References

[1] A. Aldayel and K. Alnafjan. Challenges and best practices for mobile application development. In *Proceedings of the International Conference on Compute and Data Analysis*, pages 41–48, 2017.

[2] Q. Ashfaq, R. Khan, and S. Farooq. A comparative analysis of static code analysis tools that check java code adherence to java coding standards. In *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*, pages 98–103. IEEE, 2019.

[3] F. S. Barbosa and A. Aguiar. Removing code duplication with roles. In *2013 IEEE 12th International Conference on Intelligent Software Methodologies, Tools and Techniques (SoMeT)*, pages 37–42. IEEE, 2013.

[4] X. Chen, A. Y. Wang, and E. Tempero. A replication and reproduction of code clone detection studies. In *Proceedings of the Thirty-Seventh Australasian Computer Science Conference-Volume 147*, pages 105–114, 2014.

[5] A. W. R. Emanuel, R. Wardoyo, J. E. Istiyanto, and K. Mustofa. Modularity index metrics for java-based open source software projects. *arXiv preprint arXiv:1309.5689*, 2013.

[6] E. Gamma, R. Helm, R. Johnson, J. Vlissides, and D. Patterns. Elements of reusable object-oriented software. *Design Patterns. massachusetts: Addison-Wesley Publishing Company*, 1995.

[7] F. B. HARUN. Review of ios architectural pattern for testability, modifiability, and performance quality. *Journal of Theoretical and Applied Information Technology*, 97(15), 2019.

[8] I. Heitlager, T. Kuipers, and J. Visser. A practical model for measuring maintainability. In *6th international conference on the quality of information and communications technology (QUATIC 2007)*, pages 30–39. IEEE, 2007.

[9] ISO. Iso/iec 25010:2011(en) systems and software engineering — systems and software quality requirements and evaluation (square) — system and software quality models.

[10] E. Juergens, F. Deissenboeck, and B. Hummel. Code similarities beyond copy & paste. In *2010 14th European Conference on Software Maintenance and Reengineering*, pages 78–87. IEEE, 2010.

[11] C. J. Kapser and M. W. Godfrey. "cloning considered harmful" considered harmful: patterns of cloning in software. *Empirical Software Engineering*, 13(6):645–692, 2008.

[12] V. Lenarduzzi, A. Sillitti, and D. Taibi. A survey on code analysis tools for software maintenance prediction. In *International Conference in Software Engineering for Defence Applications*, pages 165–175. Springer, 2018.

[13] I. Malavolta, R. Verdecchia, B. Filipovic, M. Bruntink, and P. Lago. How maintainability issues of android apps evolve. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 334–344. IEEE, 2018.

[14] A. Monden, D. Nakae, T. Kamiya, S.-i. Sato, and K.-i. Matsumoto. Software quality analysis by code clones in industrial legacy software. In *Proceedings Eighth IEEE Symposium on Software Metrics*, pages 87–94. IEEE, 2002.

[15] B. S. Panca, S. Mardiyanto, and B. Hendradjaya. Evaluation of software design pattern on mobile application based service development related to the value of maintainability and modularity. In *2016 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–5. IEEE, 2016.

[16] M. D. Papamichail, T. Diamantopoulos, and A. L. Symeonidis. Measuring the reusability of software components using static analysis metrics and reuse rate information. *Journal of Systems and Software*, 158:110423, 2019.

[17] D. Rodriguez and R. Harrison. An overview of object-oriented design metrics. 2001.

[18] A. A. Saifan and A. Al-Rabadi. Evaluating maintainability of android applications. In *2017 8th International Conference on Information Technology (ICIT)*, pages 518–523. IEEE, 2017.

[19] F. E. Shahbudin and F.-F. Chua. Design patterns for developing high efficiency mobile application. *Journal of Information Technology & Software Engineering*, 3(3):1, 2013.

[20] A. Shvets. What's a design pattern?

[21] P. Tomas, M. J. Escalona, and M. Mejias. Open source tools for measuring the internal quality of java software products. a survey. *Computer Standards & Interfaces*, 36(1):244–255, 2013.

Supplements

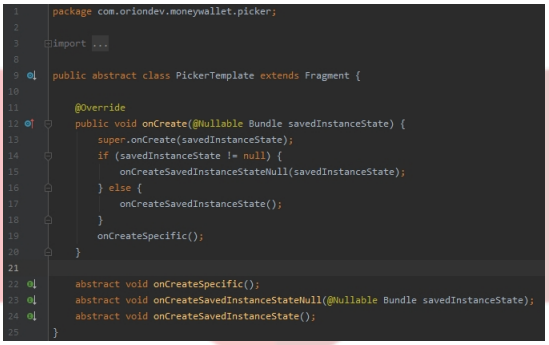

Fig 9. Template Pattern Group 3

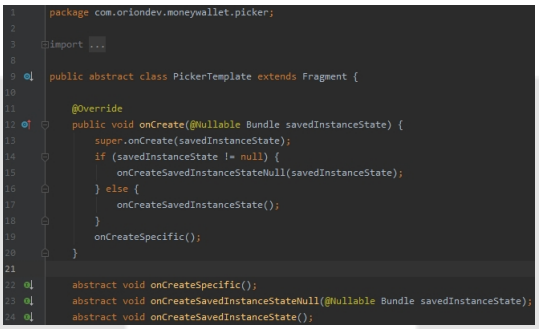

Fig 10. Template Pattern Group 4

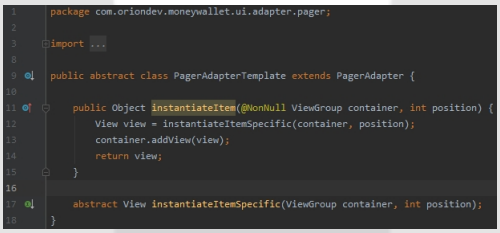

Fig 11. Template Pattern Group 31

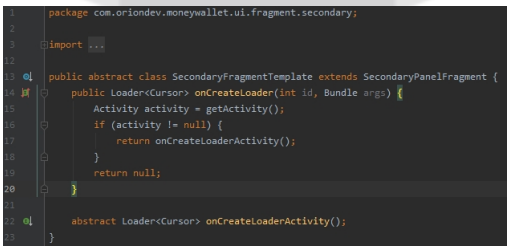

Fig 12. Template Pattern Group 51