|  |  |  |  |
| --- | --- | --- | --- |
| İş Yerinde Çalışılan Birim: Tum sirket | | | Çalışma Tarihi: 03.07.2017 |
| Yapılan iş: Oryantasyon programı yapıldı. | | | |
| İlk gün yapılanları anlatmadan önce staja alınış sürecini anlatacağım. Şirket hem internet üzerinden, hem üniversitelerdeki afişler ile hem diğer yöntemler ile birçok öğrenciye ulaştı. Sadece kariyer.net sitesindeki ilanda 5 bin başvuru vardı. Stajda CEO Zafer Şen’den öğrendiğimize göre bu başvurular arasından önce mülakat ve internet üzerinden yazılım sınavı yapıldıktan sonra oluşturulan 300 kişi arasından staja alınacak kişiler seçiliyor.  Mülakatta okulumuzun kullandığı programlama dilleri, gördüğümüz diğer zorunlu dersler, seçmeli olarak aldığımız dersler, okul içinde ve dışında uğraştığımız programlama dilleri ve diğer konular soruldu.  İşin ilk günü tüm staj programlarına katılan kişiler ve şirketten bir çok kişi buluşuldu ve oryantasyon eğitimi yapıldı. Çeşitli şirket çalışanları çeşitli konularda konuştu, bilgi aktardı.  Mobil ve java stajları vardı ben ilk tercihim olarak java stajını yazdım. Bunun sebebinide anlatayım. 2016 yılında iki donem java dersi aldım. Okulun bittiği gün kitapçıdan Java How to Program isimli kitabı aldım. Pdf olarak gelen kısımlarıda matbaaya gidip kitap haline getirttim. O yaz boyunca java çalıştım ve çok büyük yardımı oldu. Java dilinden zevk aldığım için ilk uzmanlaşacağım konu olarak onu seçtim. (C# ve perl gibi programlama dillerini seçmeli ders olarak aldım ama Java ve C kadar sevmedim) Gerçekten öğrenmeye yönelik çalışınca çok yavaş ilerliyor kitap. Ayrıca kıtap 1600 sayfa civarında. Dolayisiyla 2017 yazindada o kitaba çalışmaya devam ettim. Staj için bazi şirketlerle konuştum ama onlar benim çok iyi bilmediğim ve çok ilgimi çekmeyen dilleri kullanıyorlardı. Dolayisı ile OBSS’te tam olarak istediğim stajı gördüm, başvurdum ve kabul edildim.    Oryantasyon’a devam edecek olursak bazi kişiler şirketlerin nasıl iş bitirici insanlar aradığı hakkında konuştu. Bu iş bitirici olma üzerinde duruldu. Yüksek egolu olmamak gerektiğinden bahsedildi.  Bazı kişiler şirketin yaptığı işler ve birlikte çalıştığı kurumlarla ilgili bilgi verdi. Şirketin ortadoğuda Atlassian servislerinin tek partneri olduğu, ne gibi sektörlerde ne gibi alanlarda çalıştıkları hakkında bilgi verildi.  Bazı kişiler iletişim konusunda konuştu. Bazı kişiler şirketin yapsı ile ilgili konuştu. Bazı kişiler iş güvenliği konusunda konuştu. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Bir iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 04-07.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Age isimli program kişinin yaşını alır, eğer kullanıcı bir yaş değeri değilde başka bir değer girmişse hata yazısını gösterip tekrar giriş ister. Doğru giriş yapıldığında mantikli bir yaş girilmediyse onunla ilgili hatayı gösterir. Eğer mantıklı aralıkta bir yaş girildiyse bu şirkette çalışmak için çok genç veya yaşlı olup olunmadığına bakılıp kullanıcıya gösterilir. Programdan çıkmak içinde -1 e basılır.  Standart input object(scanner), exception handling, structured programming gibi java özelliklerinden yararlanıldı.  Gradebook isimli program sinav notu sayisini alir ve ortalamayi hesaplar. 101 girişi programı kapatır.  Programın birince bölümü giriş değerlerini alır. Eğer değer 101 ise programın ikinci bölümüne geçilir. Eğer değer 100 den büyük veya 0 dan küçükse yanlış değer girdiniz hatası verilir. Eğer başka bir değer girildiyse bu değer sınav sonucu toplamına eklenir ve okunan sınav sayısı bir artar.  Programın ikinci kısmına geçildiğinde eğer okunan sınav sayısı sıfır ise hiçbir sınav sonucu girilmedi sonucunu alırız. Eğer sıfır değilse bize kaç sınav okunduğunu, notlar toplamının ne olduğunu ve not ortalamasının ne olduğunu söyler. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 04-07.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| GuessTheNumber programı. Birden bine kadar bir sayı rastgele olarak oluşturulur. Kullanıcı tahmin etmeye çalışır. Bilgisayar kullanıcının tahmin ettiği sayının oluşturulan rastgele sayıdan büyükmü küçükmü olduğunu söyler.  SecureRandom sınıfını kullandık. randomNumber metodunda 1 ile 1000 arasi rastgele sayımizi oluşturduk. Kullanıcı ya 0 girerek programdan çıkabilir yada sayıyı tahmin etmeye çalışabilir. Tahmin edilen sayı ile rastgele sayı karşılaştırılır. Tahmin ettiğiniz sayı rastgele sayıdan küçüktür yada büyüktür uyarısı alırsınız yada doğru sayıyı tahmin ettiniz sonucunu alırsınız.  GuessTheNumber programı ikinci versiyonunda yine 1 den 1000 e kadar bir rastgele sayı oluşturulur ama bu sefer kullanıcının tahmın sayısını sayiyoruz.  Bir adet dagha fazla değişken kullanıyoruz. Önceki versiyondaki gibi programin başında ve kullanıcı sayıyı doğru tahmin ettikten sonra rastgele sayıyı sıfırlıyoruz. Bu sefer do while yerine while “repetition structure” ‘ ini kullandik. Kullanicinin girdiği sayıyı rastgele sayı ile karşılaştırdık ama bu sefer tahmin sayısınıda arttırdık her seferinde. Doğru sayı bulunduğunda önceki versiyon gibi kazandınız mesajı verdik ve rastgele sayıyı sıfırladık. Onun dışında bu sefer eğer tahmin sayısı 10 dan az ise “ya algoritayı biliyorsunuz yada şanslısınız” dedik. 10 ise “algoritmayı biliyorsunuz” dedik. 10 dan fazla ise “daha iyisini yapabilirsiniz” dedik. Çünkü her tahminde sayıların yarısını eleyebiliyoruz eğer tam ortadaki sayıyı seçersek. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 04-07.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| ArrayReverse programı bir dizi veri yapisindaki verilerin indexlerini ters çevirir.  Dizimizi uzunluğunu söyleyerek değil ilk değerlerini vererek oluşturuyoruz. İki adet metodumuz var. displayArray metodu dizimizin içeriğini baştan sona gösterir. reverseArray metodumuz dizimizin elemanlarının pozisyonlarını ters çevirir. Program once displayArray’i cagirir sonra reverseArray’i cağırır sonra tekrar dısplayArray’i cağırır.  displayArray metodu baştan sonra tüm elemanları yazdırır. reverseArray bastan sonra tüm dizi elemanlarını baştan sonra ikinci bir diziye yazar ve bu ikinci diziyi döndürür.  TwoDArray programı belirli bölgelerdeki şehirler isimlerini kullanıcıdan alır. Daha sonra sırayla bu bölgelerdeki şehirleri sırayla listeler.  Bu programda çok-boyutlu dizi kullandık. Dizimizi bu sefer farklı bir yöntem kullanarak deklare ettik. Dizinin boyutunu içerik vererek değil direk boyut vererek belirledik. Daha sonra initializeArray ve displayArray metodlari çağırılır. InitalizeArray metodu nested repetition structure kullanarak iki boyutlu diziyi doldurur kullanıcı girişleriyle. displayArray metoduda nested repetition structure kullanarak iki boyutlu dizinin içeriğini gösterir.  Çok boyutlu dizileri kullanırken boyutları homojen olmak zorunda değil. İki boyutlu bir dizi kullanırken dizinin son boyutunu belirmek zorunda değiliz. Buda bize değişken uzunlukta dizilerden oluşan bir dizi verir. İngilizcede buna array of arrays denir. Dizimizin tuttuğu eleman bir primitive variable değil, başka bir dizidir.  ArrayOfArrays programı ilk boyutu 5 olan ikinci boyutu değişken olan bir dizidir. İlk boyut 5 olarak belirlenerek dizi deklare edildikten sonra dizinin elemanları(tek boyutlu diziler) atanır. Son olarak da iki boyutu dizimizin içeriği yazdırılır. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 04-07.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Pen isimli program Circle ve Rectangle programlarınıda kullanır. Pen programı çeşitli geometrik şekilleri işler.  Başka programları kullanmak için import ile o sınıflarıda eklememiz gerekir. Pen sınıfının drawRectangle ve drawCircle, changeColorRectangle, changeColorCircle metodları vardır.  drawRectangle bir dikdörtgenin alanını hesaplar verilen uzunluklarla.  drawCircle bir çemberin alanını hesaplar verilen uzunluklarla.  changeColorRectangle dikdörtgen objesinin renk instance variable’ını değiştirir.  changeColorRectangle dikdörtgen objesinin renk instance variable’ını değiştirir.  Pen sınıfı main metodunda bir Pen, bir dikdörtgen, bir çember yaratır. Daha sonra drawRectangle ve drawCircle metodları çağırılır, renkleri yazdırılır. Bu dikdörtgen ve çember objelerinin setColor metodları istediğimiz renk argümanıyla çağırılır ve en sonunda tekrar renkleri yazdırılır ve değişiklik gözlemlenir.  Circle sınıfında radius ve color instance variable’ları vardır. Constructor, setRadius, setColor, getRadius, getColor metodlari vardir.  Rectangle sinifinda width, height, color instance variable’ları vardır. Constructor ve instance variable’lar için set ve get metodları vardır. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 10-13.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| AirlineReservationsSystem is a ticket buying software for a plane of 10 seats. 5 First class and 5 economy. Asks customer what he wants. If they are out of that seat, offers other seats.  We as user what kinda ticket he wants;  1 invokes sellFirstClass method,  2 invokes sellEconomy method,  3 invokes exit from program,  Other values invoke error message.  If there is no first class seat left program says when the next flight is leaving.  If there is no economy seat is left the program says would you like to buy a first class seat. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 10-13.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| FileIO programı. Kullanıcıdan 10 çift ve -1000 ile 1000 arasında sayı alır ve bu sayıları bır dosyaya yazdırır. Daha sonra 10 adet 0 ile 100 arasında sayıyı rastgele oluşturup tek ise yazdır.  Java.io ve SecureRandom sınıflarını kullandık.  Kullanıcıdan önce 10 adet giriş aldık, yanlış değer girilirse belirttik ve tekrar giriş istedik. Doğru giriş yapıldığında bu girişleri dosyaya yazdık.  Do while kullanarak 10 kere rastgele sayı ürettik belirtilen aralıkta. Ve bu sayılarıda dosyaya yazdık.  randomNumber = randomGenerator.nextInt(1000);  Üstteki kod 0 dan 1000 e kadar tüm tam sayılardan rastgele sayı seçer.  write.println(randomNumber);  Bu kod ise değeri dosyaya yazar. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 10-13.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| DiceRolling programıç Crabs isimli oyunu 1 milyon kere oynar. İki zar atıp toplamlarına bakar. Bir dizi kullanarak hangi zar toplamının kaç gere geldiğini gözlemler.  for (int counter = 0; counter < 1\_000\_000; counter++)  {  sumOfTwoDice = rollDice();    array[sumOfTwoDice]++;  }  Bu işlemden sonra kullanıcıya gösterir sonuçları.  0: 0  1: 0  2: 27,991  3: 55,817  4: 83,051  5: 111,451  6: 139,118  7: 166,140  8: 138,105  9: 111,205  10: 83,491  11: 55,686  12: 27,945 | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 10-13.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| WordLetterHistogram program.  Konsoldan kelimeler okunur. Ve hangi kelime kaç kere girildi sayılır. E harfi girildiğinde program durur.  Daha önce girilen farklı sayı sayısı kadar elemana bakar. Eğer onlardan biri ise dizinin o indexindeki sayıyı arttırır. Değilse yeni bir eleman 0 dan 1 e yükseltilirç (yeni karşılaşılan bir kelime)  while (counter <= wordCounter2)  {  if (s.equals(wordHistogram[counter]))  {  wordCounter[counter]++;  wordCounter2++;  En sonda ise kelimeler ve frekansları yazdırılır.  t: 1  tt: 1  ttt: 1 | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 14.07.2017 |
| Yapılan iş: Stajlerlerin javada en kötü olan yarısının elenmesi ve ankara ile istanbuldaki java staj programlarının istanbulda birleştirilmesi için sınav yapıldı. | | | |
| Sinav şu şekildeydi;  C:\Users\Anil\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2-5.png | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Bir iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 24-28.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Servlet kullanarak her açıldığında kaç kere açıldığını gösteren web sayfası yapıyoruz.  Bilgisayarınızdaki cache’i temizlediğinizde sayı tekrar bire iner. Yada cache tutmayan incognito mode ile girerseniz yine sıfırlar.  Servlet sınıfınızın counter instance variable’ı vardır.  private int counter; // Default value is 0.  Bu değer web server’ında durur. Bir http request geldiğinde verilen cevabın bizim yazdığımız kısmı şudur.  counter++;  out.println("Since loading, this servlet has been accessed " + counter + " times");  Client kısmına sadece print metodunun sonucu döndürülür. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 24-28.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Biz web server’a bazı verileri client side’a gönder dediğimizde direk gönderilmez. Bellekte tutulur ve belli bir büyüklüğe ulaşana kadar bekletilir. Bekletilmemesi için  out.flush();  özelliğini kullanırız.  out.write("test");  out.write("test2");  out.write("test3");  out.println("<br>");  out.flush();  Bu kod ile bazı verileri direk kullanıcıya yollarız. Bunu göstermek için bu verilerden sonra gönderilecek verileri geçici olarak bekletebiliriz. Bunu yapmak için  Thread.sleep();  özelliği kullanılır.  Daha sonra tekrar bilgi yollarız client’a ilk gönderdiklerimizin direk gittiğini görmek için. İkinci set bilgiler ilk set bilgilerden 5 saniye sonra ulaşır.  out.write("test4");  out.write("test5");  out.write("test6"); | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 24-28.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Önceki programlarda Servlet denilen Java teknolojisini\API’ını kullandık. Bu teknoloji java kullanarak http request ve respondlarını kullanmamızı sağlıyor. Biz okulda Servlet ve JSP yi görmedik JSF yi gördük dolayısı ile bu konular bana yeni konulardı ama öğrenmem çok iyi oldu. Şimdi göstereceğim programda ise JSP kullanıldı. JSP ve JSF programları program çalışırken alt yapı olarak Servlet leri kullanır. JSP bize dinamik internet sayfası yapmamızı sağlar.  Bu programda index.jsp sayfamız iki adet sayı alır kullanıcıdan. Eğer doğru girişler yapılırsa process.jsp sayfasına gidilir ve bu iki sayının bölümünün sonucu gösterilir. Eğer yanlış giriş yapılırsa mesela sayı yerine harf girilirse process.jsp belgesindeki java kodunda hata olur ve bu hata error.jsp sayfasına gidilmesine ve bu belgenin çalıştırılmasına sebep olur.  <form action="process.jsp">  No1:<input type="text" name="n1" /><br/><br/>  No2:<input type="text" name="n2" /><br/><br/>  <input type="submit" value="divide"/>  </form>  Yukardaki kodda doldurduğumuz form bilgileri butona bastığımızda process.jsp belgesine gönderilir.  <%@ page errorPage="error.jsp" %>  Yukaridaki kodda process.jsp nin başındaki exception handling parçasını görüyoruz.  <%@ page isErrorPage="true" %>  Yukaridaki kodda error.jsp nin error sayfalarindan birisi olduğunu gösteriyoruz ve bir isim veriyoruzki hata alınınca birden çok hata sayfası çağırabilelim. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 24-28.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Önceki birkaç programda Servlet ve JSP teknolojilerini kullandık. Şimdiki program MultiThreading teknolojisini kullanıyor. Bazı işlemlerin aynı anda yapılmasını bazı işlemlerin sırayla yapılmasını kontrol etmek için threadleri bazı işlemler için kilitleyebiliyoruz.  Mesela bir kişi hesabında 1000 dolar varken iki kere 600 dolar çekebilir eğer iki işlem aynı anda yapılırsa veya yüksek ping gibi durumlar varsa.  String errorString = "";    if("withdraw".equals(request.getParameter("action")))  {  errorString = withdraw(Integer.parseInt(request.getParameter("amount")));  }  else if ("deposit".equals(request.getParameter("action")))  {  errorString = deposit(Integer.parseInt(request.getParameter("amount")));  }    out.print(errorString);  out.print("Balance is: " + balance);  Kullanıcı yapacağı işlemi ve miktarı seçer. Gerekli metod gerekli miktar ile çağırılır. Gerekli sonuç döndürülür ve ekrana basılır.  synchronized(lock) {  try  {  Thread.sleep(3000);  } catch (InterruptedException ex) {  Logger.getLogger(NewServlet.class.getName()).log(Level.SEVERE, null, ex);  }    balance -= withdrawAmount; } | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 24-28.07.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
|  | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 31.07.2017-04.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Önceki programlarda Servlet http API’ını, JSP dynamic web page API’ını, Multithreading API’ını tek başlarına veya birlikte kullandık. Bu programda javanın database kabiliyetlerinide ekliyoruz.  Database programı iki adet html sayfasına, bir adet servlet’e ve bir database’e sahip.  index.html sayfası bize iki seçenek veriyor. Ya veritabanına bakabiliriz yada ekleme yapabiliriz. Veritabanını göster dersek servlet’i çağırır ve veri tabanı içeriğini gösterir. Veritabanına ekleme yap seçeneğini seçersek insertDb.html belgesi çağırılır. Ve o belgeye girdiğimiz bilgilerle servlet çağırılır. Servlet aldığı aksiyon seçeneğine göre işlem yapar. Eğer insert aksiyonu aldıysa onu yapar, view aksiyonu aldıysa o işlemi yapar.  <form action="insertDb.html">  <input type="submit" value="Insert to Database"/>  </form>  <br>  <form action="NewServlet">  <input type="hidden" value="view" name="operation">  <input type="submit" value="View Database."/>  </form>  <form action="NewServlet">  Name:<input type="text" name="name" /><br/><br/>  Tel number:<input type="text" name="tel\_number" /><br/><br/>  <input type="hidden" value="insert" name="operation">  <input type="submit" value="Add to database"/>  </form>  Servlet kodunuda çıktı olarak göstereyim çünkü büyük. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 31.07.2017-04.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Sıradaki programımız cookie özelliğini kullanarak kullanıcı tarafında bilgi saklayan bir program.  index.html sayfası üzerinden cookie yaratılır.  <form action="createCookieServlet">  Name: <input type="text" name="name" /><br/><br/>  <input type="submit" value="Create cookie"/>  </form>  Aşağıdaki kod ile kullanıcıdan parametre olarak aldığımız veriyi name isimli cookie’ye değer olarak veririz. Bu cookie yaratılır ve kullanıcıya http response’unda gönderilir.  out.printf("Welcome %s", request.getParameter("name"));    Cookie myCookie1 = new Cookie("name", request.getParameter("name"));  response.addCookie(myCookie1);  Aşağıdaki kod ile ise kullanıcının cookie’leri alınıp yazdırılır.  try (PrintWriter out = response.getWriter())  {  Cookie ck[]=request.getCookies();  for(int i=0;i<ck.length;i++)  {  out.print(ck[i].getName() + " " + ck[i].getValue() + "<br>");  }  } | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 31.07.2017-04.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Bu programda Session özelliğini kullanıyoruz. Session bilgiyi client side’da değil, server side’da tutar.  Önce session alınır. Sonra session içindeki count bilgisi alınır. Ayrıca bir lokal değişkenimiz vardır. Her seferinde bu lokal değişkene 1 değeri atanır. Eğer sessionda tutulan count değişkeni bir değere sahip değilse bu lokal değişkendeki 1 döndürülür, eğer bir değere sahipse o değer lokal değişkene atanır ve lokal değişkendeki değer döndürülür. Daha sonra sessiondaki count değeri bir arttırılır.  Kısaca bu sayfaya kaç kere girdiğimizi gösterir.  HttpSession session = request.getSession(true); /\* Create a session only if  the JSession cookie of client has no correspondant in session list in server \*/    Object countObject = session.getAttribute("count"); /\* Cookies hold information  as String-String(key-value), sessions hold information as String-Object(key-value).  we get the value of the count attribute in session. if it doesnt exist yet null is  returned. \*/    int count = 1; /\* We set count local variable to 1 every time \*/    if(countObject instanceof Integer){ /\* If this is not the first time we entered  then assign the current calue of count from session to local variable. if it is  the first time we enter then dont do this use the 1 we assigned above \*/  count = (Integer) countObject;  }    out.print(count);    session.setAttribute("count", ++count); | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 31.07.2017-04.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Bu programda javanın filter özelliği kullanıldı.  index.html sayfamız bir form bilgisi alır ve AdminServlet’e gönderir.  <form action="AdminServlet"><br>  Name:<input type="text" name="name"/><br/><br>  Password:<input type="password" name="password"/><br/><br>  <input type="submit">  </form>  Admin servlet’e gidicek bilgiler in belli bir filter dan geçmesi için web.xml dosyasını kullanılırız.  <filter-mapping>  <filter-name>MyFilter</filter-name>  <url-pattern>/AdminServlet</url-pattern>  </filter-mapping>  Bu filter eğer şifre “admin” ise bilgiyi gitmesi gereken yere gönderir. Eğer geçersiz bir bilgi ise indexError.html safyasına yönlendiriliriz.  String password=req.getParameter("password");  if (password.equals("admin")) {  chain.doFilter(req, resp);//sends request to next resource  }  else {  RequestDispatcher rd = req.getRequestDispatcher("indexError.html");  rd.forward(req, resp);  } | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Dört iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 31.07.2017-04.08.2017 |
| Yapılan iş: İkinci bir sınav yapıldı. | | | |
| Bu sınavda diğer sınav gibi testti. Bu sefer birinci olana ipad verildi. Ben ikinci oldum.  Soruların bir kısmı şu şekildeydi; | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Bir iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 07-11.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| StudyPlanApp çeşitli inputlara çeşitli şekilde cevap veren bir programdır.  Email, favori dil, ikinci favori dil, tecrübe, Gosling’i tanıma gibi girişler vardır.  Girdiğimiz bilgiler StudyPlanBean’e gider. Action metodu çalışır.  <h:commandButton value="Submit" action="#{studyPlanBean.action()}"/>  Eğer kişiyi tanıyorsak bir sayfa tanımıyorsak başka bir sayfa döndürülür.  if (isKnowsJamesGosling())  {  return "jamesGosling"; // goes to servlet  }  else    /\* If high experience is selected recommend he should learn all languages in list  except the ones he already knows. If not recommend just 2 languages outside the  ones he knows.  \*/    return "studyPlan"; | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 07-11.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Search engine programı kullanicidan bir arama motoru ve aranacak cümleyi ister ve sonucu gösterir.  Kullanicidan alinan bilgileri editler ve arama cümleleri oluşturulur. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 07-11.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| StudyPlanApp programinin manual validation içeren versiyonudur. StudyPlanApp çeşitli inputlara çeşitli şekilde cevap veren bir programdır.  Email, favori dil, ikinci favori dil, tecrübe, Gosling’i tanıma gibi girişler vardır.  Girdiğimiz bilgiler StudyPlanBean’e gider. Action metodu çalışır. Ama bu sefer girilen değerler manuel olarak kontrol edilir.  C:\Users\Anil\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5-3.png | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 07-11.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| StudyPlanApp programinin otomatik validation içeren versiyonudur. StudyPlanApp çeşitli inputlara çeşitli şekilde cevap veren bir programdır.  Email, favori dil, ikinci favori dil, tecrübe, Gosling’i tanıma gibi girişler vardır.  Girdiğimiz bilgiler StudyPlanBean’e gider. Action metodu çalışır. Ama bu sefer girilen değerler otomatik olarak kontrol edilir. Bunun içinde validateRegex pattern lar kullanılır. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 07-11.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| StudyPlanApp programinin user created validation içeren versiyonudur. StudyPlanApp çeşitli inputlara çeşitli şekilde cevap veren bir programdır.  Email, favori dil, ikinci favori dil, tecrübe, Gosling’i tanıma gibi girişler vardır.  Girdiğimiz bilgiler StudyPlanBean’e gider. Action metodu çalışır. Ama bu sefer girilen değerler bizim yarattığımız validator ler kullanılarak kontrol edilir. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 14-18.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Bu programida JPA sınıfları kullanarak veritabanına erişilen bir yazılım yaptık. Aşağıdada gördüğünüz gibi normal jdbc kullanmıyoruz, kullandığımız tablonun sınıfları aracılığı ile veritabanına erişiyoruz. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 14-18.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Bu programda üç sınıfımız var. Class 3 class 2 yi çağırır. 2 1 i çağırır. 1 bir exception yaratır. 2 bunu handle etmez. 3 yakalar ve belli bir exception türü haline getirir ve sonuçları bir dosyaya yazar. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 14-18.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Bu program çeşili custom annotation lar yaratır ve hangi sınıflarda kullanıldığını gözlemler. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 14-18.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Bu program kayit, giriş gibi fonksiyonları olan bir web sayfasıdır. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |
| İş Yerinde Çalışılan Birim: Yazılım | | | Çalışma Tarihi: 14-18.08.2017 |
| Yapılan iş: Java ile programlar yazıldı. | | | |
| Son programimiz ise sayısal metodlarla ilgili bir program. Pi sayısının değerini bulmak için kullanılan sonsuza kadar gider bir dizi vardır. 4 -4/3 + 4/5 şeklinde sonsuza kadar gider.  Biz bu sonsuza kadar giden seriyi 200 bin kere itere ederek ve her defasında bir önceki sayıya olan fark yüzdemize bakarak bir sayısal metod uygulaması yapıyoruz. | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| Beş iş günü |  |  | |
|  |  |  |  |