



Object-Oriented Conference 2024

Object Oriented Conference 2024 公式ガイドブック

2024-03-24 初版第1刷 OOC 2024 実行委員会 発行

はじめに -OOC 2024 開催によせて

OOC2024 にお越しくださり、ありがとうございます。
ariaki 酸が何か書く

2024 年 3 月 24 日
Object Oriented Conference 実行委員会
代表 森川晃

免責事項

- 本書の内容は、情報提供のみを目的としています。正確性には留意していますが、必ずしも保証するものではありません。この本の記載内容に基づく結果について、著者・編集者とも一切の責任を負いません。
- 会社名・商品名については、一般に各社の登録商標です。TM 表記等については記載していません。また、特定の会社・製品などについて、不当に貶める意図はありません。
- 本書の一部あるいは全部について、無断での複写・複製はお断りします。

目次

はじめに -OOC 2024 開催によせて	3
第Ⅰ部 ご案内	7
第1章 開催概要	8
1.1 理念と目的	8
1.1.1 OOC 実行委員会	8
1.2 開催概要	9
1.3 参加者向けご案内	10
1.3.1 会場アクセス	10
1.3.2 来場予約について	10
1.3.3 体調管理について	10
1.3.4 スタッフ巡回について	10
1.3.5 撮影・録画	11
1.4 注意事項	12
1.4.1 お子様連れの参加者の方へ	12
1.4.2 参加者の皆様へ	12
1.4.3 緊急時対応	12
第2章 行動規範	13
2.1 はじめに	13
2.2 改版履歴	13
2.3 行動規範	13
2.3.1 概要	13
2.3.2 本則	13
2.4 行動規範違反について	16
2.5 この行動規範の問い合わせ先	16
2.6 この行動規範のライセンス	16
第3章 協賛企業様のご紹介	17
3.1 ゴールドスポンサー	18

3.2	シルバースポンサー	18
第Ⅱ部 タイムテーブル・セッション一覧		19
第4章	サークル一覧（配置順）	20
第5章	サークルアピール	23
第Ⅲ部 協賛企業様による寄稿記事		26
第6章	君も文豪にならないか？	27
6.1	タイマーです。名前だけでも覚えて帰ってください。	27
6.2	エンジニアの成長を支援する「DevEnable 室」	27
6.2.1	1. 個人の力の向上 学習・環境に対する支援	27
6.2.2	2. 深い挑戦のある打席 事業成果の創出	27
6.2.3	3. 実体験をもとにした再現性のある言語化 キャリア・コミュニティへの還元	28
6.3	ユニークな「文豪待遇制度」	28
6.3.1	専属編集	30
6.3.2	執筆活動費 全額負担	30
6.3.3	資料デザイン全面サポート	31
6.3.4	文豪缶詰プラン	31
6.3.5	文豪締切催促サービス	32
6.3.6	文豪差し入れサービス	32
6.4	君も文豪にならないか？	33
第7章	SOLID原則をギュッとまとめた	34
7.1	ギュッとまとめました。	34
7.2	解説	35
7.3	人間はミスをする生き物	36
第8章	コードレビューガイドラインの導入と背景	37
8.1	Design（設計）	37
8.1.1	定義	37
8.1.2	具体例	38
8.2	Simplicity（理解容易性）	40
8.2.1	定義	40
8.2.2	具体例	40

目次

8.3	Naming (命名)	41
8.3.1	定義	41
8.3.2	具体例	41
8.4	Style (コードスタイル)	41
8.4.1	定義	41
8.4.2	具体例	41
8.5	Functionality (機能要求)	41
8.5.1	定義	41
8.5.2	具体例	42
8.6	Test (テスト)	42
8.6.1	定義	42
8.6.2	具体例	42
8.7	Document (文章)	42
8.7.1	定義	42
8.7.2	具体例	42
8.8	指摘対応の要否	43
8.8.1	具体的な利用方法	43
8.8.2	指摘例	43
8.9	最後に	43
第Ⅳ部 イベントを支える技術-スタッフ寄稿—		44
第9章 搬入・宅急便関係の準備いろいろ		46
9.1	事前準備	46
9.1.1	事務局の事前準備	46
9.1.2	一般サークルの事前準備	48
9.2	まとめ	48
9.3	次回予告	49
スタッフ紹介		50
コアスタッフ		50
デザイン		51

第1部

ご案内

第 1 章

開催概要

1.1 理念と目的

目的と理念

1.1.1 OOC 実行委員会

私たち技書博運営事務局は、有志のボランティアによって成り立っています。

少ない人数で試行錯誤を繰り返しながら皆さまにより良い価値を提供できればと考え、日夜精一杯がんばっています。もし至らないところがありましたら、どうかご容赦いただき、改善点を遠慮なくご教示ください。イベントを通じて私たちも成長し、皆さまと共によいエンジニアコミュニティを築けることを願います。

また、私たちと一緒に技書博の楽しさを作っていく仲間を募集しています。興味のある方はお近くのスタッフにお声がけいただくか、公式 Twitter アカウントまでご連絡ください。

1.2 開催概要

催事名

Object Oriented Conference

催事略称

OOC 2024

開催日時

2024 年 3 月 24 日（日）11:00～16:00

開催会場

お茶の水女子大

会場住所

〒112-8610 東京都文京区大塚 2-1-1

開催規模

トーク 40 本参加者 1000 人 (想定)

運営団体名

OOC 実行委員会

公式 Web サイト

<https://ooc.dev/>

公式 Twitter アカウント

<https://twitter.com/ooc>

ハッシュタグ

#OOC_2024

1.3 参加者向けご案内

1.3.1 会場アクセス

<https://www.ocha.ac.jp/access/index.html>

- ・ 東京メトロ丸ノ内線「茗荷谷」駅より徒歩 7 分
- ・ 東京メトロ有楽町線「護国寺」駅より徒歩 8 分
- ・ 都営バス「大塚 2 丁目」停留所下車徒歩 1 分

1.3.2 来場予約について

受付・予約について。

ただし、各種ノベルティは先着順で配布しますので、遅い時間にご来場された場合にはお渡しきれない場合があることをご了承ください。

1.3.3 体調管理について

新型コロナウイルス、インフルエンザなどの流行が予想されます。体調のすぐれない方は無理をしないようにして下さい。

体調不良時は近くのスタッフまで遠慮なくお声がけください。

1.3.4 スタッフ巡回について

会場内の安全確保のため、スタッフが常時巡回します。会場内では、スタッフの誘導にしたがってください。

不審物、トラブル、迷惑、危険行為、不審な行動をする者等に気づいた場合は、すぐにお近くのスタッフまでご連絡ください。その他にも不明点があれば、事務局・スタッフまで遠慮なくご相談ください。地震・火災・テロなどの際は、スタッフの誘導にしたがって、慌てず避難ください。

1.3.5 撮影・録画

会場内にて撮影・録画を行う場合は、個別に被撮影者の許可を得てください。会場には不特定多数の方が来場するため、十分な配慮をお願いします。不審な点がみられる場合、運営事務局が撮影内容を確認いたします。なお、運営スタッフがイベント開催資料のため、会場内を撮影する場合があります。

1.4 注意事項

1.4.1 お子様連れの参加者の方へ

- 迷子、ケガなどのトラブルを防止するため、絶対に目を離さないようにしてください。
- 会場の出入りにあたっては、必ず手をつないで出入りください。
- 迷子等防止のため出入口スタッフにて確認・声掛けする運用としています。
- 混雑が予想されます。他の参加者の目線に入らないことがあるため、混雑している区画を避ける、周囲に声掛けをするなど、お子様の安全確保にご留意ください。
- 午前中は混雑が予想されます！ お子様連れの参加者の方は、比較的空いている午後にご参加いただくのがおすすめです。
- 申し訳ありませんが、スペースの都合上、会場にベビーカー置き場を用意することが叶いませんでした。
- 通路が狭くなってしまっている部分もございますので、ベビーカーでの参加は非推奨となっております。ご理解いただけますと幸いです。
- 3歳以下のお子様連れの参加者のみなさんは、通路でぶつかったりする危険を踏まえ、お子様から目を離さないようお気をつけください。
- 運営事務局として、全力でサポートするように努めますが、あらゆるトラブルは故意の有無に関わらず、運営事務局はその責を免れるものとします。

1.4.2 参加者の皆様へ

- 背の低いお子様が歩いていることがあります！
- 足元にご注意いただき膝下や荷物などがぶつからないようご注意ください。
- ひとりでいるお子様に気付かれた場合は、スタッフにお声がけください。全力で対応します。
- その他トラブル、迷子等に気づいた場合、直ちに運営事務局、会場スタッフにご連絡ください。

1.4.3 緊急時対応

迷惑行為・危険行為等に遭遇した場合、お近くのスタッフにご連絡ください。お近くにスタッフが見つからない場合、受付スタッフにお声がけください。地震・火災・テロなど災害発生時はスタッフが誘導いたします。慌てずスタッフの指示に従ってください。

行動規範

2.1 はじめに

技術書同人誌博覧会（以下、「当会」といいます）は、すべてのエンジニアが自身の知見を共有することで成長と幸福をもたらすことをコミュニティ理念とし、この使命を実現するためにコミュニティ活動を行います。当会の理念に基づき、すべての関係者（参加者・登壇者・出展者・スタッフ・スポンサーを含む）の判断の拠り所や取るべき行動を定めたものが、行動規範です。すべての関係者がこの行動規範を理解・遵守し、高い倫理観をもって、誠実で公正に行動することを望みます。この行動規範は、当会が主催するイベント会場やイベントに関する活動、オンラインでのコミュニケーション空間など、当会のあらゆる活動に対して適用されます。当会の主催者ならびにスタッフは、関係者に対してこの行動規範の遵守を徹底することで誰にとっても安全かつ安心な環境を確約し、違反者に対してはあらゆる手段をもって排除に努めます。

2.2 改版履歴

- 2019-08-07 第 1.0.0 版 公開

2.3 行動規範

2.3.1 概要

私たちは、自らのとった行動や発言に責任をもち、常に他者に敬意と礼節をもって接します。また、特定の個人や団体に対する攻撃的な発言を慎み、あらゆるハラスマントや差別を排除し、安全かつ安心に過ごすことができるよう徹底します。悪質な勧誘やなりすまし行為、個人情報の不正利用、性的コンテンツの掲示などの迷惑行為を許さず、反社会的勢力の構成員または協力者を加入させません。本規範の違反者に対して主催者はあらゆる手段を講じて排除し、円滑なコミュニティ運営に努めます。

2.3.2 本則

敬意を持った行動の徹底

私たちは、コミュニティ参加を通じて、常に他者への敬意と礼節をもって接します。また、特定の個人または団体に対する暴言や誹謗中傷など貶める発言を慎み、他者への攻撃的な言動を行わな

いように務めます。とくにイベントにおいては、運営者・登壇者・出展者などに対して進行を妨げないよう注意するとともに、攻撃的発言を慎み、円滑な運営に協力します。

ハラスメント行為の禁止

私たちは、他者を許容し、他者の人格を尊重する為、あらゆるハラスメント行為をしないよう徹底します。他者の人格とは、性差・性自認と表現・性指向・障がい・容姿・外見や身体的特徴・年齢・健康状態・人種・民族・出身国・宗教・政治・思想などあらゆることがらを指し、すべての人があつアイデンティティを尊重します。また、ここでのハラスメントとは、公的空間での性的な画像や類する表現・ナンパ行為（容姿に関する発言、恋愛・性的興味を目的とした発言）・誹謗・中傷・脅迫・暴力・暴力の助長・威力行為・ストーキングやつきまとい・不適切な身体的接触・写真撮影や録音によるいやがらせ・コミュニティ運営に対して繰り返し中断や混乱を目的とした行為、および社会通念に照らし嫌がらせと認められる一切の言動を指します。私たちは、他者に対する威嚇、品位を貶める行為など、敵対的な状況を生み出す言動を控え、自らのとった行動や発言に責任をもちます。

エンジニア尊重主義

私たちは、すべてのエンジニアがあつスキルや技術、職業や職責、および、その背景と選択を尊重します。初心者と上級者、知っていることと知らないこと、使用する言語やフレームワークなどの技術的素地によって差別されるべきではありませんし、あらゆる技術や職業に貴賤はありません。

勧誘の禁止

私たちは、コミュニティを健全に運営するため、一切の勧誘行為を禁止します。ここでの勧誘とは、営利を目的とした宣伝や営業、エンジニアコミュニティ以外への誘導、政治活動、宗教活動、求人募集など、当会の目的と直接合致しない行為を指します。ただし、次の場合はその限りではありません。

1. 相手方が直接望まれる、個人間での情報提供
2. 協賛企業による宣伝告知
3. その他、主催者が特別に認めるもの

なりすましの禁止

私たちは、他者へのなりすましを許容しません。ここでのなりすましとは、自身以外の特定の人物、自身が所属しない団体、自身が組織内で意思決定権を持つように見せかける行為など、あらゆる詐称行為を含み、なりすましを用いたあらゆる言動を禁止します。

性的コンテンツの禁止

私たちは、公共空間において性的コンテンツを掲示しません。ここでの性的コンテンツとは、裸体表現・ポルノグラフィー・わいせつ行為・性行為など、性を連想する画像、映像や文章表現、あるいはその他の物品を指します。また、性的な関心を引き起こすような服装・制服・コスチュームを使うべきではありませんし、その他の手段で性的な関心を引き起こすような環境を作ることもしません。

個人情報不正利用の禁止

私たちは、他者のプライバシーを侵害せず、個人情報を不正に利用しません。当会への参加・関与によって得られた個人情報を外部へ共有または持ち出すことは固く禁じます。

主催者は、当会の円滑な運営および業務遂行に必要と判断する場合のみ、外部業者またはサービスに対して個人情報の一部または全部を委託する場合があります。

反社会的勢力の排除

私たちは、自ら反社会勢力の構成員または協力者でない事を表明および確約します。ここでの反社会勢力とは、暴力団、暴力団関係企業、総会屋若しくはこれらに準ずる者又はその構成員ではないことを指します。また、あらゆる犯罪や暴力に対して毅然たる態度で臨み、その要求には一切応じません。

撮影や録画などの記録

私たちは、写真撮影・動画録画・音声録音など（以下、記録）において必ず被記録者に許可をとり、原則として私的利用にとどめます。撮影時は相手の配慮を忘れず、また相手や主催者から撮影中止を命じられた場合には速やかに従います。

なお、当会のイベントは、主催者・スタッフ・スタッフに委託された者によって記録し、参加者に対して同意を得たうえでコンテンツを公開する場合があります。

著作権保護

コミュニティより提供されるすべてのコンテンツは、提供者（主催者・登壇者・出展者など）に著作権が帰属します。私たちは、著作権者を尊重し、厳格な著作権保護に努めます。

最善の努力

私たちは、行動規範を遵守し、誇り高く誠実で公正な行動に努めます。当会におけるすべての関係者は、主催者とともに最善の努力を尽くすことで、誰にとっても安全かつ安心な環境を作りだします。私たちの行動指針は、常に「Don't be evil（邪悪になるな）」かつ「Do the right thing（正しい行動をとろう）」であり続けます。

2.4 行動規範違反について

行動規範違反者への対処

主催者は、違反行為を行った者に対する警告・退出指示・参加停止・追放などを含むあらゆる処置を行う権利を有し、適切と判断した範囲で行います。また、必要に応じて法的機関に相談し、早急な解決と被害者の保護に努めます。

行動規範違反を発見した場合

もしあなたが行動規範違反を発見した場合、あなたや他の人がハラスメントに遭っている場合、あなたや他の参加者に対して危険または不寛容な態度を示された場合、その他に懸念や質問がある場合は、すぐに主催者またはスタッフに連絡してください。お知らせ頂いた内容は、法的な情報開示の必要がある場合を除いて、あなたの同意がない限り匿名で扱いますのでご安心ください。スタッフは、ハラスメントを経験した人がイベント会場内および往復路において安全かつ安心を感じられるよう、警察への通報連絡、付き添い者の提供、およびその他のあらゆる支援を行います。

行動規範違反を指摘された場合

もしあなたが行動規範違反を指摘された場合、ただちに主催者の指示にしたがってください。もし不適切な行動が続いた場合、主催者は全参加者に安全かつ快適な環境を維持するため、あなたに追放を含むあらゆる処置を行う場合があります。

スタッフの識別

当会が主催するイベントにおいて、主催者およびスタッフは、自身がスタッフであることを誰もが識別できるよう、専用の名札・Tシャツ・エプロン・バッジなどによって表示します。

2.5 この行動規範の問い合わせ先

この行動規範に関してご質問がある場合は、以下までお問い合わせください。

技術書同人誌博覧会運営事務局 : info@gishohaku.dev (担当: 森川)

2.6 この行動規範のライセンス

この行動規範は「行動規範 - エンジニアの登壇を応援する会」をもとに作成され、CC-BY-4.0に基づいてライセンスされます。

<https://portal.engineers-lt.info/guideline>

第3章

協賛企業様のご紹介

3.1 ゴールドスポンサー



エンジニアフレンドリーシティ福岡

<https://efc.fukuoka.jp/>

エンジニアフレンドリーシティ福岡は「エンジニアが集まる、活躍する、成長する街、福岡」の実現に向け、エンジニアと福岡市が協力して実施している取組みです。

エンジニアやコミュニティが交流、活動できるエンジニアカフェ (<https://engineercafe.jp/ja/>) の運営をはじめ、エンジニアを取り巻く環境の充実に貢献されたコミュニティや企業の表彰制度など、エンジニアファーストの精神で様々なことに取り組んでいます。

3.2 シルバースポンサー

某社

第Ⅱ部

タイムテーブル・セッション一覧

第4章

サークル一覧（配置順）

配置番号	サークル名	ジャンル
あ-01	福岡市(エンジニアフレンドリーシティ福岡)	スポンサー
あ-02	親方 Project	理工系全般-その他-コミュニティ
あ-03	ゆめみ大技林製作委員会	IT-プログラミング-言語
あ-04	Just1factory	IT-デザイン-UI
あ-05	Type D4 Lab	IT-プログラミング-言語
あ-07	Rails Girls わんだーらんど	IT-開発-Web
あ-08	システム工学研究会	IT-開発-Web
あ-09	電腦世界	IT-開発-Web
あ-10	のみぞーん	IT-コンピュータサイエンス-セキュリティ
あ-11	Auth 屋	IT-開発-Web
あ-12	脆弱性診断研究会	IT-コンピュータサイエンス-セキュリティ
あ-13	半田技術研究所	IT-プログラミング-アーキテクチャ
あ-14	Aurantifolia	IT-開発-Web
い-01	へにゃべんて	IT-開発-OS/低レイヤ
い-02	みやこ電子工房	IT-ハードウェア-同人ハード
い-03	徒然メカ	理工系全般-理論・技術-工学系
い-04	まぬる猫の工作部屋	理工系全般-理論・技術-工学系
い-05	CNMTC	IT-開発-IoT
い-06	きんとーん・らぼ	IT-開発-Web
い-08	いもあらい。	理工系全般-理論・技術-理学系
い-09	たいら屋	理工系全般-理論・技術-理学系
い-10	Mathematica 研究会葛飾支部	IT-プログラミング-言語
い-12	からあげ	IT-コンピュータサイエンス-人工知能・AI
い-13	モウフカブール	IT-PC-全般
い-14	フトンカブーレ	IT-PC-全般

あ-01 エンジニアフレンドリーシティ福岡	あ-02 親方Project	あ-03 ゆめみ大技林製作委員会	あ-04 Just1factory	あ-05 Type D4 Lab
あ-07 Rails Girls ゆめみ大技林	あ-08 システム工学研究会	あ-09 電脳世界	あ-10 のみせーん	あ-11 Auth屋
あ-12 脆弱性診断研究会	あ-13 半田技術研究所	あ-14 Aurantfloria	い-01 へにやべんて	い-02 みやこ電子工房
い-03 徒然火力	い-04 まぬる猫の工作部屋	い-05 CNMTC	い-06 きんぐーくんばほ	い-08 いもあらい。
い-09 たいら屋	い-10 Mathematica研究会 茅野支部	い-12 からあげ	い-13 モウカブール	い-14 フトンカブーレ
上記の既刊本のデータをBoothにて公開中です。	Mathematica & Object Oriented Programming	やばい本売ってるよ	新刊 ニヤコロコロ全書籍 & 大澤文孝 明後日から使える AI・機械学習入門	音楽 CD データベース 三文題歌 動物園 水族館写真集 AliceSystem ・メールシステムワーク、時々 NEOGE ・共同効果 ・芳香族を核心で弄った狂気の本 ・Djangoなどで自己用化学構造式検索ツールを作った本



第 5 章

サークルアピール

サークルカットだけでは物足りない！

頒布物やサークル活動についてもっと余すことなく魅力を語っていただくため、技書博では「サークルアピール」として各サークルに1/2ページの枠を用意しました。

サークルは配置順に並んでいます。ジャンルごとに配置していますので、お目当てのジャンルはもちろんのこと、他のジャンルもチェックしてください。きっとすてきな出会いがあるでしょう。

あ-02

親方Project

@oyakata2438



アジャイルに関するネタ 100本
知識、成功、失敗、コミュニティ etc
450ページ 27mm ¥2,000-



基板積んでませんか？
積み基板解消テクニックいろいろ
160ページ ¥1,500-

あ-03

ゆめみ大技林製作委員会



株式会社ゆめみのメンバーが作るアンソロジー本

iOS・Android開発、クロスプラットフォーム開発や開発環境など、
いろんなネタが盛りだくさん！お好みの順番で自由に読んでね♪

い-14

Futonkabure／フトソカ ブーレ



狂気のDB音楽CD500円と写真集800円

委託

AliceSystem

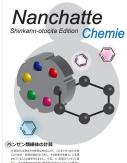
メールと
ネットワーク
時々NEOGEO



委託



共鳴効果



←芳香族を無心で
弄った狂気の本
Djangoなどで自家用
化学構造式検索システム
を作った本→

the
System of
Registration,
Estimation and
Access for organic
Chemists



各1,000円

第 III 部

協賛企業様による寄稿記事

君も文豪にならないか？

タイミー 河又

6.1 タイミーです。名前だけでも覚えて帰ってください。

今回初めて技書博にスポンサーさせていただくタイミーです。

私たちは「働きたい時間」と「働いてほしい時間」をマッチングするスキマバイトサービスです。事業者は工数を掛けずに人手が確保可能になり、働き手は面接・履歴書なしで今日からでも働く新しい体験を提供しています。

有難いことに強烈な Product Market Fit を迎え、プロダクトローンチから 5 年で登録者が 500 万人に到達するなど急成長中の会社です。

6.2 エンジニアの成長を支援する「DevEnable 室」

急成長中のタイミーですが、2023 年 9 月時点でアルバイトを含む社員数 876 名に対してプロダクト本部所属のメンバーが 70 名。この数は EM や PdMなどを含んでおり、実際にコードを書いている Developer の数でいうと 40 名前後になります。

エンジニアを採用することも大切ですが、一人一人がレベルアップすることも等しく重要です。そこでタイミーではエンジニアの成長を支援する DevEnable 室（以下、DE 室）を立ち上げました。

DE 室ではエンジニアの成長を支援するべく 3 つの観点別に施策を作成しました。

6.2.1 1. 個人の力の向上 | 学習・環境に対する支援

DE 室では、開発業務に集中するために社内手続き等ビジネス業務の代行支援や、スキルアップの機会・選択肢を増やす学習費用の補助などのサポートを行います。

エンジニアが開発に集中できること、学習し続けることで個人としてのスキルを伸ばせる環境を目指します。

6.2.2 2. 深い挑戦のある打席 | 事業成果の創出

DE 室では、柔軟な配置転換や抜擢のための機会提供、メンターのアサインを通してエンジニアの一人ひとりの多様な志向に合わせたキャリアアップのための支援を行います。

「テックリードとして専門性を深めていく」「エンジニアリングマネージャーとして開発組織のマネジメントを行う」「PdM として開発全体をリードする」

エンジニアとしてキャリアを積み重ねるなかで、やってみたいことや任せられる役割は変化していきます。

こうした支援を通してエンジニアの成長・活躍できる環境をつくり、事業成果の創出を目指します。

6.2.3 3. 実体験をもとにした再現性のある言語化 | キャリア・コミュニティへの還元

DE 室では、国内外問わず開催される技術カンファレンスへの参加機会の提供やテックブログ、登壇資料作成に集中するためのサポートを行います。エンジニアが仕事を通じて得た知識や経験をそのままにすることなく、体系化・言語化、発信をサポートし、本人のキャリアや属する技術コミュニティの前進を目指します。

6.3 ユニークな「文豪待遇制度」



今回は様々な施策の中でも今回の技書博参加の皆さんにぴったり（?）な施策の紹介をさせてください。

文豪制度は「実体験をもとにした再現性のある言語化 | キャリア・コミュニティへの還元」の観点で作られた施策です。

エンジニアがキャリアを形成していく上では、多種多様なプロジェクトに参加し、経験を積んで

いくことが重要です。同時に自身が携わったプロジェクトで経験したことを体系化・言語化し、アウトプットをすることで再現性を高めることができます。加えてアウトプットしたものが、他のエンジニアやコミュニティにとって有益な情報として共有されることで、自身のエンジニアとしてのプランディングにつながるなど、体験をアウトプットするメリットは多岐に渡ります。とはいえ、仕事が忙しくて手が回らない、そもそも資料作りが得意ではないといったエンジニアもいるでしょう。

そこで発表資料・ブログ記事を執筆する上で生じるあらゆる課題や障壁を解消し、気持ちよく執筆活動にのぞめるようサポートする「文豪待遇制度」をスタートしました。



図: 文豪待遇制度の概要

6.3.1 専属編集

専属編集

自分専用の
執筆編集スタッフをアサイン



タイミーが契約する外部パートナーが編集業務を担当します。誤字脱字、技術的な正誤についてだけでなく、よりわかりやすい表現についての提案も含めて編集と技術広報チームが支援します。編集は進行、スケジュール管理、必要な人（レビューや壁打ちなど）のアサインを担当します。関連資料を集めてきたり、登壇資料のレイアウト相談、口頭記述や初期の壁打ちなども担当します。編集と社内の技術広報チームが校正（レビュー）します。

6.3.2 執筆活動費 全額負担

執筆活動費 全額負担

経費による取材費等の支援



技術検証のためのインフラ費用をはじめとした執筆活動において必要な活動費をタイミーが全額経費負担します（技術検証を会社の AWS 環境、GCP 環境の使用が可能です）

経費の利用例

- ・ 技術検証のためのインフラ費
- ・ 図やイラストの外注費
- ・ 作業場所までの交通費

6.3.3 資料デザイン全面サポート

資料デザイン 全面サポート



資料デザイン作業の外注支援

執筆する登壇資料やテックブログのテンプレデザイン、資料内で使用する図解資料を社内のデザイナーチームに依頼することができます。

6.3.4 文豪缶詰プラン

文豪缶詰プラン



締切直前に
ホテル・旅館で缶詰執筆

カンファレンスでの登壇やブログの締め切りが近いときはホテル・貸し会議室などを利用し、缶詰状態で執筆活動に専念することができます。費用の上限は、1つの記事・登壇につき1万円です。

- 利用可能ホテル・旅館一覧
 - スーパーホテル Premier 赤坂
 - センチュリオンホテル上野
 - ロワジールホテル 品川シーサイド
 - ホテル アマネク 浅草駅前
 - ホテルマンデープレミアム豊洲
 - ホテル福田屋

- アパホテル&リゾート 両国駅タワー
- THE KANZASHI TOKYO ASAKUSA
- ホテルアマネク蒲田駅前～大浴場「四季の湯」～
- 住友不動産ホテル ヴィラフォンテーヌ東京大手町
- ホテルウィングインターナショナルプレミアム東京四谷
- ホテルエミシア東京立川
- アパヴィラホテル

6.3.5 文豪締切催促サービス



希望があれば専属編集から催促の電話をもらうことができます。頻度もリクエスト可能です。

6.3.6 文豪差し入れサービス



2,000 円を上限に、エナジードリンクやお菓子、軽食など希望する「差し入れ」をもらうことができます。

6.4 君も文豪にならないか？

技書博に参加されているあなたも是非タイミーで文豪になりませんか？
ご興味を持って頂けた方は是非、お話ししましょう。

タイミープロダクトエントランスブック



図: タイミープログラムエントランスブック

カジュアル面談



図: カジュアル面談

第 7 章

SOLID 原則をギュッとまとめた

シキム@sikimuOji

7.1 ギュッとまとめました。

SOLID 原則を理解してもらうために、誰しも必ず通る HelloWorld にまとめてみました。

リスト 8.1: Code1:HelloWorld

```
// メイン(実行クラス)
public class Main {
    public static void main(String[] args) {
        Printer printer = new SystemPrinter();
        printer.print(new Hello(), new World());
    }
}

// 出力情報
interface FirstData { String create(); }
interface SecondData { String create(); }
class Hello implements FirstData {
    public String create(){
        return "Hello";
    }
}
class World implements SecondData {
    public String create(){
        return "World";
    }
}

// 出力処理
abstract class Printer {
    abstract void print(FirstData first, SecondData second);
}
class SystemPrinter extends Printer{
    public void print(FirstData first, SecondData second){
        System.out.println(first.create() + " " + second.create());
    }
}
```

7.2 解説

それぞれの、クラスに埋め込まれた SOLID 原則について解説していきます。

開放閉鎖の原則

ここを変更するだけで、出力方法や出力情報を簡単に変更することができます。

リスト 8.2: Code2:開放閉鎖の原則

```
// メイン
public class Main {
    public static void main(String[] args) {
        Printer printer = new SystemPrinter();
        printer.print(new Hello(), new World());
    }
}
```

インターフェース分離の原則

FirstSecondData というように統合することができますが、"Hello Japan"も表示するようにしたいときなどに無駄な実装が発生してしまいます。変更する最小の単位で分けてあるほうが、柔軟な組み合わせで利用することができます。

リスト 8.3: Code3:インターフェース分離の原則

```
// 出力情報
interface FirstData { String create(); }
interface SecondData { String create(); }
class Hello implements FirstData {
    public String create(){
        return "Hello";
    }
}
class World implements SecondData {
    public String create(){
        return "World";
    }
}
```

単一責任の原則

1つのクラスは、1つの変更で同時に使う処理の集まりとしてまとめましょう。複数のタイミングで変更するものが混じっていると、いずれかの処理を変更したときにどちらも確認が必要となってしまいます。

依存性逆転の原則

変更することの少ない出力処理が、変更の多い出力情報の実装を気にしなくていいように、出力情報のインターフェースを利用しましょう。

リスコフの置換原則

Printerで決めたルール以外のことを実装してしまうと、変更の際に実装を確認する必要が出てきてしまい、バグの要因になるのでやらないようにしましょう。

リスト 8.4: リスコフの置換原則

```
// 出力処理
abstract class Printer {
    abstract void print(FirstData first, SecondData second);
}
class SystemPrinter extends Printer{
    public void print(FirstData first, SecondData second){
        System.out.println(first.create() + " " + second.create());
    }
}
```

7.3 人間はミスをする生き物

人間はすぐミスをします。特に変更は、変更した場所以外の影響も考える必要があるためミスをしやすいです。しかし、システム開発は、変更の連続です。コーディング中の気付きや、考慮漏れの発覚、ちゃぶ台返しなど大小様々な変更をしなければいけません。そのため、人間のミスを減らし変更をしやすいコードにすることは、非常に重要になってきます。SOLID原則を意識して作ることで、変更を最小限にミスを減らすことができると思います。

今回は、ここまでとさせていただきます。またいつかどこかで('・ω・')



シキム @sikimuOji <https://twitter.com/sikimuOji>

所属とか(任意)

ひとこと

コードレビューガイドラインの導入と背景

akkie76@akkie76

現在、私のチームは技術育成を目的として、コードレビューガイドラインを導入しています。特に若手エンジニアが抱える技術的な課題に対処するため、レビュー指摘を類型化し、メンバーの弱点を定量化するガイドラインを作成しコードレビューに臨んでいます。

実際のガイドライン作成では、Google が公開するレビューガイドライン^{*1}を参考にし、7つの観点を設けています。

表 8.1: レビューガイドライン 7つの観点

No	観点	概要
1	Design	アーキテクチャ、設計
2	Simplicity	理解容易性
3	Naming	クラス、メソッド、変数名などの命名
4	Style	コードスタイル
5	Functionality	機能（要件）を充足しているか
6	Test	テストの記述、パターンが適切
7	Document	コメント、ドキュメントに関連

特に、No.1 ~ No.4 は、オブジェクト指向の観点で非常に重要な観点といえます。ここからは具体例を踏まえて、実際の使用例を紹介したいと思います。

8.1 Design (設計)

8.1.1 定義

アプリケーションとして一貫性のあるアーキテクチャ、設計となっているか。また、アプリケーションにとって適切な責務・振舞いになっているか。

^{*1} レビューガイドライン (<https://fujiharuka.github.io/google-eng-practices-jp/ja/review/reviewer/standard.html>)

8.1.2 具体例

基本的には、以下のケースが指摘の対象になります。アーキテクチャ原則違反はアプリケーションで導入しているアーキテクチャを前提にしており、これに違反している場合は指摘の対象になります。他には、以下のような基本的な SOLID 原則に反するような場合は指摘の対象になります。

- アーキテクチャ原則違反
- 単一責任の原則違反
- 密結合
- 低凝集
- DRY 原則違反 etc.

単一責任の原則違反のケース

以下のコードでは、「add()」で2つのバリデーションと2つのビジネスロジックが実装されています。単純に「商品を追加する」以外の責務を1つの関数の責務として行われています。この関数自体複数の責務を持ってしまっている（責務超過）ため、Design 指摘の対象となります。

リスト 9.1: kotlin:SpecialDiscountManager.kt

```
class SpecialDiscountManager {

    // 商品の割引に関するビジネスロジッククラス
    lateinit var manager: DiscountManager

    fun add(product: Product): Boolean {
        if (product.id < 0) { // バリデーション 1
            throw IllegalArgumentException()
        }
        if (product.name.isEmpty()) { // バリデーション 2
            throw IllegalArgumentException()
        }
        val tempPrice: Int = if (product.canDiscount) { // 条件分岐 1
            manager.totalPrice + manager.getDiscountPrice(product.price)
        } else {
            manager.totalPrice + product.price
        }
        return if (tempPrice < 3000) { // 条件分岐 2
            manager.totalPrice = manager.discountProducts.add(product)
            true
        } else {
            false
        }
    }
}
```

このように1つの関数に複数の責務が実装されるような場合、以下のようになるべく関数の責務を分離することが望ましいでしょう。

リスト 9.2: kotlin:SpecialDiscountManager.kt

```
class SpecialDiscountManager {  
  
    /** 商品の割引に関するビジネスロジッククラス */  
    lateinit var manager: DiscountManager  
  
    /**  
     * 商品を追加する  
     *  
     * @param product 商品  
     */  
    fun add(product: Product) {  
        // 不正な商品をバリデートする  
        validateInvalidProduct(product)  
        // 商品を追加する  
        if (canAdd(product)) {  
            manager.totalPrice = manager.discountProducts.add(product)  
        }  
    }  
  
    /**  
     * 不正な商品をバリデートする  
     *  
     * @param product 商品  
     * @throws IllegalArgumentException 不正な商品である場合の例外  
     */  
    private fun validateInvalidProduct(product: Product) {  
        if (product.id < 0 || product.name.isEmpty()) {  
            throw IllegalArgumentException()  
        }  
    }  
  
    /**  
     * 商品が追加可能か  
     *  
     * @param product 商品  
     * @return 追加可能かどうか  
     */  
    private fun canAdd(product: Product): Boolean {  
        val tempPrice = getTempPrice(product)  
        return tempPrice >= 3000  
    }  
  
    /**  
     * 商品の一時金額を取得する  
     *  
     * @param product 商品  
     * @return 商品の一時金額  
     */  
    private fun getTempPrice(product: Product): Int {  
        return if (product.canDiscount) {  
            manager.totalPrice + manager.getDiscountPrice(product.price)  
        } else {  
            manager.totalPrice + product.price  
        }  
    }  
}
```

8.2 Simplicity (理解容易性)

8.2.1 定義

可読性のあるコーディングになっているか。処理ができるだけシンプルな振る舞いになっているか。

8.2.2 具体例

以下のように実装が複雑になる場合、指摘の対象になります。

- ネストが深い if 文
- 複雑な三項演算子文
- 冗長な SQL
- stream, filter, map を多用した Object 整形文 etc.

このように分岐が多い if 分は指摘の対象になります。

リスト 9.3: kotlin:Before.kt

```
val powerRate: float = member.powerRate / member.maxPowerRate
var currentCondition: Condition = Condition.DEFAULT
if (powerRate == 0) {
    currentCondition = Condition.DEAD
} else if (powerRate < 0.3) {
    currentCondition = Condition.DANGER
} else if (powerRate < 0.5) {
    currentCondition = Condition.WARNING
} else {
    currentCondition = Condition.GOOD
}
return currentCondition
```

実際のレビューコメントでは、以下のようにネストを解消するように指摘をする場合などに使用します。

リスト 9.4: kotlin:After.kt

```
val powerRate: float = member.powerRate / member.maxPowerRate
if (powerRate == 0) {
    return Condition.DEAD
}
if (powerRate < 0.3) {
    return Condition.DANGER
}
if (powerRate < 0.5) {
    return Condition.WARNING
}
return Condition.GOOD
```

8.3 Naming (命名)

8.3.1 定義

変数やクラス、メソッドに責務を意図した明確な名前が付けられているか。英語文法に誤りがないか。typo もこれに含まれる。

8.3.2 具体例

このような命名に関する指摘をする場合に使用します。

* 振る舞いと一致しない変数名、関数名* 責務と一致しない関数名* 英文法の誤り etc.

8.4 Style (コードスタイル)

8.4.1 定義

コードスタイルが言語仕様に準拠しているか。—

8.4.2 具体例

コードスタイルは、言語仕様やフレームワークに準拠することが基本です。公式にガイドラインが公開されている場合、これに反することは避けるべきです。

- 静的解析違反
- 不適切なアクセス修飾子
- 表記違反 (スネーク、キャメルなど) etc.

8.5 Functionality (機能要求)

8.5.1 定義

システムとして外部仕様を充足しているか。作者が意図した通りの振る舞いであるか。また、システムの通信量、パフォーマンスに懸念がないか。

8.5.2 具体例

主な観点としては、外部機能を充足しているかという点が対象になります。また、パフォーマンスやセキュリティといった非機能要件に懸念がある場合も指摘の対象となります。

- 外部仕様の充足していない（不具合）
- 概要設計書のフローチャートと異なるフローになっている
- パフォーマンスの低下が考えられる
- 不要データを送信している etc.

8.6 Test (テスト)

8.6.1 定義

システムとして適切な自動テストを兼ね備えているか。自動テストの内容で品質を担保できているか。

また、システムを担保するパラメータ群を備えているか。__

8.6.2 具体例

テストコードが期待になっていない場合や、テストでのパラメータに考慮漏れがある場合、指摘の対象になります。

- 対象のメソッドがテストされていない
- テストパターンが網羅できていない（パラメータテスト、閾値テストの不足）
- 分岐がパターンが網羅されていない
- 実装上宣言している静的定数値が直接ハードコードされている
- アーキテクトに準じたテストになっていない

8.7 Document (文章)

8.7.1 定義

ソースコード上に記載されている doc、コメントが適切な内容であるか。また、関連するドキュメントは更新されているか。その内容は適切か。__

8.7.2 具体例

ソースコードに関連するコメントだけでなく、プロジェクトで管理している関連ドキュメント (README) も対象になります。

- 関連ドキュメントの更新漏れ (README など)
- doc やコメントの内容が不適切、内容が不適切

8.8 指摘対応の要否

コードレビューでは上記の 7 つの観点に加えて、指摘修正の要否を 4 つの類型に分けてコメントをします。

表 8.2: 修正要否

観点	概要
MUST	PR、MR をマージするためには必ず修正が必要
SHOULD	修正なしにマージすることはできるが、リリースまでには修正が必要
IMO	レビューアー観点の意見。修正不要
NITS	IMO より細かい意見など。修正不要

コードレビューでマージするために必要な修正は MUST 指摘となります。MUST と SHOULD の使い分けは難しい部分もありますが、SHOULD での指摘事項は別途リリースまでに対応するなど、レビューアーを相談して対応を進めます。一方、IMO や NITS は修正は不要ですが、修正しない場合はその旨をコメントに返信してもらい、コメントを閉じてからマージする運用をしています。

8.8.1 具体的な利用方法

実際にコードレビューをするとき、上記の 7 つの観点と修正の温度感をこのように交えた Prefix を付けて、コメントをします。

8.8.2 指摘例

MUST(Design): ドメインロジックが Controller クラスに実装されています。 domain 層の対象 package に新しくクラスを作成して実装を移してください。

8.9 最後に

このように、レビューコメントを分析し PDCA サイクルを回すことで、技術的課題へのアプローチ方法も見えてきます。また、コードレビューでオブジェクト指向が学べるといった恩恵を受けることできるので、ぜひチームに合ったコードレビューガイドラインを作成してみてはいかがでしょうか。

第Ⅳ部

イベントを支える技術-スタッフ寄稿—

カンファレンスを実現するには、様々な準備があります。スタッフのワーク、考えたことの一端をまとめました。

興味を持った方は、ぜひスタッフの扉をたたいてみてください。

第9章

搬入・宅急便関係の準備いろいろ

おやかた@oyakata2438

搬入関係を（も）担当します、おやかたです。

ここでは、搬入に関わるいろいろを述べてみたいと思います。裏側を知っていただけすると幸いです。

搬入搬出としてはサークル対応と、事務局の荷物の対応があります。また、事前準備、当日の搬入、当日の搬出と3つありますから、ぜんぶで6パターンになりますね。

9.1 事前準備

事前準備編です。

9.1.1 事務局の事前準備

事務局の荷物は、一般参加者から見える部分と見えない部分とありますが、それなりの物量があります。もちろん会場で借りるものもありますが、会場に「直接」届くもの、そして手持ちで搬入するものがあります。

会場で借りるものと手配する

会場で借りるものと手配する代表格は、机といいます。会場に並んでいる机や椅子は、当たり前ですが会場から借ります。もちろんコミケなどのように超でかい規模であれば外部のレンタル業者に手配する必要があるでしょうが、今のところの規模では会場備え付けの備品で事足ります。

会場を予約する段階でおおよその数を申請書に書きます。100サークル前後を想定していれば、机100本、椅子200本、など。予約の段階では記入する枠がない場合は、後日「使用計画書」的な書類を提出することになります。ある程度サークル数などが固まっているはずですから、それなりの精度で書くことができるでしょう。

それ以外に会場から借りるものとしては、音響設備、電源、パーティションや掲示関係のいろいろ。会場ごとに所定の手続き書類がありますので、そのテンプレートに記入しましょう。

音響設備は、有線マイク、無線マイク、など。BGMをかけるための有線放送の設備があつたりすることもあります。いわゆるPA卓のようなものが備え付けのこともありますね。いずれにせよ、有償なのか無償なのかは確認しておきましょう。

電源は容量計算や配線計画なども必要になるのでこれ以上は触れません。基本的に同人誌即売会

ではあまり使わないため、サークルが自由に使ってよいというシチュエーションにはならないかと思いますが、ハードウエア系のデモをするサークルに電源を提供する場合などはそれなりに準備が必要です。これはまた別途、「同人誌即売会を作ろう」の方で書くかもしれません。

会場に届くものを手配する。

会場に直接届くものもいろいろあります。

事務局のもので会場に直接届くものの代表は、ガイドブック、そしてポスターなどの会場掲示用の印刷物です。

ガイドブックは入場者全員に配布するものですから、それなりの部数があります。技書博の場合、無料配布ですが、即売会に醉っては入場料の代わりとして有償配布されたりしますね。

ガイドブックなどは、印刷所から直接搬入されます。印刷所の人（またはそこが手配した運送業者さん）が持ってきてくれます。ガイドブックを注文した印刷所さんと、搬入について調整しておきましょう。

具体的な調整事項は、搬入希望時間と、駐車場/荷捌き所の利用可否、です。

11時開会として、11時に来てては遅いわけです。一方で、会場引き渡し前に来てもらっても対応できない/待たせてしまうので、準備スケジュールをもとに希望時間帯を連絡しておきましょう。

以下は技書博の例です

8:00 会場引き渡し/開錠

9:00～10:00 搬入希望

9:30 サークル入場開始

11:00 一般入場開始

こんな感じで、スケジュールと希望時間を伝えましょう。他の印刷所さんも同じです。

直接搬入するもの

事務局の資材には、スタッフが手持ちで搬入するものがあります。受付用の資材、測量用のメジャーやマーキング用のテープ、文房具、最近だと体温計や消毒キット、などがあります。

測量用のメジャーやマーキング用テープなどは、準備の最初の段階で必要なものですから、会場引き渡しの直後に必要となるものです。ですから、車で搬入する場合、その搬入を担当する人の到着スケジュールをきちんと押さえておきましょう。

それ以外にも、来場者に配布する入場パスなどで、スタッフ作業を経由するもの、あるいは運送業者の都合で会場直送ができなかったものが含まれます。また、来場者に配布するものとして、過去のガイドブックの残部なども、ここに含まれます。基本的に過去のガイドブックはもう使うことはないのですが、ゴミとしてはもったいないです。初めて技書博に参加した人、あるいは前回参加できなかった人がもらってくれることも少なくありませんので、受付などに置いておくと無駄にななりません。これらは、スタッフが手持ち搬入する必要があります。あるいは、宅配便等で搬入するなど、荷物の振り分けを考えておいてください。

さて、これらのものをどうやって搬入するかですが、基本的には車（自家用車またはレンタカー）で搬入できると便利です。手持ち搬入には物量的にも限界がありますしね。

たいていの会場には駐車場があります。ですから、予約の段階で「搬入用に2台」といった形で申請をしておきます。一般サークルまで車搬入できるようにするかは議論がありますが、事務局枠は押さえておきましょう。場合によっては、他の催事との調整が必要となる場合もあります。しかもその日程は1-2か月前であることがざらです。事務局荷物を車搬入する場合は、この事前調整が必須であると考えておきましょう。

また、事務局資材は時々見直しましょう。使わないものを持って行くのは作業効率の点でもよくありません。

9.1.2 一般サークルの事前準備

一般サークル向けには、宅急便の業者さんの選定、搬入予定時刻の調整、伝票記入要領の決定と、サークルへの案内です。

業者選定と搬入予定時刻の調整は前節で完了していますので、伝票記入要領の決定についてです。

大量(おおむねサークル数×1.5箱程度)の荷物が届くので、どの荷物がどのサークルのものなのかはっきりさせる必要があります。サークル数が超多い場合は、搬入票を作ることがあります。搬入票は、箱の側面に貼り付ける紙で、サークル名や配置番号を記載したものです。箱の側面に張り付けることで、箱を積み上げたままでもわかりやすいですし、より大きな文字で表示することができます。

サークル数が少ない場合、伝票の備考欄でこれに代えることもできます。備考欄に、イベント名、内容物、サークル名、配置を記入してもらいます。

書籍 技書博 8

あ-01 親方 Project

そして、これらの情報を各サークルに案内しましょう。どんなに遅くても10日前には案内を出してください。

週末に準備、発送手配をする、なんていうスケジュールはあるあるですよね。ですからこの時に案内がないとサークルは困ってしまいます。

案内の中には、送り先住所、催事名、到着指定日時(当日、時間指定なし、など)、備考欄の記載要領を明記します。備考欄(あるいは搬入票)の記載がない場合、荷物の紛失や遅配などのリスクが上がります。また、不着などのトラブルに備え、伝票番号は必ずメモしておく、あるいは当日伝票控えを持っておくように依頼して下さい。

それにあわせて、戻りの荷物の発送の案内などもできるといいですね。

Web等に記載するとともに、サークル、スポンサー、その他関係者にメールをするとよいでしょう。

9.2 まとめ

荷物の対応は地味ですが、重要度が高い調整項目です。

特に技術同人誌は一般にページ数が多く、かなりの重量になります。ということは、手搬入はあまり現実的ではありませんから、おのずと宅配便での搬入が増えます。ここでの準備をミスると当日配布するもの届かないなどの大きな事故につながります。事前調整を忘れないようにしましょうね！

9.3 次回予告

思ったより分量が多くなったので、当日の対応、撤収の対応については次回の記事に回します。
おたのしみに！

スタッフ紹介

コアスタッフ

ariaki @ariaki4dev



スタートアップの開発責任者を拝命し、刺激的な仕事で毎日楽しく駆けめぐり回っています。

おやかた（親方 Project） @oyakata2438



技術書を生やすお兄さんとして、LT 登壇や合同誌主催で新しい著者の背中を押すのが楽しい毎日です。技術同人誌を書くことはメリットしかない！ 楽しいぞ！ 本を書く場所を作れるということで、コアスタッフとして参画しました。スタッフもたのしー。

ロジウム @__rhodium__



計算量爆発と戦う日曜プログラマー(ただし爆発に巻き込まれて死亡する)。機械学習には興味はあるが、何もわからないでいる……。専門は有機化学。有機化学の顔をしていたら、私も?

デザイン

ささくら @ssakura_sn



エンタメと同人誌即売会が好きなグラフィックデザイナーです。普段はデザイナー向けの同人誌即売会「デザイン読書日和」のコアスタッフ。今回ご縁あって参戦。広がる! 同人の輪!!

OOC2024 公式ガイドブック

2024年3月24日 OOC 2024

デザイン hoge

編 集 おやかた (@oyakata2438)、ありあき (@ariaki4dev)

発行所 OOC2024 実行委員会

印刷所 K-9