислимо за формулою:

$$C_a = 0,01 * N_a * K_{\text{тм}} * C_d,$$

де C_d – договірна ціна приладу, грн;

$K_{\text{тм}}$ – коефіцієнт, який враховує витрати на транспортування та монтаж приладу у користувача;

N_a – річна норма амортизації, %.

За амортизації 25% та вартості ЕОМ 15000 грн. маємо:

$$C_a = 1,15 * 0,25 * 15000 = 4312,5 \text{ грн.}$$

Витрати на ремонт та профілактику розрахуємо за формулою:

$$C_{\text{рем}} = (K_p / 100\%) C_d * K_{\text{тм}},$$

де $K_{\text{рем}}$ – відсоток витрат на поточні ремонти. У розрахунках значення цього нормативу орієнтовно візьмемо 5%.

$$C_{\text{рем}} = 1,15 * 15000 * 0,05 = 862,5 \text{ грн.}$$

Ефективний годинний фонд часу ПК за рік розрахуємо за формулою:

$$T_p = (D_k - D_v - D_c - D_p) * t_z * n_z * K_v,$$

$$T_{\text{еф}} = (365 - 10 - 52 * 2) * 8 * 0,9 = 1706,8 \text{ год.}$$

Витрати на оплату електроенергії знаходимо за формулою:

$$C_{\text{ел}} = C_{\text{ен}} * P * T_{\text{еф}},$$

де P – потужність,

$P = 0,5$ кВт.

$$C_{\text{ел}} = 0,465 * 0,5 * 1706,8 = 391,83 \text{ грн.}$$

Накладні витрати в розмірі 67 % від витрат на оплату праці:

$$C_{\text{накл}} = 960 * 0,67 = 643,2 \text{ грн.}$$

Тоді, річні експлуатаційні витрати складуться у формулу:

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		65

$$C_{\text{екс}} = (1080 + 237,6) * 12 + 4312,5 + 862,5 + 391,83 + 643,2 = 22014,03 \text{ грн}$$

Собівартість однієї машино-години ЕОМ дорівнюватиме:

$$C_{\text{мг}} = C_{\text{екс}} / T_{\text{еф}} = 22014,03 / 1706,8 = 12,9 \text{ грн/час.}$$

Оскільки в цьому випадку всі роботи, пов'язані з розробкою програмного продукту, ведуться на ЕОМ, витрати на оплату машинного часу для кожного обраного варіанта реалізації пакета, становитимуть:

$$I : C_{\text{мч}} = T * C_{\text{мг}} = 1237,52 * 12,9 = 15964,01 \text{ грн.}$$

$$II : C_{\text{мч}} = T * C_{\text{мг}} = 1322,16 * 12,9 = 17055,86 \text{ грн.}$$

Накладні витрати становлять 67 % від заробітної плати:

$$I : C_{\text{накл}} = C_3 * 0,67 = 88388,63 * 0,67 = 59220,38 \text{ грн.}$$

$$II : C_{\text{накл}} = C_3 * 0,67 = 94433,96 * 0,67 = 63270,75 \text{ грн.}$$

Функціонально необхідні затрати на створення ПП обчислюються за формулою:

$$C_{\text{пп}} = C_3 + C_{\text{від}} + C_{\text{мч}} + C_{\text{накл}}$$

Визначимо вартість розробки програмного продукту для кожного варіанта:

$$I : C_{\text{пп}} = 88388,63 + 19445,5 + 15964,01 + 59220,38 = 183018,52 \text{ грн.}$$

$$II : C_{\text{пп}} = 94433,96 + 20775,47 + 17055,86 + 63270,75 = 195536,04 \text{ грн.}$$

5.6.4 Розрахунок показників економічної ефективності

Показник економічної ефективності розраховується за формулою:

$$K_{\text{тер}j} = K_{\text{тр}j} / C_{\text{пп}},$$

де $K_{\text{тр}j}$ – коефіцієнт технічного рівня j-го варіанта. Розрахуємо для кожного варіанта реалізацію функцій ПП коефіцієнт техніко-економічного рівня:

$$K_{\text{тер}1} = K_{\text{тр}1} / C_{\text{пп}} = 4,488 / 183018,52 = 2,45\text{E-}04;$$

$$K_{\text{тер}2} = K_{\text{тр}2} / C_{\text{пп}} = 3,376 / 195536,04 = 1,73\text{E-}05;$$

Таким чином, найбільш ефективним є другий варіант реалізації функцій ПП, який має максимальну величину коефіцієнта техніко-економічного рівня.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		66

5.7 Висновки

В результаті виконання економічного розділу були систематизовані і закріплені теоретичні знання в галузі економіки та організації виробництва використанням їх для техніко-економічного обґрунтування розробки методом функціонально-вартісного аналізу.

В результаті виконання аналізу були виділені основні функції та розглянуто два варіанти реалізації програмного продукту:

I : F1в – F2б – F3а – F4а – F5б;

II: F1в – F2б – F3а – F4б – F5б;

Згідно результатів розрахунків, найбільш економічно вигідним варіантом реалізації є перший варіант. Оскільки розрахований критерій ефективності першого варіанту($K_{тер1} = 2,45E-04$) більший за критерій ефективності другого варіанту($K_{тер2} = 1,73E-05$).

Функціональна ціна розробки програмного продукту, за найбільш вигідним варіантом реалізації, становить 183018,52 грн, у тому числі витрати на заробітну плату програмістам - 88388,63 грн. Накладні витрати становлять 59220,38 грн, а витрати на оплату машинного часу 15964,01 грн.

Програмний продукт розроблено за допомогою мови високого рівня JAVA. Завдяки тому, що розроблений програмний продукт ще не має аналогів, а також достатньо універсальний і легкий в модифікації, він являється рентабельним та економічно обґрунтованим. Саме тому даний проект вартий уваги та фінансування.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		67

ЗАГАЛЬНІ ВИСНОВКИ

У результаті виконання дипломної роботи, було розроблено та створено програмне забезпечення системи аналізу фізичного стану пацієнта до та після реабілітації. Ця робота призначена для того, щоб надати можливість вести електронну документацію лікарю реабілітологу, а також спілкуватись пацієнту з лікарем на відстані за допомогою мережі Інтернет.

Робота лікаря в сучасних умовах нерозривно пов'язана з використанням інформаційних технологій, які дозволяють проводити збір інформації про пацієнтів в їх електронних картках, проходити дистанційне підвищення кваліфікації, отримувати нову інформацію про методи лікування і т.п..

Створена система дозволяє зменшити навантаження на лікаря, так як пацієнт може самостійно заносити результати тренування в свою особисту картку. Дана дипломна робота відображає процес створення системи аналізу та надає змогу вдосконалити у майбутньому функції системи.

Створене програмне забезпечення може застосовуватись як лікарями початківцями так і кваліфікованими робітниками поліклінік та стаціонарів.

Розвиток даної системи можна вбачати в допомозі лікарю працювати із пацієнтами за допомогою віртуальних технологій. Коли він бачить знайомі симптоми, описані на форумі, він може припустити діагноз, та обов'язковою рекомендацією буде – прийти на обстеження.

При виконанні роботи, результати окремих етапів були опубліковані у вигляді статті на міжнародних конференціях

- «Научные исследования и их практическое применение. Современное состояние и пути развития '2015», тема доповіді: «Клиометрия. Анализ скоринг-систем при реабилитации опорно-двигательного аппарата »;

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		68

- «Inżynieria i technologia. Priorytetowe obszary nauki», тема доповіді:
«Клиометрия. Анализ скоринг-систем при реабилитации опорно-
двигательного аппарата ».

Робота виконана на замовлення ТОВ «Артіфішіал Кор» та складено акт
впровадження.

Пояснювальна записка до дипломної роботи оформлена відповідно до
Положення про державну атестацію студентів НТУУ «КПІ» [27].

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		69

ПЕРЕЛІК ПОСИЛАНЬ

1. Опорно-руховий апарат. Кістки, система скелета. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: http://y-ra.com/book_anatomya-lyudini-1-tom_679/17_oporno-ruhovij-aparat.-kstki-sistema-skeleta.
2. Будова і функції опорно-рухового апарату. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://disted.edu.vn.ua/courses/learn/2587>.
3. Опорно-руховий апарат людини.. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://osvita.ua/vnz/reports/biolog/23053/>.
4. Очкуренко О.М., Федотов О.В. Анатомія людини: Навч. посібник. – 2-ге вид., – К.: Вища шк., 1992. – 290 с.
5. Свиридов О.І. «Анатомія людини» – Київ, Вища школа, 2001.. – 90 с. – 314с.
6. Біологія: Навч. посіб. / А. О. Слюсарєв, О. В. Самсонов, В.М.Мухін та ін.; За ред. та пер. з рос. В. О. Мотузного. — 3тє вид., випр. і допов. — К.: Вища шк., 2002. — 622 с.: іл. Попов, А. О. Методичні вказівки до організації виконання та оформлення атестаційних робіт бакалавра, спеціаліста та магістра для студ. спец. "Фізична та біомедична електроніка" / А. О. Попов, Л. П. Медяний. – К. : 2009. – 90 с.
7. Розвиток опорно-рухового апарату дітей. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: http://pidruchniki.com/19570411/meditsina/rozvitok_oporno-ruhovogo_aparatu_ditey.
8. Мова програмування Java. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://live-box.net/movy-prohramuvannia/54-mova-prohramuvannia-java>.
9. Язык Java. Введение. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://www.java-study.ru/java-uchebnik/2-vvedenie>.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		70

10. Создание приложений и апплетов на языке Java. Часть 1. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: http://www.frolov-lib.ru/books/bsp/v30/ch2_2.htm.

11. Богатая объектная среда. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://bius.ru/java-book/30-CHto-takoe-Java/383-bogataya-obektnaya-sreda.html>

12. Применение информационных технологий в современной реабилитологии. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://www.rdc.org.ua/download/inform-kniga.pdf>.

13. Информатика / інформаційні системи і технології на підприємствах - Плєскач В.Л.. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://westudents.com.ua/glavy/27183-32-avtomatizatsya-proektuvannya-s.html>.

14. Робоче місце лікаря. Інформаційні технології в медицині. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: http://kiev-medical.ru/ukr_service_in.htm.

15. Основні діаграми моделі програмної системи. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://mybiblioteka.su/3-33662.html>.

16. Інформаційна система підтримки прийняття рішень сімейного лікаря. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://nure.ua/uk/university/structure/science/innovacijni-proekti-i-rozrobki/1-7-informacijna-sistema-pidtrimki-prijnyattya-rishen-simejnogo-likarya-zdorovya-sim%D1%97-1-0/>.

17. АРМ ЛІКАРЯ. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <https://sites.google.com/site/armlikar2014/>.

18. Компьютерные программы для врачей. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: http://www.livemedical.ru/tags/programmy_dlja_vrachej/.

19. Программы для врачей. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://softmedlist.narod.ru/>.

20. Автоматизоване робоче місце лікаря. [Электронный ресурс]. – 30.01.2016 – / Режим доступа: <http://ukrbukva.net/page,4,93080-Avtomatizirovannoe-rabochee-mesto-vracha.html>.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		71

21. UI для Firebird на Java. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://m.habrahabr.ru/post/145531/>.

22. Учебник по JavaFX 8. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://code.makery.ch/library/javafx-8-tutorial/ru/part2/>.

23. Работа с базами данных с помощью JDBC драйвера. [Електронний ресурс]. – 30.01.2016 – / Режим доступу: <http://devcolibri.com/477>.

24. Эльянов М.М. Медицинские информационные технологии. Каталог. Вып. 4 [Текст] / М.М. Эльянов. М.://Третья медицина, 2004. 350 с.

25. Попов, А. О. Методичні вказівки до організації виконання та оформлення атестаційних робіт бакалавра, спеціаліста та магістра для студ. спец. "Фізична та біомедична електроніка" / А. О. Попов, Л. П. Медяний. – К. : 2009. – 90 с.

26. Ассоциация развития медицинских информационных технологий АРМИТ [Електронний ресурс] / Режим доступу: <http://www.armit.ru/>

27. Методичні вказівки до виконання розділу «Охорона праці» в дипломних проектах для студентів Межрегіонального медико-інженерного факультету / Уклад.: Демчук Г.В., Демчук Т.В., К.: НТУУ «КПІ», 2011. – 55 с.

28. Основы охраны труда: Підручник. 2-ге видання / К.Н.Ткачук, М.О.Халімовський, В.В.Зацарний та ін. – К.: Основа, 2006 – 448 с.

29. Охрана труда в в медичних закладах: Довідник / Укладачі, Зеркалов Д.В., Теленгатор О.Я., Ушкевич Б.А., Дериземля І.О.; За ред. Зеркалова Д.В. - К. : Основа, 2008. - 728 с.

30. Кузін А.Г. Охрана труда [Текст] / Кузін А.Г. – М.: Біном, 2005. – 552 с.

31. Кривоніс Ф.М. Економіка підприємства [Текст] / Кривоніс Ф.М. – М.: Мир, 2005. – 523 с.

32. Прокоф'єв В.А. Економіка ХХ століття [Текст] / Прокоф'єв В.А. – М.: Мир, 2003. – 467 с.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		72

33. Пересада А.А. Инвестиционный процесс в Украине. – К. Либра, 1998. – 392 с.
34. Новицкий В.Е. Экономическая деятельность Украины: КНЕУ, 2003. – 948 с.
35. Методичні вказівки до виконання організаційно-економічного розділу дипломних проектів (за ред.А.Т.Чернявського та інших. К.НТУУ”КПІ”, 1999 – 66 с.)
36. Методичні вказівки до курсово роботи і організаційно-економічного розділу дипломних робіт та проектів. (Укл. М.М.Дученко, В.С.Сичов, Л.В.Ямкова. – К.КПІ, 1994 – 32с.)
37. Бобровникова Р.Г. Формування фінансових ресурсів підприємства./ Фінансово-кредитні відносини в Україні: стан та перспективи розвитку: матеріали Всеукраїнської науково-практичної конференції, 22-23 лютого 2013., Запоріжжя : ЗНТУ, 2013.- с.267. – С.245-249.
38. Данілов ОД. Інвестування: Навч. посіб./О.Д. Данілов, Г.М. Івашина, О.Г. Чумаченко. — К., 2001.
39. Загородній А.Г. Менеджмент реальних інвестицій: Навч. посібник/ А.Г. Загородній, Ю.І. Стадницький. – К.: Знання КОО, 2000. – 209 с.
40. Клименюк М.М., Брижань І.А. Управління ризиками в економіці. : Навч. посібник.-К.: Просвіт, 2000. – 256 с.
41. Курочкин А.С. Организация производства, Учебн.пособие для студентов вузов. – К.: МАУП, 2001. – 216с.
42. Онищенко В.О., Редкін О.В., Чевганова В.Я. Економіка та організація інноваційної діяльності. Практика організаційно-економічного управління проектами і програмами. Навч. посіб. – Полтава: ПолтНТУ, 2006.
43. Організація виробництва: Підручник/за заг.ред. П.В. Круша, В.І.Подвігіної, В.О.Гулевич. - К.: ЦУЛ, 2008.- 348 с.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		73

44. Пересада А.А. Проектное финансирование: Учебник / А.А. Пересада, Т.В. Майорова. – К.: КНЭУ, 2007. – 767с.

45. Подвігіна В.І., Гуревич В.О. Организация виробничого процесу в часі та просторі. Потокowe виробництво., – К.: ЦУЛ, 2007. – 136с.

46. Тіховська Т.М. Економіко-математичні моделі управління підприємством силової електроніки: теоретичні і прикладні аспекти/Т.М.Тіховська//Сучасні технології управління промисловими ринками та підприємствами: монографія/за ред. проф. П. В. Гудзя, Л. О. Кримської. – Запоріжжя : ЗНТУ, 2012. – 208 с. (С.124 - 137).

47. Тянь Р.Б., Багрова І.В. Організація виробництва: Навчальний посібник/За ред. д.ек.н., проф.Багрової І.В. – Київ, ЦНЛ, 2005. – 248с.

48. Фатхутдинов Р.А. Организация производства. Учебн. для вузов. – М.: ЗАО «Бизнес-школа «Интел-Синтез», 2002. – 344с.

49. Яловий Г.К., Пашин В.П., Сичов В.С. Методичні вказівки до виконання організаційно-економічного розділу дипломних проектів та курсових робіт з дисципліни «Економіка і організація виробництва» (для студентів факультету електроніки). – К.: НТУУ «КПІ», 2003 - 98с.

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		74

ДОДАТОК А

Код класу main

```
package sample;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
import sample.controller.LoginController;

public class Main extends Application {

    @Override
    public void start(Stage primaryStage) throws Exception{
        Parent root =
FXMLLoader.load(getClass().getResource("view/LoginBlock.fxml"));
        primaryStage.setTitle(" ");
        primaryStage.setScene(new Scene(root, primaryStage.getWidth(),
primaryStage.getHeight()));
        LoginController.currentStage = primaryStage;
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Код класу DBConnect

```
package sample.model;

import java.sql.*;

public class DBConnect {
    private static final String DRIVER =
"org.firebirdsql.jdbc.FBDriver";
    private static final String URL =
"jdbc:firebirdsql:localhost/3050:C:\\DB.FDB";
    private static final String LOGIN = "SYSDBA";
    private static final String PASSWORD = "masterkey";

    public static Connection getDBConnection(){
        Connection dbConnection = null;
        try {
            Class.forName(DRIVER);
            dbConnection = DriverManager.getConnection(URL, LOGIN,
PASSWORD);
        } catch (ClassNotFoundException ex) {
            ex.printStackTrace();
        } catch (SQLException e) {

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		75

```

        e.printStackTrace();
    }
    return dbConnection;
}
}

```

Код класу Patient

```

package sample.model;

import javafx.beans.property.SimpleStringProperty;
import javafx.beans.property.StringProperty;
import javafx.beans.value.ObservableValue;

public class Patient {

    private String numberOfCard;
    private String firstName;
    private String lastName;
    private String secondName;
    private String sex;
    private int age;

    public Patient (String firstName, String lastName){
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public Patient (String numberOfCard, String firstName, String
lastName) {
        this.numberOfCard = numberOfCard;
        this.firstName = firstName;
        this.lastName = lastName;
    }

    public Patient (String numberOfCard, String firstName, String
lastName, String secondName, String sex, int age) {
        this.numberOfCard = numberOfCard;
        this.firstName = firstName;
        this.lastName = lastName;
        this.secondName = secondName;
        this.sex = sex;
        this.age = age;
    }

    public Patient(){
    }

    public String getNumberOfCard() { return numberOfCard; }

    public void setNumberOfCard() { this.numberOfCard = numberOfCard;
}

    public String getFirstName() {
        return firstName;
    }
}

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		76

```

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName(){
    return lastName;
}

public void setLastName(String lastName){
    this.lastName = lastName;
}

public String getSecondName() {
    return secondName;
}

public void setSecondName(String firstName) {
    this.secondName = secondName;
}

public String getSex() { return sex; }

public void setSex(String firstName) {
    this.sex = sex;
}

public int getAge() {
    return age;
}

public void setAge(String firstName) {
    this.age = age;
}
}

```

Код класу MainBlockController

```

package sample.controller;

import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.stage.Modality;
import javafx.stage.Stage;
import sample.model.Patient;

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		77

```

import sample.model.DBConnect;
import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class MainBlockController {

    private ObservableList<Patient> patientData =
FXCollections.observableArrayList();

    private Stage stage = new Stage();

    @FXML
private TableView<Patient> patientTable;

    @FXML
private TableColumn<Patient, String> firstNameColumn;

    @FXML
private TableColumn<Patient, String> lastNameColumn;

    @FXML
private Label firstNameLabel;

    @FXML
private Label numberOfCardLabel;

    @FXML
private Label sexLabel;

    @FXML
private Label ageLabel;

    @FXML
private Label adressLabel;

    @FXML
private Label phoneLabel;
// инициализируем форму данными
    @FXML
private void initialize() {
    initData();

    // устанавливаем тип и значение которое должно храниться в
колонке
    firstNameColumn.setCellValueFactory(new
PropertyConnectionFactory<Patient, String>("firstName"));
    lastNameColumn.setCellValueFactory(new
PropertyConnectionFactory<Patient, String>("lastName"));

    // заполняем таблицу данными
    patientTable.setItems(patientData);

    // чистим инфу о пациенте (справа)

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		78

```

        showPatient(null);
        // слушаем выбор для изменения и выводим инфу поосле изменения

patientTable.getSelectionModel().selectedItemProperty().addListener(
        (observable, oldValue, newValue) ->
showPatient(newValue));
    }
    // подготавливаем данные для таблицы
    public void initData() {
        String selectTableSQL = "SELECT * from PATIENT";

        try {
            Connection dbConnection = DBConnect.getDBConnection();
            Statement statement = dbConnection.createStatement();

            // выбираем данные с БД
            ResultSet rs = statement.executeQuery(selectTableSQL);

            // И если что то было получено то цикл while сработает
            while (rs.next()) {
                patientData.add(new
Patient(rs.getString("NUMBER_OF_CARD"), rs.getString("FIRSTNAME"),
rs.getString("LASTNAME")));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void showPatient(Patient patient){
        if (patient !=null) {
            String selectTableSQL = "SELECT * from PATIENT WHERE
NUMBER_OF_CARD='" + patient.getNumberOfCard() + "'";

            try {
                Connection dbConnection = DBConnect.getDBConnection();
                Statement statement = dbConnection.createStatement();

                // выбираем данные с БД
                ResultSet rs = statement.executeQuery(selectTableSQL);

                rs.next();

numberOfCardLabel.setText(rs.getString("NUMBER_OF_CARD"));
                firstNameLabel.setText(rs.getString("LASTNAME") + " "
+ rs.getString("FIRSTNAME") + " " + rs.getString("SECONDNAME"));
                sexLabel.setText(rs.getString("SEX"));
                ageLabel.setText(Integer.toString(rs.getInt("AGE")));
                adressLabel.setText(rs.getString("ADRESS"));
                phoneLabel.setText(rs.getString("PHONE"));

            } catch (SQLException e) {
                e.printStackTrace();
            }
        } else{

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		79


```

        numberOfCardLabel.setText(" ");
        firstNameLabel.setText(" ");
        sexLabel.setText(" ");
        ageLabel.setText(" ");
    }

}

    public void clickNewPatient(ActionEvent actionEvent) throws
IOException {
        Parent root =
FXMLLoader.load(getClass().getResource("../view/AddBlock.fxml"));
        stage.setScene(new Scene(root, stage.getWidth(),
stage.getHeight()));
        stage.setTitle("Додати пацієнта");
        AddBlockController.currentStage = stage;
        stage.show();
    }

    public void clickDeletePatient(ActionEvent actionEvent) {

}

    public void clickEditPatient(ActionEvent actionEvent) throws
IOException {
        Parent root =
FXMLLoader.load(getClass().getResource("../view/AddBlock.fxml"));
        stage.setScene(new Scene(root, stage.getWidth(),
stage.getHeight()));
        stage.setTitle("Редагувати пацієнта");
        AddBlockController.currentStage = stage;
        stage.show();
    }
}

```

Код класу LoginController

```

package sample.controller;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.stage.Modality;
import javafx.stage.Stage;
import sample.model.DBConnect;
import sample.model.Patient;

import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		80

```

import java.sql.SQLException;
import java.sql.Statement;

public class LoginController {
    @FXML
    private TextField loginField;
    @FXML
    private TextField passField;
    @FXML
    private Label errorLabel;
    @FXML
    public static Stage currentStage;

    private Stage stage = new Stage();
    public void onClickCancel(ActionEvent actionEvent) throws
IOException {
        currentStage.close();
    }

    public void onClickLogin(ActionEvent actionEvent) throws
IOException {

        String login = loginField.getText();
        String password = passField.getText();

        String selectTableSQL = "SELECT LOGIN, PASSWORD FROM USERS
WHERE LOGIN='"+login+"' AND PASSWORD='"+password+"'";

        try {
            Connection dbConnection = DBConnect.getDBConnection();
            Statement statement = dbConnection.createStatement();

            // вибираем данные с БД
            ResultSet rs = statement.executeQuery(selectTableSQL);

            if (!(login.isEmpty()) & !(password.isEmpty())) {
                if (rs.next()) {
                    Parent root =
FXMLLoader.load(getClass().getResource("../view/MainBlock.fxml"));
                    stage.setScene(new Scene(root, stage.getWidth(),
stage.getHeight()));
                    stage.setTitle("My modal window");
                    stage.show();

                    currentStage.close();

                } else {
                    errorLabel.setText("Не вірний логін або пароль");
                }
            } else {
                errorLabel.setText("Заповніть всі поля");
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		81

Код класу AddBlockController

```
package sample.controller;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.CheckBox;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import sample.model.DBConnect;
import sample.model.Patient;

import java.io.IOException;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class AddBlockController {
    @FXML
    private TextField firstNameField;
    @FXML
    private TextField lastNameField;
    @FXML
    private TextField secondNameField;
    @FXML
    private TextField sexField;
    @FXML
    private TextField ageField;
    @FXML
    private TextField adressField;
    @FXML
    private TextField phoneField;
    @FXML
    private CheckBox registrationCheckBox;
    @FXML
    private TextField loginField;
    @FXML
    private TextField passwordField;

    @FXML
    public static Stage currentStage;

    @FXML
    private void initialize(){
        loginField.setDisable(true);
        passwordField.setDisable(true);
        registrationCheckBox.setSelected(false);
    }

    public void OnClickCancel(ActionEvent actionEvent) {
        currentStage.close();
    }
}
```

					БС42с.4208.1300.3166.ПЗ	Лист
Змін.	Лист	№ докум.	Підпис	Дата		82

```

    public void OnInsertClick(ActionEvent actionEvent) throws
    IOException {

        String insertTableSQL = "INSERT INTO PATIENT (FIRSTNAME,
LASTNAME, SECONDDNAME, SEX, AGE, " +
            "ADDRESS, PHONE) VALUES
('"+firstNameField.getText()+"', '"+lastNameField.getText()+"', '"+
            secondNameField.getText()+"', '"+
sexField.getText()+"', "+Integer.parseInt(ageField.getText())+" , '"+
            adressField.getText()+"',
 '"+phoneField.getText()+"')";

        try {
            Connection dbConnection = DBConnect.getDBConnection();
            Statement statement = dbConnection.createStatement();

            if (registrationCheckBox.isSelected()){
                statement.executeUpdate(insertTableSQL);
            }else{
                statement.executeUpdate(insertTableSQL);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        currentStage.close();
        //MainBlockController.initData();
    }

    public void OnClickCheck(ActionEvent actionEvent) {
        if (registrationCheckBox.isSelected()){
            loginField.setDisable(false);
            passwordField.setDisable(false);
        }else{
            loginField.setDisable(true);
            passwordField.setDisable(true);
        }
    }
}
}

```