

Міністерство освіти і науки України

Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інформаційних технологій проектування (№ 105)

ЗАТВЕРДЖУЮ

Голова НМК


(підпис)

Д.М.Крицький
(ініціали та прізвище)

« 31 » 08 2021 р.

**РОБОЧА ПРОГРАМА ВИБІРКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Інструментальні засоби візуального програмування

(назва навчальної дисципліни)

Галузі знань: 12 “Інформаційні технології”
(шифр і найменування галузі знань)

Спеціальності: 122 “Комп’ютерні науки”, 126 “Інформаційні системи та технології”
(код та найменування спеціальності)

Освітні програми: Інформаційні технології проектування,
Інформаційні системи та технології
підтримки віртуальних середовищ
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2021 рік

Робоча програма Інструментальні засоби візуального програмування
(назва дисципліни)
для студентів за спеціальностями 122 “Комп’ютерні науки”,
126 “Інформаційні системи та технології”

Освітні програми: Інформаційні технології проектування
Інформаційні системи та технології
підтримки віртуальних середовищ
(найменування освітньої програми)

«08» 06 2021 р., - 13 с.

Розробник: Овсяннік В.М., доцент к.т.н.

(прізвище та ініціали, посада, наукова ступінь та вчене звання)


(підпис)

Робочу програму розглянуто на засіданні кафедри
інформаційних технологій проектування
(назва кафедри)

Протокол № 1 від «31» серпня 2021 р.

Завідувач кафедри

к.т.н., доцент
(науковий ступінь та вчене звання)


(підпис)

Д.М.Крицький
(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки (спеціальність, спеціалізація), рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)	
Кількість кредитів – 8	Галузь знань <u>12 «Інформаційні технології»</u> <small>(шифр і назва)</small> Спеціальності <u>122 “Комп’ютерні науки”,</u> <u>126 “Інформаційні системи та технології”</u> <small>(шифр і назва)</small> Освітні програми <u>Інформаційні технології проектування,</u> <u>Інформаційні системи та технології підтримки віртуальних середовищ</u> <small>(найменування освітньої програми)</small> Рівень вищої освіти: перший (бакалаврський)	Цикл професійної підготовки за вибором	
Модулів – 2		Навчальний рік:	
Змістових модулів – 4		2021 / 2022	
Курсовий проект ¹⁾		Семестр	
Загальна кількість годин – 88/240		4-й	5-й
Тижневих годин для денної форми навчання: аудиторних – 4 (4-й семестр), 1.5 (5-й семестр); самостійної роботи студента – 6.3 (4-й семестр), 2.3 (5-й семестр)		Лекції²⁾	
	48 годин	-	
	Практичні, семінарські		
	16 годин		
	Лабораторні²⁾		
	32 годин	-	
Самостійна робота			
116 годин	36 годин		
Вид контролю			
	іспит	диф. залік (КП)	

Співвідношення кількості годин аудиторних занять до самостійної роботи становить для денної форми $88/152=0.58$.

¹⁾ Курсовий проект та практичні заняття виконуються у п'ятому семестрі на третьому році підготовки

²⁾ Аудиторне навантаження може бути зменшеним або збільшеним на одну годину в залежності від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення навчальної дисципліни полягає у підготовці спеціалістів з інформаційних систем до виконання робіт з розроблення програмного забезпечення з використанням принципів та методів об'єктно-орієнтованого програмування.

Основними завданнями дисципліни є вивчення засобів розробки програмного забезпечення для роботи під керівництвом ОС Windows з широким використанням можливостей об'єктно-орієнтованого програмування, правила будівництва програмних засобів в середовищах візуального програмування, налагодження програм та розв'язання типових задач

Міждисциплінарні зв'язки: навчальна дисципліна розрахована на студентів, що мають підготовку по інформатиці і програмуванню, знання основних базових алгоритмів і базових конструкцій мов програмування. Знання та уміння можуть бути використані для розробки алгоритмів і програм на базі об'єктно-орієнтованого підходу. Викладання дисципліни базується на курсах «Основи програмування», «Алгоритмізація і програмування» та «Об'єктно-орієнтоване програмування». Знання та навички, що здобувають студенти під час вивчення дисципліни, використовуються при вивченні таких дисциплін, як «Технологія розробки програмних продуктів», «Сучасні технології та інструментарій програмування», «Організація баз даних та знань»

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких загальних та фахових компетентностей:
ЗК1, ЗК2, ЗК3. ЗК4, ЗК6, ЗК7, ЗК12СК1, СК3, СК8.

Програмні результати навчання:

ПР 5, ПР9, ПР10. ПР13

У результаті вивчення навчальної дисципліни студент повинен

знати:

- алгоритмічну мову програмування C++ та C++/CLI;
- найбільш поширені інструментальні засоби для програмування у середовищі Microsoft Visual Studio;
- особливості роботи програм під керуванням Windows;
- правила будівництва програм в візуальних середовищах програмування;

уміти:

- створювати програмні засоби, що працюють під керуванням Windows;
- розробляти інтерфейси користувача за допомогою засобів середовищ візуального програмування;
- програмувати елементи інтерфейсу користувача з використанням керуючих елементів, що є типовими для сучасних застосунків;
- використовувати основні програмні механізми при програмуванні інтерфейсів користувача;
- проектувати та програмувати ієрархію класів для вирішення типових задач систем автоматизованого проектування (САПР).

Студент має надбати **навички** самостійного проектування, розробки та налагодження застосунків, призначених для розв'язання типових математичних та прикладних задач, що виникають під час розробки функціональних складових модулів САПР та застосунків іншого призначення.

3. Програма навчальної дисципліни

Модуль 1. Засоби мови С++, що значно розширюють її можливості.

Змістовий модуль 1. Перевантаження операцій

Тема 1. Вступ до дисципліни.

Ціль і задачі курсу. Види занять та їх призначення, модульно-рейтингова система. Рекомендована та додаткова література, у тому числі у електронному форматі. Основні етапи та складові розвитку візуальних систем розробки застосувань Windows.

Тема 2. Статичні компоненти класів.

Призначення статичних членів класів, правила їх опису та використання, особливі властивості. Приклади використання статичних членів класів для розв'язання типових задач, що виникають у процесі розробки модулів САПР.

Тема 3. Дружні функції та класи.

Призначення дружніх функцій класів, правила їх опису та використання, особливі властивості. Опис та використання дружніх класів. Сфери та приклади застосування дружніх функцій та класів.

Тема 4. Перевантаження операцій.

Поняття та правила перевантаження операцій. Формат операцій-функцій, способи їх реалізації. Перелік операцій, що не підлягають перевантаженню. Залежність способів оформлення операцій-функцій від арістії операцій та їх призначення. Приклади класів, у яких використовується перевантаження операцій. Переваги перевантаження операцій.

Змістовий модуль 2. Важливі властивості системи Windows та їх використання

Тема 5. Адресні простори, застосування, процеси та потоки.

Поняття та визначення адресного простору, застосування, процесу та потоку. Взаємодії та взаємовідносини процесів та потоків. Синхронізація потоків. Класифікація пріоритетів процесів та потоків Windows. Використання потоків Windows для вирішення практичних задач.

Тема 6. Оброблення виняткових ситуацій.

Види помилок у програмах. Проблеми традиційного підходу до обробки помилок. Механізм обробки виключень. Класи виключень стандартної бібліотеки мови С++. Синтаксис обробки виключень. Розробка власних класів оброблення виняткових ситуацій.

МОДУЛЬ 2. Стандартна бібліотека С++ та елементи керування

Змістовий модуль 3. Стандартна бібліотека мови С++

Тема 7. Інтелектуальні покажчики.

Недоліки використання «звичайних» покажчиків. Перелік помилок, що зазвичай виникають при використанні «звичайних» покажчиків та їх наслідки. Головна ідея інтелектуальних покажчиків та способи її реалізації. Переваги, сфери та приклади застосування інтелектуальних покажчиків.

Тема 8. Класифікація засобів стандартної бібліотеки мови С++.

Поняття та класифікація засобів бібліотеки STL (Standard Template Library). Перелік та загальні характеристики шаблонних класів бібліотеки STL. Найбільш поширені засоби бібліотеки STL, їх опис та приклади використання.

Тема 9. Контейнери стандартної бібліотеки та їх використання.

Класифікація та характеристики контейнерів стандартної бібліотеки, опис доцільності їх використання, перелік переваг та недоліків. Класифікація та опис найбільш поширених функцій контейнерів, приклади їх використання.

Тема 10. Функціональні об'єкти та їх використання у алгоритмах стандартної бібліотеки.

Поняття функціонального об'єкту, його призначення та сфера використання. Опис та реалізація функціональних об'єктів. Поняття предикатів. Приклади використання функціональних об'єктів у застосуваннях, що використовують алгоритми стандартної бібліотеки.

Змістовий модуль 4. Використання найбільш поширених елементів керування Windows

Тема 11. Реалізація та використання немодальних діалогових вікон.

Визначення та призначення немодальних діалогових вікон, їх відмінності від модальних вікон. Способи та приклади реалізації та використання немодальних діалогових вікон, їх переваги та недоліки. Взаємодія застосування Windows з немодальними діалоговими вікнами.

Тема 12. Використання елементів керування ActiveX у застосуваннях Windows.

Історична довідка. Визначення та призначення елементів керування ActiveX, їх відмінності від «звичайних» елементів керування. Способи та приклади реалізації та використання елементів керування ActiveX, їх переваги та недоліки.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		л	п	лаб	курс	с.р.
Модуль 1. Засоби мови C++, що значно розширюють її можливості						
Змістовий модуль 1. Перевантаження операцій						
Тема 1. Вступ до дисципліни	2	2	–	–		–
Тема 2. Статичні компоненти класів	12	2	–	2		8
Тема 3. Дружні функції та класи.	14	2	2	2		8
Тема 4. Перевантаження операцій	20	4	2	4		10
Разом за змістовим модулем 1	48	10	4	8		26
Змістовий модуль 2. Важливі властивості системи Windows та їх використання						
Тема 5. Адресні простори, застосування, процеси та потоки	23	2	3	6		12
Тема 6. Оброблення виняткових ситуацій	29	4	3	4		18
Разом за змістовим модулем 2	52	6	6	10		30
Усього годин за модулем 1	100	16	10	18		56
Модуль 2. Стандартна бібліотека C++ та елементи керування						
Змістовий модуль 3. Стандартна бібліотека мови C++						
Тема 7. Інтелектуальні покажчики	13	2	2	2		7
Тема 8. Класифікація засобів стандартної бібліотеки мови C++	6	4	2	–		–

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	курс	с.р.	
Тема 9. Контейнери стандартної бібліотеки та їх використання	21	2	2	2		15
Тема 10. Функціональні об'єкти та їх використання у алгоритмах стандартної бібліотеки	20	2	4	4		10
Разом за змістовим модулем 3	60	10	10	8		32
Змістовий модуль 4. Використання найбільш поширених елементів керування Windows						
Тема 11. Реалізація та використання немодальних діалогових вікон	21	3	2	4		12
Тема 12. Використання елементів керування ActiveX у застосуваннях Windows	23	3	2	2		16
Разом за змістовим модулем 4	44	6	4	6		28
Усього годин за модулем 2	104	16	14	14		60
Курсовий проект	60	–	24	–		36
Усього годин за дисципліною	240	32	24	32		152

5. Теми семінарських занять

Семінарські заняття навчальним планом не передбачені.

6. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Поставлення задачі проектування ієрархії класів. Абстрагування обраної предметної області. Аналіз вимог до ієрархії класів	2
2	Вибір предметної області. Сбір та аналіз реальних об'єктів обраної предметної області. Виявлення переліку атрибутів та методів об'єктів.	4
3	Проектування ієрархії класів обраної предметної області. Аналіз та виявлення загальних атрибутів об'єктів як головного принципу розбудови ієрархії класів. Опис ієрархії класів на мові UML та побудова діаграм класів у середовищі Microsoft Visual Studio	4
4	Вибір каркасу застосування Windows, що реалізує створення та обробку об'єктів спроектованих класів. Проектування вікон, розробка меню та панелей інструментів	2
5	Кодування функцій класів, конструкторів та деструкторів. Розробка функцій доступу до приватних атрибутів класів	2
6	Налагодження застосування, кодування обробки виняткових ситуацій	4
7	Тестування застосування, наповнення довідкової системи застосування	2
8	Остаточне оформлення пояснювальної записки до курсового проекту	2
9	Типові помилки, що виникають при оформленні пояснювальної записки. Виправлення зауважень керівника стосовно пояснювальної записки	2
	Разом	24

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Розробка типового консольного додатка	4
2	Дослідження похибок обчислень	4
3	Обробка текстового файлу	4
4	Пошук даних у масиві	4
5	Додаток з вікном виду	4
6	Розробка графічного редактора	4
7	Додаток, заснований на модальному діалоговому	4
8	Клас вектор	4
	Разом	32

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Статичні компоненти класів	18
2	<u>Дружні функції та класи</u> . Опис та використання дружніх класів. Застосування дружніх функцій та класів для обраної предметної області.	16
3	<u>Перевантаження операцій</u> . Особливості перевантаження операцій доступу до елементів масивів та операцій присвоєння. .	16
4	<u>Адресні простори, застосування, процеси та потоки</u> . Засоби синхронізації потоків, що надаються Windows	10
5	<u>Оброблення виняткових ситуацій</u> . Розробка власних класів оброблення виняткових ситуацій	10
6	Інтелектуальні покажчики	14
7	<u>Контейнери стандартної бібліотеки та їх використання</u> . Застосування контейнерів типу вектор та асоціативних масивів для обробки масивів об'єктів користувача	16
8	<u>Функціональні об'єкти та їх використання у алгоритмах стандартної бібліотеки</u> . Самостійна розробка функціональних об'єктів, що призначені для обробки масивів об'єктів користувача	14
9	<u>Реалізація та використання немодальних діалогових вікон</u> . Доступ до даних, що вводяться за допомогою немодальних діалогових вікон	18
10	<u>Використання елементів керування ActiveX у застосуваннях Windows</u> . Використання елементів ActiveX, що призначені для відображення даних у графічній формі	20
	Разом	152

9. Курсовий проект

Виконання курсового проекту є важливим етапом вивчення дисципліни. Робота над курсовим проектом сприяє систематизації, поглибленню й закріпленню знань, отриманих студентами при вивченні даної дисципліни. У процесі курсового проектування студенти розвивають навички практичного застосування отриманих знань при створенні

комплексного додатка з використанням сучасних інструментальних засобів розробки. При цьому студент повинен показати вміння користуватися спеціальною літературою, державними стандартами, довідниками та іншими матеріалами з інформаційних технологій.

У розробленому курсовому проекті студент повинен показати знання:

- основних елементів предметної області відповідно до постановки завдання;
- аналізу предметної області та її абстрагування з метою виявлення класів;
- інструментальних засобів, призначених для розроблення сучасних об'єктно-орієнтованих додатків Windows.

Студент повинен показати вміння з:

- аналізу постановки завдання;
- розроблення загальної архітектури програмної системи;
- деталізації загальної архітектури програмної системи у термінах об'єктно-орієнтованого програмування;
- програмної реалізації системи, що розробляється, на мові програмування;
- застосування стандартних бібліотек мови програмування;
- документування вихідного коду програми;
- використання засобів розроблення програм та отримання довідкової інформації;
- розроблення та оформлення пояснювальної записки до проекту.

Робота над курсовим проектом певною мірою визначає загально-теоретичну та спеціальну підготовку студента і в остаточному підсумку готує його до майбутнього виконання більш складного й завершального етапу навчального процесу – дипломного проектування. Студент повинен розглядати роботу над курсовим проектом як своєрідну "репетицію" дипломного проектування.

Якісне виконання курсового проекту вимагає чіткої організації роботи студента з моменту вибору теми проекту й до його захисту. Студентові надається право вільного вибору теми проекту з урахуванням його схильностей і можливостей найбільше повно застосувати отримані знання.

Після затвердження теми курсового проекту студентові видається завдання на курсове проектування. У ньому наводиться тема курсового проекту, зміст пояснювальної записки, завдання на розроблення додатку, строки початку й закінчення роботи над курсовим проектом, обумовлені графіком навчального процесу.

Студент розробляє зміст курсового проекту, обговорює його з керівником, підготовляє вхідні дані і приступає до проектування. У процесі проектування студент повинен регулярно відвідувати консультації керівника, подавати на перевірку йому робочі матеріали, компоненти програмного забезпечення та застосування.

Курсовий проект студент повинен виконувати самостійно. Оформлений відповідно до пред'явлених вимог проект, студент здає на перевірку керівникові за тиждень до строку захисту.

Термін виконання – тижні 1 – 14. Трудомісткість – 30 годин аудиторних занять, 30 годин самостійної роботи. Обсяг пояснювальної записки (ПЗ) – 20-30 сторінок. Типовий план-графік виконання курсового проекту наведено у наступній таблиці.

№	Етап	Тиждень семестру	Форма звіту	% обсягу
1	Вибір і затвердження теми завдання	2	Лист затвердження завдання	5

2	Проектування додатка (ієрархія класів, перелік атрибутів і методів класів)	4	Розділ записки (електронна форма)	20
3	Проектування додатка (вибір каркасу додатка; проектування вікон, меню, панелей інструментів)	7	Розділ записки (електронна форма), проект програми	40
4	Кодування додатка	9	Програма	60
5	Налагодження додатка	11	Програма	80
6	Оформлення пояснювальної записки і завершення розробки програми	12	Пояснювальна записка	90
7	Здавання пояснювальної записки і програми	13	Записка і програма	100
8	Захист проекту	14	–	–

Розподіл балів, які отримують студенти за виконання курсового проекту:

Пояснювальна записка	Ілюстративна частина	Захист роботи	Сума
до 40	до 40	до 20	до 100

10. Методи навчання

Лекції проводяться з використанням основних розділів конспекту лекцій в електронній формі.

Лабораторні роботи виконуються з використанням навчальних та ліцензованих версій Microsoft Visual Studio, методичних рекомендацій та детальних сценаріїв у друкованому та електронному вигляді.

Самостійна робота включає підготовку до лабораторних робіт, модульних контролів та іспиту, виконання позааудиторної частини індивідуального завдання з курсового проекту і вивчення вказаних вище тем за конспектом, літературними і електронними джерелами та програмною документацією.

11. Методи контролю

Поточний контроль здійснюється відповідно до повноти, своєчасності та якості виконання лабораторних робіт та курсового проектування згідно з інструкцією з модульно-рейтингового оцінювання знань студентів; проміжний (модульний) контроль – письмові контрольні роботи на 9-му та 16-му тижнях; підсумковий контроль – письмовий іспит.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	4...6	4	16...25
Модульний контроль	14...20	1	14...20
Змістовний модуль 2			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	4...6	4	16...25
Модульний контроль	14...20	1	14...20
Усього за семестр			60...100

Курсовий проект: Поточний контроль здійснюється відповідно до повноти, своєчасності та якості виконання окремих розділів, за кожний процент трудомісткості при належній повноті, своєчасному та якісному виконанні розділу нараховується один бал; підсумковий контроль – захист, залік з оцінкою.

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

- мати уявлення про весь теоретичний матеріал дисципліни;
- знати в повному обсязі не менш половини тем теоретичного матеріалу;
- знати додатковий матеріал по трьом темам.

Необхідний обсяг умінь для одержання позитивної оцінки:

- володіти технологією застосування теоретичних знань на практиці;
- уміти самостійно знаходити довідкові відомості;
- уміти застосовувати на практиці отримані знання.

12.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Знати основні положення теоретичного матеріалу. Вміти користуватися засобами інструментального середовища Visual Studio. Вміти розробляти додатки Windows, що вирішують задачі з використанням таких типів даних, як шаблонні класи.

Добре (75-89). Знати основний теоретичний матеріал в повному обсязі. Володіти технологією пошуку довідкової літератури. Вміти розробляти та налагоджувати додатки в інструментальному середовищу Visual Studio, що використовують шаблонні класи STL.

Відмінно (90-100). Знати основний і додатковий теоретичний матеріал в повному обсязі. Добре орієнтуватися в довідковій літературі. Вміти вирішувати задачі та

розробляти додатки з використанням платформи .NET Framework. Вміти розробляти додатки, що використовують бібліотек WinForms.

Переведення підсумкових семестрових оцінок в оцінки за національною шкалою та в оцінки за міжнародною шкалою ECTS здійснюється відповідно до «Положення про модульно-рейтингову систему оцінювання знань студентів» за наведеною нижче таблицею.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90–100	A	відмінно
83–89	B	добре
75–82	C	
68–74	D	задовільно
60–67	E	
1–59	FX	незадовільно з можливістю повторного складання

13. Методичне забезпечення

1. Овсянник В.Н. Инструментальные средства визуального программирования. Конспект лекций. – [Электронный ресурс] – Режим доступа: – 10.1.108.193.(01.09.2019)
2. Овсянник В.Н. Лабораторные работы по курсу «Инструментальные средства визуального программирования» – [Электронный ресурс] – Режим доступа: – 10.1.108.193.(01.09.2019)
3. Овсянник В.Н. Курсовой проект по дисциплине «Инструментальные средства визуального программирования» – [Электронный ресурс] – Режим доступа: – 10.1.108.193.(01.09.2019)

14. Рекомендована література

14.1. Базова

1. Кравець П.О. Об'єктно-орієнтоване програмування: навч. посібник/ П.О.Кравець. – Львів: Видавництво Львівської політехніки, 2014. – 624с.
2. Юрченко І.В., Сікора В.С. Інформатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с.
3. Овсянник В.Н. Язык C++ не для чайников. Учебное пособие – [Электронный ресурс] – Режим доступа: – 10.1.108.193.(01.09.2019)

14.2. Допоміжна

1. Дьюхерст С. C++. Священные знания, 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2016. – 240с.
2. Джеффри Рихтер. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Пер. с англ., 4-е изд. – Изд-во: Питер, 2013 - 896с.
3. Джеффри Рихтер, Мартен ван де Боспурт. WinRT: программирование на C# для профессионалов. Windows Runtime via C#// Пер. с англ. – М.: Вильямс, 2014. – 368 с.
4. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2007.-496с.:ил.

15. Інформаційні ресурси

1. База знань Microsoft Developer Network (MSDN) – [Електронний ресурс] – Режим доступу: – <http://msdn.microsoft.com/> (01.09.2017)
2. Офіційний сайт компанії Microsoft щодо технологій WPF та Windows Forms – [Електронний ресурс] – Режим доступу: – <http://window-sclient.net> (01.09.2016)
3. Internet-інститут інформаційних технологій – [Електронний ресурс] – Режим доступу: – www.intuit.ru (01.09.2017)