

اصول مهندسی نرم افزار پیشرفته برای مواجهه با پیچیدگی

دکتر فریدون شمس علیئی
دلارام گلپایگانی



۱۴۰۰



۸۰۹

مرکز چاپ و انتشارات دانشگاه شهید بهشتی

اصول مهندسی نرم‌افزار پیشرفته برای مواجهه با پیچیدگی
دکتر فریدون شمس علی‌ئی، دلارام گلپایگانی

حروف‌نگار و صفحه‌آرا: سمیرا دهقان

ناظر چاپ: نوید سیفان

چاپ اول: ۱۴۰۰

شمارگان: ۲۰۰

قیمت: ۱۸۲۰.۰۰۰ ریال

کلیه حقوق برای دانشگاه شهید بهشتی محفوظ است.

سرشناسه: شمس علی‌ئی، فریدون، ۱۳۳۹-
عنوان و نام پدیدآور: اصول مهندسی نرم‌افزار پیشرفته برای مواجهه با پیچیدگی / فریدون شمس علی‌ئی، دلارام گلپایگانی.
مشخصات نشر: تهران: دانشگاه شهید بهشتی، مرکز چاپ و انتشارات، ۱۴۰۰.
مشخصات ظاهری: بهشت‌ونه، ۴۲۴ ص.
فروست: مرکز چاپ و انتشارات دانشگاه شهید بهشتی: ۸۰۹.
شابک: ۹۷۸۰۵۶۴۵۵۷۵۴۵۷
وضعیت فهرست‌نویسی: فیپا
یادداشت: واژه‌نامه: کتابنامه: نمایه.
موضوع: نرم‌افزار — ارزشیابی
Computer software—Evaluation
پیچیدگی محاسباتی
Computational complexity
نرم‌افزار — مهندسی
Software engineering
شناسه افزوده: دانشگاه شهید بهشتی، مرکز چاپ و انتشارات
شناسه افزوده: گلپایگانی، دلارام، ۱۳۶۹-
شناسه افزوده: ایرانی فرد، محمود، ۱۳۶۰-، ویراستار
شناسه افزوده: Shahid Beheshti University. Printing & Publishing Center
رده‌بندی کنگره: QA۷۶/۷۶
رده‌بندی دیویی: ۰۰۵/۳۰۲۸۷۲۲
شماره کتابشناسی ملی: ۸۷۵۶۵۶۴

کد ناشر ۱۰۰۱۷۳۴

www.pub.sbu.ac.ir
unipress@mail.sbu.ac.ir

فهرست مطالب

پیشگفتار.....	بیست و هفت
فصل ۱. چرا تولید نرم افزار کار پیچیده ای است؟.....	۱
۱.۱. مشکلات تولید نرم افزار در دنیای امروز.....	۱
۲.۱. ویژگی های ذاتی نرم افزار.....	۳
۱.۲.۱. پیچیدگی.....	۳
۲.۲.۱. متابعت.....	۳
۳.۲.۱. قابلیت تغییر.....	۴
۴.۲.۱. ناپایداری.....	۴
۳.۱. عوامل پیچیدگی ذاتی نرم افزار.....	۵
۱.۳.۱. پیچیدگی دامنه مسئله.....	۵
۲.۳.۱. دشواری مدیریت فرایند تولید.....	۶
۳.۳.۱. انعطاف پذیری نرم افزار.....	۶
۴.۳.۱. مشکلات توصیف رفتار سیستم های گسسته.....	۷
۴.۱. ساختار سیستم های پیچیده.....	۷
۵.۱. مواجهه با پیچیدگی ذاتی نرم افزار.....	۱۰
۱.۵.۱. تجزیه.....	۱۰
۲.۵.۱. تجرید.....	۱۰
۳.۵.۱. سلسله مراتب.....	۱۱
۴.۵.۱. فرکتال.....	۱۱
۶.۱. ضرورت توجه به متدولوژی.....	۱۲
تمرین های فصل ۱.....	۱۳
منابع فصل ۱.....	۱۴
فصل ۲. روش های نیمه رسمی مدلسازی نرم افزار برای مقابله با پیچیدگی.....	۱۷
۱.۲. مفهوم مدل، مدلسازی و کاربرد آن در تولید نرم افزار.....	۱۸
۲.۲. نقش تجرید در مدلسازی و ضرورت استفاده از آن در مدل های نرم افزاری.....	۲۰
۳.۲. آشنایی با سطوح گوناگون مدلسازی در مهندسی نرم افزار.....	۲۱
۴.۲. آشنایی با UML.....	۲۲
۱.۴.۲. نمودار کلاس.....	۲۳
۲.۴.۲. نمودار شیء.....	۲۵

۲۵	۳.۴.۲. نمودار مورد کاربری
۲۹	۴.۴.۲. نمودارهای تعامل
۳۰	۱.۴.۴.۲. نمودار ترتیبی
۳۱	۲.۴.۴.۲. نمودار همکاری
۳۱	۵.۴.۲. نمودار حالت
۳۳	۶.۴.۲. نمودار فعالیت
۳۳	۷.۴.۲. نمودار مؤلفه
۳۵	۸.۴.۲. نمودار استقرار
۳۶	۵.۲. کاربرد مدلسازی در کاهش پیچیدگی نرم افزار
۳۸	تمرین های فصل ۲
۴۰	منابع فصل ۲
۴۱	فصل ۳. شبکه های پتری، روشی رسمی برای مدلسازی نرم افزار
۴۲	۱.۳. ضرورت مدلسازی نرم افزار با استفاده از شبکه های پتری
۴۲	۲.۳. کاربردهای شبکه های پتری
۴۳	۳.۳. آشنایی با عناصر شبکه پتری
۴۶	۴.۳. خصوصیات شبکه های پتری
۴۷	۱.۴.۳. قابلیت دستیابی
۴۷	۲.۴.۳. مطمئن بودن
۴۷	۳.۴.۳. زنده بودن
۴۷	۵.۳. انواع شبکه پتری
۴۸	۱.۵.۳. شبکه های پتری رنگی
۴۸	۲.۵.۳. شبکه های پتری زمان دار
۴۹	۶.۳. کاربرد شبکه های پتری در مهندسی نرم افزار
۵۱	۷.۳. نحوه استخراج شبکه های پتری از مدل های UML
۵۱	۱.۷.۳. تبدیل نمودار موارد کاربری به شبکه پتری
۵۲	۲.۷.۳. تبدیل نمودار ترتیبی به شبکه پتری
۵۲	۳.۷.۳. تبدیل نمودار مؤلفه به شبکه پتری
۵۴	۸.۳. چند مثال از مدلسازی رفتار اشیاء با استفاده از شبکه های پتری
۵۸	۹.۳. کاربرد شبکه های پتری در مدلسازی فرایند
۵۹	۱۰.۳. ابزارهای مدلسازی شبکه های پتری
۶۰	تمرین های فصل ۳
۶۲	منابع فصل ۳
۶۳	فصل ۴. مهندسی نرم افزار شیء گرا
۶۴	۱.۴. جهان بینی شیء گرا
۶۶	۲.۴. مفاهیم پایه شیء گرایی

۶۶	۱.۲.۴. شیء
۶۸	۲.۲.۴. کلاس
۶۹	۱.۲.۲.۴. واسط کلاس
۷۰	۲.۲.۲.۴. ارتباط کلاس‌ها
۷۲	۳.۴. اصول شیء‌گرایی برای مقابله با پیچیدگی
۷۳	۱.۳.۴. تجرید
۷۴	۲.۳.۴. محصورسازی
۷۶	۳.۳.۴. پیمانه‌بندی
۷۷	۴.۳.۴. سلسله‌مراتب
۷۸	۵.۳.۴. فرکتال
۸۱	۴.۴. روش شناسایی و یافتن کلاس‌ها
۸۲	۵.۴. رهیافت‌های کلی برای یافتن شناسایی کلاس‌ها
۸۲	۱.۵.۴. رهیافت مبتنی بر داده
۸۲	۲.۵.۴. رهیافت مبتنی بر مسئولیت
۸۳	۶.۴. روش‌های شناسایی کلاس‌های اولیه
۸۳	۱.۶.۴. طبقه‌بندی‌های پیشنهادی
۸۳	۲.۶.۴. تحلیل دامنه
۸۳	۳.۶.۴. تحلیل موارد کاربری
۸۳	۴.۶.۴. تحلیل لغوی صورت مسئله
۸۴	۵.۶.۴. استفاده از الگو
۸۴	۶.۶.۴. کارت‌های CRC
۸۷	۷.۴. روش‌های تکمیلی برای سازماندهی کلاس‌ها
۸۸	۱.۷.۴. لایه‌بندی
۹۰	۲.۷.۴. مقوله‌بندی
۹۱	۱.۲.۷.۴. کارت‌های CRC و مقوله‌بندی
۹۲	۸.۴. ویژگی‌هایی که از روش مهندسی نرم‌افزار شیء‌گرا انتظار می‌رود
۹۲	۱.۸.۴. توسعه تکراری و تدریجی
۹۴	۲.۸.۴. مدیریت نیازمندی‌ها
۹۵	۳.۸.۴. استفاده از معماری مبتنی بر مؤلفه‌ها
۹۶	۴.۸.۴. مدلسازی تصویری نرم‌افزار
۹۷	۵.۸.۴. بررسی مداوم کیفیت نرم‌افزار
۹۸	۶.۸.۴. مدیریت تغییرات و پیکربندی نرم‌افزار
۹۹	تمرین‌های فصل ۴
۱۰۴	منابع فصل ۴
۱۰۵	فصل ۵. معرفی و تحلیل یک نمونه متدولوژی شیء‌گرا
۱۰۶	۱.۵. فرایند تولید نرم‌افزار در متدولوژی USDP

۱۰۶	۱.۱.۵. محورهای فرایند یکپارچه
۱۰۶	۱.۱.۱.۵. توسعه بر مبنای موارد کاربری
۱۰۷	۲.۱.۱.۵. محوریت معماری
۱۰۷	۳.۱.۱.۵. تکرار و توسعه تدریجی
۱۰۸	۲.۱.۵. چرخه حیات فرایند یکپارچه
۱۰۹	۲.۵. معرفی اجمالی RUP (از منظر محصول و فرایند)
۱۱۱	۳.۵. ساختار ایستای RUP: معرفی اجمالی فراورده‌ها، نقش‌ها، فعالیت‌ها و نظم‌ها
۱۱۲	۱.۳.۵. نقش
۱۱۳	۲.۳.۵. فعالیت
۱۱۳	۳.۳.۵. فراورده
۱۱۵	۴.۳.۵. نظم
۱۱۶	۴.۵. ساختار یوای RUP: آغازین، تشریح، ساخت و انتقال
۱۱۸	۱.۴.۵. فاز آغازین
۱۱۹	۲.۴.۵. فاز تشریح
۱۲۰	۳.۴.۵. فاز ساخت
۱۲۰	۴.۴.۵. فاز انتقال
۱۲۱	۵.۵. معرفی تفصیلی نظم‌های RUP
۱۲۱	۱.۵.۵. نظم مدلسازی کسب‌وکار
۱۲۲	۲.۵.۵. نظم نیازمندی‌ها
۱۲۳	۳.۵.۵. نظم تحلیل و طراحی
۱۲۴	۴.۵.۵. نظم پیاده‌سازی
۱۲۵	۵.۵.۵. نظم آزمون
۱۲۵	۶.۵.۵. نظم استقرار
۱۲۶	۷.۵.۵. نظم مدیریت پیکربندی و تغییر
۱۲۸	۸.۵.۵. نظم مدیریت پروژه
۱۲۸	۹.۵.۵. نظم محیط
۱۲۸	۶.۵. آشنایی با تکنیک‌های مدلسازی در RUP
۱۲۹	۱.۶.۵. مدلسازی موارد کاربری
۱۳۱	۱.۱.۶.۵. تفاوت مدلسازی مورد کاربری و تحلیل سیستم
۱۳۱	۲.۱.۶.۵. شناسایی بازیگران سیستم در مثال تعمیرگاه
۱۳۲	۳.۱.۶.۵. نمودار موارد کاربری مثال تعمیرگاه
۱۳۳	۲.۶.۵. مدلسازی کلاس‌ها
۱۳۴	۱.۲.۶.۵. ایجاد نمودار کلاس
۱۳۴	۲.۲.۶.۵. مدلسازی کلاس‌ها در RUP
۱۳۵	۳.۲.۶.۵. مدل تحلیل
۱۳۶	۴.۲.۶.۵. مدل طراحی

۱۳۶	۵.۲.۶.۵. مدل داده‌ای
۱۳۶	۶.۲.۶.۵. نمودار کلاس مثال تعمیرگاه
۱۳۹	۳.۶.۵. مدلسازی تعامل و رفتار
۱۳۹	۱.۳.۶.۵. نمودار ترتیبی و نمودار همکاری مثال تعمیرگاه
۱۴۱	۲.۳.۶.۵. مدلسازی حالت
۱۴۱	۳.۳.۶.۵. نمودار حالت مثال تعمیرگاه
۱۴۳	۴.۳.۶.۵. نمودار فعالیت
۱۴۴	۴.۶.۵. مدلسازی مؤلفه و استقرار
۱۴۵	۱.۴.۶.۵. بسته
۱۴۶	۲.۴.۶.۵. مدلسازی مؤلفه‌ها
۱۴۶	۳.۴.۶.۵. مدلسازی استقرار
۱۴۷	۷.۵. بررسی نقدهای مطرح در مورد RUP
۱۴۸	تمرین‌های فصل ۵
۱۵۲	منابع فصل ۵
۱۵۵	فصل ۶. الگوهای طراحی و نقش آن‌ها در بهبود قابلیت استفاده مجدد نرم‌افزار
۱۵۶	۱.۶. الگو به مثابه یکی از روش‌های رایج برای دستیابی به قابلیت استفاده مجدد
۱۵۷	۲.۶. الگوهای طراحی در نرم‌افزار
۱۵۷	۱.۲.۶. تعریف الگو
۱۵۸	۲.۲.۶. طبقه‌بندی الگوها
۱۵۹	۳.۶. ساختار الگوی طراحی
۱۵۹	۱.۳.۶. ساختار پیشنهادی الکساندر
۱۵۹	۲.۳.۶. ساختار پیشنهادی پولتی
۱۶۰	۳.۳.۶. ساختار پیشنهادی باند چهارنفره (GoF)
۱۶۱	۴.۶. معرفی چند نمونه الگوی طراحی شیء گرا
۱۶۱	۱.۴.۶. الگوی Factory Method
۱۶۴	۲.۴.۶. الگوی Abstract Factory
۱۶۷	۳.۴.۶. الگوی Adapter
۱۷۱	۴.۴.۶. الگوی Observer
۱۷۳	۵.۴.۶. الگوی Template Method
۱۷۵	۶.۴.۶. الگوی Composite
۱۷۸	۵.۶. نقش الگوهای طراحی در مواجهه با پیچیدگی نرم‌افزار
۱۷۸	۶.۶. ضد الگوها
۱۸۰	۷.۶. بازبینی میزان موفقیت الگوها در دستیابی به قابلیت استفاده مجدد نرم‌افزار
۱۸۱	تمرین‌های فصل ۶
۱۸۲	منابع فصل ۶

به نام آنکه هستی نام ازو یافت
فلک جنبش، زمین آرام ازو یافت

خدائی کافرینش در سجودش
گواهی مطلق آمد بر وجودش

حکیم نظامی گنجوی

توسعه نرم افزار در دنیای امروز، بیش از پیش پیچیده شده است. لاجرم جامعه مهندسی نرم افزار و دیگر دانش پژوهان علوم مرتبط، به شناخت پیچیدگی نرم افزار و رهیافت های مواجهه با آن ترغیب شده اند. نوشتار پیش رو مجموعه ای است از رهیافت های تجربه شده برای درک بهتر پدیده پیچیدگی نهفته در توسعه نرم افزار و مواجهه آسان تر با آن.

سال هاست که در اغلب دانشگاه های کشور، بخش عمده ای از این رهیافت ها (مباحث) به شیوه های گوناگون در یک واحد درسی دوره تحصیلات تکمیلی با عنوان مهندسی نرم افزار پیشرفته تدریس می شود. ما نیز به نوبه خود در تلاش برای ارائه منسجم تر، شفاف تر و کارآمدتر این مباحث بوده ایم؛ البته به شکلی کاربردی و با هدف زمینه سازی شکل گیری مبانی فکری دانش پژوهان جامعه مهندسی نرم افزار و به ویژه دانشجویان تحصیلات تکمیلی.

تفاوت نگاه ما به موضوع مهندسی نرم افزار پیشرفته که در سراسر کتاب جاری است، به نحوه پرداختن به هر یک از رهیافت های موجود به لحاظ شیوه مواجهه با مسئله پیچیدگی در توسعه نرم افزار و طرز مدیریت آن بازمی گردد. بر همین اساس، سعی کرده ایم انسجام مطالب کتاب حول محور پیچیدگی نرم افزار حفظ شود. شاید بتوان گفت دستاورد اصلی این کتاب نیز همان نگاه متفاوت و البته منسجم به موضوع برای برقراری ارتباط کامل منطقی، معنی دار و راهگشا میان رویکردهای پیرامون مباحث پیشرفته در مهندسی نرم افزار است. لذا تلاش شده است که مذاقه و کنکاش لازم برای بررسی موشکافانه هر یک از مسائل معرفی شده از منظری نوین و کارآمد انجام شود. در همین راستا، علاوه بر دریچه هایی که نویسندگان مشهوری مانند گریدی بوچ (Grady Booch) به روی شیوه های مقابله با پیچیدگی در توسعه نرم افزار گشوده اند، به دیگر روش ها که کمتر محل توجه بوده اند (مانند نگاه فرکتال) اشارات کاربردی داشته ایم.

چیدمان فصل ها و محتوای هر فصل بر مبنای تجارب بیش از سه دهه آموزش نظری و عملی موضوع مهندسی نرم افزار در دانشگاه های داخل و خارج از کشور شکل گرفته است. در تدوین مطالب هر فصل تلاش شده است علاوه بر طرح مبانی فکری و انگیزشی برای مسئله محوری آن فصل، به معرفی رویکردهای کاربردی و نمونه خروجی های ملموس از نتایج کاربرد هر رویکرد پرداخته شود.

هر فصل با یک نقل قول کلیدی در قالب نظم یا نثر برای ایجاد جرعه اولیه در ذهن خواننده راجع به موضوع مربوطه آغاز می شود و با مقدمه ای برای روشن شدن ابعاد موضوع در نظر خواننده ادامه می یابد تا درک بهتری از آن فراهم شود. سپس به ابعاد کاربردی و ذکر نمونه از نتایج کاربرد روش های معرفی شده روی مثال های عینی می پردازیم و در پایان نیز تحلیلی از نحوه ارتباط موضوع مطرح شده در هر فصل با مفهوم پیچیدگی در توسعه نرم افزار خواهیم داشت.

برای فراهم آوردن درکی عمیق تر و نیز دسترسی به منابع استفاده شده در کتاب، سعی شده است درمورد منابع اصلی و نحوه ارتباط آن ها با مطالب هر فصل، به طور جداگانه توضیح مختصری ارائه شود. در پایان هر فصل، به منظور مرور و فهم بهتر نکات

آن فصل و نیز استفاده مدرسان و دانشجویان، تمرین‌هایی را که اغلب جنبه‌های مطرح‌شده در آن فصل را پوشش می‌دهند آماده کرده‌ایم.

روال پیشنهادی ما برای مطالعه مطالب کتاب، دنبال کردن ترتیب منطقی فصول به شرح زیر است:

طرح مسئله پیچیدگی نرم‌افزار در فصل اول بیان می‌شود. در این فصل پیچیدگی به عنوان یکی از ویژگی‌های ذاتی نرم‌افزار معرفی شده و راهکارهای کلی مواجهه با آن بررسی می‌شود. در فصل‌های آتی به نقش هر یک از رهیافت‌های مواجهه با پیچیدگی از منظر همان راهکارها پرداخته می‌شود.

فصل دوم بر نقش مدلسازی با روش‌های نیمه‌رسمی در مقابله با پیچیدگی تمرکز دارد. از این رو، به طور مختصر زبان مدلسازی UML و نحوه مدلسازی نرم‌افزار با استفاده از آن معرفی خواهد شد.

از آنجاکه مدل‌های UML امکان اجرایی شدن ندارند، باب آشنایی با شبکه‌های پتری را به عنوان روش رسمی و مبتنی بر ریاضی در مدلسازی نرم‌افزار، در فصل سوم باز خواهیم کرد. همچنین تمرکز ویژه‌ای روی کاربرد شبکه‌های پتری در مهندسی نرم‌افزار و نحوه استخراج این شبکه‌ها از روی نمودارهای UML خواهیم داشت.

در دنیای نرم‌افزار، شیء‌گرایی یکی از رهیافت‌های پذیرفته شده است که مهندسان نرم‌افزار را به سلاحی برای مقابله با پیچیدگی مجهز کرده‌اند و لذا در فصل چهارم به طور مفصل به مفاهیم شیء‌گرایی، اصول شیء‌گرایی برای مقابله با پیچیدگی و مدلسازی شیء‌گرا خواهیم پرداخت.

از آنجاکه مقابله با پیچیدگی به کمک متدولوژی آسان‌تر خواهد بود، RUP به عنوان یکی از متدولوژی‌های مطرح در حیطه شیء‌گرایی در فصل پنجم معرفی خواهد شد. همچنین در این فصل مجموعه‌ای از تکنیک‌های مدلسازی شیء‌گرا با هدف توضیح نحوه استخراج و تبدیل این مدل‌ها در چرخه متدولوژی به تفصیل تشریح خواهد شد و با طرح نمونه خروجی‌هایی از مدل‌ها به نمایش در خواهند آمد.

استفاده مجدد راهکار مؤثری در مواجهه با پیچیدگی است که به شکل‌های متنوع در توسعه نرم‌افزار خود را نشان می‌دهد. معرفی مؤلفه‌ها به عنوان یکی از سازه‌بلوک‌های اصلی در دستیابی به استفاده مجدد در مقیاسی فراتر از اشیا، به ویژه نحوه توسعه نرم‌افزار مبتنی بر مؤلفه، در دستور کار فصل ششم کتاب قرار گرفته است.

در فصل هفتم به امکانات تازه‌تر در حمایت از استفاده مجدد برای توسعه نرم‌افزار بر اساس تفکر سرویس‌گرایی می‌پردازیم، مفهوم سرویس را در ادامه مفاهیم قبلی یعنی شیء و مؤلفه تشریح کرده و خصوصیات اصلی مدل توسعه نرم‌افزار سرویس‌گرا را به همراه استانداردها، زبان و شیوه مدلسازی آن تبیین می‌کنیم.

الگوهای طراحی نرم‌افزار به عنوان یکی دیگر از شیوه‌های بهبود استفاده مجدد و البته همچنان کارآمد در امر مواجهه با پیچیدگی در فصل هشتم معرفی شده و نقش آن‌ها در توسعه نرم‌افزار با کیفیت بهتر بازبینی می‌شود.

برای فائق آمدن بر مسئله پیچیدگی، علاوه بر روش‌های معرفی شده در فصل‌های قبلی، نیازمند بهره‌مندی از اصل چابکی به عنوان یک راهکار اثبات شده و فراگیر در متدولوژی‌های نوین هستیم. بنابراین در فصل نهم روی روش‌های چابک توسعه نرم‌افزار و معرفی متدولوژی‌های چابک متمرکز خواهیم شد.

در استفاده از روش‌های چابک، فاصله مهمی میان تیم توسعه‌دهنده و تیم بهره‌بردار وجود دارد که خود باعث بروز مشکلاتی می‌شود. برای رفع این فاصله، در فصل دهم به معرفی مهندسی نرم‌افزار مبتنی بر تلفیق توسعه و عملیات (DevOps) می‌پردازیم. البته در این فصل به تشریح میکروسرویس‌ها و نقش مهم آن‌ها در DevOps و توسعه نرم‌افزار نیز نگاهی می‌اندازیم. در فصل یازدهم با نقش جنبه‌ها در جداسازی دغدغه‌ها در مقابله با پیچیدگی آشنا خواهید شد. در این فصل تمرکز روی

معرفی شیوه ساماندهی دغدغه‌های متداخلی است که به‌نوعی باعث افزایش پیچیدگی می‌شوند و مانع استفاده از پیمان‌بندی کارآمد هستند.

در نهایت، در فصل دوازدهم خواهیم دید که چگونه می‌توان بخشی از پیچیدگی‌ها را با عامل‌های هوشمند مدیریت کرد. معرفی دو نمونه متدولوژی مهندسی نرم‌افزار مبتنی بر عامل به‌همراه مثال‌هایی از خروجی این متدولوژی‌ها، پایان‌بخش مطالب این فصل خواهد بود.

نخستین بار جزوه و اسلایدهای درس مهندسی نرم‌افزار پیشرفته، با کمک آقایان سیدمصطفی کلانتر و سپس سیدمهدی زرگرنتاج تدوین و تکمیل شد و به‌عبارتی، کار ایشان پایه اولیه این کتاب بوده است. به‌نوبه خودمان از تلاش‌هایشان تشکر می‌کنیم. برای طراحی اولیه جلد نیز از آقای علی حنیفی قدردانیم. همچنین، از خانواده‌هایمان که با بردباری، گذشت و حمایت‌شان ما را در مسیر تألیف این کتاب همراهی کردند از صمیم قلب سپاسگزاریم.

آنچه در این کتاب آمده، با امید به بهره‌مندی و شکوفایی ذهن دانش‌پژوهان این عرصه همراه بوده است و فقط با دریافت نظر خوانندگان عزیز برای تکمیل و تصحیح مطالب به‌بار خواهد نشست. به‌قول فخرالدین اسعد گرگانی:

نبینی باغبان چون گل بکارد چه مایه غم خورد تا گل برآرد

از این‌رو، صمیمانه درخواست می‌کنیم کاستی‌ها و خطاهای کتاب و همچنین نظرات و پیشنهادات خود را به آدرس

f_shams@sbu.ac.ir ارسال کنید.

فریدون شمس علی‌نی، عضو هیئت علمی دانشگاه شهید بهشتی

دلارام گلپایگانی

زمستان ۱۴۰۰

www.ketab.ir