

**دانشکده مهندسی کامپیوتر**

**گزارش پروژه کارشناسی**

**نرم‌افزار ترسیم نمودار مهندسی**

نگارش:

**محمد جواد رضوانیان**

استاد راهنما:

**دکتر علیرضا باقری**

استاد راهنما:

**دکتر مصطفی چهرقانی**

شهریور 1403

# 

# سپاسگزاری

در ابتدا از خداوند منان سپاسگزارم که همواره و در هر لحظه پشتیبان بندگان خود بوده و هر کجا که آنها در بن بست قرار گرفته‌اند، آنها را راهنمایی کرده. از خدایی سپاسگزارم که توفیق شاگردی نزد اساتید گرانقدر و بزرگواری را به من عطا کرد تا بتوان از ایشان در کنار درس‌های مرسوم امروزی، درس درست زندگی کردن و دغدغه‌مندی برای آینده‌ی کشورم را بیاموزم.

در ادامه از آقای دکتر علیرضا باقری استاد راهنمای پروژه و آقای دکتر صاحب‌الزمانی استاد مشاور عزیزم کمال تشکر و قدردانی را دارم که با راهنمایی‌ها و دلسوزی‌های این اساتید بزرگوار امکان تهیه این گزارش میسر شد.

محمد جواد رضوانیان

شهریور 03

# چکیده

نمودارها ابزارهای بصری قدرتمندی هستند که به مهندسان این امکان را می‌دهند تا داده‌ها، فرآیندها، و روابط پیچیده را به‌صورتی ساده و قابل فهم نمایش دهند. اهمیت نمودارها در مهندسی به دلیل توانایی آنها در بهبود درک مفاهیم فنی است. به کمک نمودارها، می‌توان اطلاعات فنی را به‌صورت گرافیکی ارائه داد، الگوهای داده‌ای را شناسایی کرد، و مشکلات را سریع‌تر تحلیل و حل کرد. نمودارها همچنین در مستندسازی پروژه‌ها، تسهیل تصمیم‌گیری و بهینه‌سازی طراحی‌ها نقش حیاتی ایفا می‌کنند، و به این ترتیب به افزایش دقت، کارایی، و نوآوری در فرآیندهای مهندسی منجر می‌شوند. مسئله اصلی استفاده از آن‌ها، پیچیده بودن ترسیم‌ها در لایه‌ی مهندسی است؛ اعداد و ارقام در این لایه بسیار زیاد و یا بزرگ بوده و یا برخی توابع و الگوها ترسیم‌شان به راحتی توسط انسان ممکن نیست. به طور کلی، نرم‌افزارهای ترسیم نمودارهای مهندسی به دلیل سرعت، دقت و قابلیت‌های پیشرفته‌شان به یک ابزار ضروری در هر پروژه مهندسی تبدیل شده‌اند.

با توجه به اهمیت این ابزار در رشته‌های مهندسی این سوال به وجود می‌آید که «آیا با استفاده از مفاهیم موجود در رشته‌ی مهندسی نرم‌افزار می‌توان یک نرم‌افزار استاندارد با استفاده از روش‌های چابک[[1]](#footnote-1) طراحی و پیاده‌سازی کرد که نیازهای تخصصی افراد را برطرف کند؟ هزینه تولید چنین نرم‌افزاری بدون استفاده از روش‌های مهندسی نرم‌افزار چه قدر اختلاف با این روش دارد و آیا لزومی دارد که این کار توسط یک تیم آموزش دیده انجام شود؟ در پروژه‌ی به این ابعاد آیا استفاده از روش‌های آزمون نرم‌افزار می‌تواند کارآمد باشد و کیفیت کار و میزان قابل اعتماد بودن نرم‌افزار را افزایش دهد؟»

برای پاسخ به این سوال ابتدا باید مفاهیم پایه‌ای برای تولید چنین نرم‌افزاری را بررسی کنیم و سپس با استفاده از روش چابک در هر چرخه کاری[[2]](#footnote-2)، فرایند توسعه و پیاده‌سازی را متناسب با نیاز مشتری - که یکی از شرکت‌های دانش‌بنیان است - انجام دهیم.

**واژه‌های کلیدی:** چابک، چرخه‌ی کاری، ترسیم، نگهداری اطلاعات، احراز هویت، نگهداری اطلاعات.

# فهرست مطالب

عنوان صفحه

[سپاسگزاری ت‌](#_Toc175283426)

[چکیده أ‌](#_Toc175283427)

[فهرست مطالب ب‌](#_Toc175283428)

[فهرست اشکال ث‌](#_Toc175283429)

[1. مقدمه 1](#_Toc175283432)

[1-1 مقدمه 2](#_Toc175283433)

[2. روش چابک 4](#_Toc175283435)

[1-2 تعریف روش چابک 5](#_Toc175283436)

[2-2 تاریخچه‌ای کوتاه 5](#_Toc175283437)

[3-2 عملکرد 5](#_Toc175283438)

[نتیجه گیری 5](#_Toc175283439)

[3. استخراج نیازمندی‌ها 6](#_Toc175283441)

[1-3 نمودارهای مورد استفاده 7](#_Toc175283442)

[2-3 طراحی نمونه‌ی کاغذی 7](#_Toc175283443)

[3-3 معماری انتزاعی سامانه 7](#_Toc175283444)

[4-3 تجزیه اجزای پروژه 7](#_Toc175283445)

[5-3 طراحی آزمون‌های TDD 7](#_Toc175283446)

[نتیجه‌گیری 7](#_Toc175283447)

[4. طراحی و پیاده‌سازی 8](#_Toc175283449)

[1-4 رابط کاربری 9](#_Toc175283450)

[2-4 پیاده‌سازی رابط کاربری 9](#_Toc175283451)

[**1-2-4 ویرایشگر متن** 9](#_Toc175283452)

[2-2-4 لیست فرمول‌ها 9](#_Toc175283453)

[3-2-4 روابط از پیش‌نوشته شده 9](#_Toc175283454)

[3-4 ساختمان داده‌ها 9](#_Toc175283455)

[نتیجه‌گیری 9](#_Toc175283456)

[تحلیل و بررسی میزان بهره وری 10](#_Toc175283458)

[1-5 ذخیره‌سازی اطلاعات 11](#_Toc175283459)

[2-5 بررسی آزمون‌های واحد 11](#_Toc175283460)

[نتیجه‌گیری 11](#_Toc175283461)

[نتیجه‌گیری و پیشنهادات 12](#_Toc175283463)

[نتیجه‌گیری 13](#_Toc175283464)

[پیشنهادات 13](#_Toc175283465)

[مراجع 14](#_Toc175283466)

# فهرست اشکال

## عنوان صفحه

[*شکل 1-1 میزان درآمد یک کارشناس حوزه هوش تجاری در شرکت‌های مختلف در سال 2021* 2](#_Toc175233657)

[*شکل 1-2 دید ابتدایی نسبت به هوش تجاری* 8](#_Toc175233658)

[*شکل 3-1 معماری اولیه هوش تجاری* 10](#_Toc175233659)

[*شکل 4-1 مثالی از الگوریتم RLE* 18](#_Toc175233660)

# **فصل اول**

# 1. مقدمه

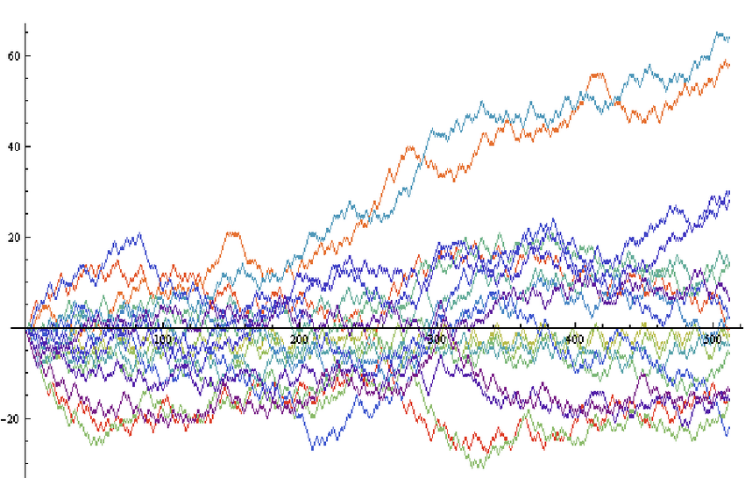
## 1-1 مقدمه

تاریخچه استفاده از نمودارها به دوران باستان بازمی‌گردد، زمانی که انسان‌ها برای درک و نمایش داده‌ها و مفاهیم مختلف به استفاده از تصاویر و نمادها روی آوردند. در طول قرون‌وسطی و رنسانس، استفاده از نمودارها به ویژه در زمینه‌های نجوم، جغرافیا، و ریاضیات پیشرفت چشمگیری داشت. با شروع انقلاب صنعتی در قرن هجدهم، نمودارها به ابزارهای مهمی در مهندسی، اقتصاد، و علوم تبدیل شدند. توسعه نمودارهای مدرن مانند نمودارهای خطی، ستونی، و دایره‌ای در قرن نوزدهم توسط افرادی مانند ویلیام هنری پلیفر[[3]](#footnote-3) به استانداردی در تجزیه و تحلیل داده‌ها و نمایش تصویری اطلاعات تبدیل شد. [1]

نمودارها در مهندسی نقش بی‌بدیلی دارند، زیرا امکان تجسم داده‌ها و تحلیل مفاهیم پیچیده را به‌صورت بصری فراهم می‌کنند. با استفاده از نمودارها، مهندسان قادرند تا الگوهای پنهان در داده‌ها را شناسایی کرده، روابط میان متغیرها را درک کنند و نتایج تحلیل‌ها را به‌طور موثر به دیگران منتقل نمایند. این ابزارهای گرافیکی، از طراحی سیستم‌های پیچیده تا مدیریت پروژه‌های بزرگ، در تمامی مراحل مهندسی کاربرد دارند. به‌ویژه در فرآیند طراحی و توسعه، نمودارها به مهندسان کمک می‌کنند تا ایده‌ها را به مدل‌های عملیاتی تبدیل کرده و از طریق شبیه‌سازی و تجزیه و تحلیل، بهینه‌سازی‌های لازم را انجام دهند. این امر منجر به افزایش دقت، کاهش خطا و تسهیل تصمیم‌گیری در پروژه‌های مهندسی می‌شود.

با پیشرفت دانش در حوزه علوم کامپیوتری، مختصصان این حوزه به این فکر افتادند که آیا این امکان وجود دارد که بتوان از فناوری‌های این حوزه در عرصه‌ی اقتصاد و کسب و کار هم استفاده کرد؟ و در اینجا ایده‌ی اولین سامانه‌ی هوشمندی کسب و کار به وجود آمد. سامانه‌ای که هدف آن آسان‌سازی فرایند جمع‌آوری، دسته‌بندی و تحلیل داده‌های مربوط به یک حوزه کاری است.

ترسیم نمودارها به‌وسیله انسان، به‌ویژه زمانی که با داده‌ها و سیستم‌های پیچیده سروکار داریم، می‌تواند فرآیندی چالش‌برانگیز و زمان‌بر باشد. دقت در ترسیم روابط پیچیده و نمایش صحیح داده‌ها نیازمند مهارت بالا و توجه دقیق به جزئیات است. هر گونه اشتباه در این فرآیند ممکن است به برداشت‌های نادرست و تصمیم‌گیری‌های اشتباه منجر شود. علاوه بر این، حجم بالای داده‌ها و نیاز به به‌روزرسانی مداوم نمودارها می‌تواند این کار را بسیار طاقت‌فرسا کند. از سوی دیگر، در برخی حوزه‌های مهندسی، نمودارها باید نه‌تنها دقیق بلکه کاملاً متناسب با استانداردهای تخصصی باشند که این امر نیاز به آشنایی کامل با قوانین و اصول طراحی دارد. به همین دلیل، ترسیم دستی نمودارها به‌خصوص در پروژه‌های بزرگ و پیچیده، علاوه بر خطاهای احتمالی، می‌تواند زمان و منابع زیادی را مصرف کند.



تصویر 1-1 نمونه ای از یک نمودار پیچیده

این نرم‌افزار با استفاده از متدولوژی چابک پیاده‌سازی شده است که باعث شده فرآیند توسعه آن به صورت تدریجی و انعطاف‌پذیر انجام گیرد. در این روش، پروژه به تعدادی چرخه یا چرخه کاری کوتاه تقسیم می‌شود که هر یک با هدف ارائه یک نسخه قابل‌استفاده از نرم‌افزار به پایان می‌رسد. این رویکرد امکان می‌دهد تا تیم توسعه به سرعت به بازخوردهای کاربران پاسخ دهد و به بهبود مداوم نرم‌افزار بپردازد. همچنین، استفاده از چابک تضمین می‌کند که نرم‌افزار با نیازهای متغیر کاربران و تغییرات محیط کسب‌وکار هماهنگ باشد و به‌روزرسانی‌های منظم و موثر انجام گیرد.

به منظور تضمین امنیت و محافظت از اطلاعات از الگوی احراز هویت در پروژه‌ استفاده شده است. احراز هویت به این معناست که تنها کاربران مجاز به دسترسی به روابط از پیش‌نوشته‌شده‌ی خود هستند و از دسترسی غیرمجاز آنها به روابط دیگران جلوگیری شود. این فرآیند نه تنها برای حفاظت از اطلاعات ضروری است، بلکه به ایجاد اعتماد میان کاربران و سیستم نیز کمک می‌کند.

هدف ما در این پروژه، ساخت یک نرم‌افزار ترسیم نمودار با استفاده از استانداردهای مهندسی نرم‌افزار و روش چابک است. اینکه بتوانیم با استفاده از چارچوب مطرح شده یک نرم‌افزار ترسیم نمودار طراحی و پیاده‌سازی کنیم و میزان هزینه‌ی آن را با زمانی که از این الگو پیروی نکنیم مقایسه کنیم.

در همین راستا در این گزارش، ابتدا بعد از تعریف مفاهیم اولیه‌ی به کار رفته از مهندسی نرم‌افزار در این پروژه، نگاهی اجمالی به ساختار چرخه‌های کاری می‌اندازیم. بعد از آن به بررسی هر یک از این چرخه‌ها پرداخته و در نهایت، عملکرد سیستم و بهره‌وری آن را بررسی می‌کنیم.

# **فصل دوم**

# 2. روش چابک

## 1-2 تعریف روش چابک

.

.

.

## 2-2 تاریخچه‌ای کوتاه

»

.

## 3-2 عملکرد

.

.

## نتیجه گیری

.

.

# **فصل سوم**

# 3. استخراج نیازمندی‌ها

## 1-3. نمودارهای مورد استفاده[[4]](#footnote-4)

.

.

## 2-3. طراحی نمونه‌ی کاغذی[[5]](#footnote-5)

.

.

## 3-3. معماری انتزاعی سامانه

.

.

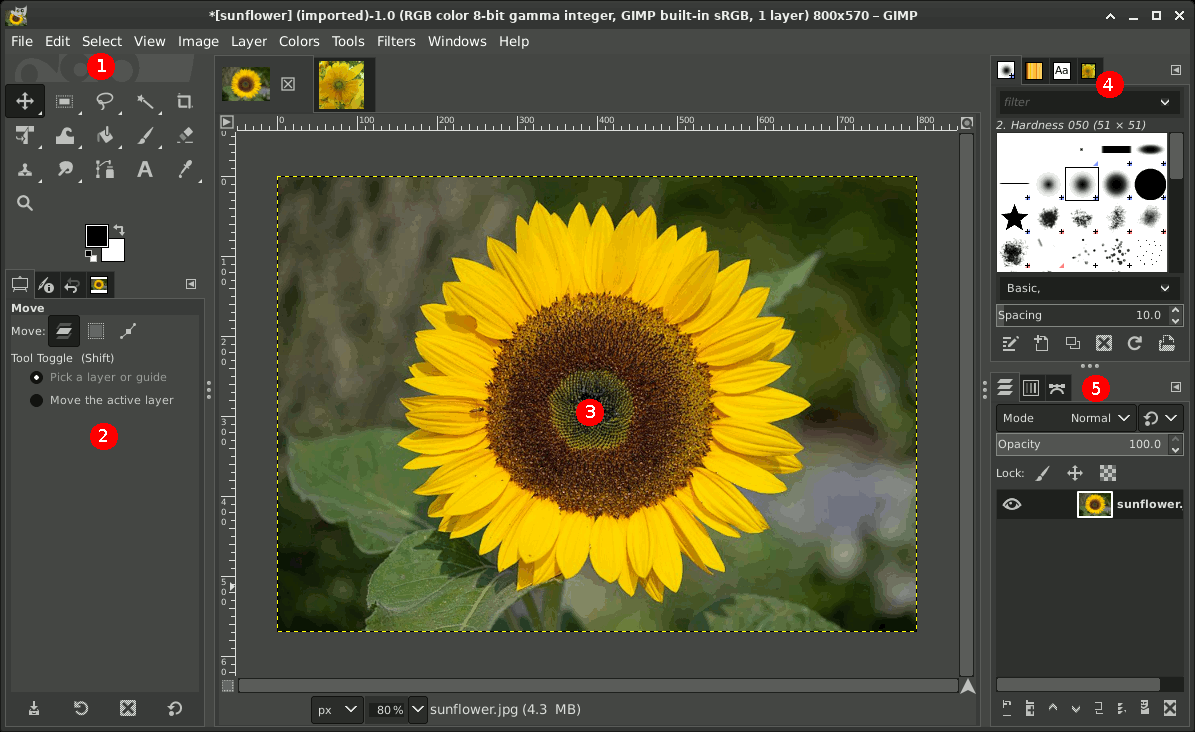
## 4-3. تجزیه اجزای پروژه

### 1-4-3. انتخاب زبان برنامه نویسی

مرحله‌ی نخست انتخاب زبان برنامه‌نویسی مناسب برای انجام پروژه است. برای اینکار باید زبان‌هایی که دارای رابط کاربری هستند را مورد نقد و بررسی قرار دهیم تا ببینیم کدام یک برای انجام پروژه‌ی ما مناسب هستند.

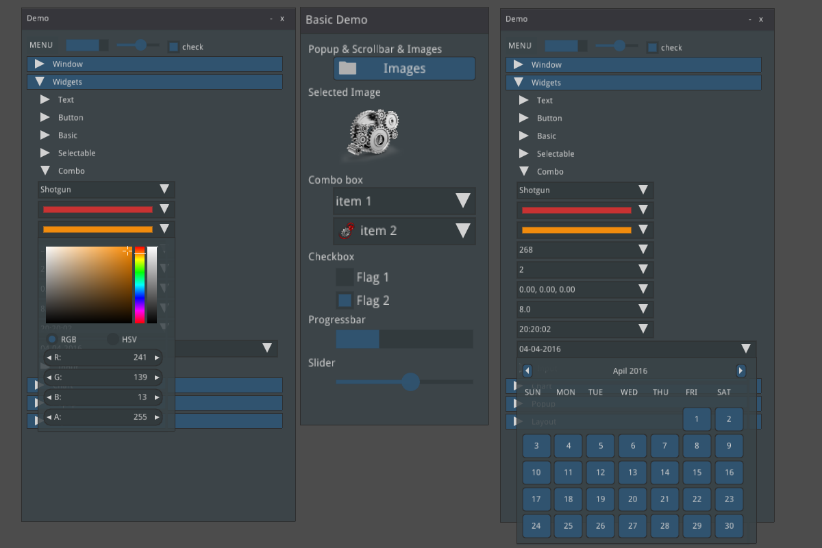
#### 1-1-4-3. زبان C و C++

در زبان‌های برنامه‌نویسی C و ++C، کتابخانه‌های مختلفی وجود دارد که می‌توان از آن‌ها برای طراحی رابط‌های کاربری استفاده کرد. یکی از معروف‌ترین آن‌ها کتابخانه GTK است. از این کتابخانه برای توسعه نرم‌افزارهای مختلفی مثل GIMP، Key sign و بسیاری از ابزارهای سیستم عامل لینوکس استفاده شده است.



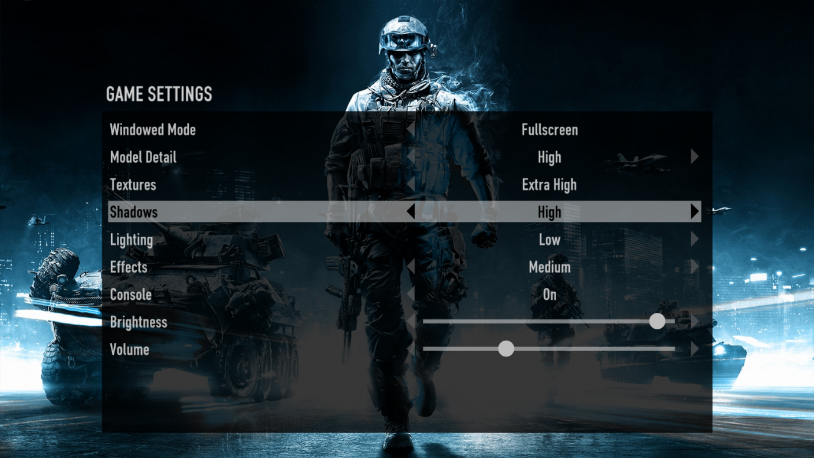
*تصویر 2 نمونه‌ای از فضای نرم‌افزار GIMP*

یکی از کتابخانه‌های این زبان، کتابخانه‌ی Nuklear است. این کتابخانه بدلیل آنکه از کتابخانه‌های رسمی نمی‌باشد، فاقد منابع غنی برای برخطایابی[[6]](#footnote-6) بوده و در صورت بروز هرگونه اشتباه یا مشکل در پیاده‌سازی رابط کاربری، زمان بسیار زیادی صرف درست کردن آن خواهد شد.



*تصویر 3 نمونه ای از فضای نرم‌افزار نوشته شده با کتابخانه‌ی Nuklear*

از مشکلات استفاده از Nuklear، می‌توان به عدم لایه‌ای بودن آن اشاره کرد و اینکه این کتابخانه به صورت اجزاء[[7]](#footnote-7) را قرار می‌دهد و امکان طراحی انتزاعی را از طراحان می‌گیرد که این کار توسعه را بسیار سخت و زمان‌بر می‌کند. از پروژه‌هایی که با این کتابخانه توسعه داده شده‌اند می‌توان به بازی Battlefield اشاره کرد.



*تصویر4 فضای بازی Battle Field*

#### 2-1-4-3. زبان جاوا

در زبان جاوا کتابخانه‌های زیادی برای طراحی رابط کاربری وجود دارد که می‌توان به موارد زیر اشاره کرد:

* JavaFX
* Swing
* AWT (Abstract Window Toolkit)
* SWT (Standard Widget Toolkit)
* JGoodies Forms
* JIDE Common Layer

کتابخانه‌ی JavaFX یکی از قدرتمندترین کتابخانه‌ها در زمینه ترسیم و به تصویرسازی در زبان جاوا است؛ اما بدلایلی که در آینده به آن‌ها خواهیم پرداخت (پردازش‌های محاسباتی) از زبان پایتون استفاده می‌کنیم.

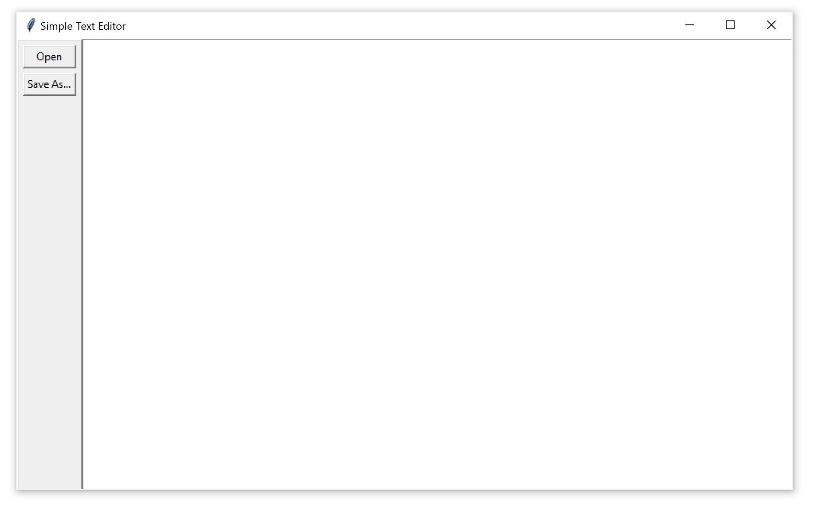
بعضی از کتابخانه‌ها بدلیل کم‌بود منابع (JGoodies Forms و JIDE Common Layer) و باقی بدلیل قدیمی شدن و عدم کاربرپسند بودن رابط‌های کاربری آن‌ها و هزینه زیاد بابت ترمیم این نقص از لیست پیشنهادات ما حذف می‌شوند.

#### 3-1-4-3. زبان پایتون

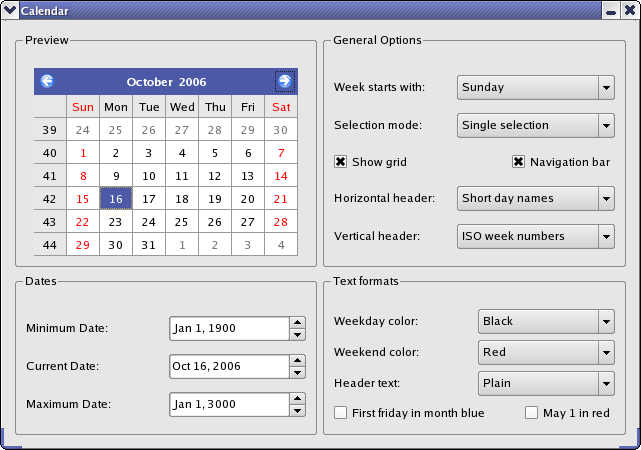
زبان برنامه نویسی پایتون، دارای پنج کتابخانه‌ی معروف برای طراحی رابط کاربری است:

* Tkinter
* PyQt
* PySide
* Kivy
* wxPython

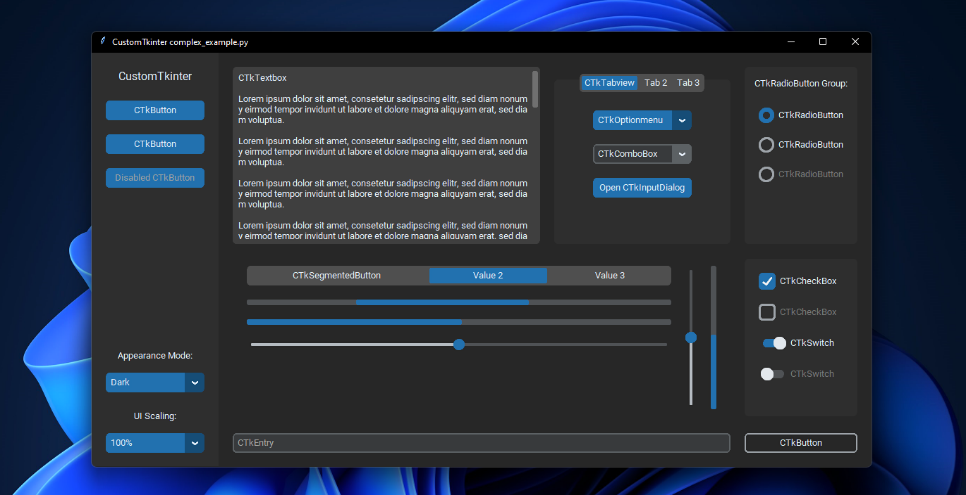
یکی از کتابخانه‌های غیر معروف زبان پایتون، کتابخانه‌ی Custom-Tkinter است که از آن می‌توان برای طراحی رابط‌های کاربری حرفه‌ای استفاده کرد. در ادامه به معرفی کتابخانه‌های مذکور خواهیم پرداخت.



*تصویر 5 نمونه ساخته شده به وسیله Tkinter*

ا

*تصویر 6 نمونه ساخته شده به وسیله Qt*



*تصویر 7 نمونه ساخته شده توسط کتابخانه Custom-Tkinter*

## 5-3 طراحی آزمون‌های TDD

.

.

## نتیجه‌گیری

.

.

|  |  |  |  |
| --- | --- | --- | --- |
| زبان برنامه نویسی | رفع اشکال و اصلاح | قدرت محاسباتی | زیبایی |
| C/C++ | سخت | متوسط | بله |
| Java | ساده | پایین | بله |
| Python | ساده | بسیار بالا | بله |

# **فصل چهارم**

# 4. طراحی و پیاده‌سازی

## 1-4 رابط کاربری

.

.

## 2-4 پیاده‌سازی رابط کاربری

.

.

### **1-2-4 ویرایشگر متن[[8]](#footnote-8)**

.

.

### 2-2-4 لیست فرمول‌ها

.

### 3-2-4 روابط از پیش‌نوشته شده

.

.

## 3-4 ساختمان داده‌ها

.

.

## نتیجه‌گیری

.

.

# فصل پنجم

# تحلیل و بررسی میزان بهره وری

## 1-5. ذخیره‌سازی اطلاعات

.

.

## 2-5. بررسی آزمون‌های واحد

.

.

## نتیجه‌گیری

.

.

# فصل ششم

# نتیجه‌گیری و پیشنهادات

## نتیجه‌گیری

.

.

## پیشنهادات

.

.

# مراجع

[1]. Tufte, Edward .R. *The Visual Display of Quantitative Information.* 1983.

[2]. Friendly, Michael. "A Brief History of Data Visualization." (2008).

[3]. Harvey Deitel, Paul J. Deitel, Jonathan P. Liperi, Ben Wiedermann. Python How to Program (Parts a & B). PAP/CDR ed. Vol. 1: Pearson, February 14, 2002, 2002.

[4]. Roger S. Pressman, Bruce Maxim. Software Engineering: A Practitioner's Approach. 1 vols. Vol. 1: McGraw Hill, January 23, 2014, 2019. Engineering.

[5]. Rubin, Kenneth. Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)). 1 st ed. Vol. 1: Addison-Wesley Professional, July 26, 2012.

1. Agile [↑](#footnote-ref-1)
2. Sprint [↑](#footnote-ref-2)
3. William Henry Playfair [↑](#footnote-ref-3)
4. Use Case [↑](#footnote-ref-4)
5. Sketch [↑](#footnote-ref-5)
6. Debugging [↑](#footnote-ref-6)
7. Modular [↑](#footnote-ref-7)
8. Text Editor [↑](#footnote-ref-8)