

BİLİM VE EĞİTİM BAKANLIĞI
ULUSAL TEKNİK ÜNİVERSİTESİ
“KHARKIV POLYTECHNIC ENSTİTÜSÜ”
YAZILIM MÜHENDİSLİĞİ VE YÖNETİMİ BÖLÜMÜ
BİLİŞİM TEKNOLOJİLERİ

METODİK TAVSİYE
YAZILIM MÜHENDİSLİĞİ
LABORATUVAR UYGULAMASI

Kharkiv

2016

Sayfa 2

2

içerik

Laboratuar № 1	3
Laboratuar № 2	10
Laboratuar № 3	14
Laboratuar № 4	23

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Sayfa 3

3

Laboratuar № 1

Amaç: **Görsel paradigmayı kullanarak Durum ve Etkinlik Diyagramlarını Kullanmayı Öğrenme**

Görevler:

1. Görevi alın ve hesaplama algoritmalarını analiz edin.
2. Etkinlik diyagramını geliştirin.
3. Kullanım Örneği diyagramını geliştirin.
4. Çalışmanın raporunu hazırlayın

Laboratuarın ilerlemesi.

1. Görevi alın ve hesaplama algoritmalarını analiz edin.

Tablo 1'deki göreve göre öğrenci hesaplamayı analiz etmelidir.

algoritmaları.

Tablo 1 - Laboratuar1 için Görev

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Örneğin, aşağıdaki görevi aldık:

Uygulama aşağıdaki formülü hesaplamak zorundadır:

$$S = \sum_{i=1}^n i \cdot 2^{i-1}$$

Uygulama ayrıca klavye ve dosya ve çıktıdan veri girişi sağlamalıdır
ekrana ve dosyaya.

İfadede gördüğümüz gibi, bilinmeyen değişkenlere sahibiz a ve n . Hesaplamak için
Formül öğrencisi önce a ve n değerlerini bulmalıdır.

let $n = 5$; $a = 4$. Elde ettiğimizden:

$$S = \sum_{i=1}^n i \cdot 2^{i-1} = \sum_{i=1}^5 i \cdot 2^{i-1} = 1 \cdot 2^0 + 2 \cdot 2^1 + 3 \cdot 2^2 + 4 \cdot 2^3 + 5 \cdot 2^4 = 1 + 4 + 12 + 32 + 80 = 129$$

2. Etkinlik diyagramını geliştirin.

Ücretsiz özel yazılım **Visual Paradigm Community Edition**

Etkinlik şemasını çizmek için [<https://www.visual-paradigm.com/>] önerilir

(Şekil 1.1 - 1.2).

Şekil 1.1 - Görsel Paradigma ana menüsü

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Şekil 1.2 - Visual Paradigm'in farklı sürümleriyle sayfa

Görsel Paradigmayı Başlat. Menüü kullanarak yeni proje oluştur: "Dosya → Yeni Proje". Projenin adını tanımlayın. Ardından aktivite şeması oluşturun ve kullanarak ismi menu: "Dosya → Yeni Diyagram → Yeni Etkinlik Diyagramı"; veya "Etkinlik" i tıklayın. Diyagram Gezgini'nde Diyagram "menüsünü ve ardından" Yeni Etkinlik'i seçin. Diyagram".

Aktivite diyagramı sistemin dinamik özelliklerini tanımlamak için kullanılır. İçin örneğin, gerekli olan farklı algoritmaları tanımlamak için kullanmak mümkündür sistemde uygulamak. Aşağıdaki düğümler ve kenarlar genellikle çizilir UML aktivite diyagramları (Şekil 1.3): aktivite, bölüm, eylem, nesne, kontrol, etkinlik kenarı.

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Şekil 1.3 - Temel Etkinlik Şeması öğeleri

Bizim durumumuzda hesaplamalı
Algoritma üzerinde gösterilen gibi görünmelidir
Şekil 1.4.

Yazı tipi değiştirme olasılığı vardır
aktivite şemasındaki değer: sağ tıklama
bir öğe veya bir diyagram, sonra "Stilleri"
ve Biçimlendirme "menü öğrencisi
"Format" menüsünü seçiniz. Kullanıcı ayarlayabilirsiniz
Diyagram şeffaflık aynı şekilde:
"Stiller ve Biçimlendirme → Şeffaf".

Şekil 1.4 –Aktivite Şeması

seçilen görev için

Yazar: Melnik KV, doktora asistanı

3. Kullanım Örneği diyagramını geliştirin.

Kullanım Örneği diyagramı oluşturun ve adı aşağıdaki gibi ayarlayın: "Dosya → Yeni Diyagram → Yeni Kullanım Durum Diyagramı" menüsü; veya içindeki "Durum Diyagramı Kullan" menüsünü tıklayın. Diyagram Gezini ve ardından "Yeni Kullanım Durum Diyagramı" nı seçin.

Kullanım durum diyagramları , birtakım eylemleri tanımlamak için kullanılır.

Sistem bir veya daha fazla harici kullanıcıyla işbirliği yapmalı veya gerçekleştirmelidir

Sistem (aktörler). Kullanım Örneği diyagramı işlevsel gereksinimleri yansıtır

yazılım. Kullanım Örneği diyagramının tüm ana elemanları tablo 2'de açıklanmıştır.

Tablo 2 - Kullanım Durum Diyagramının Ana Unsurları

Kullanım durumları işlevselliği tanımlamak
Sistemler tarafından sağlanır ve
Sistemlerin gereksinimlerini karşılar
ortamı.

Bir **oyuncu** bir kişi, organizasyon veya
birinde rol oynayan harici sistem
veya sisteminizle daha fazla etkileşim.

Sistem sınır kutuları (isteğe bağlı)
sistemin kapsamını gösterir.

Bir aktör ile bir oyuncu arasındaki **ilişki**
kullanım durumunda aktör ve
bir şekilde etkileşimi kullan veya
birbirleriyle iletişim kurabilir. Bir aktör
bir veya birkaç kullanımla ilişkilendirilebilir
vakalar.

uzatmak

Kullanım kapsamının genişletilmesi isteğe bağlıdır.
Tamamlayıcı.

Dahil etmek

İsteğe bağlı değil, gerekli kullanım durumu.

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Sayfa 8

8

genelleme

Özel kullanım durumu gerekli, değil

İsteğe bağlı, baz kullanım durumu soyut ise.

Ücretsiz özel yazılım Visual Paradigm Community Edition

Use Case diyagramını çizmeniz önerilir. Verilen Kullanım Örneği diyagramı

Görev Şekil 1.5'de gösterilmiştir.

Şekil 1.5 - Seçilen görev için Durum Diyagramı kullanın

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Sayfa 9

9

4. Çalışmanın raporunu hazırlayın

Rapor için gerekenler:

1. Rapor bölüm standartlarına göre hazırlanacaktır.

Şablon aşağıdaki yolla bulunabilir:

\\SELENA \ Method \ Kağıt çalışması \ STVUZ.dot

2. Rapor, tüm görevlerin yerine getirilmesinin sonuçlarını içermelidir.

Laboratuvara bireysel göre göre.

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Sayfa 10

10

Laboratuar № 2

Hedef: C ++ 'nın temel prensiplerini öğrenmek

Görevler:

1. Önceki laboratuardaki görev için kod geliştirin.
2. Çalışmanın raporunu hazırlayın

Laboratuvarın ilerlemesi.

1. Önceki laboratuardaki görev için kod geliştirin.

Bir önceki laboratuvarın görevine göre hesaplama geliştirmelisiniz

Visual Studio'da algoritma.

Örneğin, aşağıdaki görevi aldınız:

Uygulama, aşağıdaki formülü kullanarak değerleri hesaplamak zorundadır.

önceden tanımlanmış bir adımla aralığı:

$$y = \begin{cases} \frac{2n}{7x} & \text{ix} (2 < 7x) \\ \frac{2i}{7x} & \text{ix} (7x \geq 2) \end{cases} \quad (2.1)$$

Çözüm

İlk olarak, formülü hesaplamak için x ve n değerlerini *girmelisiniz*.

kısıtlamanın ardından: n değeri 4'ten büyük olmalıdır:

$$\frac{2n}{2i} \text{ix} (2 = \frac{2^4}{2i}) \text{ix} (2 = \frac{2^2}{2i}) \text{ix} (2 = \frac{2^2}{2i}) \text{ix} (2 = \frac{2^2}{2i}) - 2$$

Aşağıdakiler birkaç örnektir:

$$\begin{aligned} 6x; 5n &= \Rightarrow y = \frac{2^5}{2i} \text{ix} (2 = \frac{2^3}{2i}) 36 () 26 () \text{ix} (2 + - 2 = 25 \\ 8x; 5n &= \Rightarrow \frac{2i}{15787xy} = + \frac{2i}{2i} \end{aligned}$$

Yeni bir Visual Studio C++ projesi oluşturun. Menü çubuğunda

Dosya → Yeni → Proje (Şekil 2.1.) .

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Visual C ++ kategorisinde, Win32 Console Application şablonunu seçin.
ve sonra projeyi adlandırın (Şekil 2.2).

Şekil 2.2 - Boş bir proje oluşturma süreci

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Win32 Uygulama Sihirbazı görüldüğünde, Finish düğmesini seçin.
Görevi önceki laboratuardan uygulayın.

Fonksiyon y hesaplanması için bir program, Şekil 2.3 verilmiştir. Numuneler

Program tarafından hesaplanan Şekil 2.4'te bulunabilir.

Şekil 2.3 - Seçilen görev için program

Şekil 2.4 - Yürütme sonucu program

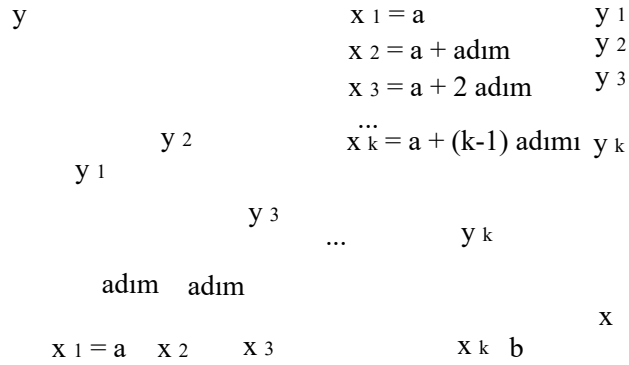
Bir sonraki adım, programı değiştirmek. Aşağıdaki olmalı
olasılıklar:

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

- Önceden tanımlı aralığı girmelisiniz) b, a (ve adım;
- Program (b) aralığından her bir değer, x için fonksiyon y hesaplamalıdır

(Şekil 2.5);

- Her adımın sonuçları ekranda görüntülenmelidir.



Şekil 2.5 - Hesaplamanın grafiksel gösterimi

2. Çalışmanın raporunu hazırlayın

Rapor için gerekenler:

1. Rapor bölüm standartlarına göre hazırlanacaktır.

Şablon aşağıdaki yolla bulunabilir:

\\SELENA \ Method \ Kağıt çalışması \ STVUZ.dot

2. Rapor, tüm görevlerin yerine getirilmesinin sonuçlarını içermelidir.

Laboratuvara bireysel göre göre.

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Laboratuvar № 3

Hedef: C ++ kodunu test etmenin temel prensiplerini öğrenmek

Görevler:

1. C ++ 'da fonksiyonların kullanım prensiplerini incelemek.
2. C ++ 'da İstisna İşleme Çalışması.
3. Önceki laboratuvardaki kodu 1 ve 2 görevlerine göre değiştirin.
4. Geliştirilmiş program için birim testini uygulayın.
5. Çalışmanın raporunu hazırlayın

Laboratuvarın ilerlemesi.

1. C ++ 'da fonksiyonların kullanım prensiplerini incelemek.

C ++ 'daki işlevleri kullanarak kodu önceki laboratuardan değiştirmelisiniz.

Bir işlev , bir isim verilen ve olabilecekleri bir grup ifadedir.

Programın bir noktasında çağrıldı. Fonksiyonlar, programların yapılandırılmasına izin verir

bireysel görevler gerçekleştirmek için kod bölümleri. Ana avantaj bir işlevdir

aslında bir program içinde birden çok kez çağrılabilir.

Bir fonksiyonun programcı tarafından önceden tanımlanmış veya oluşturulmuş olmasına bağlı olarak;
Orada
iki tür işlev vardır:

1. Kütüphane Fonksiyonu . Programcı, çağırma ile kütüphane işlevini kullanabilir

doğrudan işlev; Kendilerini yazmaları gerekmiyor.

2. Kullanıcı tanımlı Fonksiyon. Kullanıcı tanımlı bir işlev gerçekleştirmek için grup kodu

belirli bir görev ve bu kod grubuna bir isim (tanımlayıcı) verilir.

Bir işlevi tanımlamak için en yaygın sözdizimi şöyledir:

tip adı (parametre1, parametre2, ...) {statement}

nerede:

- tipi , işlev tarafından döndürülen değerin türüdür; eğer hiçbir değer iade edilmezse

arama fonksiyonu o zaman, boşluk kullanılmalıdır;

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Sayfa 15

15

- ad , işlevin çağrılabilceği tanımlayıcıdır;

- parametreler (gerektiği kadar): Parametrelerin amacı geçmesine izin vermektir

çağrıldığı yerden işlevine argümanlar;

- ifadeler fonksiyonun bedeni.

2. C ++ 'da İstisna İşleme Çalışması.

Bir istisna programın devam etmesini engelleyen bir hata koşulu.

düzenli yürütme yolu boyunca. Nesne oluşturma, dosya dahil olmak üzere belirli işlemler

giriş / çıkış ve diğer modülden yapılan işlev çağrıları, tüm potansiyel kaynaklardır

Programınız düzgün çalıştığında bile istisnalar. Sağlam kod

istisnaları öngörür ve işler. C ++ dili için yerleşik destek sağlar

istisnalar atmak ve yakalamak. C ++'da istisna işlemlerini uygulamak için

deneyin , atmak ve ifadeleri yakalamak kullanın .

```
Deneyin
{
    // ...

    1 atmak;

}
yakalamak( ... )
{
    // Burada istisna tut.
}
```

Atılabilecek bir veya daha fazla ifadeyi kapatmak için bir try bloğu kullanın.

istisna. Bir atma ifadesi istisnai bir duruma işaret eder - genellikle

hata - bir try bloğunda meydana geldi . Throw deyimi geri dönüş gibi davranır.

Herhangi bir türdeki bir nesneyi atma ifadesinin işleneni olarak kullanabilirsiniz . Tipik,

Bu nesne, hata hakkında bilgi iletmek için kullanılır. İşlemek için

atılabilir istisnalar, hemen bir veya daha fazla catch blokları uygulamak

bir try bloğu izleyerek . Her catch bloğu, istisna tipini belirtir

sap.

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

3. Önceki laboratuardaki kodu 1 ve 2 görevlerine göre değiştirin.

"Hesapla" işlevi nasıl olduğunu gösterir

Belirli bir aralıktaki değeri hesaplamak

Önceki laboratuardan ifade (2.1)

(Şekil 3.1). Bu ifadenin bir tane var

Belirli bir koşul: n 'in değeri gerektiği

4'ten büyük olmak

gelişmiş içinde fonksiyon

"CheckValidParams" (Şekil 3.1).

Kullanıcı n için yanlış değer girerse ve

x , örneğin, karakter veya çift

değer, sonra program üretmelidir

istisna. Bu davalar

"checkValidInput" işlevi.

Şekil 3.1 - Kodun bir kısmı

Geliştirilen kodun ana işlevi Şekil 3.2'de.

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Şekil 3.2 - “Ana” işlevi

4. Geliştirilmiş program için birim testini uygulamak

Bu andan itibaren bir Win32 Konsol Uygulaması geliştirmiş olmalısınız.

denklemini çözmek için gerekli tüm fonksiyonlar ile. Bu işlevleri test etmek için

Her fonksiyon için ayrı ayrı birim testleri oluşturmalıdır. Bazen gerekli

Tek bir fonksiyon için birkaç ünite testi oluşturun. Tamamen test etmek amacıyla yapılır

Her fonksiyonun olası tüm giriş parametreleriyle davranışı.

Mevcut proje için birim testleri oluşturmak için Dosya → Ekle → Yeni Proje'yi seçin.

(Şekil 3.3). Yerli Birim Test Projesi'ni seçin (Şekil 3.4).

Şekil 3.3 - Birim test projesi oluşturma süreci

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU “KhPI”
Kharkiv, 2016

Şekil 3.4 - Birim test projesi oluşturma süreci

Ünite test projesini Çözüm Gezgini'nde görebilirsiniz (Şekil 3.5).

Şekil 3.5 - Çözüm Gezgini

Yazar: Melnik KV, doktora asistanı
Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"
Kharkiv, 2016

Unittest1.cpp dosyasını açın. Test edilen birim buna yerleştirilecektir. Sağ tıklayın

kaynak cpp dosyası ve "Kopya Yolu" menü öğesini seçin (Şekil 3.6). Bu yolu yapıştır

Ünite testlerinizi içerecek dosya (Şekil 3.7).

Şekil 3.6 - Test projesinde kaynak cpp dosyasını kullanma (1)

Şekil 3.7 - Test projesinde kaynak cpp dosyasının kullanılması (2)

Kodunuz için testler oluşturun. Her test kullanılarak tanımlanır.

TEST_METHOD (YourTestName) {...} .

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Geleneksel bir işlev imzası yazmanız gerekmez. İmza

TEST_METHOD makrosu tarafından oluşturulmuş. Makro bir örnek işlevi oluşturur

void döndürür. Ayrıca, hakkında bilgi döndüren statik bir işlev oluşturur

test yöntemi. Bu bilgi test gezgininin yöntemi bulmasına izin verir.

Test yöntemleri, TEST_CLASS (YourClassName) {...} kullanılarak sınıflara ayrılır .

Testler çalıştırıldığında, her test sınıfının bir örneği oluşturulur. Testi oluştur

yöntemleri isimler içinde takip etme tavır:

testedFunction_functionArguments_expectedResult .

Test amaçlı olarak seçiminize göre birkaç farklı yöntem kullanın.

Assert sınıfı. MSDN'de bu sınıfı ve yöntemlerini okuyabilirsiniz.

<https://msdn.microsoft.com/en->

[tr / kütüphane / microsoft.visualstudio.testtools.unittesting.assert.aspx](https://msdn.microsoft.com/en-tr/kütüphane/microsoft.visualstudio.testtools.unittesting.assert.aspx) .

Testlerin örneği Şekil 3.8-3.9'da gösterilmiştir.

Şekil 3.8 - Hesaplama fonksiyonu için birim testleri

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Şekil 3.9 - checkValidParams fonksiyonu için birim testleri

Test yöntemleri geliştirildiğinde,

Onları çalıştırabilirsiniz. Bunu yapmak için sen

çözüm kurmak zorundalar (Build → Build)

Çözüm / Ctrl + ÜstKrkr + B), Açık Test

Explorer açılmamışsa

otomatik olarak (Test → Windows → Test

Explorer) ve tüm testleri çalıştırın. Eğer herşey

sorunsuz git, tüm testleri göreceksin

özet alt penceresine geçirildi

(Şekil 3.10).

Şekil 3.10 - Başarılı birim testleri

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Sayfa 22

22

Ayrıca, ünite testlerinde, onları geçilmez hale getirmek için bazı değişiklikler yapmalısınız. İçin

örneğin, beklenen sonucu yanlış değere değiştirin ve testi tekrar çalıştırın

(Şekil 3.11 a). Test Gezgini'nde hangi testlerin geçmediğini göreceksiniz

(Şekil 3.11 b).

(a)

(b)

Şekil 3.11 - Hata ile birim testi

5. Çalışmanın raporunu hazırlayın

Rapor için gerekenler:

1. Rapor bölüm standartlarına göre hazırlanacaktır.

Şablon aşağıdaki yolla bulunabilir:

\\ SELENA \ Method \ Kağıt çalışması \ STVUZ.dot

2. Rapor, tüm görevlerin yerine getirilmesinin sonuçlarını içermelidir.

Laboratuvara bireysel göre göre.

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Laboratuar № 4

**Hedef: Sistem aktivitelerini planlayın. Temel deneyim ve bilgi birikimine sahip olmak
proje takip sistemi ile çalışmak**

Laboratuarın ilerlemesi.

1. Ücretsiz xp-dev.com hesabına kayıt olun

Kayıt olmak için bir e-posta hesabınız olmalıdır (aynı zamanda ücretsiz bir e-posta hesabı sağlayın). Kayıt olmak için <https://xp-dev.com/register> adresine gidin. ve yeni ücretsiz hesap oluşturun Ücretsiz hesap size 10Mb depolama alanı sağlar. depo

2. Yeni proje oluşturun

"Yeni proje oluşturun" u seçin ve için bir isim, kısaltma ve açıklama girin.

Proje konusu. İzleme Türü XPDev olmalıdır. Oluşturmak için gerekli 1

1 takım için xpdev hesabı. XpDev İçinde yeni proje oluştur

3. Projeye yeni kullanıcı ekleyin.

Davet göndermek için kullanıcı adı ve şifre seçin. Yeni kullanıcı giriş yapabilir e-posta hesabına gönderilen kimlik bilgileri. Kullanıcının rolünü seçin (tümünü inceleyin roller ilk). Yönetici veya yazar kabul edilebilir.

4. Yeni Iterasyon oluşturun.

Modern yazılım geliştirme yaklaşımları proje faaliyetlerini aşamalar (yineleme). Her yinelemede birkaç yazılım veya sistem uygulanmalıdır işlevsellikler (kullanıcı hikayeleri). Böylece her laboratuvar yeni iterasyonla başlayabilir. birkaç kullanıcı hikayesi var.

Proje takibine gidin ve "Yeni yineleme oluşturun" u seçin.

İsim: kurulum işlemini açıklayan herhangi biri

Güne Başla - bugün

5. Yeni kullanıcı hikayeleri ekleyin ve kullanıcı hikayesinin uygulanmasına görev ekleyin

6. Görev ekibinizin kullanıcılarına görev atayın

7. svn deposu oluşturun

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU "KhPI"

Kharkiv, 2016

Havuz sekmesine git

Yeni depo oluşturun seç

Havuzun adını ayarla

İlk Dizinler Oluştur düğmesini ayarlayın (bu, ek dizinleri oluşturur.

Müşteri gereksinim değişikliklerini ele almak için projenin birden çok sürümünü ele alalım)

8. svn.exe üzerinden erişilebilirlik testi (eğitmenden alabilirsiniz)

Boş izin oluşturun

Svn komutunu kullan <url> <dizine giden yol> - username = <xpdev 's

kullanıcı adı> yapmak

ilk ödeme (tüm ekip üyelerini kontrol edin)

Gövde dizininde test metni dosyası oluştur

Yeni dosya eklemek için svn komutunu <dizin dizinine ekle> kullan

Verileri depoya koymak için svn commit –m „message 'komutunu kullanın

Yerel kasadaki verileri güncellemek için svn update <dizinine dizin> komutunu kullan

9. Çalışmanın raporunu hazırlayın

Rapor için gerekenler:

1. Rapor bölüm standartlarına göre hazırlanacaktır.

Şablon aşağıdaki yolla bulunabilir:

\\SELENA \ Method \ Kağıt çalışması \ STVUZ.dot

2. Rapor, tüm görevlerin yerine getirilmesinin sonuçlarını içermelidir.

Laboratuvara bireysel göre göre.

Yazar: Melnik KV, doktora asistanı

Yazılım mühendisliği ve yönetim bilişim teknolojileri bölümü, NTU “KhPI”

Kharkiv, 2016