

بازی صفرویک

نویسنده:

زهرا فرشچی

سرشناسه : فرشیچی، زهرا، ۱۳۵۴ - Farshchi, Zahra, 1975
 عنوان و نام : بازی صفر و یک/ نویسنده زهرا فرشیچی؛ ویراستار حسین مطیع .
 پدیدآور :
 مشخصات نشر : قمر: انتشارات گل پوش، ۱۴۰۳.
 مشخصات : ۲۴۰ ص؛ ۱۴/۵×۲۱/۵ سم.
 ظاهری :
 شابک : 978-622-6878-21-0 ۳۴۰۰۰۰۰ ریال
 وضعیت فهرست : فیبا
 نویسی :
 موضوع : برنامه نویسی
 Computer programming
 زبان های برنامه نویسی کامپیوتر
 (Programming languages (Electronic computers
 رده بندی کنگره : ۶/Q۷۷۶
 رده بندی دهی : ۱/۰۰۵
 کتابشناسی ملی : ۹۵۵۸۷۴۸
 رکوردکبشناسی : فیبا



انتشارات گلدان

«بازی صفر و یک»

نویسنده زهرا فرشیچی
 ویراستار حسین مطیع
 ناشر انتشارات گل پوش
 تیراژ ۱۰۰۰
 چاپ شریعت
 نوبت و تاریخ چاپ دوم ۱۴۰۳
 شابک ۹۷۸-۶۲۲-۶۸۲۸-۲-۰
 قیمت ۲۴۰۰۰۰ تومان

مراکز توزیع:

قم. خیابان معلم. مجتمع ناشران. طبقه دوم. پلاک: ۲۱۹. تلفن: ۰۲۵۳۷۸۳۱۲۱۵
 تهران: میدان انقلاب. خیابان شهدای ژاندارمری. پلاک: ۵۸. طبقه اول. تلفن: ۰۲۱۶۶۴۸۴۱۷۹

فهرست

۱۵	مقدمه
۱۷	تاریخچه
۱۸	مفسر بیسمک
۲۲	نگاه اولیه به QBASIC
۲۲	صفحه اصلی
۲۵	استفاده از منوهای QBASIC
۲۷	ایجاد و اجرای یک برنامه
۲۸	وارد کردن یک برنامه QBASIC
۳۲	PRINT i:
۳۲	معرفی زبان‌های برنامه نویسی
۳۹	معرفی BASIC
۴۰	پیش زمینه
۴۲	سالهای نخست: دوره مینی کامپیوترها
۴۵	رشد خیره کننده: دوره کامپیوترهای خانگی
۴۷	بلوغ: دوره کامپیوترهای شخصی
۵۱	معرفی زبان C
۵۶	مراحل اجرای یک برنامه C
۵۷	۸ ویژگی برنامه نویسی C

۲. کلمات کلیدی کم ۶۰
۳. عدم وجود رشته‌های صریح ۶۰
۴. دست‌کاری اشاره‌گر ۶۱
۵. قابلیت تعویض محل اشاره‌گرها و آرایه‌ها ۶۲
۶. استفاده گسترده از ماکروهای «define» ۶۳
۷. ماژول‌های کپسوله‌شده در فایل‌ها ۶۳
۸. کتابخانه‌های خارجی ۶۴
- زبان برنامه‌نویسی C# ۶۴
- شی‌گرایی به چه معنا است؟ ۶۶
- C# برای من مناسب است؟ ۶۷
- با C# چه نرم‌افزارهایی را می‌توانم توسعه دهم؟ ۶۸
- Net. چه ارتباطی با C# دارد؟ ۶۹
- C# و net. چیست و آیا مناسب من است؟ ۶۹
- جاوا ۷۰
- تاریخچه زبان برنامه‌نویسی Java ۷۲
- اصول اولیه زبان برنامه‌نویسی Java ۷۵
- کاربردهای زبان برنامه‌نویسی Java ۷۶
- پیاده‌سازی‌ها ۷۷
- عملکرد ۷۹
- مدیریت حافظه خودکار ۸۰
- قاعده نحوی ۸۳
- جاوا اسکریپت ۸۷

۸۹	تاریخچه
۹۰	ویژگی ها
۹۱	زبان امری و ساخت یافته
۹۱	پویایی
۹۲	تابعی بودن
۹۳	ساختار شماتیک «محوری»
۹۳	امکانات دیگر
۹۴	literal های آرایه و شی
۹۴	عبارات منظم
۹۵	سی پلاس پلاس
۹۵	موارد استفاده از زبان سی پلاس پلاس
۹۸	پی اچ پی
۹۹	تاریخچه
۱۰۳	پرل (PERL)
۱۰۴	تاریخچه
۱۰۷	پرل (PERL)
۱۱۰	پونی
۱۱۰	نام
۱۱۱	نماد شتر
۱۱۲	نماد پیاز
۱۱۲	ویژگی ها
۱۱۳	طراحی

۱۱۳	پایاده‌سازی
۱۱۷	دسترسی
۱۱۷	ویندوز
۱۱۹	پایگاه داده
۱۲۱	بهینه‌سازی
۱۲۱	پرل ۶
۱۲۴	قابلیت‌ها
۱۲۴	کاربرد پرل
۱۲۴	پرل به عنوان یک زبان چسبنده (Glue Language)
۱۲۵	پرل به عنوان یک زبان کراس پلتفرم (Cross Platform)
۱۲۵	رل و CGI
۱۲۶	سوئیفت
۱۲۷	تاریخچه
۱۲۹	سی اس اس
۱۲۹	استانداردهای تازه
۱۳۰	برتری‌ها
۱۳۱	پهنای باند
۱۳۱	نسخه‌ها
۱۳۱	اس کیو ال
۱۳۲	پایگاه داده رابطه‌ای (Relational Database)
۱۳۵	ویرایش اینترپرایز
۱۳۶	استاندارد

۱۳۶	وب
۱۳۷	Workgroup
۱۳۷	Express
۱۳۸	آزور دیتابیس
۱۳۸	آزور
۱۳۸	(Compact) SQL CE
۱۳۹	Developer
۱۴۰	(Embedded) SSEE
۱۴۰	Evaluation
۱۴۰	Fast Track
۱۴۰	LocalDB
۱۴۱	(Analytics Platform System (APS
۱۴۱	Data warehouse Appliance Edition
۱۴۱	MSDE
۱۴۲	ویرایش شخصی
۱۴۲	دیتاسنتر
۱۴۲	چرا از دیتابیس استفاده کنیم؟
۱۴۵	چرا از دیتابیس استفاده نکنیم؟
۱۴۵	سیستم مدیریت
۱۴۶	نتیجه گیری
۱۴۶	جی کوثری
۱۴۷	ویژگی‌ها

۱۴۸		اچ تی ام ال (HTML)
۱۴۹		آشنایی با برخی تگ های html
۱۵۱		مای اسکيوال
۱۵۳		اليكسیر
۱۵۵		توسعه پذیری و DSL ها
۱۵۵		اکو سیستم در حال رشد
۱۵۶		سائگاری با Erlang
۱۵۶		راست
۱۵۶		اهداف اولیه طراحی زبان Rust
۱۵۷		Rust از چه زبان ها؛ سیستم هایی الهام گرفته است؟
۱۵۸		اسکالا
۱۵۹		آشنایی دلایل محبوبیت زبان اسکالا
۱۶۰		آر (R)
۱۶۰		تعریف زبان R
۱۶۰		تاریخچه زبان R
۱۶۱		طراحی این زبان
۱۶۲		ویژگی های زبان R
۱۶۳		محبوبیت زبان برنامه نویسی R
۱۶۴		نیم نگاهی به امکانات R
۱۶۵		روبی
۱۶۸		کاربرد زبان برنامه نویسی روبی
۱۶۹		برخی از امکانات Ruby

۱۶۹	جمع بندی کلی از برنامه نویسی روبی
۱۷۱	پاسکال
۱۷۲	تاریخچه
۱۷۳	توربو پاسکال
۱۷۵	پاسکال دلفی
۱۷۵	چکیده
۱۷۷	پیاده سازی
۱۸۳	ساختارهای زبان
۱۸۳	انواع داده
۱۸۴	کامپایلرها و مترجم
۱۹۰	استانداردها
۱۹۲	دسته بندی ها
۱۹۴	انتقادات
۱۹۶	مزایا و معایب زبان برنامه نویسی پاسکال
۱۹۸	اجزای تشکیل دهنده یک برنامه
۲۰۰	ساختار کلی برنامه در زبان پاسکال
۲۰۲	تورینگ
۲۰۳	پیاده سازی باز
۲۰۴	Tplus
۲۰۴	OpenT
۲۰۵	مهارت های برنامه نویسی اندروید
۲۰۵	۱- با ساختار داخلی فریم ورک اندروید بیشتر آشنا شوید

- ۲- ترس از «دست دادن» را کنار بگذارید ۲۰۶
- ۳- تا می‌توانید کدهای بیشتری بخوانید ۲۰۶
- ۴- زبان‌های بیشتری یاد بگیرید ۲۰۷
- ۵- الگوهای طراحی جاوا را یاد بگیرید ۲۰۷
- ۶- در پروژه‌های متن‌باز مشارکت کنید ۲۰۸
- ۸- برنامه‌ی خود را بر مبنای یک معماری درست بنا کنید ۲۰۹
- ۹- کدنویسی تمیز را بیاموزید ۲۰۹
- ۱- زمانی را برای یادگیری بهترین تمرین‌های برنامه‌نویسی اندروید صرف کنید ۲۱۰
- ۱۱- از زبان خالی خود برای شنیدن پادکست استفاده کنید ۲۱۰
- ۱۲- واقع‌نگر باشید؛ بیش از حد مهندسی نکنید ۲۱۰
- ۱۳- در مورد طراحی هم مطالعه کنید ۲۱۱
- ۱۴- کمال‌گرا باشید ۲۱۲
- ۱۵- تداوم و پشتکار، کلید موفقیت شماست ۲۱۲
- ۱۶- قدم‌های کوچک بردارید. آهسته و پخته حرکت کنید ۲۱۳
- ۱۷- همیشه یک پروژه تفریحی داشته باشید ۲۱۳
- ۱۸- تست‌های بیشتری بنویسید ۲۱۴
- ۱۹- خودتان را به TDD عادت دهید ۲۱۴
- ۲۰- یک مکانیزم اتوماسیون مناسب برای نسخه‌های مختلف برنامه تدوین کنید ۲۱۵
- ۲۱- نگاهی هم به رویکرد برنامه‌نویسی واکنش‌گرا داشته باشید ۲۱۵
- ۲۲- استفاده از Kotlin برای توسعه اندروید را فرا بگیرید ۲۱۶

- ۲۳- ارتباط فعالی با دیگر توسعه‌دهندگان برقرار کنید ۲۱۶
- ۲۴- به استفاده از کلیدهای میانبر صفحه کلید عادت کنید ۲۱۷
- ۲۵- سعی کنید هر هفته حداقل یک چیز جدید در مورد اندروید یاد بگیرید ۲۱۷
- ۲۶- هر چیزی که زمانتان را می‌بلعد خودکار کنید ۲۱۸
- ۲۷- از دو نسخه اندروید استودیو استفاده کنید ۲۱۸
- ۲۸- هر از گاهی تمامی کتابخانه‌ها را بررسی کنید ۲۱۸
- ۲۹- برای رکتور کردن کدهای قدیمی از روش‌های جدید استفاده کنید ۲۱۹
- ۳۰- همیشه کد را بر روی دیوایس‌های رده‌پایین تست کنید ۲۱۹
- ۳۱- بهترین سیستم اگزیر برای کنید ۲۲۰
- هفت دلیل رشد نکردن برنامه نویسی در ایران ۲۲۰
- ۱- مدارس ۲۲۱
- ۲- منابع فارسی محدود و قدیمی! ۲۲۱
- ۳- رعایت نکردن حقوق برنامه نویس ۲۲۲
- ۴- کمی برداری در روز روشن! ۲۲۲
- ۵- فریلنسری ۲۲۳
- ۶- ما اهل اوپن سورس و نرم افزار آزاد نیستیم! ۲۲۳
- ۷- تحریم ۲۲۴
- ۱۵ ویژگی یک برنامه نویس حرفه ای ۲۲۴
- ۱- مهارت‌های تکنیکی موثر ۲۲۵
- ۲- تمایل به یادگیری ۲۲۶

۲۲۶	۳- مهارت‌های اشکال زدایی
۲۲۷	۴- خود را با محیط کار وفق دهد
۲۲۷	۵- توانایی حل مشکل
۲۲۸	۶- شور و شوق برای کار
۲۲۸	۷- آرامش داشتن
۲۲۹	۸- مهارت‌های اجتماعی
۲۲۹	۹- تنبلی
۲۳۰	۱۰ چشم انداز کسب و کار
۲۳۰	۱۱- توانایی برنامه ریزی
۲۳۱	۱۲- توانایی مدیریت شکست
۲۳۱	۱۳- ذهنیت کارکنان
۲۳۱	۱۴- تمایل به پژوهش
۲۳۲	۱۵- احترام به مهلت داده شده (DeadLine)
۲۳۲	چگونه برنامه نویس خوبی باشیم
۲۳۹	فهرست منابع

مقدمه

اگرچه برنامه نویسی یک اختراع جدید محسوب می شود اما این در حالی است که ایده ی مجبور کردن یک ماشین یا دستگاه برای انجام یکسری دستورات خاص به سال ها پیش - چیزی در حدود یک قرن پیش - باز می گردد. در واقع، یکی از ابتدایی ترین طرح ها برای یک ماشین قابل برنامه ریزی - یا همان کامپیوتر - از کسی به نام Charles Babbage در سال ۱۸۰۲ گرفته شد.

در واقع وی پیشنهاد ساخت یک ماشین بخار که به صورت مکانیکی یکسری کارها را انجام دهد - که تحت عنوان موتور تحلیلیگر شناخته می شود - را داده بود.

برخلاف سیستم های محاسبه ی آن روز که تنها می توانستند یک عمل محاسباتی را انجام دهند، موتور تحلیلیگر آقای چارلز بابیج چندین کار همزمان را می توانست انجام دهد.

دستوراتی که به ماشین داده می شد از طریق یکسری کارت ها انجام می پذیرفت که با تغییر تعداد این کارت ها و نوع آن ها، هر کسی می توانست دستورات مد نظر خود را به ماشین برای محاسبه ی آن ها بدهد.

ایده‌ی ماشین‌های قابل برنامه‌ریزی توجه خانمی به نام Ada Lovelace را به خود جلب کرد که یک ریاضیدان بود. لذا او برنامه‌ای نوشت -یا بهتر است بگوییم کارت‌هایی ترتیب داد- که تعدادی عدد خاص را پشت سر هم چاپ می‌کرد (با توجه به این که این خانم اولین کسی بود که کار خاصی را با Analytical Engine یا همان موتور تحلیلگر انجام داد، وی را می‌توان به نوعی اولین برنامه‌نویس دنیا نیز نامیدی کرد که نامگذاری زبان برنامه‌نویسی Ada هم به احترام وی صورت گرفته‌ست).

چیزی پس از گذشت یک‌صد سال، اولین کامپیوتر به معنای واقعی کلمه در سال ۱۹۴۳ توسط ارتش ایالات متحده طراحی شد. این ماشین که Electronic Numerical Integrator and Computer یا به اختصار ENIAC نامیده شد، از کابل‌ها و سویچ‌های مختلفی تشکیل شده بود و برای دستور دادن به آن، برنامه‌نویس می‌بایست جای کابل‌ها و سویچ‌ها را تغییر دهد (جالب است بدانیم که اولین برنامه‌نویسان این کامپیوتر همگی خانم بوده‌اند!) نکته اصلی اینجا است که کامپیوترهای آن موقع بسیار حجیم و گران بودند و کار کردن با آن‌ها واقعا تخصص نیاز داشت!