



T.C.
KIRIKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
STAJ DOSYASI

Ö Ğ R E N C İ N İ N

Adı Soyadı	Muhammet Ali KÖKER
Okul No	120255013
Bölümü	Bilgisayar Mühendisliği
Sınıfı	3. Sınıf

T.C.
KIRIKKALE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
STAJ DOSYASI



Ö Ğ R E N C İ N İ N	
Adı Soyadı	Muhammet Ali KÖKER
Okul No	120255013
Bölümü	Bilgisayar Mühendisliği
Sınıfı	3. Sınıf

S T A J Y A P I L A N Y E R İ N	
Adı	Baykar Makina
Adresi	İkitelli OSB Metal İş Sanayi Sitesi 15. Blok Başakşehir / İstanbul
Telefonu	0 212 671 21 15
Faks Numarası	0 212 671 10 31

56 sayfa ve eklerden oluşan bu staj dosyası yukarıda adı geçen öğrenciniz tarafından 27 / 07 / 2015 - 12 / 09 / 2015 tarihleri arasında kurumumuzda yapmış olduğu staj çalışması sonucunda hazırlanmıştır.

12 / 09 / 2015

İşyeri İlgili ve Yetkilisinin

Adı, Soyadı :

Ünvanı :

İMZA-MÜHÜR

STAJ DEVAM ÇİZELGESİ

1. HAFTA: 27 / 07 / 2015 - 01 / 08 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	Programlama Standartlarının Araştırılması	1
SALI	Yazılım Geliştirme	Yazılım Birimi	C ile Standart Kontrol Yazılımı Geliştirme	2
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	C ile Kontrol Fonksiyonlarının Yazılması	3
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	C ile Kontrol Fonksiyonlarının Tamamlanması	4
CUMA	Yazılım Geliştirme	Yazılım Birimi	C ile Dosya Fonksiyonlarının Yazılması	5
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	C ile Hata Kontrol Fonksiyonunun Yazılması	6

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

2. HAFTA: 03 / 08 / 2015 - 08 / 08 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Standart Kontrol Yazılımı Geliştirme	7
SALI	Yazılım Geliştirme	Yazılım Birimi	C# ile Liste Fonksiyonlarının Yazılması	8
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	C# ile Standart Kontrol Yazılımı Düzenlemesi	9
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	C# ile Temel Fonksiyonların Yazılması	10
CUMA	Yazılım Geliştirme	Yazılım Birimi	C# ile Ayar Okuma Fonksiyonunun Yazılması	11
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Ayar Yazma Fonksiyonlarının Yazılması	12

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

3. HAFTA: 10 / 08 / 2015 - 15 / 08 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Filtre Okuma Fonksiyonunun Yazılması	13
SALI	Yazılım Geliştirme	Yazılım Birimi	C# ile Parametre Fonksiyonunun Yazılması	14
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	C# ile Form Fonksiyonlarının Yazılması	15
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	C# ile Form Fonksiyonlarının Tamamlanması	16
CUMA	Yazılım Geliştirme	Yazılım Birimi	C# ile Arayüz Fonksiyonlarının Yazılması	17
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Seçim Fonksiyonlarının Yazılması	18

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

STAJ DEVAM ÇİZELGESİ

4. HAFTA: 17 / 08 / 2015 - 22 / 08 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Dosya Fonksiyonlarının Yazılması	19
SALI	Yazılım Geliştirme	Yazılım Birimi	C# ile Kontrol Fonksiyonlarının Yazılması	20
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	C# ile Event Fonksiyonlarının Yazılması	21
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	C# ile Event Fonksiyonlarının Yazılması	22
CUMA	Yazılım Geliştirme	Yazılım Birimi	C# ile Event Fonksiyonlarının Yazılması	23
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Event Fonksiyonlarının Tamamlanması	24

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

5. HAFTA: 24 / 08 / 2015 - 29 / 08 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Filtreler Sınıfının Yazılması	25
SALI	Yazılım Geliştirme	Yazılım Birimi	C# ile Filtreler Sınıfının Tamamlanması	26
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	C# ile 1. Kontrol Sınıfının Yazılması	27
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	C# ile 1. Kontrol Sınıfının Yazılması	28
CUMA	Yazılım Geliştirme	Yazılım Birimi	C# ile 1. Kontrol Sınıfının Yazılması	29
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile 1. Kontrol Sınıfının Tamamlanması	30

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

6. HAFTA: 31 / 08 / 2015 - 05 / 09 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile 2. Kontrol Sınıfının Yazılması	31
SALI	Yazılım Geliştirme	Yazılım Birimi	C# ile 3. Kontrol Sınıfının Yazılması	32
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	C# ile Process Sınıfının Yazılması	33
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	C# ile Process Sınıfının Tamamlanması	34
CUMA	Yazılım Geliştirme	Yazılım Birimi	C# ile Thread Sınıfının Yazılması	35
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Thread Sınıfının Tamamlanması	36

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

STAJ DEVAM ÇİZELGESİ

7. HAFTA: 07 / 09 / 2015 - 12 / 09 / 2015

GÜN	STAJ ALANI	ÇALIŞILAN KISIM	YAPILAN İŞ	SAYFA
PAZARTESİ	Yazılım Geliştirme	Yazılım Birimi	C# ile Sıralama Fonksiyonlarının Yazılması	37
SALI	Yazılım Geliştirme	Yazılım Birimi	C# ile Main() Fonksiyonun Yazılması	38
ÇARŞAMBA	Yazılım Geliştirme	Yazılım Birimi	Java ile Veri Tabanı Yazılımı Geliştirme	39
PERŞEMBE	Yazılım Geliştirme	Yazılım Birimi	Java ile Veri Tabanı Yazılımı Düzenlemesi	40
CUMA	Yazılım Geliştirme	Yazılım Birimi	Java ile Veri Tabanı Yazılımı Düzenlemesi	41
CUMARTESİ	Yazılım Geliştirme	Yazılım Birimi	Java ile Veri Tabanı Yazılımı Düzenlemesi	42

KISIM SORUMLUSUNUN ADI, SOYADI VE ÜNVANI:

ONAY

STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	İçindekiler	SAYFA NO	I
İçindekiler			I
Staj Yeri Hakkında Bilgiler			III
Yasal Uyarı			IV
Giriş			V
Programlama Standartlarının Araştırılması			1
C ile Standart Kontrol Yazılımı Geliştirme			2
C ile Kontrol Fonksiyonlarının Yazılması			3
C ile Kontrol Fonksiyonlarının Tamamlanması			4
C ile Dosya Fonksiyonlarının Yazılması			5
C ile Hata Kontrol Fonksiyonunun Yazılması			6
C# ile Standart Kontrol Yazılımı Geliştirme			7
C# ile Liste Fonksiyonlarının Yazılması			8
C# ile Standart Kontrol Yazılımı Düzenlemesi			9
C# ile Temel Fonksiyonların Yazılması			10
C# ile Ayar Okuma Fonksiyonunun Yazılması			11
C# ile Ayar Yazma Fonksiyonlarının Yazılması			12
C# ile Filtre Okuma Fonksiyonunun Yazılması			13
C# ile Parametre Fonksiyonunun Yazılması			14
C# ile Form Fonksiyonlarının Yazılması			15
C# ile Form Fonksiyonlarının Tamamlanması			16
C# ile Arayüz Fonksiyonlarının Yazılması			17
C# ile Seçim Fonksiyonlarının Yazılması			18
C# ile Dosya Fonksiyonlarının Yazılması			19
C# ile Kontrol Fonksiyonlarının Yazılması			20
C# ile Event Fonksiyonlarının Yazılması			21
C# ile Event Fonksiyonlarının Yazılması			22
C# ile Event Fonksiyonlarının Yazılması			23
C# ile Event Fonksiyonlarının Tamamlanması			24
KISIM SORUMLUSUNUN ADI, SOYADI		12 / 09 / 2015 (ONAY)	

Not: Bu sahifeden gerektiği kadar çoğaltılacaktır.


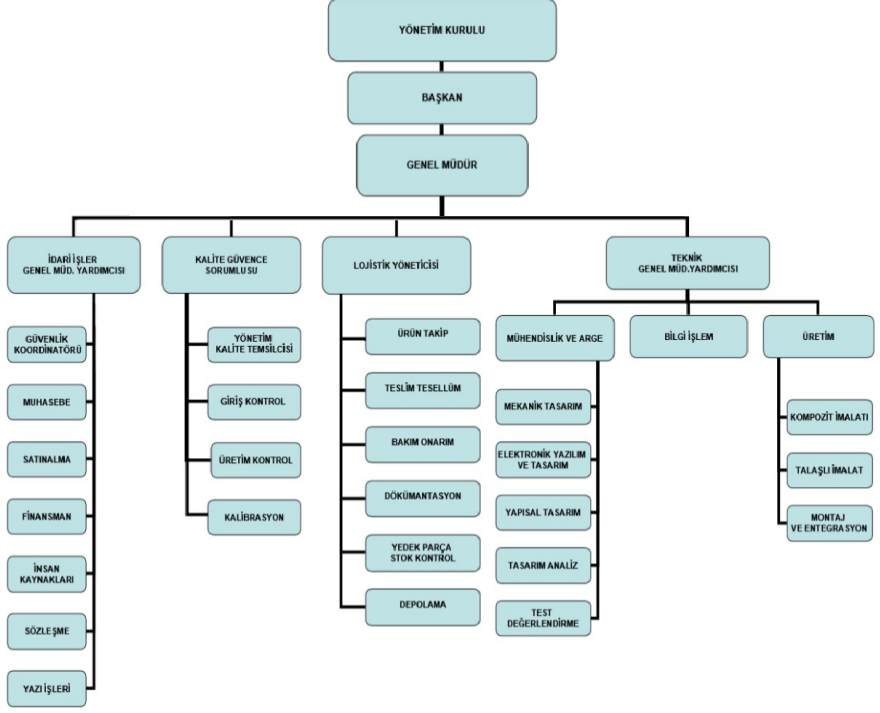
STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	İçindekiler	SAYFA NO	II
<p>C# ile Filtreler Sınıfının Yazılması25</p> <p>C# ile Filtreler Sınıfının Tamamlanması26</p> <p>C# ile 1. Kontrol Sınıfının Yazılması27</p> <p>C# ile 1. Kontrol Sınıfının Yazılması28</p> <p>C# ile 1. Kontrol Sınıfının Yazılması29</p> <p>C# ile 1. Kontrol Sınıfının Tamamlanması30</p> <p>C# ile 2. Kontrol Sınıfının Yazılması31</p> <p>C# ile 3. Kontrol Sınıfının Yazılması32</p> <p>C# ile Process Sınıfının Yazılması33</p> <p>C# ile Process Sınıfının Tamamlanması34</p> <p>C# ile Thread Sınıfının Yazılması35</p> <p>C# ile Thread Sınıfının Tamamlanması36</p> <p>C# ile Sıralama Fonksiyonlarının Yazılması37</p> <p>C# ile Main() Fonksiyonunun Yazılması38</p> <p>Java ile Veri Tabanı Yazılımı Geliştirme39</p> <p>Java ile Veri Tabanı Yazılımı Düzenlemesi40</p> <p>Java ile Veri Tabanı Yazılımı Düzenlemesi41</p> <p>Java ile Veri Tabanı Yazılımı Düzenlemesi42</p> <p>Sonuç ve Ekler43</p>			
KISIM SORUMLUSUNUN ADI, SOYADI		12 / 09 / 2015 (ONAY)	

Not: Bu sahifeden gerektiği kadar çoğaltılacaktır.

STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	Staj Yeri Hakkında Bilgiler	SAYFA NO	III

Adı	
Yeri	İkitelli OSB, Metal İş Sanayi Sitesi 15. Blok Başakşehir / İstanbul
Temel Çalışma Konusu	İnsansız Hava Araçları ve Aviyonik Sistemler
İdari Organizasyon Şeması	 <pre> graph TD YK[YÖNETİM KURULU] --> B[BAŞKAN] B --> GM[GENEL MÜDÜR] GM --> İY[İDARİ İŞLER GENEL MÜD. YARDIMCISI] GM --> KG[KALİTE GÜVENCE SORUMLUSU] GM --> LY[LOJİSTİK YÖNETİCİSİ] GM --> TY[TEKNİK GENEL MÜD. YARDIMCISI] İY --> İY1[GÜVENLİK KOORDİNATÖRÜ] İY --> İY2[MUHASEBE] İY --> İY3[SATINALMA] İY --> İY4[FİNANSMAN] İY --> İY5[İNSAN KAYNAKLARI] İY --> İY6[SÖZLEŞME] İY --> İY7[YAZI İŞLERİ] KG --> KG1[YÖNETİM KALİTE TEMSİLCİSİ] KG --> KG2[GİRİŞ KONTROL] KG --> KG3[ÜRETİM KONTROL] KG --> KG4[KALİBRASYON] LY --> LY1[ÜRÜN TAKİP] LY --> LY2[TESLİM TESELLÜM] LY --> LY3[BAKIM ONARIM] LY --> LY4[DÖKÜMANTASYON] LY --> LY5[YEDEK PARÇA STOK KONTROL] LY --> LY6[DEPOLAMA] TY --> TY1[MÜHENDİSLİK VE ARGE] TY --> TY2[BİLGİ İŞLEM] TY --> TY3[ÜRETİM] TY1 --> TY1_1[MEKANİK TASARIM] TY1 --> TY1_2[ELEKTRONİK YAZILIM VE TASARIM] TY1 --> TY1_3[YAPISAL TASARIM] TY1 --> TY1_4[TASARIM ANALİZ] TY1 --> TY1_5[TEST DEĞERLENDİRME] TY3 --> TY3_1[KOMPOZİT İMALATI] TY3 --> TY3_2[TALAŞLI İMALAT] TY3 --> TY3_3[MONTAJ VE ENTEGRASYON] </pre>
Çalışan Mühendislerin Kuruluş İçindeki Fonksiyonları	Şirket bünyesinde bilgisayar, elektronik, endüstri, kontrol, makina, mekatronik, uçak ve uzay bilimleri alanlarından mühendisler insansız hava araçları konusunda araştırma ve geliştirme yapmaktadır.
Kısa Tarihçesi	<p>Baykar Makina 1984'te Makina Yüksek Mühendisi Özdemir Bayraktar tarafından otomotiv endüstrisine yedek parça üretimi için kurulmuştur. 2000 yılına kadar bu alanda çalışmalarını sürdüren Baykar, 2001'de yapılan yatırımlar çerçevesinde milli insansız hava aracı ve aviyonik sistem üretimine başlamıştır.</p> <p>2004'te Bayraktar Mini İHA, 2006'da Malazgirt Döner Kanat İHA, 2007'de Bayraktar - Çaldıran İHA, 2012'de Bayraktar Taktik İHA isimli hava araçlarının üretimini gerçekleştirmiştir.</p> <p>Günümüzde Baykar bu alanda arge faaliyetlerini sürdürmekte ve Türk Silahlı Kuvvetleri envanteri için İHA üretimine devam etmektedir.</p>

KISIM SORUMLUSUNUN ADI, SOYADI

27 / 07 / 2015

(ONAY)

STAJ DOSYASI YAPRAĐI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	Yasal Uyarı	SAYFA NO	IV
<p>Türkiye Cumhuriyeti 5201 Sayılı Harp Araç ve Gereçleri ile Silah, Mühimmat ve Patlayıcı Madde Üreten Sanayi Kuruluşlarının Denetimi Hakkında Kanun; 5202 Sayılı Savunma Sanayii GüvenliĐi Kanunu; Harp Araç ve Gereçleri ile Silah, Mühimmat ve Patlayıcı Madde Üreten Sanayi Kuruluşlarının Denetimi Hakkında Yönetmelik; Savunma Sanayii GüvenliĐi YönetmeliĐi ve Milli Savunma Bakanlığı Savunma Sanayi GüvenliĐi Yönergesi'nce</p> <ul style="list-style-type: none">- Gizlilik dereceli bilgi ve belgeler, uygun gizlilik dereceli Kişi Güvenlik Belgesi ve Tesis Güvenlik Belgesi bulunmayan kişilere açıklanamaz ve verilemez.- Gizlilik dereceli bilgi, belge ve malzeme, üçüncü kişilere, kuruluşlara veya bunların temsilcilerine açıklanamaz.- Gizlilik dereceli bilgi, Kişi Güvenlik Belgesi'ne sahip olup bilmesi gereken kişiler dışındaki diĐer şahıslarla paylaşılabilir.- Gizlilik dereceli bilgi, belge veya malzeme, uygun güvenlik önlemleri alınmış ortamlarda muhafaza edilir veya kullanılabilir.- Gizlilik dereceli bilgiler faks veya elektronik posta ile gönderilemez.- Gizlilik derecesine sahip belgeler Baykar Makina Sanayi ve Ticaret A. Ş.'nin yazılı izni olmadan yayınlanamaz ve çoĐaltılamaz. İlgili kanun ve yönergelerde belirtildiĐi gibi üçüncü kişilere veya kuruluşlara açıklanamaz. Aksi durumlarda ilgili kişi veya kuruluş hakkında kanun maddelerinde öngörülen ceza hükümleri uygulanır. <p>İşbu belgede sadece Baykar Makina Sanayi ve Ticaret A. Ş. yetkililerince yayınlanmasına izin verilen bilgiler paylaşılabilir, söz konusu belge gizlilik dereceli bilgi içeremez.</p>			
KISIM SORUMLUSUNUN ADI, SOYADI		27 / 07 / 2015 (ONAY)	

Not: Bu sahifeden gerektiĐi kadar çoĐaltılacaktır.

STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	Giriş	SAYFA NO	V

Staj, Baykar Makina Yazılım Birimi'nde yapıldı. Staj süresince gömülü yazılım ve arayüz yazılımı alanlarında çalışılacağı, ancak bu alanlardaki çalışmaların gizlilik gerekçesiyle staj dosyasına yazılamayacağı bilgisi edinildi. Bunun üzerine ilgili mühendis ile görüşülerek “Standart Kontrol Yazılımı” projesi alındı. Burada söz konusu yazılımın ana projeden arda kalan zamanlarda geliştirileceği ve ilgili çalışmanın staj dosyasına yazılabileceği öğrenildi.

Daha sonra şirkette üretilen milli insansız hava araçları incelendi. İlk olarak Bayraktar Mini İHA incelendi. Söz konusu İHA'nın 2 m kanat açıklığında ve 1.2 m boyunda olduğu, 15 km menzil ve 55 km / saat seyir hızı özelliklerine sahip olduğu öğrenildi. Ayrıca 12000 feet irtifaya çıkabildiği ve paraşütle iniş yapabildiği bilgisi edinildi.



Resim I: Bayraktar Mini İHA



Resim II: Malazgirt Döner Kanat İHA

Malazgirt Döner Kanat İHA'nın ise 20 km menzil ve 60 km / saat seyir hızı özelliklerine olduğu öğrenildi. Ayrıca söz konusu İHA'nın 12000 feet irtifaya çıkabildiği; tamamen otomatik iniş, kalkış, seyir yeteneklerine sahip olduğu bilgisi edinildi.

Bayraktar - Çaldıran İHA'nın ise 9 m kanat açıklığı ve 450 kg kalkış ağırlığı özelliklerinin olduğu; 18000 feet irtifaya çıkabildiği öğrenildi.



Resim III: Bayraktar - Çaldıran İHA



Resim IV: Bayraktar Taktik İHA

Son olarak Bayraktar Taktik İHA'nın 6,5 m uzunluğunda ve 12 m kanat açıklığında olduğu; 24000 feet irtifaya çıkabildiği ve 30 saat havada kalabildiği öğrenildi.

KISIM SORUMLUSUNUN ADI, SOYADI

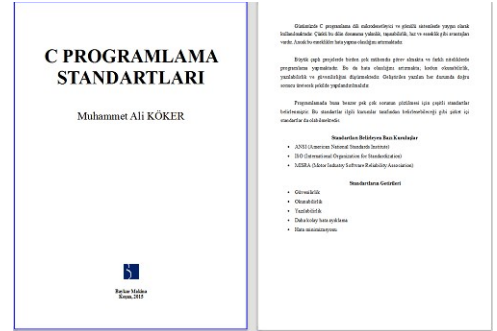
27 / 07 / 2015

(ONAY)

STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	Programlama Standartlarının Araştırılması	SAYFA NO	1

Stajın ilk gününde programlama standartları araştırıldı. Savunma endüstrisindeki şirketlerde geliştirilen yazılımlarda ANSI (American National Standards Institute), ISO (International Organization for Standardization) ve MISRA (Motor Industry Software Reliability Association) gibi kurumlar tarafından belirlenen standartların esas alındığı öğrenildi.



Resim 1: Hazırlanan belge

Bu aşamada standartlar hakkında bir belge hazırlandı. Standartların kodun güvenilirlik, okunabilirlik ve yazılabilirliğini artırdığı anlaşıldı. Ayrıca staj sürecince bu standartlarda yazılım geliştirileceği bilgisi edinildi.

- 1 (01.01) Tüm kodlar ISO 9899 standardına uygun olmalıdır.
- 2 (02.01) Assembly kodları ayrıştırılmalıdır.
- 3 (02.02) Yorum satırları sadece "/*" sembolü ile yazılmalıdır.
- 4 (02.03) Nested (iç içe) yorumlar kullanılmamalıdır.
- 5 (02.04) Mümkünse derlenecek kod parçaları yorum satırları ile değil, "#ifdef" önderleyici komutuyla yazılmalıdır.
- 6 (03.01) Tüm uygulamalar belgelendirilmelidir.
- 7 (03.02) Karakter seti ve uygun kodlama türü belgelendirilmelidir.
- 8 (03.03) Tam sayı bölme işlemi seçili derleyici için belgelendirilmelidir.
- 9 (03.04) #pragma komutları belgelendirilmelidir.
- 10 (03.05) Bitfield'lar belgelendirilmelidir.
- 11 (03.06) Üretilen kodda kullanılan tüm kütüphaneler bu belgedeki hükümlerle uyumlu olmalıdır.
- 12 (04.01) Sadece ISO C'de tanımlı escape sequence karakterleri kullanılmalıdır.
- 13 (04.02) Trigraph'lar (üçlü harf grupları) kullanılmamalıdır. (Örneğin "???" gösterimi "]" ile eşdeğerdir.)
- 14 (05.01) Identifier'lar (Değişkenler için tanımlayıcılar) 31 karakterden uzun olmamalıdır.
- 15 (05.02) Bloklar içindeki Identifier'lar (Değişkenler için tanımlayıcılar) dıştakilerle aynı isimde olmamalıdır.
- 16 (05.03) Typedef isimleri tekil olmalı, başka kısımlarda kullanılmamalıdır.
- 17 (05.04) Struct etiket isimleri tekil olmalı, başka kısımlarda kullanılmamalıdır.
- 18 (05.05) Nesne ve fonksiyon tanımlayıcıları static storage sırasında yeniden kullanılmamalıdır.
- 19 (05.06) Typedef isimleri farklı bir namespace'deki isimlerden farklı olmalıdır.
- 20 (05.07) Farklı struct bloklarındaki tanımlayıcılar farklı adlandırılmalıdır.
- 21 (06.01) char veri türü sadece karakter verisi depolamak için kullanılmalıdır.
- 22 (06.02) signed char ve unsigned char veri türleri sadece sayısal karakterler depolamak için kullanılmalıdır.
- 23 (06.03) Typedef isimlerinde boyut ve işaretlilik durumu yazılmalıdır.
- 24 (06.04) Bitfield'lar sadece unsigned int veya signed int olarak tanımlanmalıdır.
- 25 (06.05) signed tiplerin bitfield'ları en az 2 bit uzunluğunda olmalıdır.
- 26 (07.01) Sıfır dışındaki sayılar ve escape sequence karakterleri octal gösterimde yazılmamalıdır.
- 27 (08.01) Fonksiyonların prototype tanımları olmalıdır.
- 28 (08.02) Nesne ve fonksiyon tanımlarında veri türleri açık biçimde belirtilmelidir.
- 29 (08.03) Her fonksiyon için parametre verilen ve döndürülen değerler ilgili prototype ile özdeş olmalıdır.
- 30 (08.04) Nesne ve fonksiyonlar birden fazla tanımlanmış türlerine göre uyumlu olmalıdır.
- 31 (08.05) Header dosyalarında nesne veya fonksiyon tanımları bulunmamalıdır.
- 32 (08.06) Fonksiyonlar dosya kapsama alanında (file scope) tanımlanmalıdır.
- 33 (08.07) Nesnelar tek bir fonksiyondan erişilecekse ilgili tanımlar blok kapsama alanında (block scope) olmalıdır.
- 34 (08.08) Harici (extern) nesne veya fonksiyon tanımları sadece tek dosyayı ilgilendirmelidir.
- 35 (08.09) Farklı bir dosyada harici bağlantılı bir tanımlayıcının tamamen aynı özellikte bir tanımlı olmalıdır.
- 36 (08.10) Nesne ve fonksiyonlarla ilgili tüm belirtme ve tanımlamalar dosya kapsama alanında dahili bağlantılı olmalıdır.
- 37 (08.11) Dahili bağlantılı nesne ve fonksiyonlarda mümkünse static depolama sınıfı kullanılmalıdır.
- 38 (08.12) Bir dizi harici bağlantılı bağlanıyorsa ilk değer ataması yapılmalıdır.
- 39 (09.01) Otomatik oluşturulan tüm değişkenler için ilk değer ataması yapılmalıdır.
- 40 (09.02) Struct ve çok boyutlu dizi tanımlarında farklı gruplar süslü parantezlerle ayrılmalıdır.
- 41 (09.03) Enumeration'larda "=" yapısı uygun sırada kullanılmalıdır. (enum renk { sari=3, mavi=4, siyah=5 }; gibi)

Resim 2: MISRA C Standartları, 2004 Kuralları, Bölüm 1 - 9

KISIM SORUMLUSUNUN ADI, SOYADI

27 / 07 / 2015

(ONAY)

STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	C ile Standart Kontrol Yazılımı Geliştirme	SAYFA NO	2

Stajın 2. gününde bu tür standartların kontrolünü sağlayan statik kod analiz yazılımları araştırıldı. Yazılımların önemli bir kısmının ücretli olduğu, sadece belirli noktalarda ücretsiz alternatiflerinin olduğu anlaşıldı. Visual Studio, Eclipse, Code

```

Dosya: test.c
Dosya kontrol ediliyor..
9:      G++ tipi yorumlar kullanılamaz.
14:     Octal sabitler kullanılamaz.
15:     31 karakterden uzun tanımlayıcılara izin verilemez.
3 hata bulundu
    
```

Resim 3: Standart Kontrol Konsol Uygulaması

Composer Studio gibi geliştirme ortamlarında da kod analiz eklentileri test edildi. Ancak istenilen özellikte ücretsiz yazılım veya eklenti bulunamadı. Bunun üzerine C ile ilgili kontrolü gerçekleştiren bir konsol programı yazıldı.

```

#include <stdio.h>
#include <conio.h>
#define SAYILAR "0123456789"
#define HARFLER "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNQPQRSTUVWXYZ"
#define TANIMLAYICI_KARAKTERLERI HARFLER "_" SAYILAR
#define ILK_KARAKTERLER HARFLER "-"
char *hata = "";
int arabellek[64] = { 0 };
int hata_sayisi = 0;
int satir_no = 1;
int say = 0;
int tek_tirnakta_yedek, tek_tirnakta = 0;
int cift_tirnakta_yedek, cift_tirnakta = 0;
int tirnakta_yedek, tirnakta = 0;
int yorumda_onceki, yorumda_yedek, yorumda = 0;
int kodda_yedek, kodda = 0;
int onislemcide_yedek, onislemcide = 0;
    
```

Resim 4: Yapılan tanımlamalar (C)

Burada header dosyaları dahil edildikten sonra sonra gerekli tanımlamalar yapıldı. İlk olarak #define komutuyla sayı, harf ve tanımlayıcı karakterleri tanımlandı.

Ayrıca hata isimli bir char pointer'ı, arabellek isimli bir dizi, sayaçlar ve bayrak değişkenleri tanımlandı.

Bu işlemden sonra integer parametre ile aldığı karakteri arabellek[] dizisinin başına ekleyen arabellek_ekle() fonksiyonu yazıldı. Burada for döngüsüyle dizinin sağa kaydırılması sağlandı. Ayrıca ilk indise parametre ile gelen değer atandı.

```

void arabellek_ekle(int c)
{
    int i;
    for (i = 62; i >= 0; i--)
    {
        arabellek[i + 1] = arabellek[i];
    }
    arabellek[0] = c;
}
    
```

Resim 5: arabellek_ekle() fonksiyonu (C)

KISIM SORUMLUSUNUN ADI, SOYADI	28 / 07 / 2015 (ONAY)
--------------------------------	--------------------------

STAJ DOSYASI YAPRAĞI

ÇALIŞILAN KISIM	Yazılım Birimi	STAJ ALANI	Yazılım Geliştirme
YAPILAN İŞ	C ile Kontrol Fonksiyonlarının Yazılması	SAYFA NO	3

Stajın 3. gününde standart kontrol fonksiyonları yazıldı. İlk olarak parametre verilen char dizisinin arabellekte olup olmadığını kontrol eden kontrol_arabellekte() fonksiyonu yazıldı. Burada ilgili diziler için sayaç değişkenleri tanımlandı ve ilk değer atamaları yapıldı. Daha sonra bir while döngüsüyle dizideki değerlerin kıyaslanması ve eşleşmeyen değer varsa “0” değerinin döndürülmesi sağlandı. Diğer durumlarda ise “1” değeri döndürüldü.

Ayrıca kontrol_icerir() fonksiyonu ile null karakteri (\0) görülene kadar s[] dizisinin dolaşılması sağlandı. Burada dizi ilgili char değerini içeriyorsa “1”, içermiyorsa “0” değeri döndürüldü.

```
int kontrol_arabellekte(char s[], int arabellek[], int p)
{
    int say_a, say_b;
    say_a = 0;
    say_b = p;
    while (say_b >= 0)
    {
        if (s[say_a] != arabellek[say_b])
        {
            return 0;
        }
        say_a++;
        say_b--;
    }
    return 1;
}

int kontrol_icerir(char c, char s[])
{
    int i;
    for (i = 0; s[i] != '\0'; i++)
    {
        if (s[i] == c)
        {
            return 1;
        }
    }
    return 0;
}

int kontrol_icermez(char c, char s[])
{
    return !kontrol_icerir(c, s);
}
```

Resim 6: Temel kontrol fonksiyonları (C)

Bu işlemden sonra kontrol_icermez() fonksiyonu yazıldı. Söz konusu fonksiyonda kontrol_icerir() fonksiyonunun döndürdüğü değer değilinin alınması sağlandı.

Temel kontrol fonksiyonları yazıldıktan sonra kontrol_tirnak_arasinda() fonksiyonu yazıldı. Burada ilk olarak “tirnak_arasinda” değişkeni tanımlandı. İlk olarak kodun yorum kısmındayken veya kodda tırnak kapatılıyorsa değişkene “0” değerinin atanması sağlandı. Diğer durumlarda arabelleğin başındaki değere göre atama yapılarak işlem tamamlandı.

```
int kontrol_tirnak_arasinda(char tirnak_karakter, int tirnak_arasinda_yedek, int tirnak_arasinda_diger, int yorumda_yedek, int arabellek[])
{
    int tirnak_arasinda;
    if (yorumda_yedek != 0 || tirnak_arasinda_diger == 1)
    {
        tirnak_arasinda = 0;
    }
    else if (arabellek[0] == tirnak_karakter) //Tirnak basliyorsa
    {
        tirnak_arasinda = !tirnak_arasinda_yedek;
    }
    else
    {
        tirnak_arasinda = tirnak_arasinda_yedek;
    }
    return tirnak_arasinda;
}
```

Resim 7: kontrol_tirnak_arasinda() fonksiyonu (C)

KISIM SORUMLUSUNUN ADI, SOYADI	29 / 07 / 2015 (ONAY)
--------------------------------	--------------------------